HMK-CTA: A Hierarchical Multidimensional Representation for Visual Datasets

Yu-Ting Tsai* Muhammad Abu Bakar†

Department of Computer Science and Engineering, Yuan Ze University

Abstract

This paper presents a novel tensor-based representation, namely hierarchical multiway K-clustered tensor approximation, for multidimensional visual datasets. The proposed method extends a previous tensor model [20] into a hierarchical representation that can significantly reduce offline computational cost as well as provide similar approximation quality and rendering performance at the same time. We also apply the proposed method to approximate spatially-varying bidirectional reflectance distribution functions, time-varying light fields, and time-varying volume data to show its effectiveness and potential for data-driven rendering. Experimental results demonstrate that under similar performance to previous work, the proposed method can reduce offline approximation time by even an order of magnitude.

Index Terms: Realtime rendering, multidimensional data analysis, hierarchical model, multiway clustering, sparse representation.

1 Introduction

Data-driven rendering is a popular type of state-of-the-art image synthesis methods, which relies on measured or precomputed visual information to synthesize photorealistic 3D images. With the increases in the demand on high-quality 3D images in graphics applications, it usually needs to employ large-scale visual datasets for rendering. In recent years, a lot of related papers have addressed this problem by approximating visual datasets such that the results can be reconstructed for realtime image synthesis. Most of them have focused on how to make a good tradeoff among fast rendering rates, high approximation quality, and compact storage requirements for approximated data.

Nevertheless, simultaneously achieving low offline approximation time is still a major challenge that only a few papers have deeply studied. Previous sophisticated data-driven rendering algorithms may need hundreds of hours in order to approximate just one dataset with a high-performance workstation. For example, *multiway K-clustered tensor approximation* (MK-CTA) [20] takes more than 10 hours for a time-varying light field dataset with a size of 12 GB. If one would like to significantly increase rendering rates of the time-varying light field, MK-CTA may even spend a few weeks on the offline approximation process from our experience. The time-consuming offline process thus becomes a great obstacle to the use of advanced data-driven models in practical applications.

In this paper, we present a novel tensor-based method, namely *hierarchical multiway K-clustered tensor approximation* (HMK-CTA), for accelerating offline approximations of large-scale visual datasets, along with a good compromise of the above-mentioned three desired goals. The proposed algorithm particularly extends MK-CTA to not only hierarchically cluster an input tensor into disjoint smaller ones that can be progressively approximated with more efficiency, but also faithfully capture multiscale visual features within the tensor for runtime reconstruction and realtime rendering.

In brief, this paper makes the following contributions:

- Introduce a novel hierarchical tensor representation, namely HMK-CTA, to progressively preserve visual features among datasets at different scales.
- HMK-CTA can significantly reduce offline approximation time by even an order of magnitude.
- Apply HMK-CTA to approximate common large-scale visual datasets, such that an appropriate tradeoff among offline computational cost, runtime rendering performance, approximation errors, and storage requirements can be easily achieved.

2 Related Work

2.1 Data-Driven Rendering

Rendering photorealistic 3D images from visual datasets has caught a lot of attention in recent years. This data-driven idea may date back to texture mapping and has been widespread nowadays in many graphics applications. Modern visual datasets for rendering have various forms, including precomputed lighting data [11, 25, 27], (multispectral) bidirectional texture functions [4, 13], time-varying and/or spatially-varying bidirectional reflectance distribution functions [5, 7, 16, 28], (time-varying) light fields [9, 29], and more complex lighting models [19, 32], just to name a few. It is also expected to see more different forms of visual datasets in the future.

Although the data-driven idea is simple and intuitive, the large amount of visual datasets usually prevents compact storage requirements, high-quality 3D images, and fast rendering rates at the same time. Over the past decades, this issue has stimulated the development of many sophisticated algorithms for visual data approximations, for example, parametric models [6, 21, 31], spherical/zonal harmonics [2, 27], wavelets [15, 25], tensor representations [1, 20, 23], neural networks [8, 10], and many more.

Besides the above-mentioned three desired goals, we introduce a novel tensor-based algorithm in this paper to additionally provide practical offline computational cost. The proposed method is also employed to approximate common large-scale visual datasets, demonstrating its potential and effectiveness.

2.2 Tensor Representations

In data-driven rendering, tensor representations (also called multilinear models or multiway analysis) [14] are usually regarded among the most effective methods for modeling high-dimensional visual datasets. They organize a set of high-dimensional data as a multidimensional array, namely a tensor, to preserve its intrinsic structure and separately extract visually prominent components along each specified dimension/mode of the tensor. As a pioneering work, Vasilescu and Terzopoulos [24] applied a suboptimal algorithm based on *higher-order singular value decomposition* (HOSVD) [3] to approximate bidirectional texture functions for efficient runtime rendering. Wang et al. [26] then introduced an out-of-core framework for HOSVD to address large-scale visual datasets that cannot fit into main memory.

*e-mail: hieicis91@hotmail.com

†e-mail: bubakar.nuaa@gmail.com

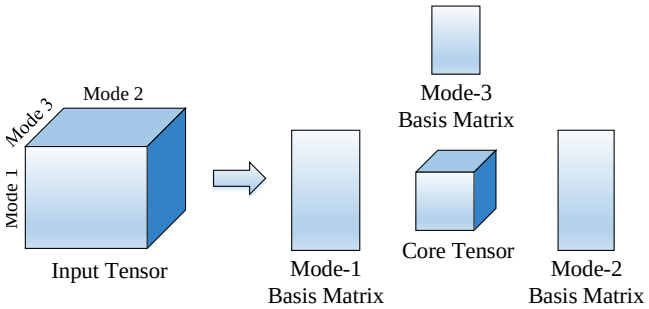


Figure 1: An example of HOSVD for a third-order tensor. The input tensor is decomposed into a third-order core tensor and three basis matrices that are associated with each mode of the input tensor.

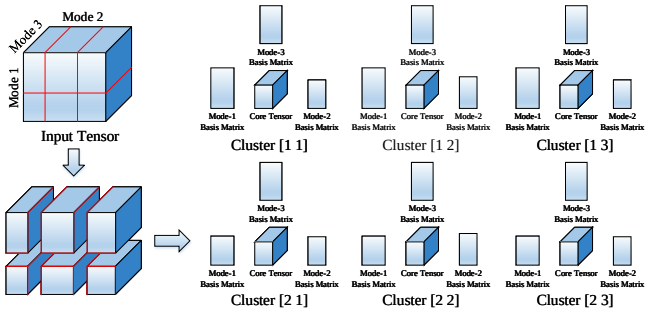


Figure 2: An example of MK-CTA for a third-order tensor. The input tensor is clustered along modes 1 and 2 into six smaller third-order tensors, each of which is further approximated using HOSVD.

Based on these two papers, many sophisticated tensor representations for data-driven rendering have been proposed over the last two decades. Clustered tensor approximation [22] successfully integrated HOSVD with clustering to partition a tensor along one specified dimension into disjoint smaller tensors, each of which can be more efficiently approximated by HOSVD. K-clustered tensor approximation [23] then extended this idea with sparse representation to additionally allow non-disjoint (or overlapped) cluster tensors [23]. Recently, MK-CTA [20] further employed multiway *soft* clustering to split a tensor along more than one dimensions, thus overcoming the major drawback of K-clustered tensor approximation.

Perhaps the most relevant research work to this paper is hierarchical tensor approximation [30]. It transformed a tensor into a hierarchy of smaller tensors and relied on HOSVD to extract multiscale visual features from tensors at different levels of the hierarchy. Nevertheless, it employed *uniform* recursive binary partition at each level to construct the hierarchy. By contrast, we take a different path to develop a novel hierarchical tensor representation based on MK-CTA, instead of just HOSVD. Our method thus allows a more flexible hierarchical framework by adaptively partitioning tensors at each level using multiway clustering. Through the merit of MK-CTA, this also leads to more efficient data approximations and runtime rendering performance.

Although there are other tensor representations for data-driven rendering, a comprehensive survey is beyond the scope of this paper. Interested readers may refer to sparse tensor decomposition [12], tensor train decomposition [1], and references therein, or even related papers in visualization [17, 18].

3 Background

3.1 HOSVD

HOSVD [3] decomposes an N-th-order tensor $\boldsymbol{\mathcal{A}}$ into a core tensor $\boldsymbol{\mathcal{Z}}$ (also N-th-order) and a set of basis matrices $\mathbf{U}_1$, $\mathbf{U}_2$, ..., and $\mathbf{U}_N$. Each basis matrix is associated to a distinct mode of $\boldsymbol{\mathcal{A}}$ and can be regarded as a basis for the subspace of that mode. The original tensor $\boldsymbol{\mathcal{A}}$ thus can be reconstructed by combining the subspaces of different modes by using mode-1, mode-2, ..., and mode-N products as

$$\boldsymbol{\mathcal{A}} \approx \boldsymbol{\mathcal{Z}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N, \tag{1}$$

where the symbol $\times_n$ represents the mode-n product between a tensor and a matrix. HOSVD relies on reducing the rank of each mode in the core tensor, while also reducing the rank of each basis matrix at the same time, to derive a locally optimal approximation of $\boldsymbol{\mathcal{A}}$. The smaller the core tensor is, the higher approximation error the decomposed result will have. Fig. 1 shows an example of employing HOSVD to decompose a third-order tensor.

3.2 MK-CTA

MK-CTA [20] integrates HOSVD with multiway clustering to partition an N-th-order tensor $\boldsymbol{\mathcal{A}}$ along more than one modes into smaller tensors and with sparse representation to assign each element in $\boldsymbol{\mathcal{A}}$ into more than one (but only a few) clusters along each mode. In this way, the smaller tensor within a cluster is expected to contain more coherent elements, such that each cluster tensor can be approximated using HOSVD with more efficiency.

Specifically, $\boldsymbol{\mathcal{A}}$ can be approximated by the summation of each the reconstructed tensor of each cluster as

$$\boldsymbol{\mathcal{A}} \approx \sum_{\mathbf{c}} \boldsymbol{\mathcal{Z}}_{\mathbf{c}} \times_1 \mathbf{U}_{1,\mathbf{c}} \times_2 \mathbf{U}_{2,\mathbf{c}} \cdots \times_N \mathbf{U}_{N,\mathbf{c}}, \tag{2}$$

where $\boldsymbol{\mathcal{Z}}_{\mathbf{c}}$, $\mathbf{U}_{1,\mathbf{c}}$, $\mathbf{U}_{2,\mathbf{c}}$, ..., and $\mathbf{U}_{N,\mathbf{c}}$ represent the extracted core tensor and basis matrices after employing HOSVD to the tensor of a cluster $\mathbf{c}$. Note that the basis matrices of each cluster are sparse, such that the overall effect is the same as approximating a single tensor element from the decomposed results of just a few clusters, but not most or all. The merit of sparse representation particularly allows better approximation quality with low storage overhead and simultaneously prevents runtime reconstruction cost from increasing significantly. Fig. 2 shows an example of applying MK-CTA to decompose a third-order tensor.

4 HMK-CTA Algorithm

4.1 Overview

Our approach was inspired by the following insights:

- The offline computational cost of MK-CTA significantly increases with larger numbers of clusters. Namely, reducing numbers of clusters can effectively speed up the offline process.
- A hierarchical tensor representation allows users to accurately capture important visual features among datasets at various scales and progressively show approximation results at runtime.
- Although the approximation quality of MK-CTA is considerably influenced by reducing numbers of clusters, a hierarchical representation still can further classify datasets at lower levels to compensate the impact.

Figure 3 illustrates an example of adopting HMK-CTA to approximate a third-order tensor. At the highest level 1, the input tensor is first approximated using MK-CTA. The residual tensor is then computed by subtracting the reconstruction of the approximation result from the input tensor, and further subdivided into smaller tensors according to the clustering result of MK-CTA. The subdivided residual tensors are passed to the next lower level 2 and respectively approximated using MK-CTA again. This process can

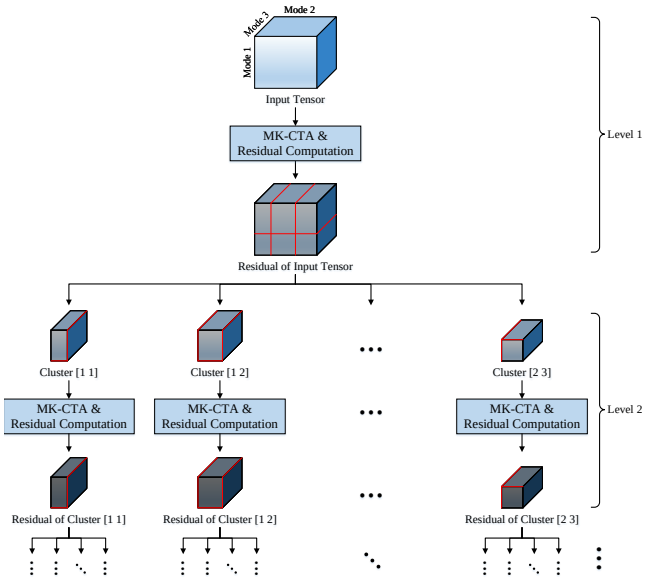


Figure 3: An example of HMK-CTA for a third-order tensor. At the highest level 1, the input tensor is approximated using MK-CTA to obtain the residual tensor, and subdivided into a few (for example, six) smaller tensors for approximations based on MK-CTA at the next lower level 2. The residuals of subdivided tensors can be further clustered and approximated at subsequent lower levels until a user-specified termination criteria is reached.

be repeated at subsequent lower levels until a user-specified termination criteria is reached, such as the maximum number of levels or the desired approximation error.

Based on the proposed HMK-CTA algorithm, we can choose to subdivide residual tensors at higher levels into less numbers of clusters along each mode to significantly reduce the offline computational cost of MK-CTA. Although this will increase approximation errors and only capture global (or large-scale) features at higher levels, local (or small-scale) features can be further extracted at lower levels. From our experience, the resulting HMK-CTA algorithm is a good compromise, since it can significantly reduce the offline computational cost, while still achieving similar runtime rendering performance, approximation errors, and storage requirements when compared with MK-CTA.

Although the framework of HMK-CTA is not complex and easy to understand, there are some implementation issues that need to be discussed in detail. In Section 4.2, we will describe how to subdivide a residual tensor for efficient approximations at the next lower level. Moreover, some parameter selection guidelines for good compromises will be suggested in Section 4.3.

4.2 Residual Tensor Subdivision

Previous hierarchical tensor representations [30, 17, 18] employ uniform blocks to organize/subdivide the input tensor for multi-scale analysis at different levels. Unlike them, we instead apply an adaptive subdivision scheme that relies on the clustering result of MK-CTA at the current level to determine the elements of subdivided tensors at the next lower level. This scheme is similar to ordinary divisive hierarchical clustering. It not only results in subdivided tensors with various sizes, but also may not maintain the spatial structure of tensor elements at the current level, namely possibly classifying adjacent tensor elements into different subdivided tensors. This characteristic particularly allows our algorithm to exploit more data coherence, other than the spatial one, among tensor elements.

Formally, consider a residual tensor $\boldsymbol{\mathcal{A}}^{(l)}$ at a level l, whose mode ranks are respectively $I_1^{(l)}$, $I_2^{(l)}$, ..., and $I_N^{(l)}$. HMK-CTA is going to subdivide it into smaller tensors for further approximation at the next lower level $l+1$. By applying MK-CTA, $\boldsymbol{\mathcal{A}}^{(l)}$ is classified into $C_1^{(l)}$, $C_2^{(l)}$, ..., and $C_N^{(l)}$ clusters along each mode, while each tensor element in $\boldsymbol{\mathcal{A}}^{(l)}$ is assigned into $K_1^{(l)}$, $K_2^{(l)}$, ..., and $K_N^{(l)}$ different clusters respectively along each mode (also called the numbers of mixture clusters). Moreover, MK-CTA approximates tensor elements in each cluster by using HOSVD with reduced ranks $R_1^{(l)}$, $R_2^{(l)}$, ..., and $R_N^{(l)}$. Intuitively, we may subdivide $\boldsymbol{\mathcal{A}}^{(l)}$ by directly following the clustering result of MK-CTA. Nevertheless, if $K_1^{(l)}$, $K_2^{(l)}$, ..., or $K_N^{(l)}$ is larger than 1, which is a quite common case, there are more than one subdivided tensors associated with a single tensor element $a^{(l)}$ in $\boldsymbol{\mathcal{A}}^{(l)}$. If we also follow the clustering results of MK-CTA at subsequent lower levels, the number of subdivided tensors associated with $a^{(l)}$ may grow exponentially. This will substantially increase offline approximation and runtime reconstruction cost, especially when runtime performance is a major concern in realtime applications.

In order to solve this issue, we propose the *most significant* subdivision scheme to associate a single tensor element with only one subdivided tensor at each level. By referring to the clustering result of MK-CTA for $\boldsymbol{\mathcal{A}}^{(l)}$, we subdivide $\boldsymbol{\mathcal{A}}^{(l)}$ such that the sum of the reconstruction error of each subdivided tensor is the lowest, while each tensor element in $\boldsymbol{\mathcal{A}}^{(l)}$ is assigned to only one subdivided tensor. As illustrated in Fig. 4, for example, elements in the blue rectangle are assigned into cluster subsets 2 (clusters [2 1], [2 2], and [2 3]) and 3 (clusters [3 1], [3 2], and [3 3]) by MK-CTA along the first mode, but our most significant scheme only associates all the elements with the cluster subset 2 for subdivided tensors. A similar subdivision choice along the second mode leads to associating elements in the red rectangle with the cluster subset 1 (clusters [1 1], [2 1], and [3 1]). Then, the overall reconstruction error of subdivided tensors is the lowest among all possible cases.

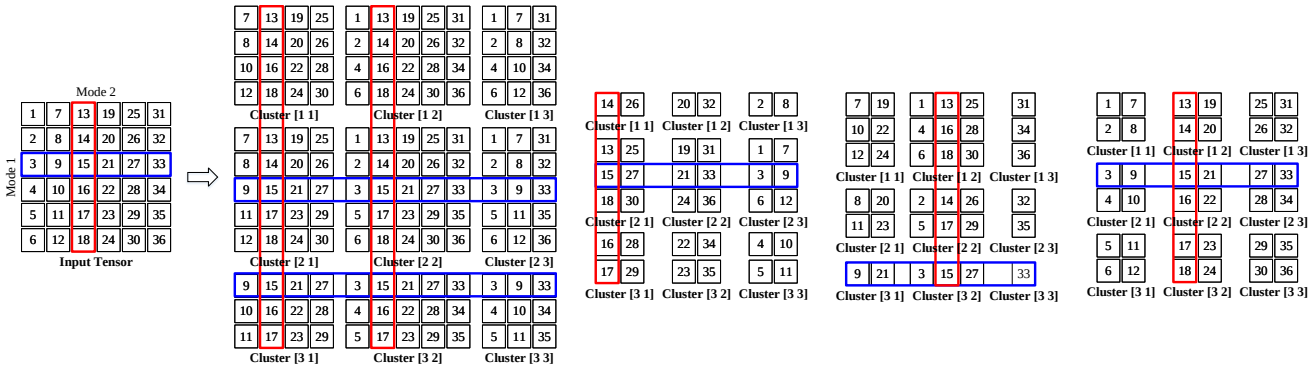
Our most significant scheme particularly assumes that when there are more coherent elements within each subdivided tensor (thus the lower overall reconstruction error), the residuals of elements within a subdivided tensor also tend to be similar to each other. If MK-CTA fails to exploit all available coherence among tensor elements at the current level (thus missing small-scale features), subdivided tensors at the next lower level are still likely to contain coherent elements for further approximations.

We also tried other subdivision schemes, including the least significant and uniform (Fig. 4). Similar to the most significant, the least significant scheme is also based on the clustering result of MK-CTA, but it leads to the highest overall reconstruction error of subdivided tensors. On the other hand, the uniform scheme partitions $\boldsymbol{\mathcal{A}}^{(l)}$ into disjoint blocks with the same size and fixed membership[1]. Among the three schemes, the most significant one outperforms the other two in terms of the final reconstruction error of HMK-CTA, while the uniform scheme is the worst. A detailed comparison will be shown in Section 5.1.

4.3 Parameter Selection Guidelines

There are various parameters of HMK-CTA, including the number of levels (L) as well as the reduced ranks ($R_n^{(l)}$), the numbers of clusters ($C_n^{(l)}$), and the numbers of mixture clusters ($K_n^{(l)}$) at each level. It is non-trivial to determine all parameters for a specific goal,

[1]The mode-n rank of a subdivided tensor is $\left\lceil I_n^{(l)}/C_n^{(l)} \right\rceil$. Nevertheless, if $I_n^{(l)}$ is not a multiple of $C_n^{(l)}$, the mode-n ranks of the last $mod(I_n^{(l)}, C_n^{(l)})$ subdivided tensors are instead $\left\lceil I_n^{(l)}/C_n^{(l)} \right\rceil - 1$, where $mod(a, b)$ denotes a modulo b.



(a) MK-CTA (b) Most Significant (c) Least Significant (d) Uniform

Figure 4: Comparison of different subdivision schemes for a second-order tensor (also known as a matrix). In each subfigure, a square represents a single tensor element with a number inside as its identity. The blue rectangle contains elements in a mode-1 subtensor (or a row for a matrix), while the red one encloses elements in a mode-2 subtensor (or a column). (a) The input tensor is classified into total 3×3 overlapped clusters based on MK-CTA. Each element is assigned into two clusters respectively along each mode, and MK-CTA restricts that elements in a mode-1/mode-2 subtensor should be classified into the same two mode-1/mode-2 cluster subsets. (b)-(c) Based on the clustering result of MK-CTA, the most/least significant scheme subdivides the tensor such that the sum of the reconstruction error of each subdivided tensor is the lowest/highest. (d) Irrelevant to the clustering result of MK-CTA, the traditional uniform scheme partitions the tensor into disjoint blocks with the same size and fixed membership.

for example, fast rendering rates and/or high-quality approximations. This subsection especially presents some parameter selection guidelines based on our experience.

The runtime rendering performance is mostly dominated by the number of levels, reduced ranks, and numbers of mixture clusters. The rendering time is roughly in proportional to $\sum_{l=1}^{L}\prod_{n=1}^{N} R_n^{(l)} K_n^{(l)}$, but only slightly increases with larger numbers of clusters.

On the other hand, increasing the number of levels, reduced ranks, or numbers of clusters often significantly improves approximation quality. For different parameter configurations, similar values of $\sum_{l=1}^{L}\prod_{n=1}^{N} R_n^{(l)} C_n^{(l)}$ generally result in similar errors. Although numbers of mixture clusters also have an impact on quality, there are no noticeable differences when their values are larger than 3.

The offline approximation time is often reduced by decreasing the number of levels, numbers of clusters, and/or numbers of mixture clusters, especially the last two. Nevertheless, increasing reduced ranks only slightly raises the offline computational cost.

In summary, an appropriate number of levels is between 2 and 4 from our experience. At each level, employing MK-CTA by setting numbers of mixture clusters to 2 is often suggested, as it provides a good tradeoff among approximation quality, rendering performance, and offline approximation cost. Moreover, if rendering performance is the most important concern, reduced ranks should be kept as small as possible. Nevertheless, numbers of clusters should be also increased in order to retain approximation quality.

5 Experiments

This section compares HMK-CTA with MK-CTA and demonstrates the experimental results of *spatially-varying bidirectional reflectance distribution functions* (SVBRDFs), *time-varying light fields* (TVLFs), and *time-varying volume data* (TVVD), which are common types of large-scale multidimensional visual datasets in computer graphics and visualization. The utilized SVBRDFs, TVLFs, and TVVD were respectively collected from the project webpage of inverse shade trees [7] (http://ist.cs.princeton.edu/), the synthetic light field archive [29] (http://web.media.mit.edu/∼gordonw/SyntheticLightFields/), and the time-varying volume data repository (http://www.cs.ucdavis.edu/ ∼ma/ITR/tvdr.html).

For MK-CTA, we employed optimal search for sparse clustering. As for HMK-CTA, we adopted the most significant subdivision scheme unless specified. Raw and approximated data were stored as 32-bit and 16-bit floating-point numbers, respectively. Approximation quality was evaluated by the *signal-to-noise ratio* (SNR). Offline approximation and runtime rendering timings were measured on a workstation with an AMD Ryzen Threadripper 2950X CPU, an NVIDIA TITAN RTX graphics card, and 128 GB main memory.

5.1 SVBRDFs

Experiment Settings. A SVBRDF describes the reflectance distributions of different points on a surface. It is often represented as a 6D function of an illumination direction ω_l, a view direction ω_v, and a surface point p (or a texel) with 2D spatial coordinates (x, y). When an incident light ray hits p in ω_l and is reflected in ω_v, a SVBRDF returns the ratio of reflected radiance to irradiance. Since a SVBRDF is a multidimensional dataset, tensor representations are suitable to approximate it for efficient runtime rendering. In our experiments, a SVBRDF was resampled and organized as a fourth-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_{\omega_l} \times I_{\omega_v} \times I_x \times I_y}$ for approximations using MK-CTA and HMK-CTA.

Results. Table 1 compares the statistics of SVBRDF approximation results for the MK-CTA baseline and HMK-CTA. Note that the MK-CTA baseline can be regarded as a special case of HMK-CTA, with only one hierarchical level. As for HMK-CTA, there are three hierarchical levels, where the input tensor (at the first level) is approximated using HOSVD (without sparse clustering and subdivision), and residual tensors at the second and third levels are approximated using MK-CTA. We especially choose the numbers of clusters at the second and third levels to match the numbers of clusters for the MK-CTA baseline. For example, $C_x^{(2)} \cdot C_x^{(3)}$ of HMK-CTA is equal to $C_x^{(1)}$ of the MK-CTA baseline. Therefore, the approximation quality of MK-CTA and HMK-CTA is expected to be similar.

When compared to the MK-CTA baseline, HMK-CTA can significantly reduce offline approximation time by more than 80% (or even more than 90%). The amount of approximated data is smaller, and rendering rates are also higher. Although the approximation quality (in terms of the SNR) of HMK-CTA may be decreased, the

Table 1: Statistics of SVBRDF approximations.

SVBRDF	Dove		Season Greetings		Wallpaper1		Wallpaper2	
$I_{\omega_l} \times I_{\omega_v} \times I_x \times I_y$	$58 \times 58 \times 510 \times 470$		$58 \times 58 \times 500 \times 523$		$58 \times 58 \times 375 \times 480$		$58 \times 58 \times 310 \times 390$	
Raw data (GB)	9.01		9.83		6.77		4.55	
Approximate method	MK-CTA	HMK-CTA	MK-CTA	HMK-CTA	MK-CTA	HMK-CTA	MK-CTA	HMK-CTA
$R^{(1)}_{\omega_l} \times R^{(1)}_{\omega_v} \times R^{(1)}_x \times R^{(1)}_y$	$3 \times 3 \times 5 \times 5$	$6 \times 6 \times 10 \times 10$	$3 \times 3 \times 4 \times 6$	$6 \times 6 \times 8 \times 12$	$2 \times 1 \times 2 \times 2$	$4 \times 2 \times 4 \times 4$	$3 \times 3 \times 5 \times 6$	$6 \times 6 \times 10 \times 12$
$C^{(1)}_{\omega_l} \times C^{(1)}_{\omega_v} \times C^{(1)}_x \times C^{(1)}_y$	$12 \times 9 \times 16 \times 16$	-	$16 \times 12 \times 16 \times 16$	-	$12 \times 16 \times 16 \times 16$	-	$12 \times 9 \times 16 \times 16$	-
$K^{(1)}_{\omega_l} \times K^{(1)}_{\omega_v} \times K^{(1)}_x \times K^{(1)}_y$	$3 \times 3 \times 3 \times 3$	-	$3 \times 3 \times 3 \times 3$	-	$3 \times 3 \times 3 \times 3$	-	$3 \times 3 \times 3 \times 3$	-
$R^{(2)}_{\omega_l} \times R^{(2)}_{\omega_v} \times R^{(2)}_x \times R^{(2)}_y$	-	$3 \times 3 \times 5 \times 5$	-	$3 \times 3 \times 4 \times 6$	-	$2 \times 1 \times 2 \times 2$	-	$3 \times 3 \times 5 \times 6$
$C^{(2)}_{\omega_l} \times C^{(2)}_{\omega_v} \times C^{(2)}_x \times C^{(2)}_y$	-	$4 \times 3 \times 4 \times 4$	-	$4 \times 4 \times 4 \times 4$	-	$4 \times 4 \times 4 \times 4$	-	$4 \times 3 \times 4 \times 4$
$K^{(2)}_{\omega_l} \times K^{(2)}_{\omega_v} \times K^{(2)}_x \times K^{(2)}_y$	-	$2 \times 2 \times 2 \times 2$	-	$2 \times 2 \times 2 \times 2$	-	$2 \times 2 \times 2 \times 2$	-	$2 \times 2 \times 2 \times 2$
$R^{(3)}_{\omega_l} \times R^{(3)}_{\omega_v} \times R^{(3)}_x \times R^{(3)}_y$	-	$3 \times 3 \times 5 \times 5$	-	$3 \times 3 \times 4 \times 6$	-	$2 \times 1 \times 2 \times 2$	-	$3 \times 3 \times 5 \times 6$
$C^{(3)}_{\omega_l} \times C^{(3)}_{\omega_v} \times C^{(3)}_x \times C^{(3)}_y$	-	$3 \times 3 \times 4 \times 4$	-	$4 \times 3 \times 4 \times 4$	-	$3 \times 4 \times 4 \times 4$	-	$3 \times 3 \times 4 \times 4$
$K^{(3)}_{\omega_l} \times K^{(3)}_{\omega_v} \times K^{(3)}_x \times K^{(3)}_y$	-	$2 \times 2 \times 2 \times 2$	-	$2 \times 2 \times 2 \times 2$	-	$2 \times 2 \times 2 \times 2$	-	$2 \times 2 \times 2 \times 2$
Approximated data (MB)	70.26	51.97	123.82	90.89	39.99	26.34	62.64	47.5
SNR (dB)	25.31	24.99	26.67	29.52	18.75	18.7	28.71	28.16
Approximation time (hr.)	363.62	65.78	663.92	99.67	617.62	84.28	416.35	20.03
Rendering rate (FPS)	1.52	4.56	1.97	4.31	33.41	97.25	2.75	7.42

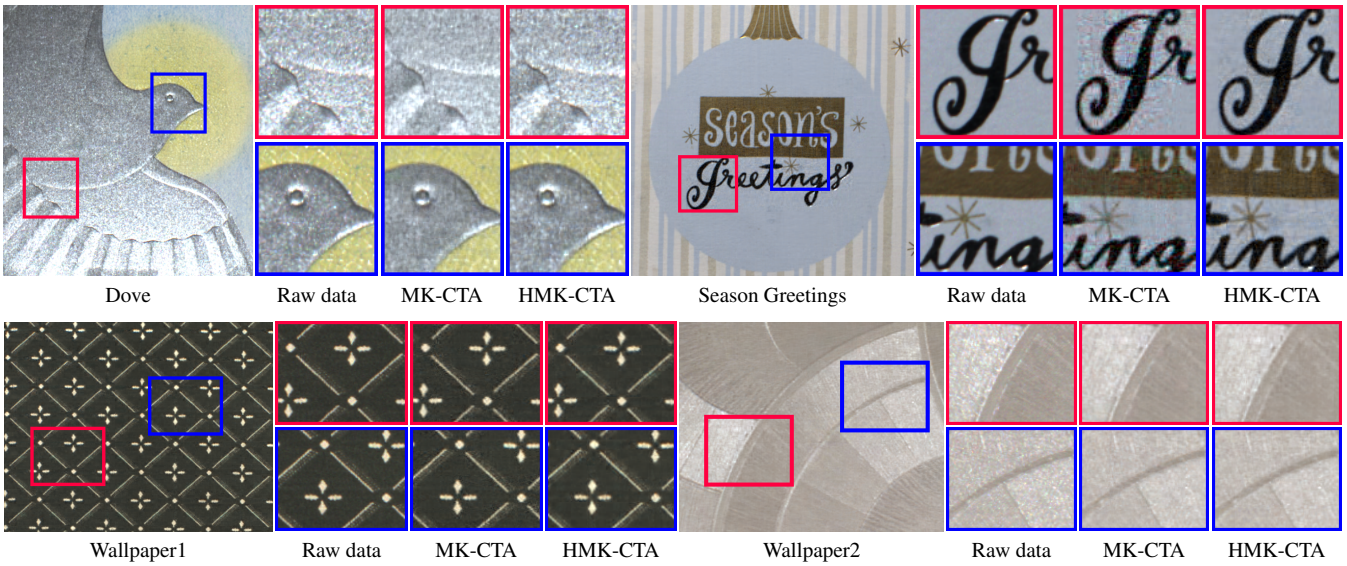


Figure 5: Reconstructed images of SVBRDF approximations. Please refer to Table 1 for statistics and parameter configurations.

difference from MK-CTA is often small. Note that for the SVBRDF "Season Greetings", its approximation quality is remarkably increased from 26.67 dB to 29.52 dB by using HMK-CTA. We found that when the numbers of clusters are larger, it is more difficult to find a good (locally) optimal solution to MK-CTA, even though increasing the numbers of mixture clusters can alleviate this issue.

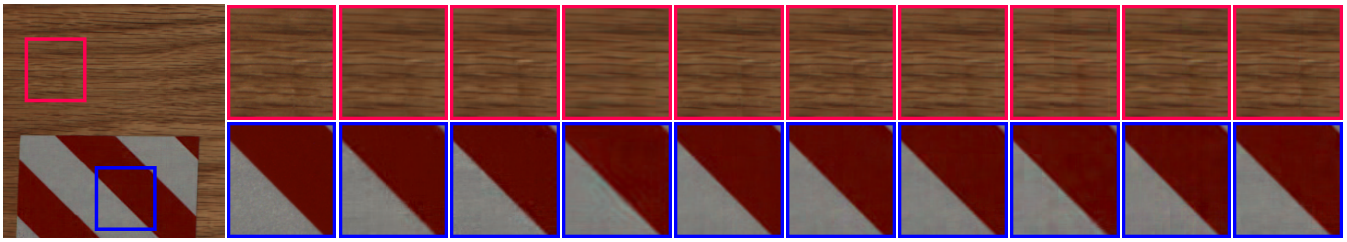
Fig. 5 further demonstrates the visual quality of reconstructed SVBRDF images for the MK-CTA baseline and HMK-CTA. From enlarged images, differences between the two methods are often not perceptible. Nevertheless, for the SVBRDF "Season Greetings", there are visible artifacts around sharp features in the reconstructed images of MK-CTA. For example, noise and ringing effects can be apparently observed at boundaries of some English letters. The images of MK-CTA are also slightly more blurry than those of HMK-CTA.

Table 2 reveals the statistics of approximation results for the SVBRDF "Wood Tape" based on different configurations of MK-CTA and HMK-CTA. For a more comprehensive comparison, we change all the numbers of mixture clusters for MK-CTA from 3 to 2. As a result, approximation time is substantially reduced (but still very long), quality is slightly decreased, and rendering performance significantly improves.

As for HMK-CTA, there are three configurations (WH1, WH2, and WH3) with only two levels. Among them, by increasing the numbers of clusters, approximation quality and time are moderately increased, while rendering rates are just slightly reduced. On the other hand, by comparing WH4 to WH2, there is an additional hierarchical level for WH4, namely the first level, where HOSVD is applied to the input SVBRDF such that the approximation time and rendering rates of WH2 and WH4 are very similar. Nevertheless, the approximation quality of WH4 is slightly decreased. From WH4 and WH7, the effects of increasing reduced ranks can be observed. Approximation quality is improved, while the rendering rate is reduced. The decrease in approximation time can be ignored,

Table 2: Statistics of SVBRDF approximations ("Wood Tape").

SVBRDF	Wood Tape								
$I_{\omega_l} \times I_{\omega_v} \times I_x \times I_y$	$58 \times 58 \times 400 \times 380$								
Raw data (GB)	5.71								
Configuration	WM1	WM2	WH1	WH2	WH3	WH4	WH5	WH6	WH7
Approximation method	MK-CTA	MK-CTA	HMK-CTA	HMK-CTA	HMK-CTA	HMK-CTA	HMK-CTA	HMK-CTA	HMK-CTA
Subdivision scheme	-	-	Most	Most	Most	Most	Least	Uniform	Most
$R^{(1)}_{\omega_l} \times R^{(1)}_{\omega_v} \times R^{(1)}_x \times R^{(1)}_y$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$4\times4\times6\times6$	$4\times4\times6\times6$	$4\times4\times6\times6$	$4\times4\times8\times8$
$C^{(1)}_{\omega_l} \times C^{(1)}_{\omega_v} \times C^{(1)}_x \times C^{(1)}_y$	$12\times9\times16\times16$	$12\times9\times16\times16$	$3\times3\times3\times3$	$4\times3\times4\times4$	$4\times3\times4\times4$	-	-	-	-
$K^{(1)}_{\omega_l} \times K^{(1)}_{\omega_v} \times K^{(1)}_x \times K^{(1)}_y$	$2\times2\times2\times2$	$3\times3\times3\times3$	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$	-	-	-	-
$R^{(2)}_{\omega_l} \times R^{(2)}_{\omega_v} \times R^{(2)}_x \times R^{(2)}_y$	-	-	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times4\times4$
$C^{(2)}_{\omega_l} \times C^{(2)}_{\omega_v} \times C^{(2)}_x \times C^{(2)}_y$	-	-	$3\times3\times3\times3$	$3\times3\times4\times4$	$4\times3\times4\times4$	$4\times3\times4\times4$	$4\times3\times4\times4$	$4\times3\times4\times4$	$4\times3\times4\times4$
$K^{(2)}_{\omega_l} \times K^{(2)}_{\omega_v} \times K^{(2)}_x \times K^{(2)}_y$	-	-	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$
$R^{(3)}_{\omega_l} \times R^{(3)}_{\omega_v} \times R^{(3)}_x \times R^{(3)}_y$	-	-	-	-	-	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times3\times3$	$2\times2\times4\times4$
$C^{(3)}_{\omega_l} \times C^{(3)}_{\omega_v} \times C^{(3)}_x \times C^{(3)}_y$	-	-	-	-	-	$3\times3\times4\times4$	$3\times3\times4\times4$	$3\times3\times4\times4$	$3\times3\times4\times4$
$K^{(3)}_{\omega_l} \times K^{(3)}_{\omega_v} \times K^{(3)}_x \times K^{(3)}_y$	-	-	-	-	-	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$	$2\times2\times2\times2$
Approximated data (MB)	21.74	31.66	8.56	22.3	28.53	22.31	22.31	22.31	29.09
SNR (dB)	27.6	28.31	24.77	27.62	28.07	27.04	26.04	25.46	27.89
Approximation time (hr.)	73.52	518.25	14.37	22.4	21.93	23.2	21.8	16.48	20.7
Rendering rate (FPS)	72.06	14.92	38.14	37.65	37.16	36.82	36.53	36.72	22.34



Wood Tape | Raw data | MK-CTA (WM1) | MK-CTA (WM2) | HMK-CTA (WH1) | HMK-CTA (WH2) | HMK-CTA (WH3) | HMK-CTA (WH4) | HMK-CTA (WH5) | HMK-CTA (WH6) | HMK-CTA (WH7)

Figure 6: Reconstructed images of SVBRDF approximations ("Wood Tape"). Please refer to Table 2 for statistics and parameter configurations.

since it is mostly due to less iterations until convergence.

Finally, WH4, WH5, and WH6 compare the three subdivision schemes, including the most/least significant and uniform ones. The most significant scheme outperforms others in terms of approximation quality. Although the approximation time of the uniform scheme is the shortest, its quality is the worst. Note that there are almost no differences in the rendering rates, which is not surprising, since runtime performance is mostly related to the number of levels, reduced ranks, and numbers of mixture clusters.

Fig. 6 shows the visual quality of reconstructed SVBRDF images for different configurations of MK-CTA and HMK-CTA in Table 2. In general, the visual quality of each configuration is consistent with its own SNR. Note that the visual quality of configurations WH1, WH5, and WH6 may be among the worst. Specifically, there are ringing effects at sharp boundaries in the reconstructed images of WH1. Moreover, blocky artifacts can be also observed in the images of WH5 and WH6 (the least significant and uniform subdivision schemes).

Discussions. Overall, HMK-CTA provides a good tradeoff among reasonable offline approximation cost, low reconstruction errors, high visual quality, fast rendering rates, and small amounts of approximated data. When compared with MK-CTA, HMK-CTA can especially reduce the offline approximation time by even an order of magnitude, without sacrificing storage space and noticeable approximation/visual quality. Note that the amount of approximated data for HMK-CTA is sometimes smaller than that for MK-CTA (Tables 1 and 2), but the approximation quality of both methods is quite similar to each other. This implies that HMK-CTA may be more effective in exploiting coherence among the input multidimensional data.

Level 1 (7.31 dB) | Level 2 (16.26 dB) | Level 3 (27.04 dB)

Figure 7: Reconstructed images of SVBRDF approximations at different levels for the configuration WH4 (Table 2). The approximation quality at each level (in terms of the SNR) is also shown in parentheses.

Moreover, the hierarchical structure of HMK-CTA also supports progressive reconstruction/rendering at runtime. Fig. 7 demon-

t]

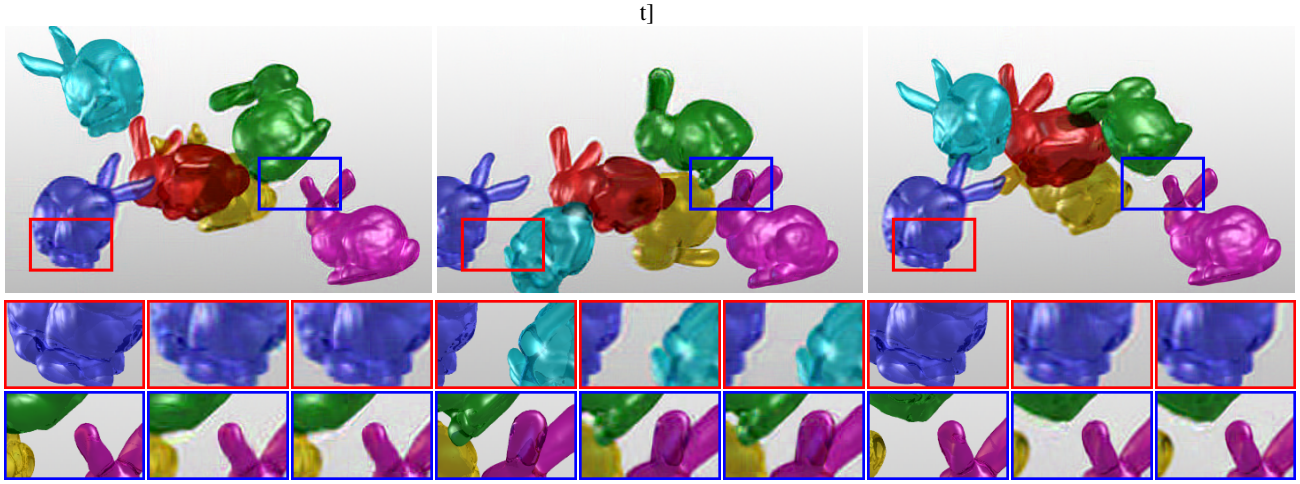


Figure 8: Reconstructed images of TVLF approximations. Please refer to Table 3 for statistics and parameter configurations.

Table 3: Statistics of TVLF approximations.

TVLF	Animated Bunnies	
$I_{\omega_v} \times I_x \times I_y \times I_t$	$81 \times 840 \times 525 \times 89$	
Raw data (GB)	11.84	
Approximation method	MK-CTA	HMK-CTA
$R^{(1)}_{\omega_v} \times R^{(1)}_x \times R^{(1)}_y \times R^{(1)}_t$	$1 \times 2 \times 2 \times 89$	$2 \times 4 \times 4 \times 89$
$C^{(1)}_{\omega_v} \times C^{(1)}_x \times C^{(1)}_y \times C^{(1)}_t$	$36 \times 90 \times 60 \times 1$	-
$K^{(1)}_{\omega_v} \times K^{(1)}_x \times K^{(1)}_y \times K^{(1)}_t$	$3 \times 3 \times 3 \times 1$	-
$R^{(2)}_{\omega_v} \times R^{(2)}_x \times R^{(2)}_y \times R^{(2)}_t$	-	$1 \times 2 \times 2 \times 89$
$C^{(2)}_{\omega_v} \times C^{(2)}_x \times C^{(2)}_y \times C^{(2)}_t$	-	$6 \times 10 \times 10 \times 1$
$K^{(2)}_{\omega_v} \times K^{(2)}_x \times K^{(2)}_y \times K^{(2)}_t$	-	$2 \times 2 \times 2 \times 1$
$R^{(3)}_{\omega_v} \times R^{(3)}_x \times R^{(3)}_y \times R^{(3)}_t$	-	$1 \times 2 \times 2 \times 89$
$C^{(3)}_{\omega_v} \times C^{(3)}_x \times C^{(3)}_y \times C^{(3)}_t$	-	$6 \times 9 \times 6 \times 1$
$K^{(3)}_{\omega_v} \times K^{(3)}_x \times K^{(3)}_y \times K^{(3)}_t$	-	$2 \times 2 \times 2 \times 1$
Approximated data (MB)	174.74	161.65
SNR (dB)	26.76	25.93
Approximation time (hr.)	530.2	24.47
Rendering rate (FPS)	324.42	470.83

strates the reconstructed images of the SVBRDF "Wood Tape" at different levels. Higher-frequency and smaller-scale visual features/details can be observed in the reconstructed images at lower levels. Although due to parameter configurations, the approximation quality at the first level is not very good both visually and in terms of the SNR, the quality at the second level improves to an acceptable level. In order to increase approximation quality, one can always apply different parameter configurations at the first level, such that the employed method will be changed from HOSVD to MK-CTA. Nevertheless, runtime rendering rates may be slightly reduced.

5.2 Time-Varying Light Fields

Experiment Settings. A TVLF describes temporal radiance distributions at different positions on a image/plane. It is often represented as a 5D function of a view direction ω_v, an image pixel p with 2D spatial coordinates (x, y), and a time/frame index t. When viewing a 3D scene in ω_v from p at frame t, the output value of a TVLF is the received radiance at p. In our experiments, a TVLF was organized as a fourth-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_{\omega_v} \times I_x \times I_y \times I_t}$ for approximations using MK-CTA and HMK-CTA. For both methods, we follow the suggestion in [20] not to decompose the time mode due to low temporal coherence.

Results. Table 3 lists the statistics of TVLF approximations for the MK-CTA baseline and HMK-CTA. We also determine parameters of both algorithms such that their approximation quality is similar. From Table 3, the approximation time of HMK-CTA is substantially lower than that of MK-CTA, even less than 5%. Note that the rendering performance of HMK-CTA is obviously better. Although determining parameters of HMK-CTA is more complicated, it also simultaneously provides more flexibility for fine-tuning rendering rates. Moreover, the differences in approximation quality and the amount of approximated data between both methods are not significant. This indicates that HMK-CTA can achieve a better compromise, especially for reducing offline computational cost.

The reconstructed TVLF images of both algorithms are shown in Fig. 8. Even from enlarged images, it is still difficult to identify which algorithm can lead to visually more pleasant results. In general, there are some blocky artifacts in the reconstructed images of HMK-CTA, while the results of MK-CTA are more blurry and exhibit more ringing effects at boundaries of bunnies. Please refer to the supplemental video for the animated result of HMK-CTA.

5.3 Time-Varying Volume Data

Experiment Settings. A TVVD set describes a dynamic field with respect to different positions in a 3D space. In this paper, since we only consider scalar fields, it can be regarded as a 4D function of a voxel v with 3D spatial coordinates (x, y, z) and a time/frame index t. For v at frame t, a TVVD set returns a scalar value that represents a certain (physical) quantity, for example, vorticity, density, or energy. In our experiments, a TVVD set was organized as a fourth-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_x \times I_y \times I_z \times I_t}$ for approximations using MK-CTA and HMK-CTA. Note that we also did not decompose the time mode due to low temporal coherence.

Results. The statistics of TVVD approximations for the MK-CTA baseline and HMK-CTA are shown in Table 4. Under similar approximation quality, the amount of approximated data and rendering rates of both algorithms are also close to each other. Never-

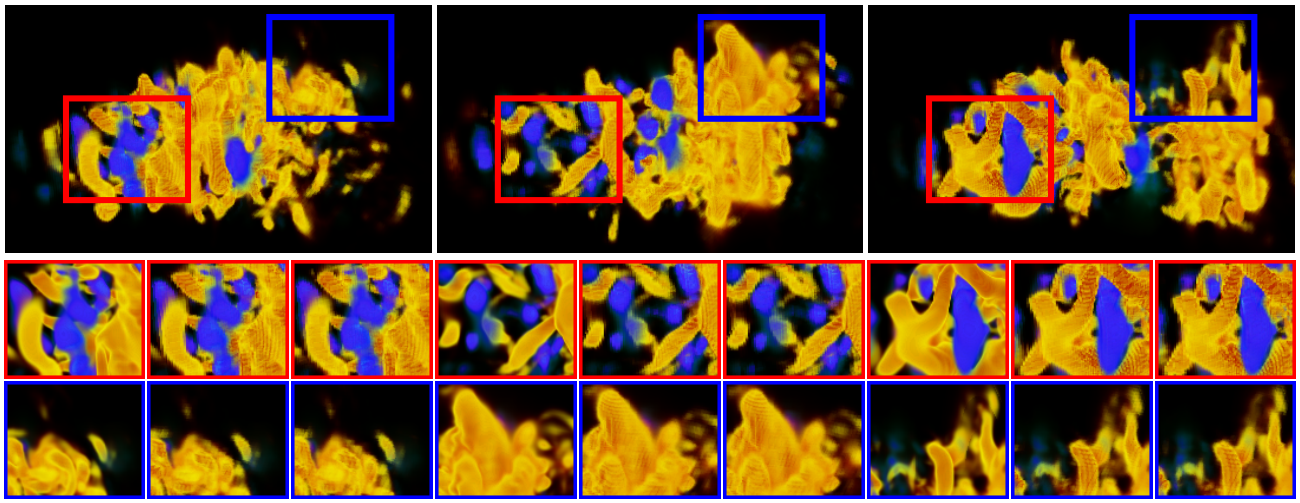


Figure 9: Visualization results of TVVD approximations. Please refer to Table 4 for statistics and parameter configurations.

Table 4: Statistics of TVVD approximations.

TVVD	Turbulent Jet	
$I_x \times I_y \times I_z \times I_t$	$258 \times 258 \times 208 \times 150$	
Raw data (GB)	7.74	
Approximation method	MK-CTA	HMK-CTA
$R_x^{(1)} \times R_y^{(1)} \times R_z^{(1)} \times R_t^{(1)}$	$1 \times 1 \times 1 \times 150$	$2 \times 2 \times 2 \times 150$
$C_x^{(1)} \times C_y^{(1)} \times C_z^{(1)} \times C_t^{(1)}$	$36 \times 36 \times 64 \times 1$	-
$K_x^{(1)} \times K_y^{(1)} \times K_z^{(1)} \times K_t^{(1)}$	$3 \times 3 \times 3 \times 1$	-
$R_x^{(2)} \times R_y^{(2)} \times R_z^{(2)} \times R_t^{(2)}$	-	$1 \times 1 \times 1 \times 150$
$C_x^{(2)} \times C_y^{(2)} \times C_z^{(2)} \times C_t^{(2)}$	-	$6 \times 6 \times 8 \times 1$
$K_x^{(2)} \times K_y^{(2)} \times K_z^{(2)} \times K_t^{(2)}$	-	$2 \times 2 \times 2 \times 1$
$R_x^{(3)} \times R_y^{(3)} \times R_z^{(3)} \times R_t^{(3)}$	-	$1 \times 1 \times 1 \times 150$
$C_x^{(3)} \times C_y^{(3)} \times C_z^{(3)} \times C_t^{(3)}$	-	$6 \times 6 \times 8 \times 1$
$K_x^{(3)} \times K_y^{(3)} \times K_z^{(3)} \times K_t^{(3)}$	-	$2 \times 2 \times 2 \times 1$
Approximated data (MB)	23.56	23.53
SNR (dB)	17.73	17.52
Approximation time (hr.)	106.95	14.12
Rendering rate (FPS)	340.32	364.28

theless, HMK-CTA can reduce more than 85% of the approximation time of MK-CTA. As for visual quality, the enlarged images in Fig. 9 reveal that there are no significant differences in the visualization results of both algorithms. Although some high-frequency and sharp visual features are not faithfully captured by both algorithms, one can always adjust parameters, such as increasing numbers of clusters, in order to reduce approximation errors. Please also refer to the accompanying video for the animated visualization result of HMK-CTA.

6 Conclusions

This paper introduces a novel hierarchical multilinear model, namely HMK-CTA, to achieve efficient, progressive, compact, and high-quality approximations for large-scale multidimensional visual datasets. When compared to MK-CTA, HMK-CTA can significantly reduce offline approximation time by even an order of magnitude, while retaining similar approximation quality, rendering rates, and storage requirements. Its hierarchical structure particularly contributes to capturing multiscale visual features among datasets.

In the future, we would like to develop an automatic method for determining various parameters of HMK-CTA based on a few specified constraints, for example, a desired rendering rate and/or SNR. We are also interested in accelerating the offline approximation process of HMK-CTA by utilizing graphics processing units or parallel/cloud computing techniques.

Acknowledgments

This work was supported in part by the National Science and Technology Council of Taiwan under Grant Numbers MOST109-2221-E-155-039-MY3 and NSTC112-2221-E-155-022.

References

[1] R. Ballester-Ripoll and R. B. Pajarola. Compressing Bidirectional Texture Functions via Tensor Train Decomposition. In *Proc. of PG '16 - Short Papers*, pp. 19–22, 2016. doi: 10.2312/pg.20161329 1, 2

[2] L. Belcour, G. Xie, C. Hery, M. Meyer, W. Jarosz, and D. Nowrouzezahrai. Integrating Clipped Spherical Harmonics Expansions. *ACM Trans. Graph.*, 37(2), 2018. doi: 10.1145/3015459 1

[3] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the Best Rank-1 and Rank-$(R_1, R_2, \ldots, R_n)$ Approximation of Higher-Order Tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000. doi: 10.1137/S0895479898346995 1, 2

[4] J. Filip and M. Haindl. Bidirectional Texture Function Modeling: A State of the Art Survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):1921–1940, 2009. doi: 10.1109/TPAMI.2008.246 1

[5] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong. Deep Inverse Rendering for High-Resolution SVBRDF Estimation from an Arbitrary Number of Images. *ACM Trans. Graph.*, 38(4), 2019. doi: 10.1145/3306346.3323042 1

[6] E. Heitz, J. Dupuy, S. Hill, and D. Neubelt. Real-Time Polygonal-Light Shading with Linearly Transformed Cosines. *ACM Trans. Graph.*, 35(4), 2016. doi: 10.1145/2897824.2925895 1

[7] J. Lawrence, A. Ben-Artzi, C. DeCoro, W. Matusik, H. Pfister, R. Ramamoorthi, and S. Rusinkiewicz. Inverse Shade Trees for Non-Parametric Material Representation and Editing. *ACM Trans. Graph.*, 25(3):735–745, 2006. doi: 10.1145/1141911.1141949 1, 4

[8] S. Lombardi, J. M. Saragih, T. Simon, and Y. A. Sheikh. Deep Appearance Models for Face Rendering. *ACM Trans. Graph.*, 37(4), 2018. doi: 10.1145/3197517.3201401 1

[9] R. S. Overbeck, D. Erickson, D. Evangelakos, M. M. Pharr, and P. E. Debevec. A System for Acquiring, Processing, and Rendering Panoramic Light Field Stills for Virtual Reality. *ACM Trans. Graph.*, 37(6), 2018. doi: 10.1145/3272127.3275031 1

[10] G. Rainer, W. Jakob, A. Ghosh, and T. Weyrich. Neural BTF Compression and Interpolation. *Comput. Graph. Forum*, 38(2):235–244, 2019. doi: 10.1111/cgf.13633 1

[11] R. Ramamoorthi. Precomputation-Based Rendering. *Foundations and Trends in Computer Graphics and Vision*, 3(4):281–369, 2009. doi: 10.1561/0600000021 1

[12] R. Ruiters and R. Klein. BTF Compression via Sparse Tensor Decomposition. *Comput. Graph. Forum*, 28(4):1181–1188, 2009. doi: 10.1111/j.1467-8659.2009.01495.x 2

[13] M. Rump, R. Sarlette, and R. Klein. Groundtruth Data for Multispectral Bidirectional Texture Functions. In *Proc. of CGIV '10*, pp. 326–331, 2010. 1

[14] A. Smilde, R. Bro, and P. Geladi. *Multiway Analysis: Applications in the Chemical Sciences*. Wiley Press, 2004. 1

[15] B. Sun and R. Ramamoorthi. Affine Double- and Triple-Product Wavelet Integrals for Rendering. *ACM Trans. Graph.*, 28(2), 2009. doi: 10.1145/1516522.1516525 1

[16] B. Sun, K. Sunkavalli, R. Ramamoorthi, P. N. Belhumeur, and S. K. Nayar. Time-Varying BRDFs. *IEEE Trans. Vis. Comput. Graph.*, 13(3):595–609, 2007. doi: 10.1109/TVCG.2007.1013 1

[17] S. K. Suter, J. A. Iglesias Guitián, F. Marton, M. Agus, A. Elsener, C. P. E. Zollikofer, M. Gopi, E. Gobbetti, and R. B. Pajarola. Interactive Multiscale Tensor Reconstruction for Multiresolution Volume Visualization. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2135–2143, 2011. doi: 10.1109/TVCG.2011.214 2, 3

[18] S. K. Suter, M. Makhynia, and R. B. Pajarola. TAMRESH - Tensor Approximation Multiresolution Hierarchy for Interactive Volume Visualization. *Comput. Graph. Forum*, 32(3):151–160, 2013. doi: 10.1111/cgf.12102 2, 3

[19] A. Toisoul and A. Ghosh. Practical Acquisition and Rendering of Diffraction Effects in Surface Reflectance. *ACM Trans. Graph.*, 36(5), 2017. doi: 10.1145/3012001 1

[20] Y.-T. Tsai. Multiway K-Clustered Tensor Approximation: Toward High-Performance Photorealistic Data-Driven Rendering. *ACM Trans. Graph.*, 34(5), 2015. doi: 10.1145/2753756 1, 2, 7

[21] Y.-T. Tsai, K.-L. Fang, W.-C. Lin, and Z.-C. Shih. Modeling Bidirectional Texture Functions with Multivariate Spherical Radial Basis Functions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(7):1356–1369, 2011. doi: 10.1109/TPAMI.2010.211 1

[22] Y.-T. Tsai and Z.-C. Shih. All-Frequency Precomputed Radiance Transfer Using Spherical Radial Basis Functions and Clustered Tensor Approximation. *ACM Trans. Graph.*, 25(3):967–976, 2006. doi: 10.1145/1141911.1141981 2

[23] Y.-T. Tsai and Z.-C. Shih. K-Clustered Tensor Approximation: A Sparse Multilinear Model for Real-Time Rendering. *ACM Trans. Graph.*, 31(3), 2012. doi: 10.1145/2167076.2167077 1, 2

[24] M. A. O. Vasilescu and D. Terzopoulos. TensorTextures: Multilinear Image-Based Rendering. *ACM Trans. Graph.*, 23(3):336–342, 2004. doi: 10.1145/1015706.1015725 1

[25] B. Wang, X. Meng, and T. Boubekeur. Wavelet Point-Based Global Illumination. *Comput. Graph. Forum*, 34(4):143–153, 2015. doi: 10.1111/cgf.12686 1

[26] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja. Out-of-Core Tensor Approximation of Multidimensional Matrices of Visual Data. *ACM Trans. Graph.*, 24(3):527–535, 2005. doi: 10.1145/1186822.1073224 1

[27] J. Wang and R. Ramamoorthi. Analytic Spherical Harmonic Coefficients for Polygonal Area Lights. *ACM Trans. Graph.*, 37(4), 2018. doi: 10.1145/3197517.3201291 1

[28] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo. All-Frequency Rendering of Dynamic, Spatially-Varying Reflectance. *ACM Trans. Graph.*, 28(5), 2009. doi: 10.1145/1618452.1618479 1

[29] G. Wetzstein, D. Lanman, M. Hirsch, and R. Raskar. Tensor Displays: Compressive Light Field Synthesis Using Multilayer Displays with Directional Backlighting. *ACM Trans. Graph.*, 31(4), 2012. doi: 10.1145/2185520.2185576 1, 4

[30] Q. Wu, T. Xia, C. Chen, H.-Y. S. Lin, H. Wang, and Y. Yu. Hierarchical Tensor Approximation of Multidimensional Visual Data. *IEEE Trans. Vis. Comput. Graph.*, 14(1):186–199, 2008. doi: 10.1109/TVCG.2007.70406 2, 3

[31] K. Xu, W.-L. Sun, Z. Dong, D.-Y. Zhao, R.-D. Wu, and S.-M. Hu. Anisotropic Spherical Gaussians. *ACM Trans. Graph.*, 32(6), 2013. doi: 10.1145/2508363.2508386 1

[32] L.-Q. Yan, C.-W. Tseng, H. W. Jensen, and R. Ramamoorthi. Physically-Accurate Fur Reflectance: Modeling, Measurement and Rendering. *ACM Trans. Graph.*, 34(6), 2015. doi: 10.1145/2816795.2818080 1