Selection of Optimal Salient Time Steps by Non-negative Tucker Tensor Decomposition

J. Pulido\textsuperscript{1} and J. Patchett\textsuperscript{1} and M. Bhattarai\textsuperscript{1} and B. Alexandrov\textsuperscript{1} and J. Ahrens\textsuperscript{1}

\textsuperscript{1}Los Alamos National Laboratory, USA

Abstract
Choosing salient time steps from spatio-temporal data is useful for summarizing the sequence and developing visualizations for animations prior to committing time and resources to their production on an entire time series. Animations can be developed more quickly with visualization choices that work best for a small set of the important salient timesteps. Here we introduce a new unsupervised learning method for finding such salient timesteps. The volumetric data is represented by a 4-dimensional non-negative tensor, $X(t,x,y,z)$. The presence of latent (not directly observable) structure in this tensor allows a unique representation and compression of the data. To extract the latent time-features we utilize non-negative Tucker tensor decomposition. We then map these time-features to their maximal values to identify the salient time steps. We demonstrate that this choice of time steps allows a good representation of the time series as a whole.

CCS Concepts
\begin{itemize}
\item Human-centered computing \rightarrow Visualization design and evaluation methods;
\end{itemize}

1. Introduction
Supercomputers get ever larger with equitably larger potential for data extracts that allow simulation scientists the ability to analyze and study the evolution of the phenomena they model. Outputs are frequently high spatial resolution timesteps, sampled regularly in simulated time or simulation cycles, leaving hundreds or thousands of timesteps on persistent storage representing terabytes to petabytes of data to read and process for summary results. Depending on the maturity and novelty of the model, those data may be repeatedly analyzed for different purposes. We are concerned with the process of efficiently developing and viewing imagery derived from full spatial resolution timesteps “dumped” from simulations.

In this work, we are interested in identifying a subset of key timesteps, i.e. salient timesteps, that efficiently represent an entire time series. This subset can be used both for developing visualization parameters to process the entire time series, to use as keyframes to summarize important moments in the simulation, and to more efficiently communicate to collaborators key timesteps worth investing further analysis.

We introduce the application of products of unsupervised machine learning based on non-negative tucker factorization (NNTF). This novel NNTF process is adapted to produce a set of features and a set of weights for each of those features. For each feature, we identify the specific timestep that it is most heavily associated with, and tag that time step as a keyframe. For comparison, we propose a set of evaluation criteria to measure the quality of selected key timesteps of our experimental method against other traditional methods.

2. Related work
For workflows, Ma et al. [MLF\textsuperscript{*}12] covers the importance of streamlining the visualization process through scientific storytelling, where the selection of important points in a simulation, i.e. keyframes, can improve the process of production visualization.

A variety of techniques exist for identifying keyframes. Meyers, et al. [MLF\textsuperscript{*}16] look at salient time step selection for in situ use cases, they assume they have an incoming stream of time steps and have to decide whether to keep each time step or not as it is produced. They produce a linear piecewise model and evaluate each new timestep against that model until a new timestep falls out of the precision of that model at which point they store a new timestep and start a new model. Tong, et al. [TLS12] find the best K time steps from a time varying dataset with a user specified K. They define a cost between time steps and using Dynamic Time Warping (DTW) and then use dynamic programming to find the minimum cost for a given k timesteps. Frey and Ertl use a flow based distance metric to quantify the distance to neighboring time steps and minimize the selected time steps difference from the entire time series. Our work is similar to Porter, et al. [PXv\textsuperscript{19}] who use an unsupervised machine learning autoencoder to develop a feature set based on all time steps, then select keyframes based on those features. We use a Non-negative Tucker Factorization to identify features and their weights, then select keyframes based directly on those weights.
In practice, there has been usage of more traditional methods for identifying keyframes. In this paper, we make use of such methods to perform comparative analysis against NNTF.

Wavelets for keyframe selection have traditionally been in use in selecting important frames from a series of videos [CLN99]. By using the Haar basis function, a multiscale wavelet decomposition is performed for every keyframe for all keyframes in a dataset. More recent improvements to this method have been made by the introduction of K-means clustering on the wavelet coefficients [HNMK04]. By performing K-means on the high-pass, fine coefficients for a target K frames, a K amount of clusters are computed that represent keyframe groups. Withing each keyframe group cluster, the keyframe that has the shortest distance to the center cluster is considered the most important keyframe.

For large data simulations, it is important to estimate important keyframes or timesteps within a simulation for post-analysis and in-situ analysis routines. In Meyers et al. a new system of unique float binning is used to extract important keyframes as time-steps in a simulation [MLF 16].

In information theory, it is believed that an entropy-based approach can provide valuable data points when selecting time steps from a simulation. When evaluating multiple timesteps, the frames of which the highest entropy are selected but may have an unfortunate outcome of selecting subsequent frames in a row that relate to a cluster of keyframes with high activity.

Finally, the most trivial method of selecting time steps from a simulation is to pick every nth cycle from the simulation. This method is most commonly used in AMR codes or Lagrangian codes where the cell sizes can change from timestep to timestep.

### 3. Non-negative Tucker Tensor Decomposition for Salient Timestep Selection

![Figure 1: The weights for each feature plotted over timestep. A rendering of the feature that lines up with each time step is numbered. Vertical lines on the plot show the keyframe (time index) that will be chosen based on that feature.](image)

#### 3.1. Non-negative Tucker Factorization

In big-data analysis, it is often difficult to link directly the data to the parameters of the generating processes, since the datasets are formed exclusively by directly observable quantities, while the underlying processes/features remain unobserved, hidden, or latent [Eve13]. Extracting these latent (not directly observable) features not only reveals valuable information about hidden causality and mechanisms, but it also reduces the dimensionality by revealing the low-dimensional latent structure that represents the whole dataset.

#### 3.2. Matrix Factorization

One of the most powerful tools for extracting latent features is factor analysis. In two dimensions factor analysis can be performed by various versions of Principle Component Analysis (PCA) [Jol86], Independent Component Analysis (ICA) [ACY96], or Non-negative Matrix Factorization (NMF) [PT94]. The presence of the non-negativity constraint in NMF makes the extracted latent features physically interpretable, since they are parts of the data [LS99]. Importantly, many variables, e.g., pixels, density, counts, etc., are naturally non-negative and the extracted features will not have physical meaning if the non-negativity is not at place.

#### 3.3. Tensor Factorization

Most datasets are high-dimensional and are represented by tensors, or multidimensional arrays. Such tensors typically describe multiple concurrent latent processes imprinting their signatures in the observable state variables in different dimensions.

![Figure 2: Illustration of nTD-1 for a 3-dimensional tensor. A) Unfolding of the tensor \( X(t, x, y) \). B) Decomposition of the unfolding, \( A^{(1)}(t, x \ast y) \), via NMF. C) Reshaping the columns of the matrix \( H \) to obtain the space-factors.](image)

Tensor factorization, which is the higher-dimensional analog of...
matrix factorization, is an unsupervised learning method that represents a cutting-edge approach for factor analysis. Its main objective is to decompose a high-dimensional tensor into factor matrices and a core-tensor, where the factor matrices carry the latent features in each tensor dimension [KB09]. Tucker decomposition is a classical tensor factorization [Tuc06]. Tucker decomposition is not unique but by Tucker decomposition one can extract the minimal subspaces [FH12], which represent the latent features in the corresponding dimensions [ASVR19]. If the data is non-negative, we perform non-negative Tucker decomposition and the minimal subspaces become cones [ADMS19].

### 3.4. Non-negative Tucker-1 decomposition

In this paper we utilize non-negative Tucker-1 decomposition (nTD-1) [CZPA09], illustrated on Figure (2). nTD-1 unfolds/flatten the original volumetric data-tensor, \(\mathcal{X}(t,x,y,z)\), along its time dimension. Thus, nTD-1 converts the 4-dimensional tensor, \(\mathcal{X}(t,x,y,z)\), to a 2d representation, \(\mathcal{A}^{(1)}(t,x+y+z)\), see Figure (2), panel A. Further, nTD-1 uses NMF to extract the latent structure of \(\mathcal{A}^{(1)}(t,x+y+z)\). Specifically, NMF approximates, \(\mathcal{A}^{(1)} \approx \mathcal{W}^{(1)} \otimes \mathcal{H}^{(1)}\), with a product of two non-negative factor matrices, \(\mathcal{W} \in \mathbb{R}^{n \times K}\) and \(\mathcal{H} \in \mathbb{R}^{K \times n \times n} \times \mathbb{R}_{+}\) (Figure (2), panel B), such that the difference,

\[
O = ||\mathcal{A}^{(1)}(t,x+y+z) - \sum_{i=1}^{K} W_i(t) H_i(x+y+z)||_{\text{dist}}
\]

\[
W_i(t) \geq 0; \quad H_i(x+y+z) \geq 0.
\]

is minimal under some distance, \(\|..\|_{\text{dist}}\) and for a given small latent dimension \(K\). Here, for \(\|..\|_{\text{dist}}\) we use Kullback–Leibler divergence: \(D_{KL}(X|Y) = \sum_{j} X_j \log \left( \frac{X_j}{Y_j} \right) - X_j + Y_j\). We utilize the distributed NMF algorithm presented in [BCS20b] to perform the matrix factorization of the large-scale datasets for the keyframe selection.

### 3.5. The Salient Time Step Selection

NMF is underpinned by a statistical generative model of superimposed components that can be treated as latent variables of Gaussian, Poisson, or other mixed model [FC09]. In the case, the \(K\)-columns, \(W_i(t)\) of \(\mathcal{W}\), represent the latent time-features, while the \(K\)-columns, \(H_i(x+y+z)\) of the transposed matrix \(\mathcal{H}\), are the corresponding space-factors. After the extraction of the factor-matrices, \(\mathcal{W}(t,K)\) and \(\mathcal{H}(K,x+y+z)\), we reshape the matrix \(\mathcal{H}(K,x+y+z)\) to construct the 3-dimensional tensors, \(\mathcal{H}_{s}(x,y,z)\). Each \(\mathcal{H}_{s}(x,y,z)\) corresponds to a 3-dimensional space-feature, and we have,

\[
\mathcal{X}(t,x,y,z) = \sum_{s=1}^{K} W_i(t) H_i(x,y+z) + \mathcal{E}(t,x,y,z).
\]

Figure (2), panel C. In Eq. (2) \(\mathcal{E}\) is the tensor error of minimization.

We leverage this observation, (Eq. 2), to select the timesteps that are most strongly associated with each space-feature. This is done simply by finding the index of the largest value in each of the \(K\) columns of \(\mathcal{W}\). This mapping provides us with \(K\) easy interpretable features, \(\mathcal{H}_{s}(x,y,z)\), each associated with a specific “influential” time point, the set of which we call the optimal latent salient timesteps. The code for this method is available [BCS20b].

### 4. Results

#### 4.1. Evaluation Criteria

To evaluate the selection of \(K\) number of keyframes, we use the comparative analysis framework named Foresight [GBP20]. To evaluate the quality of selected \(K\) keyframes, we perform a full temporal dataset reconstruction using only those \(K\) keyframes. To reconstruct time-steps in between the selected keyframes, we perform a linear interpolation to the nearest frame. For this initial study, we’ve selected \(K=19\) because it gives us reasonable results for all of the methods tested in this paper, and provides fair results for traditional methods such as regular and random sampling.

To evaluate the quality of a fully reconstructed temporal dataset from a number of keyframes, we use traditional statistical and image quality algorithm metrics: Total absolute error (TAE), mean-square error (MSE), Peak signal-to-noise ratio (PSNR) [HZ10], signal-to-noise ratio (SNR), the structural similarity index measure (SSIM) [ZBSS04], the multi-scale structural similarity index measure (MS-SSIM) [WSB03], and a Universal-image Quality Index (UQI) [ZB02]. The quality of reconstructed datasets are evaluated by the minimization of TAE and MSE, while maximizing SNR, PSNR, SSIM, MS-SSIM, and UQI metrics.

There are key differences between statistical metrics and image quality algorithm metrics. Traditional statistical methods, i.e. TAE, MSE, SNR, and PSNR, aim in measuring differences between the data points (cells) of the original and approximated datasets. In contrast, image quality metrics, i.e. SSIM, MS-SSIM, and UQI, do the opposite and quantify similarities between data points. Unlike the former, image quality metrics are standardized, i.e. values between 0 to 1, making it easier to evaluate and understand regardless of dataset type. These differences make image quality metrics more suitable for targeting visualization-oriented applications.

#### 4.2. Deep-water impact dataset

![Figure 3: Volume renderings of six Deep Water Impact time steps chosen by our method.](image)

The Deep Water Impact Ensemble dataset [PG17], is a set of simulations created to study the propensity of asteroids impacting deep ocean water to create tsunamis. Simulated by xRage [GWC08], a parallel multi-physics Eulerian hydrodynamics code developed at the Los Alamos National Laboratory, the dataset is produced using an Adaptive Mesh Refinement (AMR) technique that allows higher and lower resolution areas of the simulation grid for computation. We use the yb31 ensemble member- a 250 meter diameter asteroid entering the atmosphere at 45 degrees, and...
Table 1: Results for the key-frame selection methods on the Deepwater impact dataset. MSE* is \(10^{-3}\). Wavelets M1 is coefficient thresholding, while Wavelets M2 is K-Means clustering.

<table>
<thead>
<tr>
<th>Method</th>
<th>TAE</th>
<th>PSNR</th>
<th>SNR</th>
<th>MSE*</th>
<th>SSIM</th>
<th>MS-SSIM</th>
<th>UQI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>4.598</td>
<td>48.086</td>
<td>39.742</td>
<td>0.9291</td>
<td>0.9294</td>
<td>0.7099</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>9.526</td>
<td>45.091</td>
<td>36.715</td>
<td>0.0099</td>
<td>0.9321</td>
<td>0.9330</td>
<td>0.0653</td>
</tr>
<tr>
<td>Entropy</td>
<td>44.011</td>
<td>31.936</td>
<td>23.594</td>
<td>0.719</td>
<td>0.9294</td>
<td>0.9293</td>
<td>0.3789</td>
</tr>
<tr>
<td>Wavelets(M1)</td>
<td>25.374</td>
<td>38.894</td>
<td>27.552</td>
<td>0.3409</td>
<td>0.9466</td>
<td>0.9293</td>
<td>0.5071</td>
</tr>
<tr>
<td>Wavelets(M2)</td>
<td>6.0985</td>
<td>49.225</td>
<td>33.806</td>
<td>0.0023</td>
<td>0.9497</td>
<td>0.9393</td>
<td>0.7292</td>
</tr>
<tr>
<td>Unique Floats</td>
<td>11.584</td>
<td>42.345</td>
<td>34.003</td>
<td>0.339</td>
<td>0.9281</td>
<td>0.9293</td>
<td>0.6471</td>
</tr>
<tr>
<td>NNTF</td>
<td>8.112</td>
<td>45.202</td>
<td>36.700</td>
<td>0.064</td>
<td>0.9398</td>
<td>0.9405</td>
<td>0.6472</td>
</tr>
</tbody>
</table>

Table 2: Results for the key-frame selection methods on Antarctic Ice-sheet ocean dataset. TAE* is \(10^{-4}\). Wavelets M1 is coefficient thresholding, while Wavelets M2 is K-Means clustering.

<table>
<thead>
<tr>
<th>Method</th>
<th>TAE*</th>
<th>PSNR</th>
<th>SNR</th>
<th>MSE</th>
<th>SSIM</th>
<th>MS-SSIM</th>
<th>UQI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>318.5</td>
<td>29.09</td>
<td>35.08</td>
<td>735.8</td>
<td>0.8991</td>
<td>0.9029</td>
<td>0.9001</td>
</tr>
<tr>
<td>Random</td>
<td>529.11</td>
<td>27.84</td>
<td>32.53</td>
<td>1358</td>
<td>0.8557</td>
<td>0.9011</td>
<td>0.8972</td>
</tr>
<tr>
<td>Entropy</td>
<td>17.007</td>
<td>19.60</td>
<td>25.04</td>
<td>878.3</td>
<td>0.8921</td>
<td>0.8966</td>
<td>0.8948</td>
</tr>
<tr>
<td>Wavelets(M1)</td>
<td>57.87</td>
<td>23.07</td>
<td>29.09</td>
<td>1832</td>
<td>0.8971</td>
<td>0.9021</td>
<td>0.8989</td>
</tr>
<tr>
<td>Wavelets(M2)</td>
<td>220.9</td>
<td>29.99</td>
<td>34.17</td>
<td>517.3</td>
<td>0.9062</td>
<td>0.9044</td>
<td>0.9022</td>
</tr>
<tr>
<td>Unique Floats</td>
<td>1086.1</td>
<td>20.70</td>
<td>25.17</td>
<td>13802</td>
<td>0.8524</td>
<td>0.8989</td>
<td>0.8847</td>
</tr>
<tr>
<td>NNTF</td>
<td>611.3</td>
<td>25.99</td>
<td>30.56</td>
<td>1984</td>
<td>0.8989</td>
<td>0.9071</td>
<td>0.9038</td>
</tr>
</tbody>
</table>

4.3. Ocean modeling dataset

The ABUMIP Land ice modeling dataset [SPS+20] represents a simulation from the CMIP6 Ice Sheet Model Intercomparison Project (ISMIP6) [NPL+16] and was run using the MPAS Albany Land Ice (MALI) simulation code. ABUMIP investigates an extreme scenario where all ice shelves around Antarctica are removed instantaneously and prevented from reforming over a period of 500 years. The ABUMIP experiment is climatologically unrealistic. However, it provides an estimate for the upper-bound response of the Antarctic ice sheet to the loss of its ice shelves. For this model, MALI uses a mesh with 2 km resolution in dynamically important areas near the coast and coarsens to 30 km resolution in the slow moving ice sheet interior, with a total of 1.8 million grid cells in the horizontal over 200 time-steps. The mesh uses ten vertical layers preferentially concentrated near the ice sheet base, where vertical shearing tends to be greatest. The simulation discussed here was run on about 6000 processors at the National Energy Research Scientific Computing Center (NERSC).

5. Conclusions

In this paper, we’ve shown initial results in keyframe selection methods for explorative visualization. We introduce a set of evaluation criteria that considers statistical and image quality metrics for keyframe quality evaluation. We present a new method, NNTF, that produces the best visual quality metrics compared to the other keyframe selection methods. Specific to this dataset, we found all methods to generally perform well in image-quality metrics (>0.89) and NNTF to excel above the rest. This can be attributed to slowly evolving numerical data in the original dataset, where hardly significant changes occur compared to the asteroid impact dataset.

6. Acknowledgements

This research was funded by the LANL Laboratory Directed Research and Development (LDRD) grant 20190020DR and Los Alamos National Laboratory Institutional Computing Program, supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001.
References


[KB09] Kolda T. G., Bader B. W.: Tensor decompositions and applications. SIAM review 51, 3 (2009), 455–500. 3


