

A Surrogate Visualization Model Using the Tensor Train Format

R. Ballester-Ripoll*, E. G. Paredes*, and R. Pajarola*
Department of Informatics, University of Zürich

Abstract

Complex simulations and numerical experiments typically rely on a number of parameters and have an associated score function, e.g. with the goal of maximizing accuracy or minimizing computation time. However, the influence of each individual parameter is often poorly understood a priori and the joint parameter space can be difficult to explore, visualize and optimize. We model this space as an N -dimensional black-box tensor and apply a cross approximation strategy to sample it. Upon learning and compactly expressing this space as a surrogate visualization model, informative subspaces are interactively reconstructed and navigated in the form of charts, images, surface plots, etc. By exploiting efficient operations in the tensor train format, we are able to produce diagrams such as parallel coordinates, bivariate projections and dimensional stacking out of highly-compressed parameter spaces. We demonstrate the proposed framework with several scientific simulations that contain up to 6 parameters and billions of tensor grid points.

Keywords: multidimensional data, surrogate visualization models, tensor decompositions, simulation parameter spaces

Concepts: •Computing methodologies → Modeling and simulation; Simulation evaluation;

1 Introduction

Parameter settings can critically influence the results of complex simulations and numerical experiments. The set of all possible parameter combinations constitutes a *parameter space*. Interactive exploration and knowledge discovery of such spaces can greatly assist in the understanding of a simulation’s behavior; however, such spaces grow exponentially in size with the number of parameters N . While many approaches exist for sparse data sets, visualizing high-dimensional dense data is more challenging from a computational point of view. In this paper, we tackle this visualization problem under the paradigm of black-box sampling and, in particular, cross approximation (CA) methods for multidimensional arrays (also known as tensors). Tensor decomposition is central to all stages of our pipeline and as of yet has not been used to explore and understand parameter spaces in an interactive visualization application. We propose a framework that:

1. Provides a compact representation of the complete parameter space to operate with. We have chosen the *tensor train* (TT) format, a rather recent decomposition model well-suited for high dimensions: its storage space grows only linearly with the number N of parameters. Complex inter-variable dependence is encoded in the form of *tensor ranks*.

2. Can approximately predict simulation results for one or many combinations of parameters. In this context, the tensor formulation acts as a *surrogate model* for interactive exploration. This is a crucial feature since, often, many different parameter combinations are queried during the visualization process (either because of a high resolution sampling is desired, or because the sheer number of parameters) and computing and evaluating each is too expensive.

The basic features of our framework include parameter tensor sampling at a prescribed accuracy and interactive reconstruction of arbitrary subspaces. The user can select and modify parameter points and subspaces and visualize predictions at real time. These capabilities are efficiently supported by the TT format, even when the dense parameter space contains billions of points. As an evaluation we showcase our framework with some examples of selection queries in parameter spaces using a global-to-local navigation strategy to browse different focus + context visualization diagrams.

Notation Throughout this paper, vectors are denoted with bold lowercase (\mathbf{a}), matrices with bold uppercase (\mathbf{A}) and tensors (multidimensional arrays) with calligraphic letters (\mathcal{A}). A space of N parameters forms a Cartesian grid arising from a tensor product and has size $I_1 \times \dots \times I_N$. For simplicity, sometimes $I = I_n \forall n$ is assumed. The symbol “:” indicates axis-aligned subspaces (free indices); e.g. $\mathbf{A}(i, :)$ is the i -th row of \mathbf{A} . We consider visualization diagrams that can be expressed as a set of M freely-moving indices, usually with $M \leq 3$. For example, a univariate plot is a subspace with $M = 1$, a surface plot (or an overlay of several 1D plots) has $M = 2$, etc.

2 Related Work

2.1 Visualizing Parameter Spaces

The traditional approach for estimating appropriate parameter values for complex simulation models consists in running the experiment with a specific set of parameters, analyze the obtained results, and repeat the simulation with new settings modified according to the simulation objectives. This informed trial-and-error approach requires a deep *a priori* knowledge of the model and a considerable amount of time and user intervention.

Many different strategies for systematic analysis of parameter spaces have been proposed in particular application domains. For example, a visualization library [Nocke et al. 2007] for the exploration of climate simulation models with different comparative techniques. In the context of diffusion tensor imaging, a system for parameter sensibility analysis [Brecheisen et al. 2009] was developed using an unattended pre-computation stage where streamlines for the full space of threshold values are generated first, and later the user explores the results interactively assisted by plots of multiple model features. Ensemble-Vis [Potter et al. 2009] is an interactive framework consisting of a collection of linked overview and statistical displays. The authors state that the combination of multiple linked displays obtains better results in visual data analysis tasks. For the optimal placement of objects in a 3D computer tomography scan, a visual stability analysis tool [Amirkhanov et al. 2010] combined a 3D computer tomography simulation algorithm with Radon space analysis to find the optimal placement on the rotatory

*e-mail: {rballester|egparedes|pajarola}@ifi.uzh.ch

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). © 2016 Copyright held by the owner/author(s).

SA '16 Symposium on Visualization, December 05-08, 2016, Macao

ISBN: 978-1-4503-4547-7/16/12

DOI: <http://dx.doi.org/10.1145/3002151.3002167>

plate. For high-dimensional datasets, the iLAMP method [dos Santos Amorim et al. 2012] allows the interactive selection of points and subspaces in high-dimensional datasets by using a multidimensional projection technique, although it does not provide any automatic way to determine relevant parameter values. Another system for finding patterns in high-dimensional datasets [Anand et al. 2012] used iteratively refined random projections to maximize the visual disparity in different subspaces. A recent interactive system for the segmentation of medical images [Pretorius et al. 2015] works with previews and thumbnails to assist in the estimation of the best segmentation parameters.

Some efforts aim to categorize and classify the field according to a coherent set of tasks and strategies. For example, a recent survey [Kehrer and Hauser 2013] provides a broad compilation of techniques for visual analysis of high-dimensional scientific data according to different facets of the data: spatiotemporal, multivariate (different attributes), multimodal (different data sources), multi-run or ensemble (multiple simulations), multi-model (different simulations of coupled interacting phenomena). The authors roughly follow a previous categorization of the literature [Bertini and Lalanne 2010] identifying common groups of visualization, analysis, and interaction methods.

Another categorization of the parameter space analysis problem [Sedlmair et al. 2014] identifies three key components of the domain (data flows, navigation strategies, analysis tasks) and provides a comprehensive conceptual framework for the characterization of the most relevant methods found in the literature. The authors also note that surrogate models are frequently used to allow interactive exploration of the parameter space, specially in the global-to-local navigation strategy, which was the most widely used strategy in all the examined applications. This strategy consists in generating first a global overview of the parameter space and then allow the refinement and filtering of details on demand.

Additionally, the use of surrogate models can be merged with multidimensional visualization techniques [Chan 2006] to provide meaningful visual representations of the parameter space. ParaGlide [Bergner et al. 2013], for example, is a region-based parameter sensitivity analysis tool which subdivides the parameter space into partitions that represent distinct output behavior. This partitioning helps users in their understanding of qualitative differences among possibly high-dimensional model outputs. Another system [Tatu et al. 2012] for the interactive exploration of specific sets of subspaces in high-dimensional datasets uses an interestingness-guided subspace search algorithm and various similarity functions to guide the users during the visualization. Focusing on high-dimensional scalar functions, Morse-Smale complex-based approximation can be used together with dimensionality reduction techniques to produce a geometric summary of the space [Gerber et al. 2010].

When data is sparse (i.e. point cloud-type data sets), high dimensionality results in high complexity but not necessarily in large space requirements. On the other hand, data sets that are both dense and high-dimensional require vast amounts of memory if stored explicitly. In practice, black-box situations are usual: instead of pre-computing and saving the data, a routine is available to compute any given element on demand. This principle also stands behind compressive sensing, in which signals are assumed to be sparse in some domain and are thus recovered from the transformed space. In this context, we aim for a tool that can be used for acquiring and compactly representing high-dimensional black-box spaces as well as for efficient subspace selection, sample prediction and visualization.

2.2 Tensor Train Decomposition

The singular value decomposition (SVD) has many applications for bivariate data: signal processing, big data problems, dimensionality reduction, etc. Multidimensional data sets, however, possess a much richer structure than their 2D counterparts. Research efforts have produced several higher-order generalizations of the SVD [Kolda and Bader 2009] which are collectively known as *tensor decompositions*. In the context of visualization and visual computing, these methods have been used for interactive volume rendering [Suter et al. 2011; Suter et al. 2013; Balsa Rodríguez et al. 2014], compression [Ballester-Ripoll et al. 2015; Ballester-Ripoll and Pajarola 2015], texture synthesis [Costantini et al. 2008; Wu et al. 2008], denoising [Rajwade et al. 2013], data recovery and inpainting [Chen et al. 2014; Filipović and Jukić 2015], and more. Tensor-based techniques have been shown to outperform other approaches at high compression rates (such as, notably, the wavelet transform [Wu et al. 2008; Suter et al. 2010]). Some approaches for parameter space analysis and exploration leverage tensor decomposition. However, they are usually domain-specific, e.g. [Ballani and Grasedyck 2014] for parameter-dependent differential equations (using the hierarchical Tucker model) and [Liao et al. 2015] for parameters of gene regulatory networks (without an adaptive sampling strategy).

Classical tensor models have a number of drawbacks regarding algorithmic complexity. No stable procedure exists for decomposing a tensor into CANDECOMP/PARAFAC (CP) [Harshman 1970], and determining the exact number of CP ranks is known to be an ill-posed problem. On the other hand, stable algorithms that obtain the Tucker decomposition do exist [de Lathauwer et al. 2000], but the format still requires exponential space R^N with respect to the number of dimensions N (where R is the tensor rank). To overcome these issues, a new decomposition model named tensor train (TT) was proposed by Oseledets [Oseledets 2011]. In the TT model each dimension is encoded with one 3D tensor *core*, so that the total number of elements grows only linearly with the input dimensionality. An element (x_1, \dots, x_N) is obtained as a product of matrices (see Fig. 1): $\mathcal{A}(x_1, \dots, x_N) = \mathcal{A}^{(1)}(:, x_1, :) \cdot \dots \cdot \mathcal{A}^{(N)}(:, x_N, :)$, with $\mathcal{A}^{(k)}(:, x_k, :)$ denoting a $R_{k-1} \times R_k$ matrix for each $x_k = 1, \dots, I_k$ (the first and last being a row and a column vector, respectively).

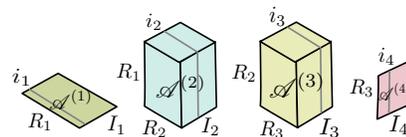


Figure 1: The TT decomposition in the 4D case. The subspaces (a row, two matrices, and a column) that are multiplied together to produce an element (i_1, i_2, i_3, i_4) are highlighted in gray.

If the input data is explicitly available as a full tensor, a TT expression can be obtained by the stable TT-SVD algorithm in $O(NIR^3)$ operations [Oseledets 2011]. It is based on a sequence of singular value decompositions (SVD), and it generates a tensor core at each step. By separating one dimension at a time, the TT format aims to circumvent the *curse of dimensionality*. Several toolboxes and numerous algorithms have been implemented to operate on the TT format.

3 Sampling a Black-Box Parameter Space

3.1 Cross Approximation Algorithms

Cross approximation (CA) has recently emerged as a powerful multidimensional paradigm for efficient data sampling and manipulation in 2 or more dimensions. The main supporting idea comes from the so-called *pseudo-skeleton* decomposition for matrices [Bebendorf 2000; Tyrtshnikov 2000]: a rank- R matrix \mathbf{A} of size $I \times I$ can be exactly recovered from R of its rows and columns. Only $O(IR)$ elements (instead of $O(I^2)$) must be stored to represent the full matrix \mathbf{A} , which can be interpolated from its rows \mathbf{R} , columns \mathbf{C} , and cross entries $\hat{\mathbf{A}}$ as $\mathbf{A} = \mathbf{C}\hat{\mathbf{A}}^{-1}\mathbf{R}$. Many numerical methods become very efficient with the rank- R assumption, since they have to operate on just a subset of the initial I^2 elements. Rows and columns can be determined and sampled adaptively (Fig. 2) based on the *maxvol* principle: a submatrix with maximal volume (i.e. determinant in modulus) will yield the best approximation.

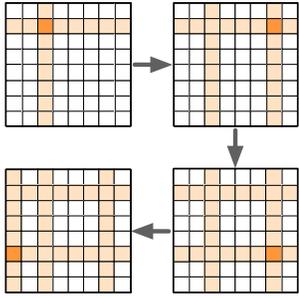


Figure 2: Four steps of a 2D pivoting scheme. A possible heuristic is: before each pivot-centered fiber sampling step, predict the values along the targeted fiber. After sampling it, place the next pivot on the sample that differs the most from its prediction.

This construction has been generalized to N dimensions and successfully blended in with this TT formulation [Oseledets and Tyrtshnikov 2010; Savostyanov and Oseledets 2011], and only $O(NIR^2)$ operations must be computed in order to recover (interpolate) a full I^N tensor [Savostyanov and Oseledets 2011]. Analogously to the 2D case, many useful transformations in a tensor train require manipulating only the cross entries and fibers of the input. CA schemes have been developed also for other tensor models (CP in [Espig et al. 2009]; Tucker in [Oseledets et al. 2008]); we have chosen the implementation of the TT-Toolbox [Oseledets et al.] (open-source, written in Python), since it also includes many manipulation routines for the TT format.

3.2 Sampling by Subspace Traversal

We assume we have a computer simulation or experiment s that depends on $M \geq 3$ parameters and can be discretized onto a domain of size $I_1 \times \dots \times I_N$. The domain of s is a Cartesian grid arising from a tensor product $\mathbf{u}_1 \otimes \dots \otimes \mathbf{u}_N$, where each \mathbf{u}_n is a vector with I_n samples. In our experiments we use equidistant sampling, but this is not mandatory. In order to explore and understand the simulation’s response on this discretized parameter space, manual approaches typically follow this *trial-and-error* procedure:

1. Fix all but a few parameters x_{p_1}, \dots, x_{p_M} , with $M < N$ (usually $M \leq 3$; often $M = 1$).
2. Generate query points by varying the M free parameters. Their space is the M -dimensional grid $\mathbf{u}_{p_1} \otimes \dots \otimes \mathbf{u}_{p_M}$.

3. Sample the black-box function on these query points (i.e., run the simulation with the corresponding parameters).
4. Visualize the results over the query points (which is feasible, since they have a low dimensionality M) and gain intuition.
5. Fix a different set of parameters and repeat the procedure.

The free parameters are usually generated by sequences of M nested loops in which each x_{p_m} varies between $\mathbf{u}_m(1)$ and $\mathbf{u}_m(I_m)$ for $m = 1, \dots, M$. When $M = 1$, the collection of samples has the form of a set of plots. Out of this set, the most informative plots (from a subjective point of view) are selected and presented by the user.

These manual sampling strategies can be regarded as a local-to-global navigation strategy [Sedlmair et al. 2014]. We observe that this subspace sampling approach can be viewed also as a particular case of cross approximation. If only one parameter at a time is fixed (that is, if $M = 1$), the strategy is then a fiber-based cross sampling scheme. Instead of such a trial-and-error exploration, we propose to apply an existing automated CA algorithm. We assume the simulation’s behavior with respect to the input parameters follows approximately a low-rank structure, so that it can be recovered from a relatively small subset of its samples. Using such an adaptive cross-interpolated scheme (in particular, one based on tensor decomposition) instead of a trial-and-error offers several advantages:

- Experiment results are recorded and analyzed on the fly, and at each step the sampling points are objectively determined. Irregular regions of the underlying black-box tensor increase the overall tensor rank and require more interpolating points during the cross-sampling scheme. Parameters with simple or no influence on the simulation output, on the other hand, tend to entail a lower rank structure and make the surrogate model need less samples along their corresponding modes to achieve the specified target accuracy.
- A compressed representation can be directly generated from the observed values. In the case of the TT representation, the resulting memory space is roughly proportional to the amount of samples needed and, in particular, not exponential with respect to the number of parameters.
- The resulting representation can be efficiently used to predict, explore and visualize the large parametric space as detailed in later sections.

Fig. 3 shows two simple examples of rank-1 cross-approximation of $\mathbb{R}^2 \rightarrow \mathbb{R}$ functions.

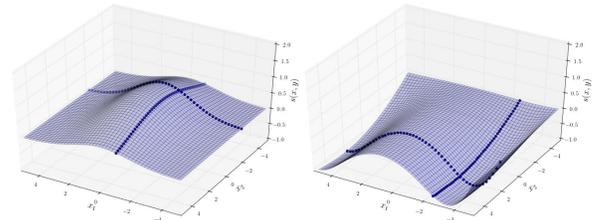


Figure 3: Two fibers are sufficient to exactly interpolate any rank-1 surface, reducing the amount of necessary samples from $O(I^2)$ to $O(I)$.

For the rest of the paper, \mathcal{A} will denote a TT low-rank approximation of the simulation output: $\mathcal{A}(\mathbf{x}) \approx s(\mathbf{x})$.

4 Visualization in the TT Format

4.1 Reconstructing Compressed Subspaces

Standard visualization diagrams such as 1D and surface plots, tables, and combinations thereof (hierarchical axis, dimensional stacking, overlaid plots, etc.) can be viewed as discretized axis-aligned subspaces of \mathcal{A} . It is straightforward to fix or restrict the movement of multilinear indices within any tensor decomposition. In the TT format in particular, we can set the k -th parameter to a by substituting the tensor core $\mathcal{A}^{(k)}$ with a $R_{k-1} \times 1 \times R_k$ core, namely $\mathcal{A}^{(k)}(:, a, :)$. The asymptotic reconstruction cost is $O(I^M R + R^2)$, where M is the number of free parameters. Such tensor decomposition is obtained by a sequence of matrix products and multi-array reshaping operations for which efficient parallel routines exist (such as the Linear Algebra PACKage, LAPACK). Since our simulations' behavior can be usually captured accurately using only a few TT ranks, in our experiments the subspace reconstruction cost was always in the order of a few milliseconds.

4.2 From Tensor Train to Parallel Coordinates

In order to further illustrate the versatility of such a TT parameter representation, we detail here how to obtain a parallel coordinates diagram from our compressed tensor \mathcal{A} . While this type of diagrams is applied to sparse data mostly, one can extend it to a dense space by varying the polyline styling. We make the opacity of each polyline proportional to the value at its corresponding position in the tensor (in the spirit of density-based parallel coordinates [Heinrich and Weiskopf 2013]). We assume linearity, so the total opacity of a segment equals the sum of the opacities of all polylines that include it. Thus the opacity of all $I_n I_{n+1}$ segments between two adjacent coordinates n and $n+1$ in the diagram is determined by the sum of \mathcal{A} along all indices except the n -th and $(n+1)$ -th ones. This sum yields an $I_n \times I_{n+1}$ matrix \mathbf{S} : for each entry (i, j) , a segment with opacity $\mathbf{S}(i, j)$ connecting abscissas i and j is to be drawn in the diagram. In total we have to display $(N-1)I^2$ segments and perform $O(NI^2R^2)$ operations, as opposed to the $O(NI^N)$ operations that a naive traversal of the full uncompressed tensor would take. The steps are summarized in Alg. 1.

Note that the pairs $(n, n+1)$ traverse the dimensions in their original order, but any reordering of coordinates is possible if we sum over the appropriate sets of indices.

Algorithm 1 Plot a parallel coordinates diagram from a TT

Require: \mathcal{A} is a low-rank TT approximation of the simulation s
 $\mathcal{B} :=$ empty N -dimensional TT-tensor
for n in $1, \dots, N$ **do**
 $\mathcal{B}^{(n)} := \sum_{i=1}^{I_n} \mathcal{A}^{(n)}(:, i, :)$
end for $\triangleright \mathcal{B}$ has size $1 \times \dots \times 1$
for n in $1, \dots, N-1$ **do**
 $\mathcal{C} := \mathcal{B}$
 $\mathcal{C}^{(n)} := \mathcal{A}^{(n)}$
 $\mathcal{C}^{(n+1)} := \mathcal{A}^{(n+1)}$ $\triangleright \mathcal{C}$ has now size $1 \times \dots \times I_n \times I_{n+1} \times \dots \times 1$
 $\mathbf{S} := \text{decompress}(\mathcal{C})$ $\triangleright \mathbf{S}$ has now size $I_n \times I_{n+1}$
 for i in $1, \dots, I_n$ **do**
 for j in $1, \dots, I_{n+1}$ **do**
 source := (n, i)
 target := $(n+1, j)$
 opacity := $\mathbf{S}(i, j)$
 drawSegment(source, target, opacity)
 end for
 end for
end for

4.3 Bivariate Projections

In Alg. 1 the projections onto all consecutive pairs of variables $(n, n+1)$ are computed. It is straightforward to compute the projection of \mathcal{A} onto every other possible pair of parameters. Each projection forms a 2D array (image), and one can arrange the $N^2 - N$ possible images as a matrix with an empty diagonal. An example using the simulation described later in Sec. 6.3 is shown in Fig. 4. Note that, since the colormap is normalized to the data range, displaying projections is equivalent to displaying averages.

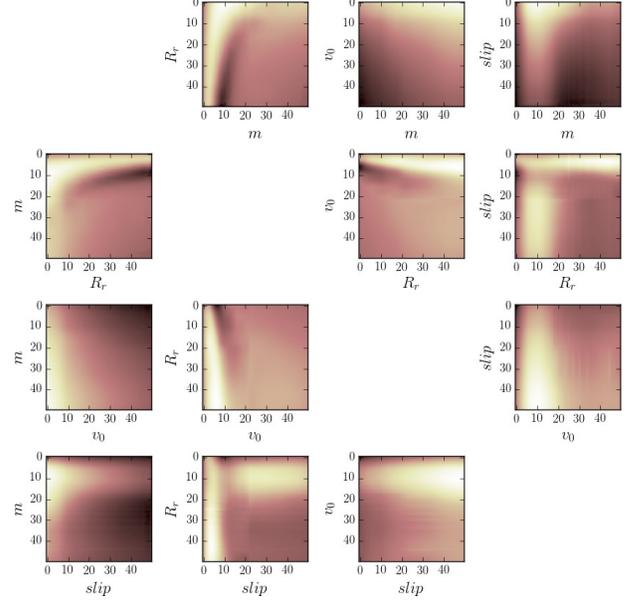


Figure 4: Example of bivariate projections diagram (from the simulation in Sec 6.3). Computing the projections from the compressed tensor took only 9.4ms, while the overall plot took 0.59s.

5 User Interaction

The previous sections have covered the building blocks of our navigation system: sampling paradigm and efficient reconstruction and visualization tools that can be leveraged from a TT decomposition. Here we describe the interaction elements available for the user for the overall knowledge discovery process.

5.1 Data Acquisition

The sampling stage is straightforward for the user, as it always follows three sequential steps:

1. Specify a numerical simulation to interface with.
2. Define the space: number of parameters and sampling density along each axis.
3. Run the cross-approximated adaptive sampling procedure with chosen accuracy.

5.2 Interactive Exploration

There are two main diagram types: the ones that display a general summary of the space (global visualization), and the ones that allow a local-to-global visualization strategy.

- Global visualization:

- Show a parallel coordinates diagram (Sec. 4.2). The ordering of dimensions can be varied, and each parameter’s movement can be restricted to a subinterval.
- Show uni- or bivariate projections (Sec. 4.3).
- Compute and display statistical moments and averages: global extrema, mean, variance, etc.
- Localized navigation: a series of subspace tensors are shown, each grouping one or more parameters together. The navigation window is centered on one focus point, signaled by vertical bars for 1D diagrams and red markers for 2D diagrams). Each diagram shows how the simulation will behave when its parameter(s) is/are moved while the rest are fixed on the focus (see e.g. Fig. 6). The focus point can be interactively modified by displacing its bars and markers.

Fig. 5 shows the building blocks of our sampling and visualization system.

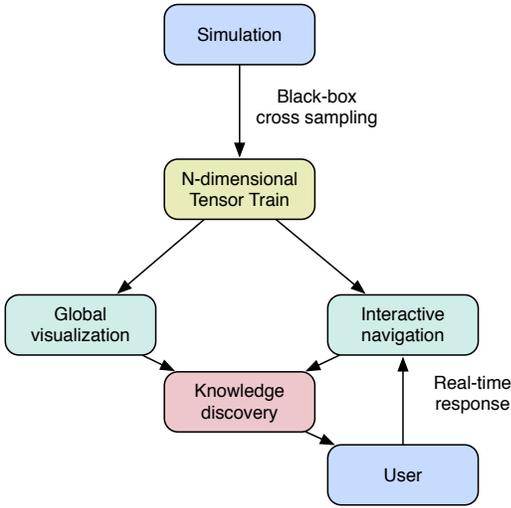


Figure 5: Visualization workflow.

6 Results

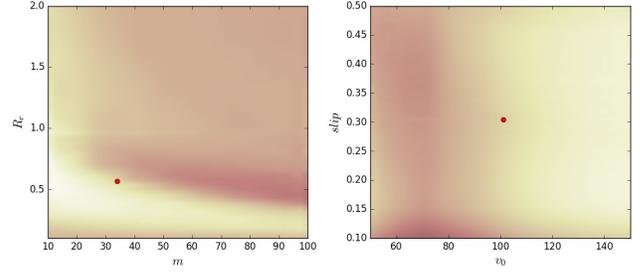
We have tested the CA sampling paradigm and the proposed visualization criteria on a number of practical application domains that we detail in this section. We used Python 2.7 on a 16-core Intel Xeon CPU E5-2670 with 2.60GHz and 32GB.

6.1 Synthetic Simulation

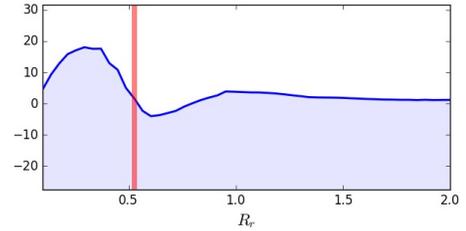
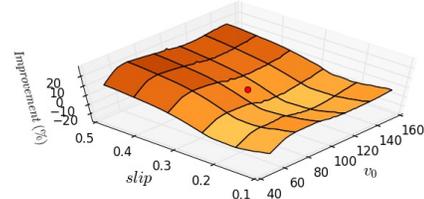
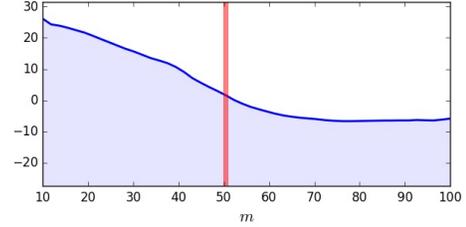
We defined the following 5-parametric synthetic model as a first example:

$$s(x, y, z, w, t) = \begin{cases} x + 10 \cdot \sin(y) - z^2/100 + \varepsilon & \text{if } 20 \leq w \leq 30 \\ \varepsilon & \text{otherwise} \end{cases}$$

where ε follows a normal distribution $N(0, 1)/2$ mimicking experimental error. Note that t is a *non-essential dimension*; in other words it has no influence on the model output. To discretize our grid we set $\mathbf{u}_n = (1, \dots, 50)$ for $n = 1, \dots, 5$. The resulting tensor product space has $\prod_{i=1}^5 I_n = 50^5 \approx 3.1 \cdot 10^8$ elements, out of which the CA method sampled 904 fibers (45200 elements) to produce a compressed TT-tensor with (12, 13, 13, 13) ranks and 25950



(a) Two 2D color plots



(b) Combining a 2D plot with a surface plot

Figure 7: Arbitrary pairs of parameters can be grouped together to form bidimensional plots. The focus point is indicated by red markers which can also be moved.

double-precision values. We targeted a relative compression error of 1%. Being a synthetic model composed of elementary mathematical functions, the tensor would have an exact rank of only 3 if it were not for the error ε .

The parallel coordinates diagram in Fig. 8 gives a good macroscopic overview of each variable’s influence, e.g. the linear contribution from x , the sinusoidal pattern of y , and the bandpass condition imposed by w . Computing the opacities for the $\sum_{n=1}^{N-1} I_n I_{n+1} = 10000$ segments from the compressed tensor took only 6.0ms.

6.2 KNN Classifier

The rest of our examples originate from real computer experiments. We consider now a K-nearest neighbors classifier (KNN), with its classification accuracy as the simulation output. This is a relatively simple model that, having only 3 parameters, can be displayed with a volume renderer. We used a subset of the MNIST Handwritten Digit Database [mni] consisting of 1000 samples (100 for each digit). Each digit image has 28×28 pixels. The samples are split

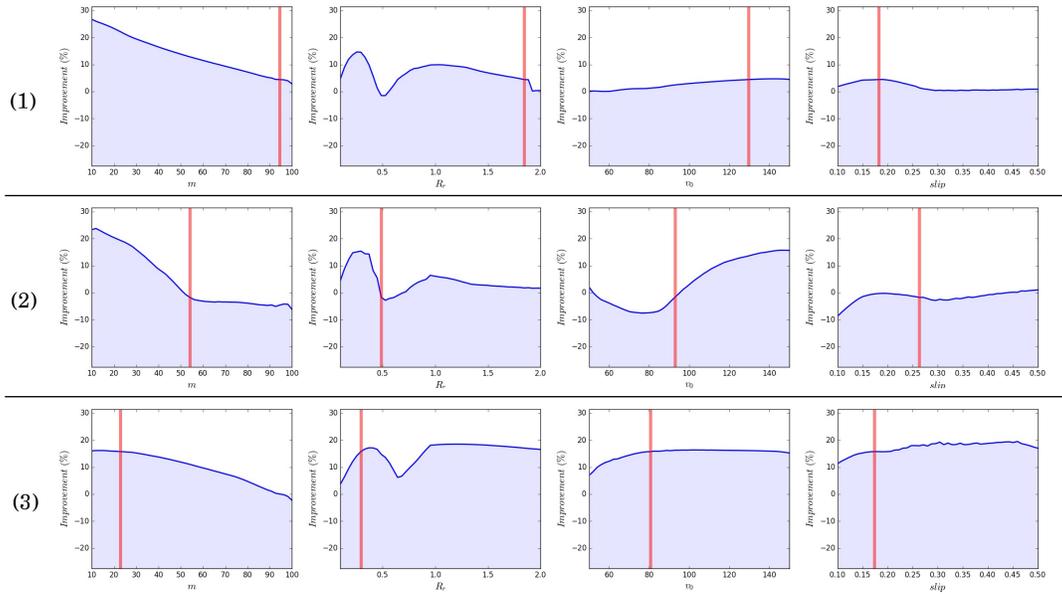


Figure 6: 3 instants in the navigation (one per row), showing the response of a simulation centered on 3 different focus points. Such points consist of four parameters, each of which is indicated by a vertical red bar. Note that each bar intersects each fiber plot at the same height; such height corresponds to the predicted simulation value at the focus point. During navigation the bars can be dragged left or right to vary the focus, allowing interactive visualization of the surrogate model and aiding in the knowledge discovery process.

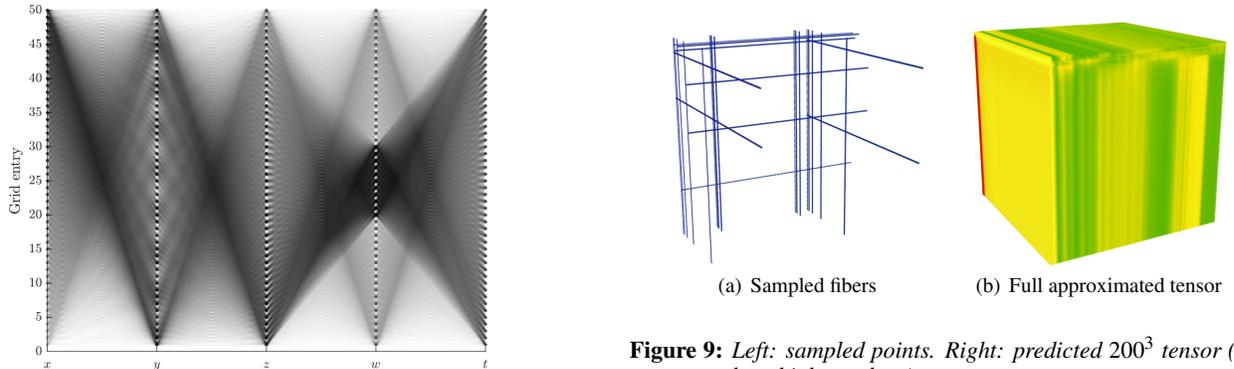


Figure 8: Parallel coordinates diagram for the synthetic simulation, directly extracted from the compressed \mathcal{A} .

evenly into a training and a test set. Given that filtering preprocessing operations can often improve performance in computer vision applications, we apply a bivariate Gaussian filter with standard deviations σ_1 and σ_2 on all training and query (test) images. As an additional parameter we also include the number of neighbors K for the classifier. Each experiment run $s(\sigma_1, \sigma_2, K)$ maps a parameter combination to the average performance (classification accuracy) of the KNN over the 500 test images using that combination. We set the grids $\mathbf{u}_1 = \mathbf{u}_2 = (0, \dots, 10)$ and $\mathbf{u}_3 = (1, \dots, 200)$ with each interval containing 200 equally spaced points, for a total of $200^3 = 8 \cdot 10^6$ possible combinations of parameters. One run takes approximately 0.1s of computing time, so sampling the full space would require several days of simulation.

Fig. 9 shows the 5600 samples taken by the TT-cross algorithm (targeting 5% error) and the resulting full tensor after explicit reconstruction. Sampling plus compression took 7.3 minutes, while decompression took 144ms. The tensor ranks are (3,4).

6.3 Anti-Lock Braking System

Next, we consider a 4-parametric simulation of a vehicle equipped with an ABS system for reducing its stopping distance when braking. This is one of the standard automotive models offered along the Simulink platform. We modified it to leave the on-board controller's target wheel slip factor as a parameter (instead of being fixed to 0.2). The other 3 inputs are vehicle mass m , wheel radius R_r and initial speed v_0 . The output is the reduction (%) of the distance traveled until full stop between a vehicle with ABS and one without it. A (10, 13, 10)-rank tensor was computed from 27800 simulation samples (2.6 hours, targeting 3% error). Fig 6 shows the navigation window of this tensor for 3 different focus points.

6.4 Suspension System

Our final model example has 6 parameters and is also an automotive application from Simulink. It studies the behavior of a vehicle as it circulates at a constant speed over a bump of fixed height. The inputs include the distances d_f and d_r from the front and rear axes

to the vehicle’s center of mass, total vehicle mass m , its initial moment of inertia I_{yy} , and the stiffness k and damping factors c of the suspension system. We define the output value as the overall discomfort perceived by the driver, measured as the total vertical acceleration of the vehicle (in modulus) during the simulation,

$$s(d_f, d_r, m, I_{yy}, k, c) = \int_0^T |\ddot{z}(t)| dt$$

The simulation is stopped at instant T , as soon as \ddot{z} stabilizes to 0.

The grid ranges are $d_f, d_r \in [0.5, 1.5]$, $m \in [500, 2500]$, $I_{yy}, c \in [1000, 3000]$ and $k \in [15000, 40000]$ with again 50 equidistant samples per interval. Out of the $50^6 \approx 1.56 \cdot 10^{10}$ grid points, 44700 samples were taken during 1.8 hours to produce a (9, 9, 9, 10, 7)-rank tensor (aiming again for 3% error). Using the navigation system the user can quickly identify regions of low or high discomfort. The global maximum is shown in Fig. 10.

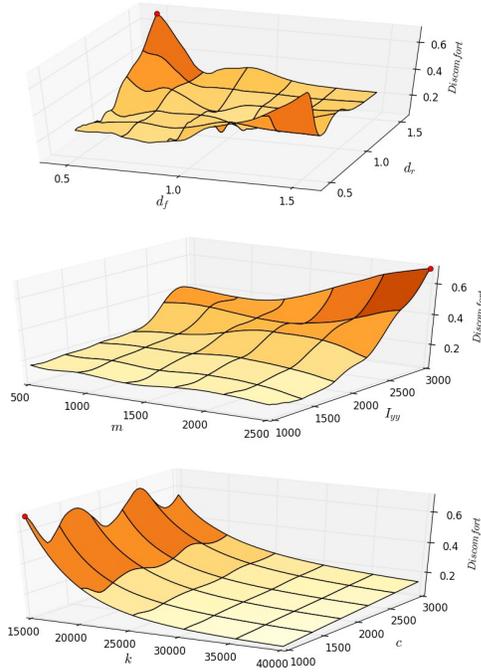


Figure 10: Navigation of a 6-dimensional tensor arising from the suspension system simulation. Each surface plot encompasses 2 out of the 6 space parameters. The red markers (focus point) indicate the global maximum of the tensor.

7 Conclusion

We have described a visualization system that creates and operates on a surrogate model in order to predict and help analyze the behavior of scientific simulations. Cross-approximated sampling algorithms relieve the user from tedious trial-and-error explorations, while tensor decompositions offer many possibilities for manipulating and visualizing multidimensional data. Thanks to the tensor train model one can combine these two aspects into one single framework. In our system, the output of the simulation with respect to its parameters was considered as a high-dimensional space and compressed into a low-rank tensor. TT constructions scale linearly with the number of parameters and the amount of sampled points. We have shown how the user can select and effectively reconstruct interesting subspaces from the compressed domain in real time and interactively. Several types of visualization diagrams and

techniques can benefit from the compactness and efficiency of the TT framework.

Future Work In the future we would like to include categorical variables in the domain of the simulation function s . CA algorithms (both sampling operations and compressed-domain optimization) are equally well-defined for categorical and ordinal variables. In our method we have quantized (discretized) cardinal parameters into ordinal ones. We also want to take into account the simulation time, in order to inform the user how long it would take to compute the ground-truth of a subspace. One possibility is to consider the simulation’s running time as an additional output, that can be predicted in the same way.

References

- AMIRKHANOV, A., HEINZL, C., REITER, M., AND GRÖLLER, M. E. 2010. Visual optimality and stability analysis of 3DCT scan positions. *IEEE Transactions on Visualization and Computer Graphics* 16, 6, 1477–1486.
- ANAND, A., DANG, T. N., AND WILKINSON, L. 2012. Visual pattern discovery using random projections. In *Proceedings IEEE Conference on Visual Analytics Science and Technology*, 43–52.
- BALLANI, J., AND GRASEDYCK, L., 2014. Hierarchical tensor approximation of output quantities of parameter-dependent PDEs. Technical Report Nr. 385, March.
- BALLESTER-RIPOLL, R., AND PAJAROLA, R. 2015. Lossy volume compression using Tucker truncation and thresholding. *The Visual Computer*, 1–14.
- BALLESTER-RIPOLL, R., SUTER, S. K., AND PAJAROLA, R. 2015. Analysis of tensor approximation for compression-domain volume visualization. *Computers & Graphics* 47, 34–47.
- BALSA RODRÍGUEZ, M., GOBBETTI, E., IGLESIAS GUITIÁN, J. A., MAKHINYA, M., MARTON, F., PAJAROLA, R., AND SUTER, S. K. 2014. State-of-the-art in compressed GPU-based direct volume rendering. *Computer Graphics Forum* 33, 6, 77–100.
- BEBENDORF, M. 2000. Approximation of boundary element matrices. *Numerische Mathematik* 86, 4, 565–589.
- BERGNER, S., SEDLMAIR, M., MOLLER, T., ABDOLYOUSEFI, S. N., AND SAAD, A. 2013. Paraglode: Interactive parameter space partitioning for computer simulations. *IEEE Transactions on Visualization and Computer Graphics* 19, 9, 1499–1512.
- BERTINI, E., AND LALANNE, D. 2010. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *SIGKDD Explor. Newsl.* 11, 2 (May), 9–18.
- BRECHSEISEN, R., VILANOVA, A., PLATEL, B., AND TER HAAR ROMENY, B. 2009. Parameter sensitivity visualization for dti fiber tracking. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov), 1441–1448.
- CHAN, W. W.-Y. 2006. A survey on multivariate data visualization. Tech. rep., Department of Computer Science and Engineering. Hong Kong University of Science and Technology.
- CHEN, Y.-L., HSU, C.-T., AND LIAO, H.-Y. M. 2014. Simultaneous tensor decomposition and completion using factor priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3, 577–591.

- COSTANTINI, R., SBAIZ, L., AND SÜSSTRUNK, S. 2008. Higher order SVD analysis for dynamic texture synthesis. *IEEE Transactions on Image Processing* 17, 1, 42–52.
- DE LATHAUWER, L., DE MOOR, B., AND VANDEWALLE, J. 2000. On the best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of higher-order tensors. *SIAM Journal of Matrix Analysis and Applications* 21, 4, 1324–1342.
- DOS SANTOS AMORIM, E. P., BRAZIL, E. V., II, J. D., JOIA, P., NONATO, L. G., AND SOUSA, M. C. 2012. iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In *Proceedings IEEE Conference on Visual Analytics Science and Technology*, 53–62.
- ESPIG, M., GRASEDYCK, L., AND HACKBUSCH, W. 2009. Black box low tensor-rank approximation using fiber-crosses. *Constructive Approximation* 30, 3, 557–597.
- FILIPOVIĆ, M., AND JUKIĆ, A. 2015. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems Signal Processing* 26, 3, 677–692.
- GERBER, S., BREMER, P.-T., PASCUCCI, V., AND WHITAKER, R. 2010. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics* 16, 6, 1271–1280.
- HARSHMAN, R. A. 1970. Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis. *UCLA working papers in phonetics* 16, 1–84.
- HEINRICH, J., AND WEISKOPF, D. 2013. State of the art of parallel coordinates. In *Proceedings Eurographics 2013 (State of the Art Reports)*, 95–116.
- KEHRER, J., AND HAUSER, H. 2013. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics* 19, 3, 495–513.
- KOLDA, T. G., AND BADER, B. W. 2009. Tensor decompositions and applications. *SIAM Review* 51, 3, 455–500.
- LIAO, S., VEJCHODSKÝ, T., AND ERBAN, R. 2015. Tensor methods for parameter estimation and bifurcation analysis of stochastic reaction networks. *Journal of The Royal Society Interface* 12, 108.
- MNIST Handwritten Digit Database. Y. LeCun, C. Cortes and C. Burges, <http://yann.lecun.com/exdb/mnist/>.
- NOCKE, T., FLECHSIG, M., AND BÖHM, U. 2007. Visual exploration and evaluation of climate-related simulation data. In *Simulation Conference*, 703–711.
- OSELEDETS, I. V., AND TYRTYSHNIKOV, E. E. 2010. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.* 432, 1, 70–88.
- OSELEDETS, I. V., ET AL. Tensor Train Toolbox. Github repository, <https://github.com/oseledets/ttpt>.
- OSELEDETS, I. V., SAVOSTYANOV, D. V., AND TYRTYSHNIKOV, E. E. 2008. Tucker dimensionality reduction of three-dimensional arrays in linear time. *SIAM Journal on Matrix Analysis Applications* 30, 3, 939–956.
- OSELEDETS, I. V. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing* 33, 5, 2295–2317.
- POTTER, K., WILSON, A., BREMER, P.-T., WILLIAMS, D., DOUTRIAUX, C., PASCUCCI, V., AND JOHNSON, C. R. 2009. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *IEEE International Conference on Data Mining Workshops.*, 233–240.
- PRETORIUS, A., ZHOU, Y., AND RUDDLE, R. 2015. Visual parameter optimisation for biomedical image processing. *BioMed Central Bioinformatics* 16, 11, 1–13.
- RAJWADE, A., RANGARAJAN, A., AND BANERJEE, A. 2013. Image denoising using the higher order singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 4, 849–862.
- SAVOSTYANOV, D. V., AND OSELEDETS, I. V. 2011. Fast adaptive interpolation of multi-dimensional arrays in tensor train format. In *Proceedings 7th International Workshop on Multidimensional Systems*.
- SEDLMAIR, M., HEINZL, C., BRUCKNER, S., PIRINGER, H., AND MÖLLER, T. 2014. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics* 20, 12, 2161–2170.
- SUTER, S. K., ZOLLIKOFER, C. P., AND PAJAROLA, R. 2010. Application of tensor approximation to multiscale volume feature representations. In *Proceedings Vision, Modeling and Visualization*, 203–210.
- SUTER, S. K., IGLESIAS GUITIÁN, J. A., MARTON, F., AGUS, M., ELSENER, A., ZOLLIKOFER, C. P., GOPI, M., GOBBETTI, E., AND PAJAROLA, R. 2011. Interactive multiscale tensor reconstruction for multiresolution volume visualization. *IEEE Transactions on Visualization and Computer Graphics* 17, 12, 2135–2143.
- SUTER, S., MAKHYNIA, M., AND PAJAROLA, R. 2013. TAM-RESH: Tensor approximation multiresolution hierarchy for interactive volume visualization. *Computer Graphics Forum* 32, 3pt2, 151–160.
- TATU, A., MAASS, F., FÄRBER, I., BERTINI, E., SCHRECK, T., SEIDL, T., AND KEIM, D. A. 2012. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings IEEE Symposium on Visual Analytics Science and Technology*, 63–72.
- TYRTYSHNIKOV, E. E. 2000. Incomplete cross approximation in the mosaic-skeleton method. *Computing* 64, 367–380.
- WU, Q., XIA, T., CHEN, C., LIN, H.-Y. S., WANG, H., AND YU, Y. 2008. Hierarchical tensor approximation of multidimensional visual data. *IEEE Transactions on Visualization and Computer Graphics* 14, 1, 186–199.