https://doi.org/10.2352/EI.2026.38.1.VDA-351

Efficient Selection of Salient Timesteps in Scientific Simulations

Roxana Bujack, Los Alamos National Laboratory, Los Alamos, NM, USA
Jesus Pulido, Los Alamos National Laboratory, Los Alamos, NM, USA
Manish Bhattarai, Los Alamos National Laboratory, Los Alamos, NM, USA
David H. Rogers, Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract

Identifying key timesteps in spatio-temporal datasets is essential for shaping the story that a simulation tells. The selected timesteps act as anchors for visualization, guiding parameter choices for rendering, animation, and analysis. While many sophisticated selection methods have been proposed, we show that the field has often leaned toward unnecessary complexity. In this work, we provide a survey of existing timestep selection strategies, illustrating their limited ability to balance quality and efficiency. Building on these insights, we introduce a deliberately simple approach based on greedy local search. Starting from uniformly spaced candidates, we iteratively shift selections to minimize reconstruction error under interpolation. Despite its simplicity, this method consistently yields high-quality subsets, enabling effective parameter tuning and exploratory visualization while achieving significantly lower computational cost than more elaborate techniques. Through quantitative comparisons across datasets and error metrics, we demonstrate that this purposeful simplicity can provide a better trade-off between quality and runtime than existing, more complex alternatives.

Introduction

As processor speeds and parallelism levels continue to rise, the significant costs associated with storage and the slow rate of data movement have emerged as principal obstacles for fully employing the computational capabilities of large-scale parallel machines. These problems are particularly acute in scientific domains such as climate modeling, nuclear reactor simulation, and combustion engine design, where the sheer volume of data produced by time-varying simulations compels researchers to limit the amount of data that is stored. The inability to analyze every timestep means that scientists are often confined to examining only a subset of data at predetermined intervals, chosen by simplistic heuristics. This selection process becomes problematic when the simulated phenomena exhibit complex behaviors at unpredictable points in time, leaving the optimal selection of timesteps for analysis and visualization as a persistent question.

In response to this challenge, various strategies have been devised to identify key timesteps in time-varying datasets. Some methods use user judgment by presenting an overview of the data in a two-dimensional layout, enabling users to pinpoint significant timesteps for closer examination. This user-centric approach has been realized through different visual representations, including spreadsheet-like layouts for time series data, storyboard-like arrangements of images, and time histogram techniques that illustrate data distribution changes over time. However, direct user intervention is not always feasible, especially for applications requiring data compression or in-situ analysis, necessitating automated solutions. Among these, algorithms that cluster similar timesteps or identify those with significant changes from their predecessors have shown promise. However, these methods typically focus on local timestep data, potentially overlooking the broader context provided by the entire time sequence, which can lead to the selection of a suboptimal set of timesteps when faced with a limit on the total number to be chosen.

Furthermore, the evolution of supercomputing capabilities has enabled scientists performing simulations to generate and analyze data at unprecedented scales, often resulting in vast numbers of high-spatial-resolution timesteps that accumulate to form extensive datasets. The task of efficiently processing and visualizing information derived from these comprehensive data collections poses a significant challenge, particularly as the volume of data stored can range from terabytes to petabytes. The goal is to refine the process of developing and viewing imagery from these full-spatial-resolution timesteps, enhancing the ability to derive meaningful insights from the data generated by simulations. This concern underscores the need for innovative approaches to data management and analysis in the era of exascale computing, where the volume and complexity of data far exceed traditional simulation outputs.

In this study, our focus is on selecting a subset of significant timesteps—referred to as *salient timesteps*—that effectively encapsulate the essence of an entire time series. This selection aids in establishing visualization parameters for processing the full series, acts as keyframes highlighting pivotal moments in a simulation, and facilitates the efficient sharing of crucial timesteps for deeper analysis with collaborators.

We evaluate several techniques, including an unsupervised machine learning method based on non-negative Tucker factorization (NNTF) [1]. All techniques are 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 with which it is most strongly associated, and tag that timestep as a keyframe. To compare methods, we define sets of evaluation criteria to measure the quality of selected key timesteps by measuring reconstruction error. In contrast to existing techniques, we use a simple but highly efficient method that employs local optimization to minimize the reconstruction error. Despite its simplicity, it is superior to all existing approximate approaches and significantly faster than exhaustive search.

For simplicity, we assume uniform-length (in time) timesteps, but conclude with an indication of how our approach readily generalizes to varying-length timesteps.

Related work

Even though it is only a side note in their paper about transfer-function design, Akiba et al. group similar timesteps and choose one representative timestep from each group [2]. One can think of their method as hierarchical clustering of the histograms of neighboring timesteps using the L^2 norm as measure of difference. They then select the midpoint of each interval as the representative.

Similarly, Wang et al. use salient timestep extraction as a small component of their paper on classification of time-varying volumetric data [6]. They evaluate an importance function based on the conditional entropy of a timestep (or blocks of it) in relation to its neighbors in time. They then segment the whole time series into intervals with the same accumulated importance and select representatives as the salient timesteps. They always include the first timestep and choose the representative from each interval that maximizes conditional entropy compared to the previous representative.

Ma et al. discuss the significance of optimizing the visualization workflow through the lens of scientific storytelling, emphasizing that identifying critical moments in a simulation—termed keyframes—can enhance the production visualization process [10]. Several methods have been developed to identify these keyframes, which are reviewed in the remainder of this section.

For data-intensive simulations, identifying essential keyframes or timesteps is crucial for both post-analysis and in-situ analysis. Meyers et al. explore the concept of selecting salient timesteps for in-situ applications, assuming a continuous flow of timesteps [9]. Their approach involves deciding on the retention of each timestep as it arrives by comparing it to a piecewise-linear model. This comparison continues until a timestep deviates sufficiently from the model's prediction, prompting the storage of this new timestep and the initiation of a new model. They further introduce the novel approach of unique-float binning to extract keyframes as timesteps within a simulation.

Frey and Ertl begin by randomly selecting one timestep from each of a set of regular partitions of the entire time series [7]. As the process unfolds, it incorporates more timesteps through random sampling based on a probability distribution that favors choosing timesteps from longer intervals. This approach is non-deterministic and aims to efficiently manage large datasets by relying on random sampling, but without taking into account the entire dataset or the timesteps that are omitted. They use the Wasserstein distance as a metric of similarity between timesteps, which they compute by connecting a discrete subset of samples of the volumes into a graph via Delauney tesselation and applying a minimum-cost flow-graph algorithm.

Tong et al. tackle the challenge of extracting the optimal K timesteps from a dataset that changes over time, with K being predetermined by the user [3]. They employ dynamic time warping and dynamic programming to identify the sequence of K timesteps with the least cumulative cost of being warped to the original dataset.

Zhou et al. [4] extend the work by Tong et al. They select the most representative timesteps by minimizing the difference in the amount of information from the original data using information-theoretic variation-of-information and dynamic programming. In contrast to earlier work, they provide a novel chart as a storyboard of the data to guide efficient selection of the needed number of timesteps.

Porter et al. use an unsupervised machine learning approach, specifically an autoencoder, to generate a feature set from all available timesteps, and then select keyframes based on this feature set [8].

Pulido et al. use a broad selection of different approaches to select salient timesteps ranging from simple methods, such as uniform, random, or maximum entropy sampling, to very sophisticated methods [1]. The complex approaches include a non-negative tensor decomposition [11, 12], wavelets with the Haar basis function facilitating a multi-scale wavelet decomposition [13], and wavelets combined with K-means clustering [14].

Wu et al. [5] combine time warping through dynamic programming [3] and in-situ piece-wise linear interpolation [9] to select timesteps in an online streaming in-situ fashion, offering efficient computing time and memory usage.

A summary of existing work on salient timestep detection is shown in Table 1. Given are a set of methods for in-core timestep selection [2, 6, 7, 3, 4, 8, 1] and a set for out-of-core timestep selection [9, 5]. Our approach is in-core. Further, we have a set of methods that use dynamic programming to find the optimal solution [3, 4, 5] with a very expensive pre-processing step, and a set of methods that use a heuristic to find a good solution quickly [2, 6, 9, 7, 8, 1]. Our approach falls in the latter category.

We consider our approach most closely related to the work by Akiba et al., Wang et al., Frey and Ertl, and Pulido et al., who all determine, in core, good solutions in a deterministic, algorithmic way without the combinatoric growth in complexity inherent to the globally optimal solutions. In this paper we conduct a thorough comparison of our heuristic and all of theirs and show that ours generates superior results for the metrics and datasets suggested by Pulido et al.

Algorithm

Our algorithm is motivated by how the quality of the selection of salient timesteps has been evaluated by existing methods. Typically, the full simulation is used as ground truth and an approximation is recovered using linear interpolation on just the selected salient timesteps. Then the quality of the selection is determined as a norm of the difference of the two series, where various candidates exist for the choice of the norm [1].

Based in this, we chose to develop a straightforward method that minimizes this norm directly to select the set of timesteps that best describes the overall behavior of the simulation. The idea is very simple: we start by selecting initial times equidistantly spaced and then we shift them forward or backward, if that decreases the overall error, until no more changes occur.

If we select $n \in \mathbb{N}$ salient timesteps from a simulation v with $N \in \mathbb{N}$ timesteps total, we denote the selected timesteps by $\tilde{t}_i, i = 1,...,n$, the actual timesteps by $t_j, j = 1,...,N$, and the reconstructed state of the simulation by $\tilde{v}$.

Reconstruction: For each timestep $t \in [\tilde{t}_i, \tilde{t}_{i+1}] = [t_j, t_k]$ between two selected timesteps $\tilde{t}_i = t_j < \tilde{t}_{i+1} = t_k$, where j,k range from 1 to the total number of timesteps of the simulation $N \in \mathbb{N}$, we reconstruct the approximate timesteps using linear interpolation for each pixel location $x \in \mathbb{R}^3$:

$$\tilde{v}(x,t) = \frac{v(x,t_j)(t-t_k) + v(x,t_k)(t_j-t)}{t_k - t_j}. \quad (1)$$

	In core	Out of core
Exact	Tong et al. [3], Zhou et al. [4]	Wu et al. [5]
Heuristic	Akiba et al. [2], Wang et al. [6], Frey and Ertl [7], Porter et al. [8], Pulido et al. [1]	Meyers et al. [9]

Table 1: In summary, the existing work on salient timestep detection can be categorized as in-core and out-of-core, and exact and heuristic, approaches. Our method falls into the largest group—in-core heuristic approaches.

Our simulations have grids that do not change over time. In the case of an evolving grid, we suggest determining the reconstructed value using an additional spatial linear interpolation step.

We do not require that the first and last timestep be chosen. The timesteps $t \in [1, \tilde{t}_1 - 1]$ before the first and $t \in [\tilde{t}_n + 1, N]$ after the last selected timestep are reconstructed as identical copies of their nearest neighbors:

$$\tilde{v}(x,t) = \begin{cases} v(x,\tilde{t}_1) & \text{if } t \in [1,\tilde{t}_1], \\ v(x,\tilde{t}_n) & \text{if } t \in [\tilde{t}_n, N]. \end{cases} \tag{2}$$

Our method is agnostic to the underlying reconstruction method. It could for example be used with cubic interpolaton, TSR-TVD [15], or HyperINR [16]. We use linear interpolation for the reconstruction in the experiments presented here to be compatible with previous work [1].

Initialization: We start with a uniform timestep selection. Typically, in uniform sampling, all intervals between chosen timesteps have the same size. However, this causes the contribution of error by the first and the last interval to be larger on average than that of the inner intervals because nearest-neighbor interpolation, which is used for them, is only of zeroth-order accuracy, while linear interpolation is of first order. To compensate for this, we initialize the intervals on both ends to be half the size of the internal ones.

Optimization: We compute the accumulated error for all $n+1$ intervals before, between, and after the selected timesteps by adding the reconstruction error $\|\tilde{v}(t) - v(t)\|$ of all timesteps t that fall into the respective interval. We then iterate through all selected timesteps and adjust each selection by moving one step forward or backward if that decreases the overall error. If such an adjustment is accepted, the stored location and error are updated. This process is repeated as long as changes occur.

Acceleration: When we accept the adjustment of one selected timestep, this changes only the accumulated errors in the two adjacent intervals. As a result, if a selected timestep and both of its neighbors have not been changed in the previous iteration, it is also not going to change in the next iteration because all numbers that go into this decision are identical. We use this observation to accelerate the algorithm: we use a boolean array with an entry for each selected timestep that is set to true if it gets changed, and to false otherwise, in each iteration. Then we only treat the selected timesteps that either have been changed or that have a neighbor that has been changed in the previous iteration.

Convergence Behavior

The algorithm is greedy in nature and is not guaranteed to converge to the optimum, and its results depend on its initialization. It is guaranteed to converge, though, because it monotonically decreases the error in each timestep.

The worst-case run time occurs if the optimal salient timesteps would be exactly the first n or the last n timesteps. In these cases, the algorithm would need to move all uniformly-spaced initially-selected timesteps to one end, which would take a number of steps asymptotic to $\Theta(Nn)$. For each step, the metric needs to be evaluated for all N timesteps, which requires N times the number of pixels operations.

For the average behavior, we assume no temporal prior and model salient timestep locations as i.i.d. uniform over the time interval. We note that in applications with temporally biased or clustered events, this assumption may not hold. In such cases convergence may require more steps, potentially approaching the worst case. To find the average asymptotic behavior for a uniform model, we need to sum the pairwise absolute differences between the initial set and assumed i.i.d. random locations. Optimal transport theory shows that the expected 1D Wasserstein distance of order 1 between two independent and identically distributed samples is on the order of $n^{-\frac{1}{2}}$ per point [17]. Therefore summing over n points and scaling from $[0,1]$ to $[0,N]$ gives total of $N\sqrt{n}$ steps on average.

Results

In this section, we describe the experiments that we performed to evaluate our algorithm, and their results.

Evaluation Criteria

The evaluation of the chosen keyframes' quality involves reconstructing the entire temporal dataset using only these keyframes. Linear interpolation towards the closest frame is used for filling in the timesteps between the selected keyframes, as described above. For comparability to the work of Pulido et al., we use $n = 19$ selected timesteps.

Analogously, the assessment of the completely reconstructed temporal dataset from the selected keyframes is conducted using a collection of traditional statistical metrics and image quality algorithms.

We evaluate the metrics for each timestep and then produce the final assessment of the quality of the selection by averaging over the entire dataset. Note that this averaging can cause metrics that are typically related monotonically, such as PSNR and MSE, to not be monotonically related.

In the following definitions, $v(i)$ denotes the original scalar value at cell i and $\tilde{v}(i)$ the reconstructed value, for i ranging over all N cells in one timestep of the dataset.

The Total Absolute Error (TAE) is defined as

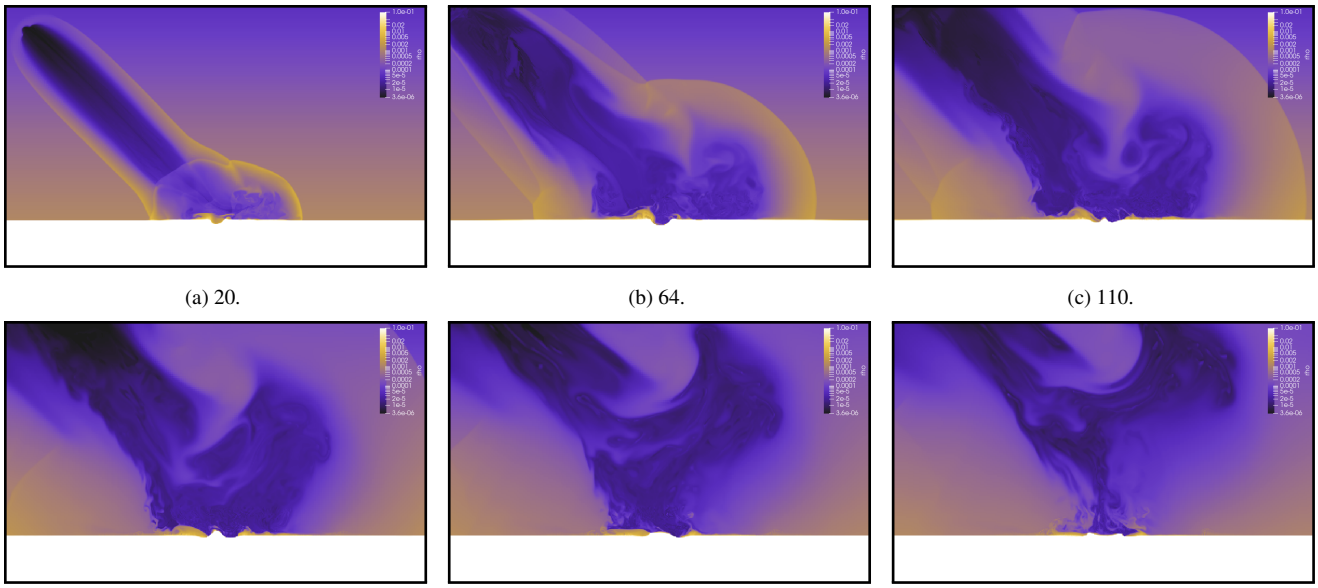
$$TAE = \sum_{i=1}^{N} |v(i) - \tilde{v}(i)|, \tag{3}$$

which is simply the L^1 norm of the difference of original and reconstructed data.

The Mean-Square Error (MSE) [18] is calculated as

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (v(i) - \tilde{v}(i))^2, \tag{4}$$

which is simply the square of the L^2 norm of the difference.



(a) 20. (b) 64. (c) 110.

(d) 157. (e) 204. (f) 247.

Figure 1: The result of our method when tasked to select six salient timesteps of the Deep Water Impact Ensemble dataset.

The Peak Signal-to-Noise Ratio (PSNR) [18] is given by

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_v^2}{MSE}\right), \tag{5}$$

where $MAX_v = \max_{i,t} v(i,t)$ represents the maximum possible value of the dataset across all timesteps t and all cells.

The Signal-to-Noise Ratio (SNR) is expressed as

$$SNR = 10 \cdot \log_{10}\left(\frac{\sum_{i=1}^{N} v(i)^2}{\sum_{i=1}^{N} (v(i)-\tilde{v}(i))^2}\right). \tag{6}$$

The Structural Similarity Index Measure (SSIM) [19] is defined by

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \tag{7}$$

where x and y are windowed signals of the original and reconstructed datasets, μ represents the mean over all scalar values, σ^2 the variance, σ_{xy} the covariance, and c_1, c_2 are constants to stabilize division with weak denominator.

The Multi-Scale Structural Similarity Index Measure (MS-SSIM) [20] extends SSIM over multiple scales:

$$MS-SSIM = \prod_{j=1}^{M} SSIM_j^{\alpha_j}, \tag{8}$$

where M is the number of scales, $SSIM_j$ is the SSIM at the jth scale, and α_j are the weights for each scale.

The Universal Quality Index (UQI) [21] is determined by

$$UQI = \frac{4\sigma_{xy}\mu_x\mu_y}{(\sigma_x^2 + \sigma_y^2)(\mu_x^2 + \mu_y^2)}, \tag{9}$$

where, as before, σ_{xy} is the covariance of the original and reconstructed datasets, μ_x and μ_y are their means, respectively, and σ_x^2 and σ_y^2 are their variances.

There is a distinct difference between statistical and image quality algorithm metrics. While traditional statistical metrics such as TAE, MSE, SNR, and PSNR measure differences between the data points of the original and reconstructed datasets, image quality metrics including SSIM, MS-SSIM, and UQI assess similarities between timesteps. Unlike statistical metrics, image quality metrics are normalized, with values ranging from 0 to 1, which simplifies evaluation and comprehension across different types of datasets. We use the same metrics as Pulido et al. for a fair comparison to their results, but our method can be used in combination with other metrics, such as LPIPS [22] and DreamSim [23].

Alternatives

For an exhaustive comparison, we have used all timestep selection methods from the work of Pulido et al. We further added the method suggested by Akiba et al. [2], an adapted version of Akiba's, Wang et al.'s [6], and an adapted version of the method suggested by Frey and Ertl [7].

Basics: Let N be the number of timesteps in the series and n the number of salient timesteps to be selected. The simple method **Regular** corresponds to the basic uniform timestep selection $i\left\lceil\frac{N}{n}\right\rceil$ as performed by Pulido et al., **Adapted Regular** is our initialization as described above, **Random** is a repeated independent random selection from a uniform distribution over the interval $[0, N-1]$, and **Entropy** selects the timesteps with the highest entropy,

$$H(X) = -\sum_{x \in X} p(x)\log_2(p(x)), \tag{10}$$

until the desired number n is achieved.

Wavelets: Wavelets in general have several vanishing moments (minima and maxima). This property allows for a sparse but accurate representation of an input dataset with only a small number of coefficients with respect to a wavelet basis. A signal is decomposed through multiple steps, i.e. multi-resolution, involving "folds" at the largest scale until it reaches the smallest scale, i.e.,

Method	TAE	PSNR	SNR	MSE*	SSIM	MS-SSIM	UQI
Regular	4598	0.015	48.084	39.742	0.999718	0.999879	0.820540
Adapted Regular	4407	0.014	48.438	40.096	0.999742	0.999913	0.825735
Random	9926	0.097	45.093	36.718	0.999052	0.999607	0.762238
Entropy	44011	0.719	31.936	23.594	0.995602	0.996703	0.449596
Wavelets (M1)	25374	0.340	35.894	27.552	0.997264	0.998214	0.577396
Wavelets (M2)	6090	0.023	46.231	37.888	0.999446	0.999841	0.779228
Unique Floats	11584	0.339	42.345	34.003	0.998803	0.999603	0.717829
NNTF	8112	0.064	45.202	36.760	0.999309	0.999734	0.773684
Akiba TAE	5882	0.028	46.890	38.548	0.999556	0.999853	0.801945
Akiba SSIM	11109	0.111	42.007	33.666	0.998873	0.999546	0.725777
Adapted Akiba TAE	4397	0.012	48.310	39.968	0.999764	0.999918	0.821436
Adapted Akiba SSIM	5078	0.019	46.994	38.652	0.999692	0.999895	0.806484
Wang	5033	0.019	47.702	39.360	0.999667	0.999845	0.814679
Adapted Frey TAE	4476	0.014	48.274	39.932	0.999748	0.999917	0.819859
Adapted Frey SSIM	4443	0.014	48.250	39.908	0.999759	0.999915	0.823714
Ours TAE	4287	0.012	48.803	40.461	0.999763	0.999917	0.824787
Ours SSIM	4323	0.012	48.512	40.170	0.999773	0.999919	0.823742

Table 2: Results for the keyframe selection methods on the deep-water impact dataset. The metrics reported are averaged over all timesteps. MSE* is $\times 10^{-3}$. Blue cells indicate the best performers, light blue cells the second best.

cell size. We compute the bi-orthogonal spline wavelet transform using Matlab's bior1.1 basis for each timestep.

As suggested by Pulido et al., the method **Wavelets (M1)** involves discarding all coefficients within the three finest folds, preserving only the largest, most important features. The salient timesteps are then chosen as the ones with the largest L^1 norm of the remaining coefficients.

The method **Wavelets (M2)**, also by Pulido et al., uses a different approach. Rather than removing coefficients, k-means clustering with $k = n$ is used on all non-zero wavelet coefficients from the transforms of all individual timesteps simultaneously. For all elements in a cluster, they compute the average distance from the centroid to all coefficients that belong to one same timestep. Then, for each of the n clusters, they select the timestep that exhibits the smallest average distance as the representative to identify the most distinctive signals, i.e., features, of a dataset.

Unique Floats: Unique Floats is a simple method used for unique dataset selection [1]. By analyzing ensembles of datasets, a count is conducted on each timestep of an ensemble to tally of the number of unique floating-point values. Very distinctive and dynamic timesteps will tend to have a large count of unique floating point values compared to a timestep in a more static region of the dataset. We add the n timesteps that exhibit the highest number of unique floats.

NNTF: We employed Non-negative Tucker-1 Factorization (NNTF) to extract keyframes from the tensor $\mathbf{X}(t,x,y,z)$, where $z = 1$ for 2D data. First, the tensor was reshaped into a matrix $\mathbf{X}(t, x*y*z)$. We then applied the non-negative matrix factorization with latent feature estimation approach termed NMFk [24, 25] to factorize this matrix, yielding factor matrices $\mathbf{W} \in \mathbb{R}^{t\times k}$ and $\mathbf{H} \in \mathbb{R}^{k\times(x*y*z)}$ for automatically estimated k extracted keyframes (see Pulido et al. for details). This factorization was performed using the distributed NMFk Python library pyDNMFk [26, 27]. The factor matrices hold latent feature information $\mathbf{W}_{i,:}$ that provides the time activation for each keyframe, while $\mathbf{H}_{:,j}$ encodes the flattened keyframes. To recover individual keyframes, we apply the reshape operation reshape$(\mathbf{H}_{:,j}, (x,y,z))$.

Akiba et al.: Akiba et al. iteratively group similar timesteps, analogously to hierarchical clustering, and choose one representative timestep from each group [2]. As the measure of similarity, they use the histograms of neighboring timesteps and the L^2 norm of their difference. It is possible to replace the L^2 norm with any of the other metrics. Finally, they select the midpoint of each interval as the representative. The numbers presented in our benchmark correspond to 100 bins for each histogram. We used Matlab's *histcounts* for the generation of the histograms. We then find the pair of consecutive histograms with the smallest difference in the L^2 norm, average the two histograms, replace the first histogram of the pair with the average and remove the second one. Finally, we update the number of histograms and record the midpoints of the merged intervals. We have implemented their method using varying numbers of bins and found that the number not to significantly change the performance. We refer to this method as **Akiba**.

Since Akiba et al. had envisioned their timestep selection to be used specifically for the generation of transfer functions based on the histograms, for them the histograms actually contain all the information that they care about. It is therefore fair to imagine that they would have adapted their method to run on the actual timesteps instead of their histograms given our problem formulation. We have therefore adapted their method slightly by allowing the algorithm to take the difference using any of our supported norms and similarity measures of the entire timesteps and added it to our benchmark suite under the name **Adapted Akiba**.

Wang et al.: We have implemented the salient timestep extraction of Wang et al. [6] They evaluate an importance function $A(x,t)$ for spatial blocks of timesteps based on a weighted sum of its conditional entropy in relation to its neighbors in time and then sum up the importance functions over the individual blocks to derive an importance function for the entire timestep, $A(t) = \sum_x A(x,t)$. The conditional entropy is defined as $H(X|Y) = H(X) - I(X;Y)$, where $H(X)$ is the entropy, and

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{11}$$

Method	TAE*	PSNR	SNR	MSE	SSIM	MS-SSIM	UQI
Regular	319	735.800	40.824	35.080	0.990240	0.994327	0.991750
Adapted Regular	197	532.313	43.881	35.387	0.996360	0.998674	0.997268
Random	529	1358.000	32.330	32.330	0.990240	0.994327	0.991750
Entropy	1730	8783.000	33.062	24.560	0.986651	0.990903	0.989307
Wavelets (M1)	579	1832.000	37.594	29.090	0.991615	0.996095	0.993399
Wavelets (M2)	221	517.700	42.665	34.170	0.995679	0.998505	0.996755
Unique Floats	1069	3802.000	33.877	25.370	0.986896	0.992867	0.989228
NNTF	611	1984.000	39.094	30.560	0.991188	0.996112	0.993026
Akiba TAE	178	418.929	43.120	34.620	0.996555	0.999092	0.997531
Akiba SSIM	238	576.449	42.733	34.233	0.995519	0.998235	0.996544
Adapted Akiba TAE	179	400.204	43.423	34.925	0.996387	0.998888	0.997356
Adapted Akiba SSIM	211	552.789	42.544	34.042	0.996289	0.998886	0.997316
Wang	388	905.829	43.287	34.792	0.992258	0.995352	0.993360
Adapted Frey TAE	192	469.931	43.642	35.146	0.996247	0.998692	0.997233
Adapted Frey SSIM	189	466.028	43.726	35.229	0.996406	0.998768	0.997326
Ours TAE	181	444.197	44.069	35.574	0.996434	0.998774	0.997364
Ours SSIM	189	499.139	43.985	35.490	0.996471	0.998752	0.997354

Table 3: Results for the keyframe selection methods on Antarctic Ice-sheet ocean dataset. The metrics reported are averaged over all timesteps. TAE* is $\times 10^6$. Blue cells indicate the best performer, light blue cells the second best.

is the mutual information from information theory. The probabilities $p(x) \in [0,1]$ are defined as the probability for a data value to fall into a specific histogram bin, or alternatively, as the count of values in the respective histogram bin divided by the total number of points in the dataset. Analogously, the joint probability $p(x,y) \in [0,1]$ is the probability for a pair of data values to fall into a bin of the joint histogram. We compute the joint histograms using Matlab's *histcounts2* and compute the importance using Matlab's *calcCondEntropy*. We calculate the target value, which is the mean of the importance, iterate through the timesteps, and check if the current sum of importance has reached or exceeded the target value. If so, we are done with the interval and reset the counter. In our implementation, we used the simplest setting of just a single block per dataset, and set the number of bins in the histograms to 100. Wang et al. also mention adding other properties into the histograms, such as the gradient or domain-specific attributes, but we use just the basic histogram of the scalar values. For the weights, we used the scheme Wang et al. provide in Fig. 1 of their paper, i.e., the importance function for the dataset X_t at timestep t takes the form

$$\begin{aligned} A(t) =& I(X_t;X_{t-3}) + 2I(X_t;X_{t-2}) + 4I(X_t;X_{t-1}) \\ &+ 4I(X_t;X_{t+1}) + 2I(X_t;X_{t+2}) + I(X_t;X_{t+1}). \end{aligned} \tag{12}$$

As Wang et al. suggest, we then segment the whole time series into intervals with the same accumulated importance and select representatives as the salient timesteps. We always add the first timestep and then choose the representative from each following interval that maximizes the conditional entropy in relation to the previously selected representative.

Frey and Ertl: Frey and Ertl's method first selects one timestep randomly from each of a set of regular partitions of the time series [7]. Later it incorporates more timesteps through random sampling based on a probability distribution that favors choosing timesteps from longer intervals. We have adapted the method in that we 1) use only a single interval for the first step, and 2) remove the randomness. We give the method foresight in the sense that we have it iteratively choose the timestep that most reduces the error instead of a random one. This of course violates the original idea of Frey and Ertl to efficiently manage large datasets by avoiding reading every timestep, and makes it very slow. We do not suggest doing this in practice, but only run it in this way to produce a deterministic best-case scenario to compare against. We start with an empty set of selected timesteps and iteratively fill it until we have reached the number of desired salient timesteps. For each iteration, we loop through all unselected timesteps, choose one to add to the current selection, generate the reconstructions, and compute the error with respect to the selected metric. The timestep that reduces the error the most is the one that is permanently added to the set. Originally, Frey and Ertl used the Wasserstein distance as their metric of choice, while we ran it with all of the metrics described above. We refer to this method as **Frey**.

Deep-water Impact Dataset

We test our method on the same datasets as Pulido et al. to allow comparison to their methods. These datasets have very different temporal behaviors, which makes them good candidates for evaluation. The first changes rapidly but uniformly and the second moderately at first and very slowly later. The Deep Water Impact Ensemble dataset [28] consists of simulations designed to explore the likelihood of asteroid impacts in deep ocean waters causing tsunamis. These simulations were conducted using xRage [29], a parallel multi-physics Eulerian hydrodynamics code developed by Los Alamos National Laboratory. The simulation uses adaptive mesh refinement (AMR) to vary resolution across different areas of the computational grid. The specific simulation we focus on is the yB31 ensemble member, which simulates a 250-meter asteroid impacting at a 45-degree angle and exploding in the air 5 km above the ocean surface. Visualization data was generated using ParaView Catalyst [30]. For our analysis, we used the same center Z-slice of the resampled dataset of a uniform grid measuring 460x280x240, comprising 269 timesteps, as did Pulido et al.

The performance of the selected keyframes with respect to all quality metrics is summarized in Table 2. Our algorithm took

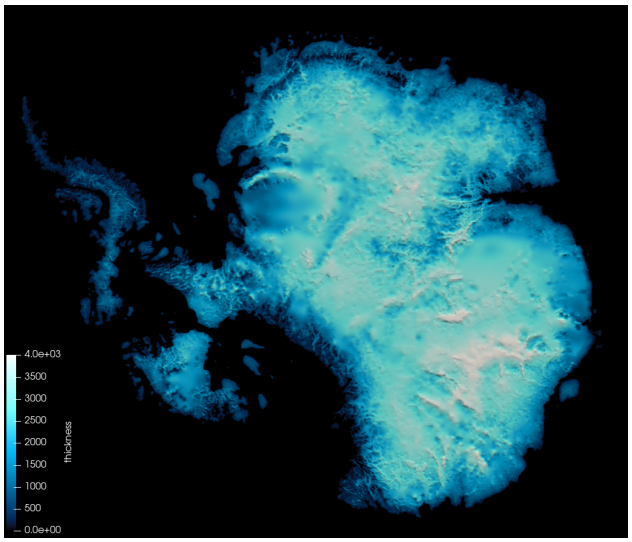


Figure 2: Early time of the ABUMIP dataset showing ice thickness with all floating ice removed.

12 iterations and 11.4 seconds to converge for TAE, and 10 iterations and 62.3 seconds for SSIM, as the optimization metrics. Its performance is compared to other algorithms in Table 4. The results of our algorithm for $K = 6$ keyframes is visualized in Fig. 1.

Note that the numbers for the statistical metrics are identical to the results reported by Pulido et al., but differ for the image quality metrics in Table 2. In Pulido et al., the inclusion of the selected timesteps themselves, i.e., perfect reconstructions, were skipped in the error accumulation across all methods, while we include them. Because these contribute perfect zeros for the statistical metrics, nothing changes. In contrast, because they contribute perfect number ones for the image quality metrics, the results are different. The reason we included them is that excluding them introduces an unintended bias towards methods that select the same timestep more than once for the image quality metrics alone, which occurred in the results for random sampling and NNTF.

Land Ice Modeling Dataset

The ABUMIP land ice modeling dataset [31] is derived from simulations carried out for the CMIP6 Ice Sheet Model Intercomparison Project (ISMIP6) [32], using the MPAS Albany Land Ice (MALI) simulation software, Fig. 2. ABUMIP explores a hypothetical extreme scenario in which all ice shelves surrounding Antarctica are instantly removed and are not allowed to reform for 500 years. While this scenario is not climatologically plausible, it serves to provide a theoretical maximum response of the Antarctic ice sheet to the complete loss of its ice shelves. In this simulation, MALI employs a variable-resolution mesh that is finely discretized to 2 km in regions near the coast, which are dynamically critical, and expands to 30 km resolution in the less dynamic interior of the slow-moving ice sheet. This mesh comprises approximately 1.8 million horizontal grid cells across 200 timesteps and incorporates ten vertical layers. These layers are strategically densified near the base of the ice sheet to accurately capture vertical shearing effects. The simulation was executed on roughly 6000 processors at the National Energy Research Scientific Computing Center (NERSC).

	Impact		Land ice	
Metric	**TAE**	**SSIM**	**TAE**	**SSIM**
Adapted Regular	0.0041	0.0041	0.0037	0.0037
Random	0.0034	0.0034	0.0035	0.0035
Akiba	0.71	0.71	2.6	2.6
Adapted Akiba	32	690	280	3100
Wang	5.0	5.0	34	34
Adapted Frey	2400	8500	9500	25000
Ours	11	62	90	110

Table 4: Runtimes in seconds for the methods for which we have implementations. The fastest are marked blue and the second fastest light blue.

The performance of the selected keyframes with respect to all quality metrics is summarized in Table 3. Our algorithm took 10 iterations and 90.4 seconds to converge for $K = 19$ keyframes for TAE, and six iterations and 109.29 seconds for SSIM, as the optimization metrics. Its performance is compared against other algorithms in Table 4.

Note that as before the numbers for the image quality metrics differ from the results reported by Pulido et al. by including the original keyframes in the image-metric computations.

Runtimes

Lastly we provide runtimes for the algorithms that we have implemented. We are not able to provide runtimes for all of the methods to which we compare our algorithm because we do not have their implementations; in particular, we do not know the performance of Entropy, Wavelets (M1), Wavelets (M2), Unique Floats, or NNTF. We know their performance with respect to the metrics because they were reported in related work but their runtimes were not provided. All of our runs were performed in serial in MATLAB on a MacBook Pro with the Apple M2 Max chip.

We again stress that Adapted Frey has very different performance than that of the original algorithm suggested by Frey and Ertl [7], which would have been significantly faster. The runtimes for regular and random sampling are dominated by loading the dataset, which is needed to count the number of timesteps N.

Conclusions

Summary: Selecting salient timesteps in spatio-temporal data can be used for data reduction, creating summaries and crafting initial visualizations for animations and overviews, as well as finding optimal parameters for visualization algorithms or transfer functions that work for the entire series, i.e, that produce good visualizations. We have presented an extremely simple, yet highly efficient, strategy that allows for the effective determination of the best salient timesteps. We start with a uniform sampling and refine it by locally adjusting the positions of selected timesteps in the direction that reduces the error.

In contrast to methods that perform exhaustive searches [3, 4, 5], our method does not guarantee the optimal selection of timesteps, but it delivers good solutions in a fraction of the time. While the globally optimal solutions require a pre-processing step that takes hours for simulations with moderately many timesteps, ours takes only from a few seconds up to two minutes. But, as Frey and Ertl have pointed out: “Tong et al.’s selection approach is also excessively expensive in a number of cases (e.g. when choos-

ing relatively few timesteps from a long series)" [7], the combinatoric character of the exhaustive methods does not scale to the resolutions of large scientific simulations.

We have compared our algorithm to a broad set of techniques suggested in related work. Our suggested heuristic uses one specific metric with respect to which it optimizes. We therefore expected it to excel in its respective column, which it did in almost all cases. We ran our method with each of the metrics and present the result of the total absolute error (TAE) and Structural Similarity (SSIM) as representatives. What we did not expect was for each variant of our heuristic to also perform extremely well with respect to the other metrics. Except for a very few cases, each of our runs outperforms every other approach in every other metric. Tables 2 and 3 show that the numbers in the last two rows are better than all others except for the methods by Akiba et al. and the one we adapted from it in the land ice modeling dataset. Examining the deep-water impact results, though, we see that only Adapted Akiba provides competitive results. In contrast, Akiba's original method performed even worse than regular sampling. We will make the source code publicly available upon acceptance.

Discussion: The fact that our method was inferior in some metrics compared to the method suggested by Akiba et al. in the ice sheet dataset reveals a limitation of our method. The ice sheet dataset has a much less uniform behavior than the deep impact dataset. This makes uniform sampling a poor choice for an initial guess and our algorithm is unable to escape the corresponding trough of the local minimum. We can only expect this behavior to worsen if the time series have larger numbers of timesteps.

Another observation is that the error measures that relate to a mathematical norm, i.e., TAE and MSE, are significantly better predictors of the behavior with respect to all measures on average than are the image quality metrics. Representative of this is that for Adapted Akiba, adapted Frey, and our method, the results with TAE perform much better than the results with SSIM (Tables 2 and 3). For example, our method running with SSIM only outperformed our method running with TAE for SSIM (and possibly MS-SSIM), whereas the TAE version outperformed the SSIM version in every other measure. We therefore strongly recommend the mathematical norms over the image quality metrics for salient timestep selection algorithms for scientific simulations.

Our experiments demonstrate that our heuristic, despite its simplicity, achieves a better balance of high-quality outcomes and computational efficiency than current approaches.

Future Work: In future work, we intend to parallelize the algorithm and extend it to sampling 2D and 3D non-uniform and evolving datasets, e.g., AMR. We also want to explore if it can be used for data reduction purposes. Also, we intend to extend the initialization to more sophisticated methods, such as applying the clustering method used by Akiba et al. [2]. Instead of allowing moving only one timestep at a time, we want to use a simulated annealing optimizer to avoid getting trapped in local minima. Finally, we plan on extending our method to run on simulations whose timesteps are not regularly spaced in time by weighting the reconstruction error with the length of the represented time interval and to DICOM-style slice-based data by treating slice indices as proxy timesteps.

Acknowledgments

We greatly thank Kei Davis for his help with the manuscript. We acknowledge the use of ChatGPT for reformulation. This work is published under LLA-UR-24-21625. It was supported by the National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) Program.

References

[1] J. Pulido, J. Patchett, M. Bhattarai, B. Alexandrov, and J. Ahrens, "Selection of optimal salient time steps by non-negative tucker tensor decomposition," in *EuroVis*, 2021.

[2] H. Akiba, N. Fout, and K.-L. Ma, "Simultaneous classification of time-varying volume data based on the time histogram.," in *EuroVis*, vol. 6, pp. 1–8, 2006.

[3] X. Tong, T. Lee, and H. Shen, "Salient time steps selection from large scale time-varying data sets with dynamic time warping," in *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 49–56, 2012.

[4] B. Zhou and Y.-J. Chiang, "Key time steps selection for large-scale time-varying volume datasets using an information-theoretic storyboard," in *Computer Graphics Forum*, vol. 37, pp. 37–49, Wiley Online Library, 2018.

[5] M. Wu, Y.-J. Chiang, and C. Musco, "Streaming approach to in situ selection of key time steps for time-varying volume data," in *Computer Graphics Forum*, vol. 41, pp. 309–320, Wiley Online Library, 2022.

[6] C. Wang, H. Yu, and K.-L. Ma, "Importance-driven time-varying data visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1547–1554, 2008.

[7] S. Frey and T. Ertl, "Flow-based temporal selection for interactive volume visualization," *Computer Graphics Forum*, vol. 36, no. 8, pp. 153–165, 2017.

[8] W. P. Porter, Y. Xing, B. R. von Ohlen, J. Han, and C. Wang, "A deep learning approach to selecting representative time steps for time-varying multivariate data," in *2019 IEEE Visualization Conference (VIS)*, pp. 1–5, 2019.

[9] K. Myers, E. Lawrence, M. Fugate, C. M. Bowen, L. Ticknor, J. Woodring, J. Wendelberger, and J. Ahrens, "Partitioning a large simulation as it runs," *Technometrics*, vol. 58, pp. 329–340, Jul 2016. Publisher: Taylor & Francis.

[10] K. Ma, I. Liao, J. Frazier, H. Hauser, and H. Kostis, "Scientific storytelling using visualization," *IEEE Computer Graphics and Applications*, vol. 32, no. 1, pp. 12–19, 2012.

[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[12] B. Alexandrov, D. DeSantis, G. Manzini, and E. Skau, "Nonnegative canonical polyadic decomposition with rank deficient factors," *arXiv preprint arXiv:1909.07570*, 2019.

[13] P. Campisi, A. Longari, and A. Neri, "Automatic key frame selection using a wavelet-based approach," in *Wavelet Applications in Signal and Image Processing VII*, vol. 3813, pp. 861–872, International Society for Optics and Photonics, Oct 1999.

[14] S. Hasebe, M. Nagumo, S. Muramatsu, and H. Kikuchi, "Video key frame selection by clustering wavelet coefficients," in *2004 12th European Signal Processing Conference*, pp. 2303–2306, Sep 2004.

[15] J. Han and C. Wang, "Tsr-tvd: Temporal super-resolution for time-varying data analysis and visualization," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 205–215, 2019.

[16] Q. Wu, D. Bauer, Y. Chen, and K.-L. Ma, "Hyperinr: A fast and pre-

dictive hypernetwork for implicit neural representations via knowledge distillation," *arXiv preprint arXiv:2304.04188*, 2023.

[17] C. Villani *et al.*, *Optimal transport: old and new*, vol. 338. Springer, 2009.

[18] A. Horé and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, 2010.

[19] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[20] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, pp. 1398–1402 Vol.2, 2003.

[21] Zhou Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.

[22] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

[23] S. Fu, N. Tamir, S. Sundaram, L. Chai, R. Zhang, T. Dekel, and P. Isola, "Dreamsim: Learning new dimensions of human visual similarity using synthetic data," *arXiv preprint arXiv:2306.09344*, 2023.

[24] R. Vangara, E. Skau, G. Chennupati, H. Djidjev, T. Tierney, J. P. Smith, M. Bhattarai, V. G. Stanev, and B. S. Alexandrov, "Semantic nonnegative matrix factorization with automatic model determination for topic modeling," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 328–335, IEEE, 2020.

[25] R. Vangara, M. Bhattarai, E. Skau, G. Chennupati, H. Djidjev, T. Tierney, J. P. Smith, V. G. Stanev, and B. S. Alexandrov, "Finding the number of latent topics with semantic non-negative matrix factorization," *IEEE Access*, vol. 9, pp. 117217–117231, 2021.

[26] M. Bhattarai, G. Chennupati, E. Skau, R. Vangara, H. Djidjev, and B. S. Alexandrov, "pydnmfk: Python distributed non negative matrix factorization with determination of hidden features," 2020. [ONLINE]. `https://github.com/lanl/pyDNMFk`, Last accessed on 2021-04-12.

[27] I. Boureima, M. Bhattarai, M. Eren, E. Skau, P. Romero, S. Eidenbenz, and B. Alexandrov, "Distributed out-of-memory nmf on cpu/gpu architectures," *The Journal of Supercomputing*, vol. 80, no. 3, pp. 3970–3999, 2024.

[28] J. Patchett and G. Gisler, "Deep water impact ensemble data set," *Technical Report LA-UR-17-21595, Los Alamos National Laboratory*, 2017.

[29] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstaedt, K. New, W. R. Oakes, *et al.*, "The rage radiation-hydrodynamic code," *Computational Science & Discovery*, vol. 1, no. 1, p. 015005, 2008.

[30] U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian, and J. Mauldin, "Paraview catalyst: Enabling in situ data analysis and visualization," in *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV2015, (New York, NY, USA), pp. 25–29, Association for Computing Machinery, 2015.

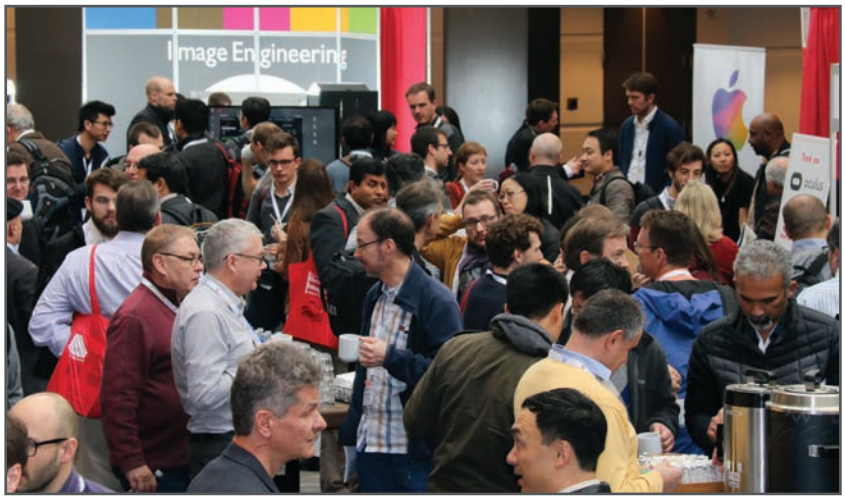
[31] S. Sun, F. Pattyn, E. G. Simon, T. Albrecht, S. Cornford, R. Calov, C. Dumas, F. Gillet-Chaulet, H. Goelzer, N. R. Golledge, R. Greve, M. J. Hoffman, A. Humbert, E. Kazmierczak, T. Kleiner, G. R. Leguy, W. H. Lipscomb, D. Martin, M. Morlighem, S. Nowicki, D. Pollard, S. Price, A. Quiquet, H. Seroussi, T. Schlemm, J. Sutter, R. S. W. van de Wal, R. Winkelmann, and T. Zhang, "Antarctic ice sheet response to sudden and sustained ice-shelf collapse (ABUMIP)," *Journal of Glaciology*, pp. 1–14, sep 2020.

[32] S. M. Nowicki, A. Payne, E. Larour, H. Seroussi, H. Goelzer, W. Lipscomb, J. Gregory, A. Abe-Ouchi, and A. Shepherd, "Ice Sheet Model Intercomparison Project (ISMIP6) contribution to CMIP6," *Geoscientific Model Development*, vol. 9, no. 12, pp. 4521–4545, 2016.

JOIN US AT THE NEXT EI!

electronic IMAGING

Imaging across applications . . . Where industry and academia meet!



• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

ADVANCING IMAGING SCIENCE 75 1947 IS&T 2022 imaging.org