K-Clustered Tensor Approximation: A Sparse Multilinear Model for Real-Time Rendering

YU-TING TSAI
Yuan Ze University
and
ZEN-CHUNG SHIH
National Chiao Tung University

With the increasing demands for photo-realistic image synthesis in real time, we propose a sparse multilinear model, which is named *K-clustered tensor approximation* (K-CTA), to efficiently analyze and approximate large-scale multidimensional visual data sets, so that both storage space and rendering time are substantially reduced. K-CTA not only extends previous work on *clustered tensor approximation* (CTA) to exploit inter-cluster coherence, but also allows a compact and sparse representation for high-dimensional data sets with just a few low-order factors and reduced multi-dimensional cluster core tensors. Thus, K-CTA can be regarded as a sparse extension of CTA and a multilinear generalization of sparse representation. Experimental results demonstrate that K-CTA can accurately approximate spatially varying visual data sets, such as bidirectional texture functions, view-dependent occlusion texture functions, and bi-scale radiance transfer functions for efficient rendering in real-time applications.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*; E.4 [**Data**]: Coding and Information Theory—*Data compaction and compression*

General Terms: Algorithms

Additional Key Words and Phrases: Real-Time Rendering, Multidimensional Data Analysis, Tensor Approximation, Sparse Representation

This work was supported in part by the National Science Council of Taiwan under Grant No. NSC100-2218-E-155-002 and NSC100-2221-E-155-078.
Authors' addresses: Y.-T. Tsai, Department of Computer Science and Engineering, Yuan Ze University, 135 Yuan-Tung Road, Chung-Li City, Taoyuan, Taiwan 320, R.O.C.; email: yttsai@saturn.yzu.edu.tw; Z.-C. Shih, Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan 300, R.O.C.; email: zcshih@cs.nctu.edu.tw.

DOI 10.1145/2167076.2167077
http://doi.acm.org/10.1145/2167076.2167077

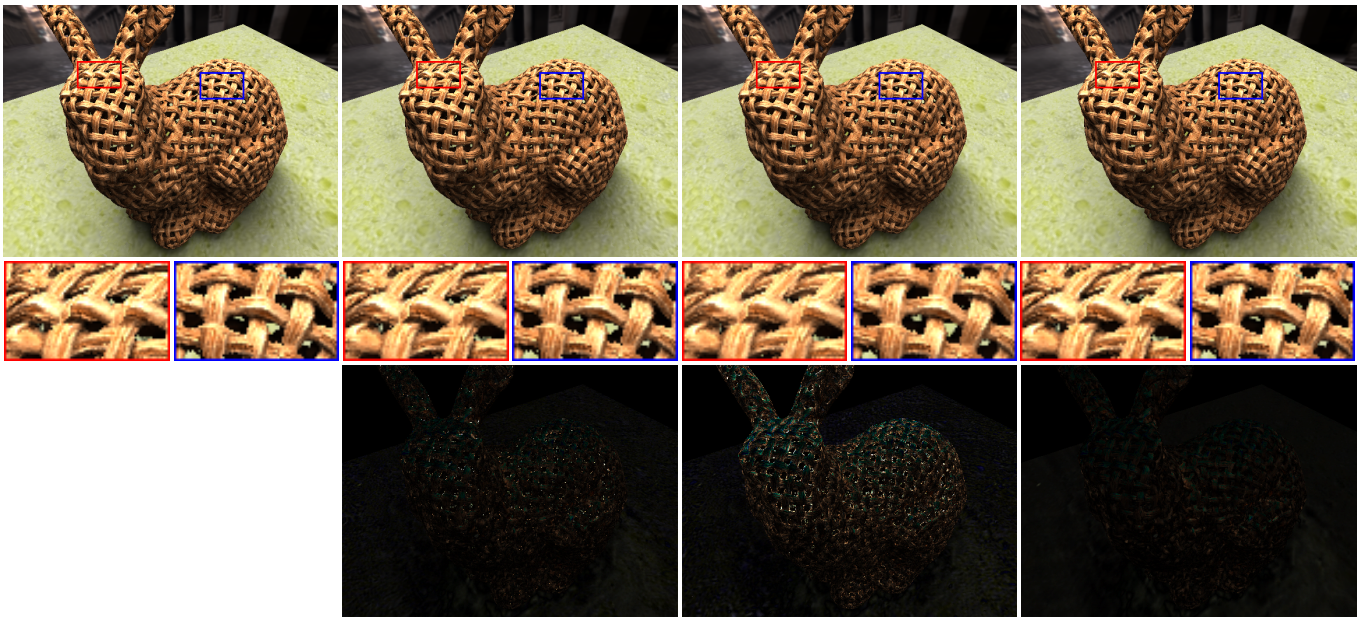
ACM Reference Format:
Tsai, Y.-T. and Shih, Z.-C. 2012. K-Clustered Tensor Approximation: A Sparse Multilinear Model for Real-Time Rendering. ACM Trans. Graph. 31, 3, Article 19 (May 2012), 17 pages.
DOI = 10.1145/2167076.2167077.
http://doi.acm.org/10.1145/2167076.2167077.

1. INTRODUCTION

Synthesizing photo-realistic images is a significant and ambitious goal in computer graphics. Researchers have conventionally focused on developing analytic models and simulation-based algorithms to achieve photo-realistic image synthesis in real time. Nevertheless, real-world object shape, surface reflectance, micro-scale appearance, and natural illumination effects are frequently too complicated to be synthesized using analytic models or simple simulations.

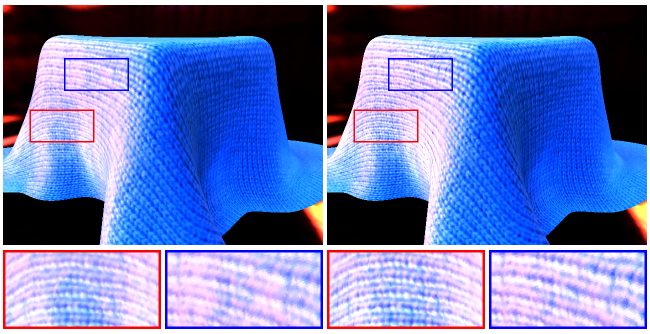
State-of-the-art methods, which are known as *data-driven models*, thus perform rendering from cached or pre-sampled data that represent the results of complex procedures or even real-world measurement. Although data-driven models can avoid computationally expensive procedures at run-time and generate high-quality images, they are usually subject to cumbersome pre-sampled observations that consume a large amount of storage space and memory bandwidth. This problem becomes even worse when we have to record more information to account for more degrees of freedom and more detailed descriptions of the desired visual effects. Nowadays, the amount of pre-sampled data often exceeds tens or hundreds of gigabytes, so that the performance of data-driven models may be even slower than directly employing complex procedures.

To solve this problem, we propose a sparse multilinear representation, namely K-CTA, for compressing and rendering large-scale visual data sets. By retaining the structures of the input data as a multidimensional array, namely a tensor, K-CTA extends CTA [Tsai and Shih 2006] to classify each sub-tensor along the clustered mode into more than one cluster, say K_m clusters, so that inter-cluster coherence can be exploited by mixing the decomposed results of these clusters. To reduce run-time computational costs for real-time applications, K-CTA also constrains that each sub-tensor belongs to exactly K_m clusters, thus providing a sparse representation in which the sparsity is totally under user control. In other words, the number of non-zero elements on which each sub-tensor depends is guaranteed as a constant. This especially leads to an easy-to-optimize and efficient shader program on *graphics processing units* (GPUs). Note that one can instead employ an approximation threshold to determine the value of K_m for different sub-tensors. Nevertheless, it usually needs to add redundant zero



(a) Raw data (b) N-SVD (16.71 FPS) (c) CTA (78.39 FPS) (d) K-CTA (71.58 FPS)

Fig. 1. Rendered images of all-frequency BRT based on different tensor representations for meso-structures, including N-SVD [De Lathauwer et al. 2000], CTA [Tsai and Shih 2006], and K-CTA. In each sub-figure, from top to bottom: rendered images; enlarged images; absolute difference images (scaled by a factor of 3). Although both CTA and K-CTA allow real-time rendering performance, CTA sometimes produces noticeable artifacts when the viewpoint/object moves. For the artifacts of CTA, please refer to our accompanying video and Fig. 2. The model *Bunny* was provided in courtesy of Stanford Computer Graphics Laboratory [2011].



(a) CTA (105.35 FPS) (b) K-CTA (101.42 FPS)

Fig. 2. Artifacts of the rendered images based on CTA. In each sub-figure, from top to bottom: rendered images; enlarged images. Although the run-time rendering rates of CTA and K-CTA are similar, CTA may produce noticeable discontinuities when the viewpoint/object moves, especially at small grazing angles of illumination/view directions. Please refer to our accompanying video for more comparisons.

entries for fast run-time rendering performance, in order to avoid dynamic branches on GPUs.

Applications of K-CTA to spatially varying surface appearance, such as *bidirectional texture functions* (BTFs) [Dana et al. 1999], *view-dependent occlusion texture functions* (VOTFs), and *bi-scale radiance transfer* (BRT) [Sloan et al. 2003b], demonstrate the effectiveness and efficiency of K-CTA. Experimental results further reveal that the inter-cluster coherence ignored by CTA is important to the approximation of sub-tensors close to cluster boundaries, since their approximation errors can be compensated by other clusters. Moreover, the sparse property of K-CTA also provides a good comprise between image quality and reconstruction costs.

The remainder of this article is organized as follows. Section 2 reviews the literature on recent advances in dimensionality reduction, tensor approximation, and sparse representation. Next, Section 3 briefly presents preliminaries and background of tensor approximation and CTA for completeness. Then, Section 4, which is the main part of this article, introduces a novel tensor decomposition algorithm, namely K-CTA, to allow a multilinear and sparse representation for multidimensional visual data sets. Section 5 also considers some practical issues of K-CTA, such as initialization, the extraction of global basis matrices, and the degeneracy and convergence problems. Applications of K-CTA to the compression of BTFs, VOTFs, and all-frequency BRT data, with experimental statistics and comparisons to related methods, are demonstrated in Section 6. Finally, Section 7 gives conclusions and future research directions on K-CTA for real-time rendering.

2. RELATED WORK

2.1 Dimensionality Reduction

The high-dimensional "curse" has driven the advances of dimensionality reduction techniques for a long time. Scientists generally assume that low-dimensional manifolds are embedded in their high-dimensional observations, and could be estimated by linear or non-linear transformations. One of the most popular linear models may be *principal component analysis* (PCA) [Jolliffe 2002]. In computer graphics, PCA has been widely adopted to analyze and compress various types of visual data, such as reflectance data [Matusik et al. 2003] and spatially varying appearance models [Sattler et al. 2003; Wang et al. 2003]. In general, linear models are compu-

tationally efficient and easy to implement, but they are inadequate to analyze data sets with non-linear structures.

Apart from linear models, numerous dimensionality reduction algorithms have been proposed to explore non-linear correlations among data, for example local PCA (or clustered PCA) [Kambhatla and Leen 1997], kernel PCA [Schölkopf et al. 1998], isomap [Tenenbaum et al. 2000], locally linear embedding [Roweis and Saul 2000], to name a few. Although the structures of real-world observations are complicated and globally non-linear, many non-linear models assume that local correlations are approximately linear. Thus, the observations could be "locally" transformed into low-dimensional linear sub-spaces without a significant loss of information. Compared to linear models, however, non-linear models are computationally more expensive, and sometimes may be intractable for large-scale data sets. Additionally, not all non-linear models are generative. This frequently prevents some of them from practical applications of data-driven rendering. Despite these disadvantages, successful applications of non-linear models are still prevalent in computer graphics, for instance *precomputed radiance transfer* (PRT) [Sloan et al. 2003a], BTF compression [Müller et al. 2003; 2004], texture synthesis [Lefebvre and Hoppe 2006], and material modeling [Matusik et al. 2003].

2.2 Tensor Approximation

In recent years, tensor approximation (also called multilinear models or multiway analysis) [De Lathauwer et al. 2000; Kolda and Bader 2009; Smilde et al. 2004] has become widespread and caught a lot of attention. It can be regarded as a generalization of *singular value decomposition* (SVD), where data samples are processed in their intrinsic form as a multidimensional array, and separate reduction is allowed along each dimension. Unlike linear and non-linear models in dimensionality reduction, tensor approximation relies on decomposing a high-dimensional space into multiple low-dimensional sub-spaces that are, respectively, associated with each mode of observations to remove the curse of dimensionality. The extracted low-order factors in each sub-space then can be combined to effectively model the original high-dimensional space. In this way, multilinear models successfully preserve the intrinsic structures and important information of observations, and thus overwhelm one of the main disadvantages of previous dimensionality reduction techniques.

In computer graphics, tensor approximation has also been successfully extended [Tsai and Shih 2006; Wu et al. 2008] and applied to various applications, such as data-driven rendering [Sun et al. 2007; Vasilescu and Terzopoulos 2004; Wang et al. 2005] and human facial processing [Vasilescu and Terzopoulos 2003; Vlasic et al. 2005]. Even some matrix factorization methods [Lawrence et al. 2006; Nayar et al. 2004; Suykens et al. 2003] are implicitly related to multilinear models. Although tensor-based methods have been shown more powerful and flexible than linear models, most of them are inadequate for real-time applications. Even after applying the popular N-mode SVD (N-SVD) algorithm [De Lathauwer et al. 2000] to derive an optimal approximation of the input tensor, the amount of compressed data is still too cumbersome for fast run-time rendering. Although CTA can reduce run-time computational costs by dividing input data into disjoint regions, it meanwhile disregards inter-cluster correlations. Moreover, similar to traditional clustering methods, CTA also suffers from an inappropriate initial guess. K-CTA thus targets at overcoming these major drawbacks of CTA by searching for an optimal reconstruction of input tensor across the sub-spaces of different clusters.

2.3 Sparse Representation

Recently, there has been a growing interest in modeling real-world observations as sparse linear combinations of atoms (or basis functions) in an *over-complete* dictionary [Elad et al. 2010; Wright et al. 2010]. Although the underlying physical process of a natural phenomenon may be a complex function or mixture of heterogeneous elements, it is frequently desirable to represent observations in a *sparse* form that allows efficient data analysis. However, this intuitive concept is far from easy to achieve in practice. Even with a fixed dictionary, searching for an optimal solution in which each signal exactly depends on a given number of atoms was proved NP-hard [Davis et al. 1997]. Therefore, many practical algorithms instead consider sub-optimal solutions, such as matching pursuit [Mallat and Zhang 1993], basis pursuit [Chen et al. 2001], and Bayesian models [Kreutz-Delgado et al. 2003].

In addition to pursuit algorithms and Bayesian models, previous studies have also reported a close connection between sparse representation and vector quantization [Kreutz-Delgado et al. 2003]. K-SVD [Aharon et al. 2006] thus generalized K-means clustering to seek sparse representations by alternating between the pursuit process and dictionary learning. Nevertheless, rather than using Bayesian inference, it applied SVD to simultaneously update dictionary atoms and non-zero basis coefficients in the dictionary learning stage, so that the convergence rate could be improved.

In computer graphics, Ruiters and Klein [2009] also developed a sparse representation based on K-SVD for BTF compression. Nevertheless, their method simply applies K-SVD to each dimension of the input BTF, and cannot derive a locally optimal solution. Furthermore, it is also difficult to achieve efficient performance for data-driven rendering applications. By contrast, we propose a novel sparse multilinear model, namely K-CTA, that can easily achieve real-time rendering rates for compressed BTFs. K-CTA also combines the advantages of CTA and K-SVD to bridge the gap between sparse representation and tensor approximation. Several important theorems of K-CTA are proved and discussed, showing that K-CTA resembles the behaviors of K-SVD in the tensor space and is a natural extension of CTA. As a result, K-SVD is further generalized to allow high-dimensional data analysis based on multilinear algebra, without destroying the intrinsic structures of observations.

3. PRELIMINARIES AND BACKGROUND

This section briefly reviews the background of tensor approximation[1] and CTA. For completeness, notation and basic definitions of tensor operators are also introduced. Although the out-of-core tensor decomposition algorithm [Wang et al. 2005] was applied in our experiments, in-core tensor notation is adopted in this article for notational simplicity. For more details about various tensor operators, interested readers may additionally refer to [De Lathauwer et al. 2000; Smilde et al. 2004].

3.1 Basic Definitions

In this article, scalars are written as italic roman lowercase letters ($a, b, \ldots$); vectors as boldface roman lowercase letters ($\mathbf{a}, \mathbf{b}, \ldots$);

[1] Throughout this article, tensor approximation is particularly referred to as the multiway analysis based on Tucker models [Smilde et al. 2004]. Readers may notice that there is another popular multilinear model named *parallel factor analysis* (PARAFAC) or canonical decomposition (CANDECOMP) [Smilde et al. 2004] in chemometrics and psychometrics, but it is beyond the scope of this article.

matrices as boldface roman capitals ($\mathbf{A}, \mathbf{B}, \dots$); tensors as boldface calligraphic capitals ($\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \dots$). The entry in row i and column j of a matrix $\mathbf{U} \in \mathbb{R}^{I \times J}$ is denoted by $(\mathbf{U})_{ij}$; similarly, the entry of an N-th order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by $(\boldsymbol{\mathcal{A}})_{i_1 i_2 \cdots i_N}$. The i-th row of $\mathbf{U}$ is written as $(\mathbf{U})_{i*}$ and the j-th column of $\mathbf{U}$ as $(\mathbf{U})_{*j}$. The transpose of a matrix $\mathbf{U}$ is denoted by $\mathbf{U}^T$.

The Frobenius norm of an N-th order tensor $\boldsymbol{\mathcal{A}}$ is defined as $\|\boldsymbol{\mathcal{A}}\|_F = \sqrt{\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle}$, where

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} (\boldsymbol{\mathcal{A}})_{i_1 i_2 \cdots i_N} (\boldsymbol{\mathcal{B}})_{i_1 i_2 \cdots i_N} \tag{1}$$

denotes the scalar product of two tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The mode-$n$ product of a tensor $\boldsymbol{\mathcal{A}}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ is written as $\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \times_n \mathbf{U}$, where the entries of the resulting tensor $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ are given by

$$(\boldsymbol{\mathcal{B}})_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} (\boldsymbol{\mathcal{A}})_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} (\mathbf{U})_{j_n i_n}. \tag{2}$$

The symbol $uf_n(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1})}$ denotes the mode-n unfolded matrix of an N-th order tensor $\boldsymbol{\mathcal{A}}$, which results from retaining the n-th mode of $\boldsymbol{\mathcal{A}}$ and flattening the others (refer to Fig. 2.1 in [De Lathauwer et al. 2000]).

Moreover, we further define a series of mode-n products by

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{A}} \bigtimes_{n=1}^{N} \mathbf{U}_n = \boldsymbol{\mathcal{A}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N, \tag{3}$$

and $\boldsymbol{\mathcal{A}}_{\langle n_i \rangle} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times 1 \times I_{n+1} \times I_{n+2} \times \cdots \times I_N}$ represents the i-th mode-n sub-tensor of a tensor $\boldsymbol{\mathcal{A}}$, whose entries are defined as

$$\left(\boldsymbol{\mathcal{A}}_{\langle n_i \rangle}\right)_{i_1 i_2 \cdots i_{n-1} 1 i_{n+1} i_{n+2} \cdots i_N} = (\boldsymbol{\mathcal{A}})_{i_1 i_2 \cdots i_{n-1} i i_{n+1} i_{n+2} \cdots i_N}. \tag{4}$$

3.2 Tensor Approximation

Given a set of reduced ranks $\left\{R_n \in \{1, 2, \dots, I_n\}\right\}_{n=1}^{N}$, where R_n is the mode-n reduced rank, tensor approximation decomposes an N-th order tensor $\boldsymbol{\mathcal{A}}$ as a series of mode-n products of a core tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ and a set of N basis matrices $\left\{\mathbf{U}_n \in \mathbb{R}^{I_n \times R_n}\right\}_{n=1}^{N}$, so that the following constrained least-squares optimization problem is resolved:

$$\min_{\left\{\boldsymbol{\mathcal{Z}}, \{\mathbf{U}_n\}_{n=1}^{N}\right\}} \left\| \boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{Z}} \bigtimes_{n=1}^{N} \mathbf{U}_n \right\|_F^2, \text{ subject to } \forall n, \mathbf{U}_n^T \mathbf{U}_n = \mathbf{I}_{R_n}, \tag{5}$$

where $\mathbf{I}_{R_n} \in \mathbb{R}^{R_n \times R_n}$ represents the identity matrix of size $R_n \times R_n$, and $\mathbf{U}_n$ is also known as the mode-n basis matrix.

N-SVD [De Lathauwer et al. 2000] derives a locally optimal solution to (5) using an iterative alternating least-squares algorithm that optimizes only one basis matrix at a time, while leaving other basis matrices unchanged. At the n-th iteration, the mode-n basis matrix $\mathbf{U}_n$ is extracted by retaining the structures of the n-th mode of $\boldsymbol{\mathcal{A}}$, projecting $\boldsymbol{\mathcal{A}}$ onto the basis matrices of other modes, and applying SVD to the mode-n unfolded matrix of the projected tensor. The preceding steps are then repeated until convergence.

3.3 Clustered Tensor Approximation

For an N-th order tensor $\boldsymbol{\mathcal{A}}$, CTA [Tsai and Shih 2006] partitions $\left\{\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}\right\}_{i=1}^{I_m}$, along the clustered mode m, into C disjoint regions to reduce the run-time reconstruction costs of real-time applications. All the mode-m sub-tensors within a cluster c then can be concatenated along the m-th mode into a new tensor and approximated using N-SVD to minimize reconstruction errors from the decomposed core tensor $\boldsymbol{\mathcal{Z}}_c \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ and N basis matrices $\left\{\mathbf{U}_{n,c} \in \mathbb{R}^{I_n \times R_n}\right\}_{n=1}^{N}$ of cluster c.

To obtain a locally optimal solution, the *dual* mode-m basis matrices $\left\{\mathbf{V}_{m,c}\right\}_{c=1}^{C}$ are derived from the core tensor of each cluster to iteratively re-classify $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ into a cluster c_i that minimizes the approximation error of $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$. This is equivalent to maximizing the following objective function[2]:

$$\left\| uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i \rangle} \bigtimes_{\substack{n=1 \\ n \neq m}}^{N} \mathbf{U}_{n,c_i}^T\Big) \mathbf{V}_{m,c_i}^T \right\|_F^2, \tag{6}$$

and the dual mode-m basis matrix of a cluster c is defined as

$$\mathbf{V}_{m,c} = \Big(uf_m(\boldsymbol{\mathcal{Z}}_c)\, uf_m(\boldsymbol{\mathcal{Z}}_c)^T\Big)^{-\frac{1}{2}} uf_m(\boldsymbol{\mathcal{Z}}_c). \tag{7}$$

In this way, CTA groups correlated mode-m sub-tensors, so that tensor approximation can exploit more coherence within a cluster.

4. K-CLUSTERED TENSOR APPROXIMATION

4.1 Algorithm Overview

4.1.1 *Motivations.* One major drawback of CTA is that it enforces *hard* clustering in which each mode-m sub-tensor is classified into just one cluster. The subsequent tensor approximation within each cluster thus can only exploit intra-cluster coherence. In addition, the results of CTA heavily depend on the initial guess of cluster membership, but estimating an appropriate initial guess is a nontrivial problem. Even if the globally optimal solution to hard clustering could be easily found, the decomposed core tensors and basis matrices of different clusters may still have strong correlations.

Our solution to this issue is to relax the hard clustering constraint into a *soft* one. Each mode-m sub-tensor now can be classified into more than one cluster and approximated by mixing the decomposed results of these clusters. To reduce run-time reconstruction costs, this soft constraint should also be a sparse one in which each mode-m sub-tensor belongs to just a few, say K_m, clusters. This not only permits K-CTA to exploit the inter-cluster coherence that cannot be analyzed by CTA, but also alleviates the influence of an inappropriate initial guess by breaking the hard cluster boundaries. Thus, K-CTA can be regarded as a sparse extension of CTA and a multilinear generalization of K-SVD [Aharon et al. 2006].

4.1.2 *Mathematical Formulation.* To allow soft and sparse clustering, K-CTA is formulated as the following constrained least-squares optimization problem:

$$\min_{\left\{\boldsymbol{\mathcal{Z}}_c, \left\{\mathbf{U}_{n,c}\right\}_{n=1}^{N}\right\}_{c=1}^{C}} \left\| \boldsymbol{\mathcal{A}} - \sum_{c=1}^{C} \Big(\boldsymbol{\mathcal{Z}}_c \bigtimes_{n=1}^{N} \mathbf{U}_{n,c} \Big) \right\|_F^2,$$

$$\text{subject to } \begin{cases} \forall i,\ \sum_{c=1}^{C} \left\| \left(\mathbf{U}_{m,c}\right)_{i*} \right\|_0 = K_m R_m, \\ \forall c,\ \forall i,\ \left\| \left(\mathbf{U}_{m,c}\right)_{i*} \right\|_0 \in \{0, R_m\}, \\ \forall c,\ \forall n,\ \mathbf{U}_{n,c}^T \mathbf{U}_{n,c} = \mathbf{I}_{R_n}, \end{cases} \tag{8}$$

[2]Note that Eq. (6) is not derived in [Tsai and Shih 2006], but we will show and prove it in Theorem 1 and Appendix A.1.

Algorithm 1: The K-CTA algorithm.

Input: An N-th order tensor $\mathcal{A}$, a set of reduced ranks $\{R_n\}_{n=1}^N$, the number of clusters C for the clustered mode m, the number of mixture clusters K_m, and the initial guess for $\left\{\mathcal{Z}_c, \{\mathbf{U}_{n,c}\}_{n=1}^N\right\}_{c=1}^C$.

Output: The core tensor and basis matrices of each cluster $\left\{\mathcal{Z}_c, \{\mathbf{U}_{n,c}\}_{n=1}^N\right\}_{c=1}^C$.

repeat

// Clustering stage

for $c \leftarrow 1$ **to** C **do**

Compute $\mathbf{V}_{m,c}$ by (7)

Initialize each entry of $\mathbf{U}_{m,c}$ to zero

for $i \leftarrow 1$ **to** I_m **do** *// Greedy search*

Obtain c_{i_1} of $\mathcal{A}_{\langle m_i\rangle}$ by solving (9)

Update $\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*}$ as (11)

for $k \leftarrow 2$ **to** K_m **do**

Obtain c_{i_k} of $\mathcal{A}_{\langle m_i\rangle}$ by solving (13)

Update $\left\{\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}\right\}_{j=1}^k$ as (15) *// Optimal projection*

for $c \leftarrow 1$ **to** C **do** *// Post-processing*

Decompose $\mathbf{U}_{m,c}$ to obtain $\mathbf{U}'_{m,c}$ and $\mathbf{W}_c$ (Section 4.2.3)

$\mathbf{U}_{m,c} \leftarrow \mathbf{U}'_{m,c}$

$\mathcal{Z}_c \leftarrow \mathcal{Z}_c \times_m \mathbf{W}_c$

// Update stage

for $c \leftarrow 1$ **to** C **do**

Compute $\mathcal{R}_c$ as (20)

$\mathcal{R}'_c \leftarrow \mathcal{R}_c \times_m \mathbf{M}_c^T$ *// Refer to* (21) *and* (22) *for* $\mathbf{M}_c$

Update $\mathcal{Z}_c$ and $\{\mathbf{U}_{n,c}\}_{n=1}^N$ by decomposing $\mathcal{R}'_c$ using N-SVD

$\mathbf{U}_{m,c} \leftarrow \mathbf{M}_c\mathbf{U}_{m,c}$

until $\sum_{c=1}^C \|\mathcal{Z}_c\|_F$ *converges*

where C is the total number of clusters for the clustered mode m, R_n specifies the mode-n reduced rank, $\mathcal{Z}_c \in \mathbb{R}^{R_1\times R_2\times\cdots\times R_N}$ and $\mathbf{U}_{n,c} \in \mathbb{R}^{I_n\times R_n}$, respectively, denote the decomposed core tensor and the mode-n basis matrix of cluster c, K_m represents the number of mixture clusters of each mode-m sub-tensor, and $\|\cdot\|_0$ denotes the ℓ^0 norm of a vector. Note that $\mathbf{U}_{m,c}$ is also called the mixing matrix of cluster c, whose i-th row contains the mixing coefficients of the i-th mode-m sub-tensor, namely $\mathcal{A}_{\langle m_i\rangle}$, with respect to cluster c.

The first and second constraints in (8) enforce that each mode-m sub-tensor belongs to exact K_m mixture clusters, and the entries of the i-th row of $\mathbf{U}_{m,c}$ must be all zeros if $\mathcal{A}_{\langle m_i\rangle}$ is not classified into cluster c. Therefore, all the mode-m basis matrices are sparse and implicitly specify the cluster membership of each mode-m sub-tensor. Note that when $K_m = 1$, K-CTA should derive the same results as CTA to permit it as a natural generalization of CTA.

4.1.3 *Solution Outline.* The proposed iterative K-CTA algorithm is an alternating least-squares approach that consists of two stages: *clustering* and *update* stages. After initializing the core tensor and basis matrices of each cluster, all variables in (8) are fixed in the clustering stage (Section 4.2), except for the mode-m basis matrix of each cluster. For each mode-m sub-tensor, a greedy approach is then applied to sequentially search for the best K_m mixture clusters that minimize its approximation error, and the mixing coefficients are obtained by the proposed optimal projection method. Next, in the update stage (Section 4.3), the core tensor and basis matrices of each cluster are then iteratively updated using N-SVD, one cluster at a time, while those of other clusters are unchanged. Note that we also fix the cluster membership in the update stage to simplify the proposed algorithm. Finally, the aforesaid two stages are iteratively executed until the sum of the Frobenius norm of each cluster core tensor converges, or a user-specified maximum iteration count is reached. The whole process of K-CTA is summarized in Algorithm 1.

4.2 The Clustering Stage

Given the core tensor and basis matrices of each cluster, we would like to find the best K_m mixture clusters of each mode-m sub-tensor and derive the corresponding mixing coefficients in this stage, so that the constraints in (8) are satisfied. This issue can be considered as a multilinear counterpart of the pursuit problem for sparse representation [Aharon et al. 2006; Chen et al. 2001; Mallat and Zhang 1993], which was proved NP-hard [Davis et al. 1997]. Indeed, it is difficult to solve the cluster membership and the nonzero mixing coefficients of a mode-m sub-tensor at the same time, since the decomposed results of different clusters are frequently correlated. Even a single change in the membership of a mode-m sub-tensor, either adding the sub-tensor to or removing it from a cluster, will affect its mixing coefficients for other clusters. Nevertheless, greedy approaches often provide a satisfactory approximate solution to the pursuit problem in both theory and practice [Davis et al. 1997]. We thus sequentially update the cluster membership and sparse mixing coefficients of all the mode-m sub-tensors, one cluster at a time.

4.2.1 *Greedy Search.*

First Mixture Cluster. To allow K-CTA as a natural generalization of CTA, the first mixture cluster of a mode-m sub-tensor is obtained by the following theorem and lemma:

THEOREM 1. *Let* $\mathbf{V}_{m,c} \in \mathbb{R}^{R_m\times(R_{m+1}R_{m+2}\cdots R_N R_1 R_2\cdots R_{m-1})}$ *be the dual mode-*m *basis matrix of cluster* c *[Tsai and Shih 2006]. The first mixture cluster* c_{i_1} *of* $\mathcal{A}_{\langle m_i\rangle}$ *can be obtained by solving the following constrained integer optimization problem:*

$$\max_{c_{i_1}} \left\| uf_m\left(\mathcal{A}_{\langle m_i\rangle}^{(c_{i_1})}\right)\mathbf{V}_{m,c_{i_1}}^T \right\|_F^2, \text{ subject to } c_{i_1} \in \{1, 2, \ldots, C\}, \quad (9)$$

where

$$\mathcal{A}_{\langle m_i\rangle}^{(c_{i_1})} = \mathcal{A}_{\langle m_i\rangle} \bigtimes_{\substack{n=1\\ n\neq m}}^{N}{}_n \mathbf{U}_{n,c_{i_1}}^T. \quad (10)$$

LEMMA 2. *The mixing coefficients for the first mixture cluster* c_{i_1} *of* $\mathcal{A}_{\langle m_i\rangle}$ *are computed as*

$$\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*} = uf_m\left(\mathcal{A}_{\langle m_i\rangle}^{(c_{i_1})}\right) uf_m(\mathcal{Z}_{c_{i_1}})^+, \quad (11)$$

where the superscript "+" specifies the Moore-Penrose pseudoinverse.

The mathematical proofs of Theorem 1 and Lemma 2 are presented in Appendix A.1. Theorem 1 states that $\mathcal{A}_{\langle m_i\rangle}$ should be classified into a cluster whose basis matrices adequately preserve its projected norm, so that the mode-m sub-tensors within a cluster are correlated with each other. Lemma 2 further indicates that the mixing coefficients for the first mixture cluster of $\mathcal{A}_{\langle m_i\rangle}$ is the projection coefficients of $\mathcal{A}_{\langle m_i\rangle}$ onto the sub-spaces of cluster c_{i_1}. Note that the objective function in (9) is equivalent to the objective

function of CTA in (6). This implies that if the number of mixture cluster is set to one, K-CTA will be identical to CTA in the clustering stage.

Remaining Mixture Clusters. After resolving the first mixture cluster of a mode-m sub-tensor, its remaining mixture clusters are then iteratively derived, one mixture cluster at each iteration, from the results of previous iteration. We thus propose the following theorem to settle the k-th mixture cluster of a mode-m sub-tensor:

THEOREM 3. *The k-th mixture cluster c_{i_k} of $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ is resolved from the mixing coefficients for the previously selected $k-1$ mixture clusters $c_{i_1}, c_{i_2}, \ldots, c_{i_{k-1}}$ by minimizing the approximation error of the residual sub-tensor*

$$\boldsymbol{\mathcal{R}}^{(k)}_{\langle m_i \rangle} = \boldsymbol{\mathcal{A}}_{\langle m_i \rangle} - \sum_{j=1}^{k-1} \left(\boldsymbol{\mathcal{Z}}_{c_{i_j}} \times_m \left(\mathbf{U}_{m,c_{i_j}} \right)_{i*} \bigtimes_{\substack{n=1 \\ n \neq m}}^{N}{}_n \mathbf{U}_{n,c_{i_j}} \right), \quad (12)$$

which is equivalent to solving the following constrained integer optimization problem:

$$\max_{c_{i_k}} \left\| uf_m\left(\boldsymbol{\mathcal{A}}^{(c_{i_k})}_{\langle m_i \rangle} \right) \mathbf{V}^T_{m,c_{i_k}} - \sum_{j=1}^{k-1} uf_m\left(\boldsymbol{\mathcal{Z}}^{(c_{i_k})}_{c_{i_j}} \times_m \left(\mathbf{U}_{m,c_{i_j}} \right)_{i*} \right) \mathbf{V}^T_{m,c_{i_k}} \right\|^2_F, \quad (13)$$

subject to $c_{i_k} \in \{1, 2, \ldots, C\}, c_{i_k} \notin \{c_{i_1}, c_{i_2}, \ldots, c_{i_{k-1}}\}$,

where

$$\boldsymbol{\mathcal{Z}}^{(c_{i_k})}_{c_{i_j}} = \boldsymbol{\mathcal{Z}}_{c_{i_j}} \bigtimes_{\substack{n=1 \\ n \neq m}}^{N}{}_n \mathbf{U}^T_{n,c_{i_k}} \mathbf{U}_{n,c_{i_j}}. \quad (14)$$

The mathematical proof of Theorem 3 can be found in Appendix A.2. For each mode-m sub-tensor, the first term of the objective function in (13) is actually the same as the objective function in (9), and the second term instead penalizes a cluster whose basis matrices are correlated to those of previously selected mixture clusters. Therefore, Theorem 3 implies that the k-th mixture cluster of a mode-m sub-tensor is determined by maximizing intra-cluster correlations and minimizing inter-cluster correlations at the same time. This interesting result is similar to the optimized orthogonal matching pursuit approach [Rebollo-Neira and Lowe 2002], where the k-th atom is resolved by simultaneously minimizing its linear dependence with previously selected atoms and maximizing the projected norm of the residual.

Moreover, it is obvious that Eq. (13) can be computed in the reduced tensor space[3] to significantly decrease computational costs. The first term of the objective function in (13) is the projection coefficients of $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ onto the basis matrices and the dual mode-m basis matrix of cluster c_{i_k}. It should be already computed when resolving the first mixture cluster of $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ and remains unchanged during the whole clustering stage. The second term instead can be interpreted as transforming the projected $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ in the sub-spaces of cluster c_{i_j}, for $j = 1, 2, \ldots, k-1$, to the sub-spaces of cluster c_{i_k}, followed by the multiplication with $\mathbf{V}^T_{m,c_{i_k}}$ to obtain projection coefficients. As a result, we can avoid computing the residual sub-tensor in the original tensor space, which needs to first reconstruct the corresponding mode-m sub-tensor from the results of previous iteration.

4.2.2 *Optimal Projection.* Since the sub-spaces of different clusters may be correlated, each time when assigning a new mixture cluster to a mode-m sub-tensor, its mixing coefficients for previously selected mixture clusters should be updated to account for the change in the cluster membership. This guarantees an optimal projection of the mode-m sub-tensor onto the sub-spaces of all selected mixture clusters. We therefore introduce the following theorem to update the mixing coefficients of a mode-m sub-tensor:

THEOREM 4. *The mixing coefficients for the k selected mixture clusters $c_{i_1}, c_{i_2}, \ldots, c_{i_k}$ of the i-th mode-m sub-tensor $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ are given by*

$$\mathbf{u}^{(k)}_{m_i} = \mathbf{Z}^{(k)+}_{m_i} \mathbf{a}^{(k)}_{m_i}, \quad (15)$$

where

$$\mathbf{u}^{(k)}_{m_i} = \left[\left(\mathbf{U}_{m,c_{i_1}} \right)_{i*} \cdots \left(\mathbf{U}_{m,c_{i_k}} \right)_{i*} \right]^T, \quad (16)$$

$$\mathbf{Z}^{(k)}_{m_i} = \begin{bmatrix} f^{(c_{i_1})}_m\left(\boldsymbol{\mathcal{Z}}^{(c_{i_1})}_{c_{i_1}} \right) & \cdots & f^{(c_{i_k})}_m\left(\boldsymbol{\mathcal{Z}}^{(c_{i_k})}_{c_{i_1}} \right) \\ \vdots & \ddots & \vdots \\ f^{(c_{i_1})}_m\left(\boldsymbol{\mathcal{Z}}^{(c_{i_1})}_{c_{i_k}} \right) & \cdots & f^{(c_{i_k})}_m\left(\boldsymbol{\mathcal{Z}}^{(c_{i_k})}_{c_{i_k}} \right) \end{bmatrix}, \quad (17)$$

$$\mathbf{a}^{(k)}_{m_i} = \left[f^{(c_{i_1})}_m\left(\boldsymbol{\mathcal{A}}^{(c_{i_1})}_{\langle m_i \rangle} \right) \cdots f^{(c_{i_k})}_m\left(\boldsymbol{\mathcal{A}}^{(c_{i_k})}_{\langle m_i \rangle} \right) \right]^T, \quad (18)$$

and $f^{(c)}_m(\cdot)$ is the unfolded core projection function of a tensor, which is defined as

$$f^{(c)}_m(\cdot) = uf_m(\cdot) uf_m\left(\boldsymbol{\mathcal{Z}}_c \right)^T. \quad (19)$$

Appendix B gives the mathematical proof of Theorem 4. Interestingly, Eq. (15) resembles the least-squares solution to the projection coefficients of an observation onto a set of basis vectors, where $\mathbf{Z}^{(k)}_{m_i}$ can be regarded as the Gram matrix that accounts for the correlations between all available cluster sub-spaces. Note that the proposed K-CTA algorithm is indeed efficient, since K_m is usually a small positive integer. Moreover, all of the operations during the clustering stage are performed in the reduced tensor space, and most of them only need to be computed once at the beginning of this stage.

4.2.3 *Post-Processing.* Note that the mode-m basis matrices derived by optimal projection are not column-orthonormal. To satisfy the orthonormal constraints on basis matrices in (8), we additionally decompose the mode-m basis matrix of each cluster using SVD to obtain $\mathbf{U}_{m,c} = \mathbf{U}'_{m,c} \mathbf{W}_c$, where $\mathbf{U}'_{m,c} \in \mathbb{R}^{I_m \times R_m}$ is a basis matrix whose columns are the orthonormal left singular vectors of $\mathbf{U}_{m,c}$, and each column of $\mathbf{W}_c \in \mathbb{R}^{R_m \times R_m}$, respectively, contains the projection coefficients of each column of $\mathbf{U}_{m,c}$ onto $\mathbf{U}'_{m,c}$. After that, we replace the mode-m basis matrix of cluster c with $\mathbf{U}'_{m,c}$ and re-compute the core tensor of cluster c by the mode-m product $\boldsymbol{\mathcal{Z}}_c \times_m \mathbf{W}_c$, so that the value of the objective function in (8) is unchanged.

4.3 The Update Stage

In this stage, the core tensor and basis matrices of each cluster are updated from the results of the clustering stage. While this problem for CTA can be easily solved by simultaneously applying tensor approximation to the member sub-tensors of each cluster [Tsai and Shih 2006], sub-space learning for all clusters at the same time

[3] In this article, the *reduced tensor space* is referred to as the union of all decomposed cluster sub-spaces, whose dimensionality is frequently much lower than the original tensor space.

would be difficult for K-CTA, since each mode-m sub-tensor now belongs to K_m different clusters that may be correlated with each other. We thus alternately decompose one cluster at a time by tensor approximation, while fixing the results of other clusters. To simplify the proposed algorithm, the cluster membership of each mode-m sub-tensor is not altered in this stage, leaving it to be updated only in the clustering stage.

At the c-th iteration, $\boldsymbol{\mathcal{Z}}_c$ and $\{\mathbf{U}_{n,c}\}_{n=1}^{N}$ of cluster c are allowed to change, while those of other clusters are fixed. The objective function in (8) thus can be rewritten as

$$\left\| \boldsymbol{\mathcal{R}}_c - \boldsymbol{\mathcal{Z}}_c \bigtimes_{n=1}^{N}{}_n \mathbf{U}_{n,c} \right\|_F^2, \ \boldsymbol{\mathcal{R}}_c = \boldsymbol{\mathcal{A}} - \sum_{\substack{j=1 \\ j \neq c}}^{C} \left(\boldsymbol{\mathcal{Z}}_j \bigtimes_{n=1}^{N}{}_n \mathbf{U}_{n,j} \right), \quad (20)$$

where $\boldsymbol{\mathcal{R}}_c$ is the residual tensor at the c-th iteration. By comparing (20) to (4.1) and (4.2) in [De Lathauwer et al. 2000], we know that the minimization of (20) can be solved with the aid of tensor decomposition. To enforce the sparse constraint on $\mathbf{U}_{m,c}$, the key idea is to only include member sub-tensors of cluster c in the tensor decomposition process and just update the non-zero entries of $\mathbf{U}_{m,c}$, which is based on the same concept in the codebook update stage of K-SVD [Aharon et al. 2006]. As a result, the membership of cluster c is fixed, and the zero entries of $\mathbf{U}_{m,c}$ remain zeros. However, the non-zero entries of $\mathbf{U}_{m,c}$ are allowed to change with the decomposed results of cluster c at the same time.

More formally, let M_c be the membership index set of cluster c defined as

$$M_c = \left\{ i \in \{1, 2, \ldots, I_m\} \mid \boldsymbol{\mathcal{A}}_{\langle m_i \rangle} \text{ is a member of cluster } c \right\}, \quad (21)$$

and $\mathbf{M}_c \in \mathbb{R}^{I_m \times |M_c|}$ denotes the membership matrix of cluster c, whose entries are

$$\forall i_1, \ \forall i_2, \ \left(\mathbf{M}_c\right)_{i_1 i_2} = \begin{cases} 1, & \text{if } i_1 = (M_c)_{i_2}, \\ 0, & \text{otherwise}, \end{cases} \quad (22)$$

where $|\cdot|$ denotes the cardinality of a set, and $(M_c)_{i_2}$ is the i_2-th element of M_c. We thus can extract the member sub-tensors of cluster c into an N-th order tensor $\boldsymbol{\mathcal{R}}'_c$ by $\boldsymbol{\mathcal{R}}'_c = \boldsymbol{\mathcal{R}}_c \times_m \mathbf{M}_c^T$. When applying tensor approximation to the *shrunken* tensor $\boldsymbol{\mathcal{R}}'_c$, non-members of cluster c are excluded from the decomposition. $\boldsymbol{\mathcal{Z}}_c$ and $\{\mathbf{U}_{n,c}\}_{n=1}^{N}$ are then updated by the decomposed core tensor and basis matrices of $\boldsymbol{\mathcal{R}}'_c$. Note that since $\boldsymbol{\mathcal{R}}'_c$ contains only members of cluster c, $\mathbf{U}_{m,c}$ should be further updated by the multiplication $\mathbf{M}_c \mathbf{U}_{m,c}$ to satisfy the constraints in (8).

5. IMPLEMENTATION ISSUES

In this section, we discuss some practical issues of K-CTA, such as the initial guess of the core tensor and basis matrices of each cluster (Section 5.1), the extraction of global basis matrices of all clusters (Section 5.2), and the degeneracy and convergence problems of K-CTA (Section 5.3).

5.1 Initial Guess

One practical issue of K-CTA is the initial guess of the core tensor and basis matrices of each cluster. A simple and heuristic method is to employ the decomposed results of hard clustering. We can execute CTA for just one iteration to obtain the initial core tensor and basis matrices of each cluster for further data analysis using K-CTA. Although this scheme will push the problem back to the initial seeds for clustering methods, various techniques for generating and fixing the initial cluster seeds usually provide satisfactory results in our experience.

The best approach may vary with the given data sets, but here we present a general method as a guideline to determine the initial cluster seeds. For a real-world data set, each mode of the input tensor is associated with a parametric space that describes its physical conditions, for example different illumination or view directions for a reflectance function. Based on the assumption that observations from nearby parameters in the mode-m parametric space are expected to be highly correlated, we can perform initial clustering in the parametric space instead. This is computationally efficient since the dimensionality of the mode-m parametric space is frequently lower than that of observations. Sophisticated or even exhaustive methods therefore can be employed to generate appropriate cluster seeds. In our experiments, this scheme generally reduces the final approximation errors of K-CTA by $2\% \sim 5\%$, when compared with directly performing K-means clustering on mode-m sub-tensors to determine the initial cluster seeds for CTA.

Interestingly, while finding a favorable initial guess is a significant issue for CTA and other iterative algorithms, K-CTA is less sensitive to the quality of an initial solution. As the number of mixture clusters increases, the importance of an appropriate initial guess for K-CTA decreases. This result is not surprising since the impact of inappropriate initial cluster membership can be compensated by additional mixture clusters. The compensation also can be regarded as an optimal interpolation scheme from other cluster sub-spaces in the least-squares sense, which particularly allows smooth transitions across different physical conditions in practical applications. In Section 6.3, we will further demonstrate the influence of this characteristic of K-CTA in the experimental results of all-frequency BRT.

5.2 Global Basis Matrices

Sometimes a single *global* mode-n basis matrix of all clusters is preferred rather than an individual *local* mode-n basis matrix of each cluster. The preference for global basis matrices may be due to computational costs, storage space, or some special purposes of an application. To account for this issue, the global basis matrices are computed by decomposing an N-th order tensor $\boldsymbol{\mathcal{A}}$ before applying K-CTA.

Let G be the index set of global basis matrices $\{\mathbf{U}_n\}_{n \in G}$, where the subscript c of a local mode-n basis matrix is omitted to denote a global one. After extracting $\{\mathbf{U}_n\}_{n \in G}$ by applying tensor approximation to $\boldsymbol{\mathcal{A}}$, we project $\boldsymbol{\mathcal{A}}$ onto $\{\mathbf{U}_n\}_{n \in G}$ to obtain an N-th order reduced tensor $\boldsymbol{\mathcal{A}}_G$ as

$$\boldsymbol{\mathcal{A}}_G = \boldsymbol{\mathcal{A}} \bigtimes_{n \in G}{}_n \mathbf{U}_n^T. \quad (23)$$

K-CTA is then performed on $\boldsymbol{\mathcal{A}}_G$ to compute the core tensor and the local basis matrices of each cluster, while the global basis matrices are fixed.

Although an iterative algorithm can be employed to alternately update the global basis matrices and the decomposed results of K-CTA, it is computationally too expensive and only reduces approximation errors by a small amount. We therefore just performed the initial tensor decomposition on $\boldsymbol{\mathcal{A}}$ to derive the global basis matrices and did not update them after K-CTA for all experimental results in this article.

5.3 Degeneracy and Convergence

When the total number of clusters is large, we have observed that sometimes all the entries in a column of the mode-m basis matrix of a cluster, for example $\mathbf{U}_{m,c}$, may become zeros at the end of the clustering stage, which implies that no mode-m sub-tensors are classified into cluster c. Although this degeneracy problem does not occur frequently in practice, an empty cluster actually consumes memory space without any contributions to final approximation errors. In the worst case, K-CTA may even derive a poor solution when there are too many empty clusters, as if the total number of clusters were set to a lower value. Our solution to this issue is to split the cluster with the largest total sum of approximation errors in half by sorting the approximation error of each mode-m sub-tensor within this cluster. If there is more than one empty cluster, we can sequentially split non-empty clusters using the preceding method until each empty cluster has been assigned at least one mode-m sub-tensor. It should be noted that our approach is in fact related to the key concept of teleportation [Cohen-Steiner et al. 2004] and enhanced LBG [Patanè and Russo 2001] in the clustering literature.

Another important question is whether the proposed iterative K-CTA algorithm always converges to a local optimum. Apparently, the answer is no. Similar to K-SVD [Aharon et al. 2006], the convergence of K-CTA is not guaranteed, since only the approximate mixing coefficients (and also the cluster membership) of each mode-m sub-tensor are derived in the clustering stage. Therefore, the total sum of approximation errors is not guaranteed to decrease when compared to the decomposed results of the previous iteration. Note that if the convergence and a locally optimal solution of K-CTA are both required, one can instead perform the brute-force search for a globally optimal solution to the pursuit problem, at the cost of longer computation time.

Fortunately, although K-CTA theoretically does not always ensure convergence, we have found that it practically converges within just a few iterations in the experiments. We currently have no idea about how to efficiently update mixing coefficients and cluster sub-spaces, so that approximation errors are always reduced. However, a simple and intuitive technique can be applied to prevent bad results due to the divergence problem. At the end of the update stage, if the total sum of approximation errors increases, we instead restore the decomposed results of previous iteration. As a result, the approximation errors of an input tensor will never increase, and the convergence criterion of K-CTA in Algorithm 1 can be always reached.

6. APPLICATIONS AND RESULTS

Three applications of the proposed K-CTA algorithm are presented in this section, including the compression of BTFs[4] (Section 6.1), VOTFs[5] (Section 6.2), and all-frequency BRT data (Section 6.3). Our experiments demonstrate that K-CTA is effective and promising when compared to previous tensor-based methods: N-SVD and CTA. The experiments and simulation timings in this article were conducted and measured on a workstation with an Intel i7-980X Extreme CPU, an NVIDIA GeForce GTX 480 graphics card, and 12GB main memory.

In our experiments, the reduced ranks of each mode are constrained to be multiples of 4 for all tensor-based algorithms, since this can maximize the efficiency of run-time texture filtering/fetching on GPUs. For N-SVD, the reduced ranks of each mode are separately obtained by decomposing only the basis matrix of one mode, say the mode n, while fixing other basis matrices to identity ones, and incrementally adding the value of R_n until an approximation threshold is reached.

For CTA and K-CTA, we simply let $R_m = 4$ for the clustered mode m, and select the reduced ranks of other modes as those for N-SVD. Moreover, the total number of clusters C can be determined from the mode-m reduced rank for N-SVD. We have found that when $R_m C$ is equal to the mode-m reduced rank for N-SVD, the signal-to-error ratios of the three tensor-based representations are roughly similar to each other. As for the number of mixture clusters K_m, it usually can be a very small integer (typically 2 or 3 from our experimental results, especially from Table I and Fig. 5). A larger value of K_m, for example 4, can slightly increase the signal-to-error ratios by a factor of less than 1%, but may substantially reduce the rendering rates by a factor of more than 20%.

6.1 Bidirectional Texture Functions

BTFs [Dana et al. 1999] generalized BRDFs to contain textural patterns of real-world object surfaces. They capture spatially varying surface appearance and reflectance that change with respect to the illumination and view directions. A BTF is thus a 6D function of the illumination direction ω_l, the view direction ω_v, and the 2D spatial coordinates, x and y, of a texel $\mathbf{t}$. Although the illumination effects of real-world object surfaces can be faithfully captured with BTFs, we usually have to tabulate several gigabytes of raw data for a single BTF data set. This is certainly impractical for photo-realistic image synthesis in real-time rendering applications.

Various approximation algorithms have been employed to solve this problem [Filip and Haindl 2009], including matrix factorization [Koudelka et al. 2003; Sattler et al. 2003; Suykens et al. 2003], non-linear dimensionality reduction methods [Müller et al. 2003; 2004], parametric models [McAllister et al. 2002; Tsai et al. 2011], to name a few. Recently, tensor-based methods have been widely applied to BTF compression [Vasilescu and Terzopoulos 2004; Wang et al. 2005; Wu et al. 2008]. We thus take a step further to conduct experiments of K-CTA on different BTFs for comparison.

In our experiments, a BTF is organized as a fourth order tensor $\boldsymbol{\mathcal{A}}^{(B)} \in \mathbb{R}^{I_{\omega_l} \times I_{\omega_v} \times I_x \times I_y}$ to retain its intrinsic structures, where I_{ω_l} and I_{ω_v} denote the numbers of sampled illumination and view directions, and I_x as well as I_y specify the spatially horizontal and vertical resolutions. Given the reduced ranks of each mode, namely $R^{(B)}_{\omega_l}$, $R^{(B)}_{\omega_v}$, $R^{(B)}_x$, and $R^{(B)}_y$, K-CTA is then applied to $\boldsymbol{\mathcal{A}}^{(B)}$ with total $C^{(B)}$ clusters for the view mode, $K^{(B)}_{\omega_v}$ mixture clusters for a mode-ω_v sub-tensor, and a single global basis matrix for the illumination mode. $\boldsymbol{\mathcal{A}}^{(B)}$ is thus approximated by

$$\boldsymbol{\mathcal{A}}^{(B)} \approx \sum_{c=1}^{C^{(B)}} \left(\boldsymbol{\mathcal{Z}}^{(B)}_c \times_{\omega_l} \mathbf{U}^{(B)}_{\omega_l} \times_{\omega_v} \mathbf{U}^{(B)}_{\omega_v,c} \times_x \mathbf{U}^{(B)}_{x,c} \times_y \mathbf{U}^{(B)}_{y,c} \right), \tag{24}$$

where $\boldsymbol{\mathcal{Z}}^{(B)}_c \in \mathbb{R}^{R^{(B)}_{\omega_l} \times R^{(B)}_{\omega_v} \times R^{(B)}_x \times R^{(B)}_y}$ denotes the core tensor of cluster c, $\mathbf{U}^{(B)}_{\omega_l} \in \mathbb{R}^{I_{\omega_l} \times R^{(B)}_{\omega_l}}$ represents the global mode-ω_l basis matrix of all clusters, and $\mathbf{U}^{(B)}_{\omega_v,c} \in \mathbb{R}^{I_{\omega_v} \times R^{(B)}_{\omega_v}}$, $\mathbf{U}^{(B)}_{x,c} \in \mathbb{R}^{I_x \times R^{(B)}_x}$,

[4] In this article, the measured BTFs, including *Corduroy*, *Sponge*, and *Wool*, were provided in courtesy of Koudelka et al. [2003] and Sattler et al. [2003], whereas the BTF of each simulated material, such as *Fiber* or *RoughHole*, was obtained by rendering a geometric surface using global illumination techniques.

[5] The VOTF of a simulated material can be precomputed over a geometric surface by using ray-tracing algorithms for visibility test.

Table I. Statistics and timing measurements of different tensor approximation algorithms for BTFs and VOTFs, including N-SVD, CTA, and K-CTA. All decomposed data were stored as half-precision (16-bit) floating point numbers. For BTFs, the quality of compressed data is measured by the signal-to-error ratio (*S/E ratio*) in dB. As for VOTFs, we list the total number of error texels for the two representations: (B) binary visibility masks; (S) signed-distance functions.

Material	Corduroy			Fiber			RoughHole			Sponge			Wool		
$I_{\omega_l} \times I_{\omega_v} \times I_x \times I_y$	81×81×128×128			100×100×96×96			100×100×96×96			120×90×128×128			81×81×128×128		
Raw data (GB)	1.2			1.03			1.03			1.98			1.2		
Algorithm	N-SVD	CTA	K-CTA	N-SVD	CTA	K-CTA	N-SVD	CTA	K-CTA	N-SVD	CTA	K-CTA	N-SVD	CTA	K-CTA
$R_{\omega_l}^{(B)} \times R_{\omega_v}^{(B)}$	20×32	20×4	20×4	16×80	16×4	16×4	28×44	28×4	28×4	16×20	16×4	16×4	20×28	20×4	20×4
$R_x^{(B)} \times R_y^{(B)}$	80×80	80×80	80×80	64×64	64×64	64×64	96×96	96×96	96×96	80×80	80×80	80×80	64×64	64×64	64×64
$C^{(B)}$	1	8	8	1	20	20	1	11	11	1	5	5	1	7	7
$K_{\omega_v}^{(B)}$	1	1	3	1	1	3	1	1	3	1	1	3	1	1	3
$R_{\omega_v}^{(O)}$	-	-	-	80	4	4	44	4	4	-	-	-	-	-	-
$R_x^{(O)} \times R_y^{(O)}$	-	-	-	80×80	80×80	80×80	80×80	80×80	80×80	-	-	-	-	-	-
$C^{(O)}$	-	-	-	1	20	20	1	11	11	-	-	-	-	-	-
$K_{\omega_v}^{(O)}$	-	-	-	1	1	3	1	1	3	-	-	-	-	-	-
Compressed data (MB)	7.87	8.13	8.14	11.07	12.06	12.09	22.26	22.54	22.56	3.96	4.11	4.11	4.42	4.6	4.6
BTF S/E ratio (dB)	19.38	18.2	19.09	17.34	16.45	17.29	14.77	12.71	14.54	24.32	23.55	24.23	20.7	19.76	20.52
VOTF error texels (B)	-	-	-	732	2222	1046	572	1938	1029	-	-	-	-	-	-
VOTF error texels (S)	-	-	-	71	2027	291	325	969	597	-	-	-	-	-	-
Compression time (min.)	19.5	30.43	54.25	8.58	15.14	30.86	2.45	8.61	11.74	13.37	17.61	18.14	8.23	8.86	12.66

and $\mathbf{U}_{y,c}^{(B)} \in \mathbb{R}^{I_y \times R_y^{(B)}}$ are, respectively, the mode-ω_v, mode-x, and mode-y basis matrices of cluster c.

The configurations of K-CTA for BTF compression are determined for the following reasons:

(1) Efficient texture filtering techniques on GPUs can be directly utilized by not clustering the x and y modes.

(2) For a complex BTF, a high mode-ω_v reduced rank is usually required to model large view-dependent variations in the appearance data, which substantially increases run-time rendering costs on GPUs. Clustering along the view mode allows us to reduce the costs in an adjustable manner.

(3) In our experience, the mode-ω_v reduced rank often dominates the perceptual quality of reconstructed BTFs and may need to be much higher than the mode-ω_l reduced rank.

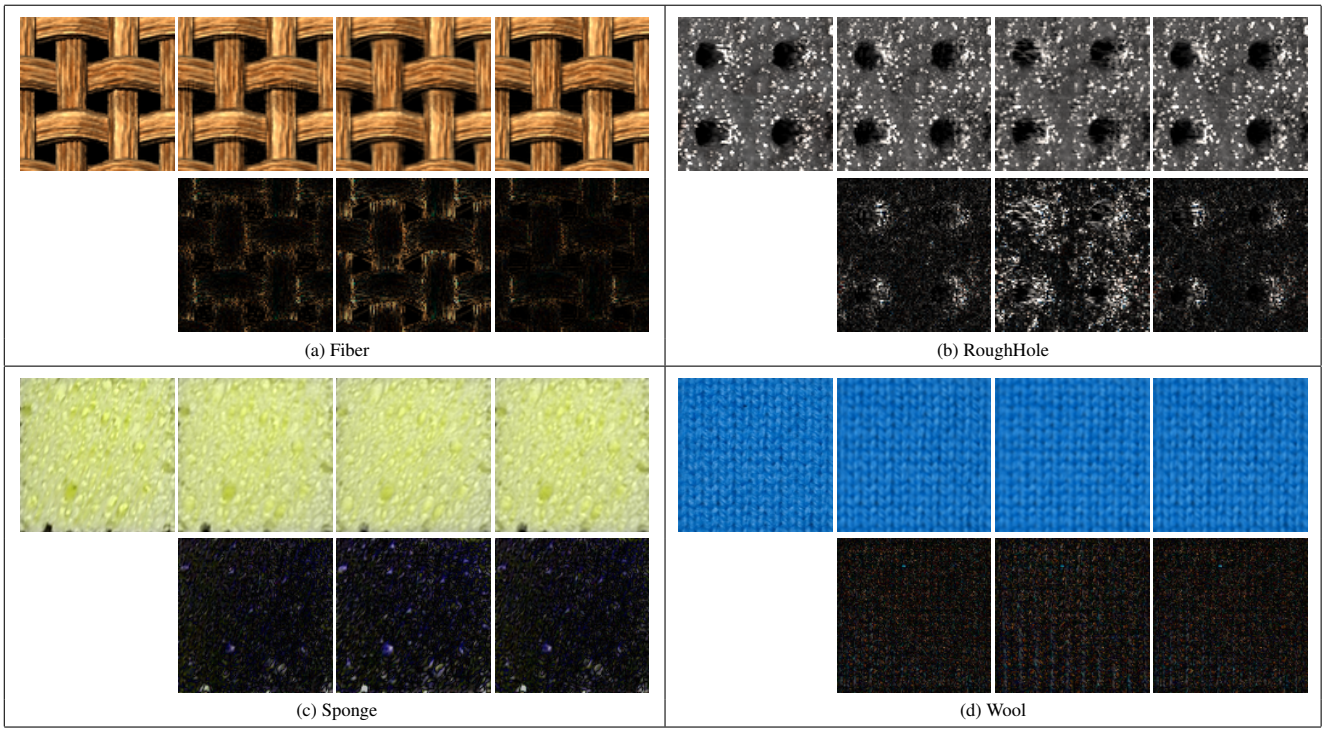
According to the reconstructed results in previous articles [Vasilescu and Terzopoulos 2004; Wang et al. 2005], we identify that if the mode-ω_v reduced rank is too low to capture the view variations in a BTF, the reconstructed images will become over-blurred or have strong ringing effects. Human eyes seem to be more sensitive to these artifacts than unrealistic illumination effects. Therefore, clustering along the view mode will reduce the performance penalty when we need a high mode-ω_v reduced rank.

In Fig. 3, we present the reconstructed BTFs of different tensor approximation algorithms, including N-SVD, CTA, and K-CTA, whose statistics and timing measurements are further compared in Table I. Fig. 5(a) also plots the signal-to-error ratio versus the mode-ω_v reduced rank for the three multilinear models with various configurations. Note that we compare them based on the same total number of mode-ω_v reduced rank, which leads to similar storage space. It can be shown that K-CTA achieves compression ratios and image quality comparable to N-SVD, while having lower approximation errors than CTA with ignorable storage overhead. For CTA, note that the view mode is also partitioned into total $C^{(B)}$ clusters, and a single global mode-ω_l basis matrix is extracted for all clusters. We have found that although extracting the global basis matrix may slightly increase approximation errors, it can significantly reduce both off-line and run-time costs in some applications (Section 6.3).

6.2 View-Dependent Occlusion Texture Functions

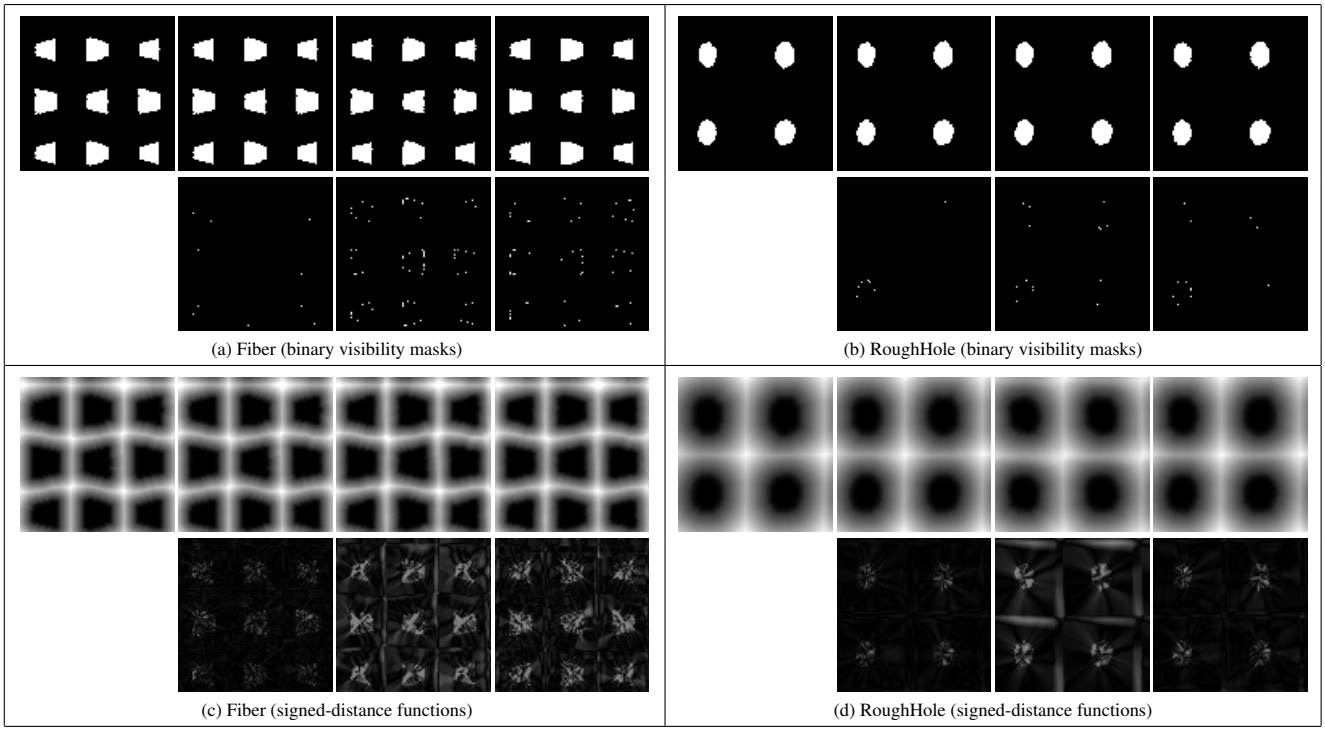
Apart from BTFs, various appearance models have also been developed to render shadows and complex illumination effects from precomputed meso-scale data or special fields, such as visibility information [Heidrich et al. 2000], view-dependent displacement mapping [Wang et al. 2003], shell texture functions [Chen et al. 2004], and relief mapping [Policarpo et al. 2005]. In this article, we propose a spatially varying appearance model, namely VOTFs, to visualize complex micro-geometry of object surfaces. A VOTF is a set of 2D textures in which each texture contains spatially varying meso-scale occlusions from a sampled view direction. It is a 4D function of the view direction ω_v and the 2D spatial coordinates, x and y, of a texel $\mathbf{t}$. Thus, VOTFs can be employed to enhance the surface appearance of objects with shape details and silhouettes. Examples of a VOTF are shown in Fig. 6(b).

Besides conventional binary visibility masks, we represent a VOTF with a set of 2D signed-distance textures. Signed-distance functions [Danielsson 1980] have been shown to successfully preserve sharp features in vector textures [Qin et al. 2006] and image structures for texture synthesis [Lefebvre and Hoppe 2006]. Recently, they were also applied to the approximation of visibility integrals for PRT [Xu et al. 2008], further showing their potentials for modeling occlusion information. According to our experiments, the advantages of signed-distance functions still hold even after compression with tensor approximation algorithms. The intuitive explanation for this outcome is that sharp boundaries in the binary visibility masks are in fact high-frequency signals. Converting them into signed-distance functions instead produces smooth and continuous signals that would facilitate subsequent approximation.



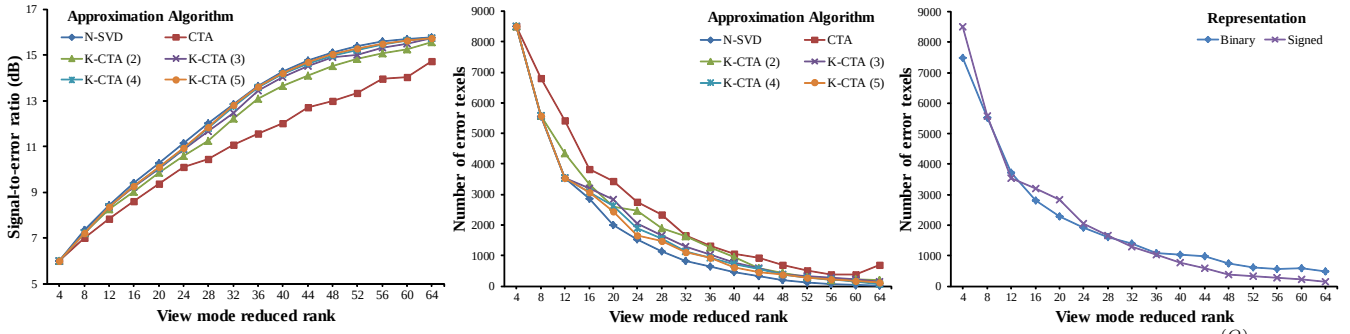
(a) Fiber (b) RoughHole (c) Sponge (d) Wool

Fig. 3. Reconstructed BTF images of different tensor approximation algorithms. In each sub-figure, from left to right: raw data; N-SVD; CTA; K-CTA, from top to bottom: reconstructed images; absolute difference images (scaled by a factor of 3).



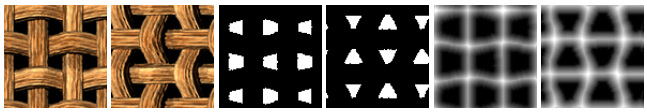
(a) Fiber (binary visibility masks) (b) RoughHole (binary visibility masks) (c) Fiber (signed-distance functions) (d) RoughHole (signed-distance functions)

Fig. 4. Reconstructed VOTF images of different tensor approximation algorithms. In each sub-figure, from left to right: raw data; N-SVD; CTA; K-CTA, from top to bottom: reconstructed images; absolute difference images (scaled by a factor of 10 for signed-distance functions).



(a) BTF (b) VOTF (signed-distance functions) (c) VOTF comparison (K-CTA, $K^{(O)}_{\omega_v}$=3)

Fig. 5. Comparisons of the approximation errors of different multilinear models and visibility representations for the material *RoughHole*. In (a) and (b), the values in parentheses for K-CTA represent the number of mixture clusters, namely $K^{(B)}_{\omega_v}$ or $K^{(O)}_{\omega_v}$. For all multilinear models, $R^{(B)}_x = R^{(B)}_y = 96$ and $R^{(O)}_x = R^{(O)}_y = 80$. For CTA and K-CTA, the horizontal axis in each chart specifies various total mode-ω_v reduced ranks whose values correspond to $R^{(B)}_{\omega_v} C^{(B)}$ for the BTF and $R^{(O)}_{\omega_v} C^{(O)}$ for the VOTF, with $R^{(B)}_{\omega_v} = R^{(O)}_{\omega_v} = 4$ for each cluster.



(a) BTF data (b) VOTF data (binary visibility masks) (c) VOTF data (signed-distance functions)

Fig. 6. Examples of the simulated material *Fiber*. For the VOTF, each image records visibility information from a sampled view direction. The values of signed-distance functions were normalized into the interval $[-1, +1]$.

Fig. 6(c) shows examples of a VOTF in the signed-distance representation.

Therefore, a VOTF is first converted into signed-distance-to-boundary textures from binary visibility masks, one texture for each view direction. For a given texel $\mathbf{t}$ in the binary mask of a view direction, we compute its nearest distance to the boundaries. A positive distance is stored if the value of texel $\mathbf{t}$ in the binary mask is equal to 0 (occluded). Otherwise, a negative distance is adopted. The transformed VOTF is then normalized into the interval $[-1, +1]$ and organized as a third order tensor $\mathcal{A}^{(O)} \in \mathbb{R}^{I_{\omega_v} \times I_x \times I_y}$.

Given the reduced ranks of each mode, namely $R^{(O)}_{\omega_v}$, $R^{(O)}_x$, and $R^{(O)}_y$, K-CTA is then applied to $\mathcal{A}^{(O)}$ with total $C^{(O)}$ clusters for the view mode and $K^{(O)}_{\omega_v}$ mixture clusters for a mode-ω_v sub-tensor. $\mathcal{A}^{(O)}$ is thus approximated by

$$\mathcal{A}^{(O)} \approx \sum_{c=1}^{C^{(O)}} \left(\mathcal{Z}^{(O)}_c \times_{\omega_v} \mathbf{U}^{(O)}_{\omega_v,c} \times_x \mathbf{U}^{(O)}_{x,c} \times_y \mathbf{U}^{(O)}_{y,c} \right), \quad (25)$$

where $\mathcal{Z}^{(O)}_c \in \mathbb{R}^{R^{(O)}_{\omega_v} \times R^{(O)}_x \times R^{(O)}_y}$ is the core tensor of cluster c, and $\mathbf{U}^{(O)}_{\omega_v,c} \in \mathbb{R}^{I_{\omega_v} \times R^{(O)}_{\omega_v}}$, $\mathbf{U}^{(O)}_{x,c} \in \mathbb{R}^{I_x \times R^{(O)}_x}$, and $\mathbf{U}^{(O)}_{y,c} \in \mathbb{R}^{I_y \times R^{(O)}_y}$, respectively, specify the mode-ω_v, mode-x, and mode-y basis matrices of cluster c.

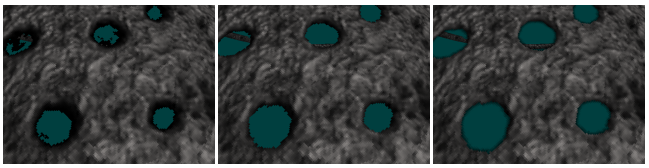
In Fig. 4, we show the reconstructed VOTF data of simulated spatially varying meso-structures. Table I also compares the number of error texels after approximation using N-SVD, CTA, and K-CTA. From Fig. 4 and Table I, K-CTA obviously outperforms CTA in perceptual quality and the number of error texels, especially for the VOTF of the material *Fiber*. Moveover, signed-distance functions are significantly better than binary visibility masks in visual quality and approximation errors (the number of error texels) for encoding VOTFs. After compression, signed-distance functions tend to reconstruct more continuous signals and less noises at silhouette boundaries (see Fig. 7 and the accompanying video).

Fig. 5(b) plots the number of error texels versus the mode-ω_v reduced rank for the three multilinear models with various configurations, when VOTFs are encoded in the signed-distance representation. Note that we compare the three models based on the same total number of mode-ω_v reduced rank, which leads to similar storage space. Furthermore, Fig. 5(c) shows that signed-distance functions begin to outperform binary visibility masks from a certain cross point, for example 36. This implies that the high-frequency signals in binary visibility masks significantly limit the approximation ability of tensor approximation algorithms. In practice, we have found that a higher mode-ω_v reduced rank than the value of this cross point is frequently necessary to render high-quality images.

6.3 All-Frequency Bi-Scale Radiance Transfer

6.3.1 *Overview.* BRT [Sloan et al. 2003b] generalizes PRT [Sloan et al. 2002] with spatially varying materials, which are called radiance transfer textures, to improve image quality with detailed surface appearance. The main concept of BRT is to separate the light transport problem into macro-scale (coarsely-sampled global illumination data) and meso-scale (spatially varying appearance models) radiance transfer. Nevertheless, most previous BRT algorithms are limited to low-frequency light transport and low-quality surface appearance at the silhouettes. The radiance transfer textures only model fine-scale lighting and shadowing effects, but neither contain shape information nor actually modify the surface geometry. Therefore, meso-scale shape details and shadow boundaries owing to complex meso-structures cannot be faithfully captured. Recently, Sun et al. [2011] proposed an all-frequency BRT algorithm based on bi-clustering. Although their approach supports high-frequency bi-scale lighting effects in real time, it only focuses on compressing the macro-scale radiance transfer data. By contrast, the proposed method can accurately approximate the radiance transfer data sets at both scales.

In our experiments, BTFs and VOTFs are combined for meso-scale radiance transfer to model not only spatially varying illumination effects but also view-dependent occlusions at the meso-scale. To obtain an accurate and compact representation, we first apply



(a) Binary visibility masks (b) Signed-distance functions (c) Signed-distance functions (anti-aliased)

Fig. 7. Comparison between different encoding schemes for VOTFs. Inside triangle faces of the model are culled to clearly show the difference. For binary visibility masks, the threshold $T^{(O)}$ and weight $W^{(O)}$ in (26) were, respectively, set to 0.5 and 1. For signed-distance functions, $T^{(O)}$ and $W^{(O)}$ were, respectively, set to 0.05 and 12.

tensor approximation algorithms to decompose BTFs and VOTFs into a few low-order factors and the reduced multidimensional core tensor(s). For VOTFs, view-dependent signed-distance functions are recommended instead of binary visibility masks to preserve sharp geometric features at the meso-scale

As for macro-scale radiance transfer, we modify the all-frequency PRT framework [Tsai and Shih 2006] to precompute BRT data for run-time rendering. Specifically, the light-dependent functions in [Tsai and Shih 2006] are replaced with the extracted mode-ω_l basis matrix from tensor approximation algorithms. For the BTF compressed using CTA or K-CTA, note that we derive only a single *global* mode-ω_l basis matrix of all clusters as in Section 6.1. This results in only one set of macro-scale BRT data, rather than $C^{(B)}$ sets of BRT data when a *local* mode-ω_l basis matrix is extracted for each cluster. Both BRT computation and rendering costs thus can be significantly reduced. Then, the raw radiance transfer matrices are computed at each object vertex, approximated with a set of uniform SRBFs, and finally compressed using CTA[6] to exploit inter-vertex coherence. During BRT computation, we also employ the meso-scale occlusion information in VOTFs to cast shadows owing to complex meso-structures. More details can be found in [Tsai 2009, Chapter 9.2].

6.3.2 *Resampling.* Due to the sparse sampling rates of the illumination and view modes, it is necessary to resample the compressed BTF data for efficient run-time rendering under novel illumination and view conditions. Although N-SVD allows us to generate observations from a novel illumination direction by resampling just the columns of the mode-ω_l basis matrix, resampling the view mode is complicated since CTA and K-CTA partition the view mode into different regions in our experiments. For CTA, we employ traditional linear interpolation from nearby view directions to solve this issue, whereas the proposed K-CTA algorithm implicitly suggests a better resampling scheme.

For a compressed BTF based on K-CTA, we can resample the view mode of the raw BTF first and employ the greedy search and optimal projection in the clustering stage of K-CTA to compute the mixing coefficients of each mode-ω_v sub-tensor of the resampled raw BTF. Thus, the overall effect is equivalent to resampling the columns of the mode-ω_v basis matrices, so that the BTF reconstruction and interpolation are combined together without reducing run-time performance. Although one can continue to update the cluster sub-spaces of the compressed BTF and follow the process of K-CTA on the resampled raw BTF until convergence, we do not recommend applying this time-consuming iterative scheme. Moreover, to accelerate the resampling of the raw BTF using sophisticated methods, such as cubic or thin-plate spline interpolation in the spherical domain, we perform N-SVD on the raw BTF without approximation errors to obtain a full-rank mode-ω_v basis matrix of size $I_{\omega_v} \times I_{\omega_v}$, and then resample columns of this matrix instead.

[6]One can instead apply the proposed K-CTA algorithm to compress the macro-scale radiance transfer matrices, but for a reasonable computational cost, we employ CTA in the current implementation.

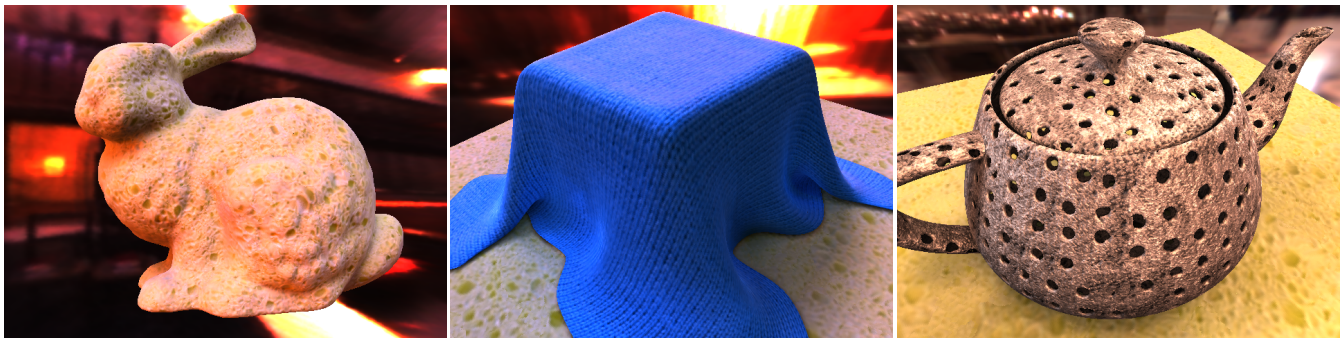
Table II. Statistics and timing measurements of the proposed all-frequency BRT algorithm. In the row *MRT compression time*, the performance of compressing the macro-scale radiance transfer matrices is shown. In the row *Frames per second*, we list the rendering performance with/without visibility anti-aliasing when the viewpoint changes. For the configurations of each meso-scale material, please refer to Table I.

Model	Bunny	BunnyPlane	Cloth	Teapot
Material(s)	Sponge	Fiber + Sponge	Wool + Sponge	RoughHole + Sponge
Vertices	36k	98k	55k	75k
SRBFs	642	2562	642	642
Raw data (GB)	40.11	108.35	69.23	123.26
Clusters	100	180	110	150
Compressed data (MB)	15.77	71.4	18.79	29.44
BRT computation time (hr)	1.78	16.41	4.95	7.21
MRT compression time (hr)	5.83	12.35	7.47	9.56
Frames per second (w/wo)	-/147.42	71.58/93.27	-/81.22	52.29/66.74

6.3.3 *Rendering.* The rendering process of all-frequency BRT based on tensor approximation algorithms is rather straightforward. To increase performance and employ texture filtering on GPUs, we reconstruct the x and y modes of each (cluster) core tensor, create their mipmaps, and then concatenate the results into one or more 2D texture arrays before rendering. The mode-ω_l and mode-ω_v basis matrices are instead stored in 2D textures in the parabolic parameterization [Heidrich and Seidel 1999]. For meso-structure synthesis, the spatial coordinate texture S is obtained by using appearance-space texture synthesis [Lefebvre and Hoppe 2006] on the raw BTF/VOTF data. The run-time process on GPUs thus consists of the following steps:

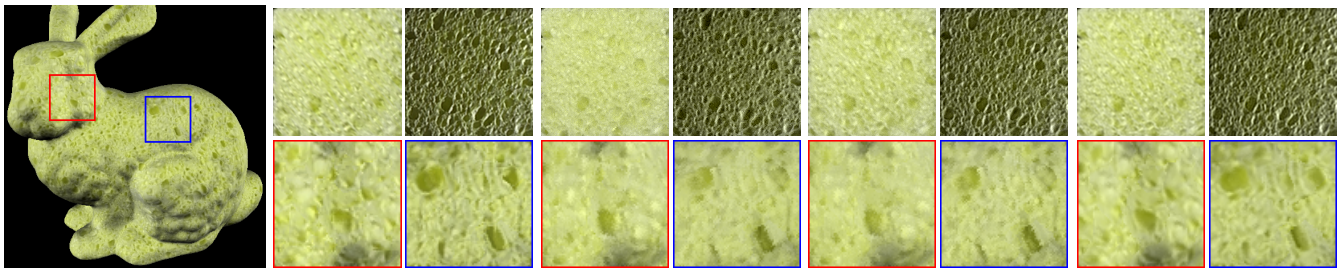
(1) Perform Steps 1–3 in the run-time process in [Tsai and Shih 2006] to obtain the per-vertex radiance transfer vector $\mathbf{r_p}$.
(2) In the pixel shader, sample the synthesized texture S for the BTF/VOTF spatial coordinates $\mathbf{t_p}$ of current pixel $\mathbf{p}$.
(3) If the meso-structure of pixel $\mathbf{p}$ does not contain VOTF, set pixel $\mathbf{p}$ as visible and go to Step 7. Otherwise, continue to execute Step 4.
(4) The approximated VOTF value of pixel $\mathbf{p}$, $\hat{O}_\mathbf{p}$, is given by reconstructing the view mode according to the employed tensor approximation algorithm.
(5) A user-defined visibility mapping function, for example (26), is then applied to map $\hat{O}_\mathbf{p}$ into a visibility value $\hat{O}'_\mathbf{p}$ within the interval $[0, 1]$. Set pixel $\mathbf{p}$ to visible if $\hat{O}'_\mathbf{p} > 0$.
(6) If pixel $\mathbf{p}$ is visible, compute its shading color by Step 7. Otherwise, discard it.
(7) Sample the texture of the mode-ω_v basis matrix for all the components of current novel view direction. The shading color of pixel $\mathbf{p}$ is then given by the dot product of the sampled results and $\mathbf{r_p}$.

The main purpose of the visibility mapping function in Step 5 is to determine final visibility values and avoid the aliasing problem that results from meso-scale visibility. It should be designed



(a) Bunny (147.42 FPS) (b) Cloth (81.22 FPS) (c) Teapot (52.29 FPS)

Fig. 8. Rendered results of the all-frequency BRT algorithm. The configurations of macro-scale and meso-scale radiance transfer are listed in Tables I and II. The model *Bunny* was provided in courtesy of Stanford Computer Graphics Laboratory [2011].



(a) Rendered image (K-CTA) (b) Raw data (<0.01 FPS) (c) STD (21.09 dB, 42.35 FPS) $k_1 = 4, k_2 = 4$ (d) STD (24.26 dB, 2.91 FPS) $k_1 = 8, k_2 = 32$ (e) K-CTA (24.23 dB, 168.47 FPS)

Fig. 9. Comparisons of reconstructed BTF images (top) and rendered results (bottom) between STD [Ruiters and Klein 2009] and K-CTA. The signal-to-error ratio and rendering performance of the compressed BTF are shown in parentheses. For STD, we let $D_1 = D_2 = 256$ and employ two configurations for k_1 and k_2 [Ruiters and Klein 2009, Section 4.2]. The model *Bunny* was provided in courtesy of Stanford Computer Graphics Laboratory [2011].

to generate visually pleasing effects at silhouette boundaries. In the experiments, the mapping function is defined as

$$\hat{O}'_{\mathbf{p}} = \begin{cases} 1 & \text{if } \hat{O}_{\mathbf{p}} \geq T^{(O)} \quad \text{(fully visible)}, \\ W^{(O)}\hat{O}_{\mathbf{p}} & \text{if } T^{(O)} > \hat{O}_{\mathbf{p}} > 0 \quad \text{(partially visible)}, \\ 0 & \text{otherwise} \quad \text{(invisible)}, \end{cases} \tag{26}$$

where $T^{(O)}$ and $W^{(O)}$ are, respectively, the user-defined threshold and weight. In this way, $\hat{O}'_{\mathbf{p}}$ can be utilized as an alpha value to blend a partially visible pixel with its background as shown in Fig. 7(c). To approximate the blending effects without sorting, fully visible pixels are rendered first. We then keep the contents of frame buffer and blend partially visible pixels that pass the depth test. This correctly captures the order of a fully visible pixel and a partially visible pixel, but disregards the blending effects between two partially visible pixels. The resulting artifacts are ignorable in practice since the amount of partially visible pixels is usually small.

For K-CTA, since mode-ω_v basis matrices are sparse, the reconstruction in Step 7 can be computed from only their non-zero entries. We achieve this by packing only the non-zero entries of mode-ω_v basis matrices in the corresponding texture, and constructing another 2D texture to identify the cluster membership of each resampled view direction. Since the reconstruction with mixture clusters and mixing coefficients already resembles an interpolation process, we currently only utilize the nearest neighbor of the novel view direction for rendering. In our experience, nearest-neighbor interpolation is usually unnecessary for K-CTA. The proposed resampling scheme effectively leads to smooth transitions when the viewpoint/object moves, at the cost of a dense resampling rate for view directions (typically 256×256 in the parabolic parameterization).

6.3.4 *Results.* Table II lists the experimental statistics of the proposed all-frequency BRT algorithm in various configurations. For macro-scale radiance transfer, we simulated light paths with at most two inter-reflections to precompute a 6×32×32 radiance transfer matrix at each vertex and approximated the raw BRT data with a set of uniform SRBFs. Due to the enormous amount of BRT data, we did not directly employ CTA to exploit inter-vertex coherence, but first applied clustered principal component analysis [Sloan et al. 2003a] to classify vertices, as suggested by Sun et al. [2007], and fine-tuned cluster membership using CTA. The reduction of the view mode was omitted to accelerate the compression process, and the reduced ranks of the light and vertex modes were, respectively, set to 64 and 12.

Fig. 1 compares the rendered all-frequency BRT images based on different tensor approximation algorithms for meso-structures. It shows that K-CTA achieves image quality comparable to N-SVD, while providing almost the same rendering performance as CTA. Although both CTA and K-CTA allow real-time rendering performance, CTA sometimes produces noticeable artifacts when the viewpoint/object moves (refer to the accompanying video).

In Fig. 8, we further present more rendered all-frequency BRT images. From these images, the proposed BRT algorithm certainly provides more reflectance and geometric details of meso-scale sur-

face appearance than previous PRT/BRT methods. Furthermore, VOTFs particularly allow rendering complex geometric features without consuming hundreds of thousands of polygons to explicitly model the surface micro-geometry.

6.4 Discussions

In general, N-SVD provides the best perceptual quality among the three tensor approximation algorithms, but CTA and K-CTA allow much more efficient rendering performance on GPUs for complex meso-structures. Although the compression time of K-CTA is slightly longer than N-SVD and CTA on average, K-CTA effectively compromises between image quality and run-time rendering performance.

Moreover, when the viewpoint/object moves at run-time, CTA with linear interpolation from the three nearest view directions sometimes produces noticeable artifacts. This is due to the intrinsic nature of CTA in which the inter-cluster coherence is ignored, so that significant discontinuities may occur at cluster boundaries, especially when inappropriate clustered results were found. A dense sampling rate for the view mode of the raw BTFs or increasing the mode-ω_v reduced rank would solve this issue for CTA, but both of them will increase the amount of compressed data and rendering time. The acquisition of densely sampled raw BTFs, nevertheless, is also a challenging problem. By contrast, K-CTA is less sensitive to the quality of clustering since approximation errors are compensated by additional mixture clusters. This compensation can be regarded as an optimal interpolation scheme in the least-squares sense to enable smooth transitions across different view directions.

It should also be noted that a related sparse representation for BTFs, named *sparse tensor decomposition* (STD), was proposed in [Ruiters and Klein 2009]. Although STD and K-CTA were both inspired by K-SVD, they are essentially different approximation algorithms. While STD simply applies K-SVD to selected dimensions of the input BTF, K-CTA instead iteratively refines the decomposed results to obtain a (locally) near-optimal solution. In Fig. 9, we compare the reconstructed BTF images and rendered results between STD and K-CTA. From this figure, it can be shown that under similar approximation errors, the run-time rendering performance of K-CTA substantially outperforms that of STD. Moreover, we also employ another configuration of STD (with a much higher approximation error) for fast rendering rates, but this still fails to match the run-time performance of K-CTA.

Note that the perceptual quality of rendered images may be significantly different between STD and K-CTA, even with quantitatively similar signal-to-error ratios as in Fig. 9(d) and Fig. 9(e). We believe that the proposed resampling scheme for K-CTA (Section 6.3.2) particularly contributes to the difference, and is also more effective and efficient at run-time than the conventional nearest-neighbor or barycentric interpolation [Ruiters and Klein 2009].

7. CONCLUSIONS AND FUTURE WORK

Data-driven models have stimulated the development of sophisticated approximation algorithms for large-scale visual data sets. This article presents a sparse multilinear model, namely K-CTA, to enable efficient rendering of complex objects for real-time applications. K-CTA introduces the concept of sparse representation into multilinear models, and effectively integrates clustering, sparse coding, and tensor approximation into a unified framework. Moreover, KCTA can also exploit inter-cluster coherence for smooth transitions across different physical conditions by mixing the decomposed results of multiple clusters. Experimental results demonstrate that K-CTA is a compact and efficient representation for spatially varying surface appearance, such as BTFs, VOTFs, and all-frequency BRT data. Furthermore, the signed-distance representation for visibility information especially preserves sharp silhouettes at the meso-scale even after compression using tensor-based methods.

In the future, we intend to improve both image quality and rendering performance by combining K-CTA with functional approximation approaches, such as specular lobe separation [Sun et al. 2007], to efficiently model highly specular spatially varying surface appearance. Moreover, the proposed K-CTA algorithm is not restricted to solve problems in computer graphics. They are indeed general approximation algorithms that can be employed to analyze various multidimensional data sets. We believe that K-CTA can be applied to other data-driven models, such as the analysis of time-varying materials and volume data sets, and may have an impact on many fields outside computer graphics.

APPENDIX

In the following sections, we present the mathematical proofs of the proposed theorems and lemma in this article. Due to limited paper length, we only describe the main ideas and key points. Interested readers may refer to [Tsai 2009, Chapters 5.5 and 6.4] for more details.

A. OBJECTIVE FUNCTIONS IN THE CLUSTERING STAGE

In the clustering stage, each mode-m sub-tensor of $\boldsymbol{\mathcal{A}}$ is sequentially classified into K_m mixture clusters using a greedy approach, while the core tensor and basis matrices, except for the mode-m basis matrix, of each cluster are fixed. Since the cluster membership and mixing coefficients of each mode-m sub-tensor are independent, Eq. (8), without the orthonormal constraints on $\mathbf{U}_{m,1}, \mathbf{U}_{m,2}, \ldots, \mathbf{U}_{m,C}$, can be separated into I_m distinct constrained least-squares optimization sub-problems as

$$\min_{\left\{\left(\mathbf{U}_{m,c}\right)_{i*}\right\}_{c=1}^{C}} \left\| \boldsymbol{\mathcal{A}}_{\langle m_i \rangle} - \sum_{c=1}^{C} \left(\boldsymbol{\mathcal{Z}}_c \times_m \left(\mathbf{U}_{m,c}\right)_{i*} \bigtimes_{\substack{n=1 \\ n \neq m}}^{N}{}_{n} \mathbf{U}_{n,c} \right) \right\|_F^2,$$
$$\text{subject to} \begin{cases} \sum_{c=1}^{C} \left\| \left(\mathbf{U}_{m,c}\right)_{i*} \right\|_0 = K_m R_m, \\ \forall c,\ \left\| \left(\mathbf{U}_{m,c}\right)_{i*} \right\|_0 \in \{0, R_m\}, \end{cases} \tag{27}$$

for $i = 1, 2, \ldots, I_m$. Note that the orthonormal constraints on $\mathbf{U}_{m,1}, \mathbf{U}_{m,2}, \ldots, \mathbf{U}_{m,C}$ in (8) can be enforced from the optimized results of (27) using the post-processing approach as described in Section 4.2.3.

A.1 First Mixture Cluster

Before proving Theorem 1, we first show the proof of Lemma 2.

PROOF OF LEMMA 2. The first mixture cluster c_{i_1} of $\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}$ is selected as if $K_m = 1$. Since the constraints in (27) can be enforced by setting $\left(\mathbf{U}_{m,c}\right)_{i*}$ to zeros for all $c \neq c_{i_1}$, the objective function in (27) can be rewritten as

$$\left\| uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i \rangle}\Big) - \left(\mathbf{U}_{m,c_{i_1}}\right)_{i*} uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}} \bigtimes_{\substack{n=1 \\ n \neq m}}^{N}{}_{n} \mathbf{U}_{n,c_{i_1}}\Big) \right\|_2^2, \tag{28}$$

where $\|\cdot\|_2$ denotes the ℓ^2 norm of a vector. Eq. (28) can be easily verified from the definitions of the mode-n unfolded matrix, the mode-n product, and the Frobenius norm. Therefore, the optimal solution of $\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*}$ to (28) is the least-squares estimation of the solution to the following linear equation:

$$uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\Big) = \left(\mathbf{U}_{m,c_{i_1}}\right)_{i*} uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}\Big)\Big(\bigotimes_{\substack{n=1\\n\neq m}}^{N}\mathbf{U}_{n,c_{i_1}}^T\Big), \quad (29)$$

where

$$\bigotimes_{\substack{n=1\\n\neq m}}^{N}\mathbf{U}_{n,c_{i_1}}^T = \mathbf{U}_{1,c_{i_1}}^T\otimes\cdots\otimes\mathbf{U}_{m-1,c_{i_1}}^T\otimes\mathbf{U}_{m+1,c_{i_1}}^T\otimes\cdots\otimes\mathbf{U}_{N,c_{i_1}}^T \quad (30)$$

denotes a series of Kronecker products, and the symbol $\otimes$ specifies the Kronecker product operator. Eq. (29) comes from the relation between the mode-n product and the Kronecker product [De Lathauwer et al. 2000]. Since each basis matrix has orthonormal columns, the least-squares solution of $\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*}$ to (28) is

$$\begin{aligned}\left(\mathbf{U}_{m,c_i}\right)_{i*} &= uf_m(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle})\Big(uf_m(\boldsymbol{\mathcal{Z}}_{c_i})\Big(\bigotimes_{\substack{n=1\\n\neq m}}^{N}\mathbf{U}_{n,c_i}^T\Big)\Big)^+\\ &= uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_1}}^T\Big)uf_m(\boldsymbol{\mathcal{Z}}_{c_{i_1}})^+. \quad (31)\end{aligned}$$

Eq. (11) is thus proved. □

Now we can finally prove Theorem 1 with the aid of Lemma 2.

PROOF OF THEOREM 1. After substituting (31) for $\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*}$ in (28), we have [Tsai 2009, Chapters 5.5 and 6.4.1]

$$\left\|\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\right\|_F^2 - \left\|uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_1}}^T\Big)\mathbf{V}_{m,c_{i_1}}^T\right\|_F^2. \quad (32)$$

Since the first term in (32) is a constant, the minimization of (32) is equivalent to the maximization of the second term in (32). Theorem 1 thus is proved by identifying that the cluster membership is implicitly specified in the solution to (27). □

A.2 Remaining Mixture Clusters

PROOF OF THEOREM 3. For the k-th mixture cluster c_{i_k} of $\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}$ other than c_{i_1}, it is determined as if $K_m = k$. When previously selected clusters $c_{i_1}, c_{i_2}, \ldots, c_{i_{k-1}}$ and the corresponding mixing coefficients are fixed, the objective function in (27) becomes

$$\left\|\boldsymbol{\mathcal{R}}_{\langle m_i\rangle}^{(k)} - \boldsymbol{\mathcal{Z}}_{c_{i_k}}\times_m\left(\mathbf{U}_{m,c_{i_k}}\right)_{i*}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_k}}\right\|_F^2, \quad (33)$$

where $\boldsymbol{\mathcal{R}}_{\langle m_i\rangle}^{(k)}$ is defined as (12). By following the same approach as in the proof of Theorem 1 and Lemma 2, the minimization of (33) is equivalent to the maximization of

$$\left\|uf_m\Big(\boldsymbol{\mathcal{R}}_{\langle m_i\rangle}^{(k)}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_k}}^T\Big)\mathbf{V}_{m,c_{i_k}}^T\right\|_F^2. \quad (34)$$

Substituting (12) for $\boldsymbol{\mathcal{R}}_{\langle m_i\rangle}^{(k)}$ in (34) then yields the objective function in (13). □

B. OPTIMAL PROJECTION FOR MIXING COEFFICIENTS

PROOF OF THEOREM 4. Suppose that total k mixture clusters $c_{i_1}, c_{i_2}, \ldots, c_{i_k}$ of $\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}$ have been selected as if $K_m = k$. Similar to (28), since the constraints in (27) can be satisfied by setting $\left(\mathbf{U}_{m,c}\right)_{i*}$ to zeros for all $c \notin \{c_{i_1}, c_{i_2}, \ldots, c_{i_k}\}$, Eq. (27) can be simplified into a standard unconstrained least-squares problem whose objective function is

$$\left\|uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\Big) - \sum_{j=1}^{k}\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_j}}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_j}}\Big)\right\|_2^2. \quad (35)$$

By taking the first-order partial derivatives of (35) with respect to each entry of $\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*}$ and setting the resulting derivatives to zeros, we have the following linear equation:

$$\begin{aligned}\sum_{j=1}^{k}uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_1}}\Big)uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_j}}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_j}}\Big)^T\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}^T\\ = uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_1}}\Big)uf_m\big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\big)^T. \quad (36)\end{aligned}$$

The right side of (36) can be rewritten as [Tsai 2009, Chapter 6.4.2]

$$uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}\Big)uf_m\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_1}}^T\Big)^T = f_m^{(c_{i_1})}\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}^{(c_{i_1})}\Big)^T, \quad (37)$$

where $f_m^{(c)}(\cdot)$ is defined in (19). Similarly, the left side of (36) is equal to

$$\begin{aligned}\sum_{j=1}^{k}uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}\bigtimes_{\substack{n=1\\n\neq m}}^{N}{}_n\,\mathbf{U}_{n,c_{i_j}}^T\mathbf{U}_{n,c_{i_1}}\Big)uf_m\Big(\boldsymbol{\mathcal{Z}}_{c_{i_j}}\Big)^T\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}^T\\ = \sum_{j=1}^{k}f_m^{(c_{i_j})}\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}^{(c_{i_j})}\Big)\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}^T. \quad (38)\end{aligned}$$

By following (36), (37), and (38) with respect to each entry of $\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}$ for all j, we have the following set of linear equations:

$$\begin{aligned}\sum_{j=1}^{k}f_m^{(c_{i_j})}\Big(\boldsymbol{\mathcal{Z}}_{c_{i_1}}^{(c_{i_j})}\Big)\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}^T &= f_m^{(c_{i_1})}\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}^{(c_{i_1})}\Big)^T,\\ \vdots &\quad\vdots\\ \sum_{j=1}^{k}f_m^{(c_{i_j})}\Big(\boldsymbol{\mathcal{Z}}_{c_{i_k}}^{(c_{i_j})}\Big)\left(\mathbf{U}_{m,c_{i_j}}\right)_{i*}^T &= f_m^{(c_{i_k})}\Big(\boldsymbol{\mathcal{A}}_{\langle m_i\rangle}^{(c_{i_k})}\Big)^T,\end{aligned} \quad (39)$$

which can be further written in a matrix form as $\mathbf{Z}_{m_i}^{(k)}\mathbf{u}_{m_i}^{(k)} = \mathbf{a}_{m_i}^{(k)}$, where $\mathbf{u}_{m_i}^{(k)}$, $\mathbf{Z}_{m_i}^{(k)}$, and $\mathbf{a}_{m_i}^{(k)}$ are, respectively, defined in (16), (17), and (18). We thus can conclude that (15) gives the least-squares solution of $\left(\mathbf{U}_{m,c_{i_1}}\right)_{i*}, \left(\mathbf{U}_{m,c_{i_2}}\right)_{i*}, \ldots, \left(\mathbf{U}_{m,c_{i_k}}\right)_{i*}$ to (35). □

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their profound comments and suggestions.

REFERENCES

AHARON, M., ELAD, M., AND BRUCKSTEIN, A. M. 2006. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Trans. Signal Process. 54,* 11, 4311–4322.

CHEN, S. S., DONOHO, D. L., AND SAUNDERS, M. A. 2001. Atomic Decomposition by Basis Pursuit. *SIAM Rev. 43,* 1, 129–159.

CHEN, Y., TONG, X., WANG, J., LIN, S., GUO, B., AND SHUM, H.-Y. 2004. Shell Texture Functions. *ACM Trans. Graph. 23,* 3, 343–353.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational Shape Approximation. *ACM Trans. Graph. 23,* 3, 905–914.

DANA, K. J., VAN GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. 1999. Reflectance and Texture of Real-World Surfaces. *ACM Trans. Graph. 18,* 1, 1–34.

DANIELSSON, P.-E. 1980. Euclidean Distance Mapping. *Comput. Graph. Image Process. 14*, 227–248.

DAVIS, G. M., MALLAT, S. G., AND AVELLANEDA, M. 1997. Adaptive Greedy Approximations. *Constr. Approx. 13,* 1, 57–98.

DE LATHAUWER, L., DE MOOR, B., AND VANDEWALLE, J. 2000. 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.

ELAD, M., FIGUEIREDO, M. A. T., AND MA, Y. 2010. On the Role of Sparse and Redundant Representations in Image Processing. *Proc. of the IEEE 98,* 6, 972–982.

FILIP, J. AND HAINDL, M. 2009. Bidirectional Texture Function Modeling: A State of the Art Survey. *IEEE Trans. Pattern Anal. Mach. Intell. 31,* 11, 1921–1940.

HEIDRICH, W., DAUBERT, K., KAUTZ, J., AND SEIDEL, H.-P. 2000. Illuminating Micro Geometry Based on Precomputed Visibility. In *Proc. of SIGGRAPH '00*. 455–464.

HEIDRICH, W. AND SEIDEL, H.-P. 1999. Realistic, Hardware-Accelerated Shading and Lighting. In *Proc. of SIGGRAPH '99*. 171–178.

JOLLIFFE, I. T. 2002. *Principal Component Analysis*, 2nd ed. Springer-Verlag.

KAMBHATLA, N. AND LEEN, T. K. 1997. Dimension Reduction by Local Principal Component Analysis. *Neural Comput. 9,* 7, 1493–1516.

KOLDA, T. G. AND BADER, B. W. 2009. Tensor Decompositions and Applications. *SIAM Rev. 51,* 3, 455–500.

KOUDELKA, M. L., MAGDA, S., BELHUMEUR, P. N., AND KRIEGMAN, D. J. 2003. Acquisition, Compression, and Synthesis of Bidirectional Texture Functions. In *Proc. of Texture '03*. 59–64.

KREUTZ-DELGADO, K., MURRAY, J. F., RAO, B. D., ENGAN, K., LEE, T.-W., AND SEJNOWSKI, T. J. 2003. Dictionary Learning Algorithms for Sparse Representation. *Neural Comput. 15,* 2, 349–396.

LAWRENCE, J., BEN-ARTZI, A., DECORO, C., MATUSIK, W., PFISTER, H., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2006. Inverse Shade Trees for Non-Parametric Material Representation and Editing. *ACM Trans. Graph. 25,* 3, 735–745.

LEFEBVRE, S. AND HOPPE, H. 2006. Appearance-Space Texture Synthesis. *ACM Trans. Graph. 25,* 3, 541–548.

MALLAT, S. G. AND ZHANG, Z. 1993. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Trans. Signal Process. 41,* 12, 3397–3415.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A Data-Driven Reflectance Model. *ACM Trans. Graph. 22,* 3, 759–769.

MCALLISTER, D. K., LASTRA, A., AND HEIDRICH, W. 2002. Efficient Rendering of Spatial Bi-Directional Reflectance Distribution Functions. In *Proc. of Graphics Hardware '02*. 79–88.

MÜLLER, G., MESETH, J., AND KLEIN, R. 2003. Compression and Real-Time Rendering of Measured BTFs Using Local PCA. In *Proc. of VMV '03*. 271–279.

MÜLLER, G., MESETH, J., AND KLEIN, R. 2004. Fast Environmental Lighting for Local-PCA Encoded BTFs. In *Proc. of CGI '04*. 198–205.

NAYAR, S. K., BELHUMEUR, P. N., AND BOULT, T. E. 2004. Lighting Sensitive Display. *ACM Trans. Graph. 23,* 4, 963–979.

PATANÈ, G. AND RUSSO, M. F. 2001. The Enhanced LBG Algorithm. *Neural Netw. 14,* 9, 1219–1237.

POLICARPO, F., OLIVEIRA, M. M., AND COMBA, J. L. D. 2005. Real-Time Relief Mapping on Arbitrary Polygonal Surfaces. In *Proc. of I3D '05*. 155–162.

QIN, Z., MCCOOL, M. D., AND KAPLAN, C. S. 2006. Real-Time Texture-Mapped Vector Glyphs. In *Proc. of I3D '06*. 125–132.

REBOLLO-NEIRA, L. AND LOWE, D. 2002. Optimized Orthogonal Matching Pursuit Approach. *IEEE Signal Process. Lett. 9,* 4, 137–140.

ROWEIS, S. T. AND SAUL, L. K. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science 290,* 5500, 2323–2326.

RUITERS, R. AND KLEIN, R. 2009. BTF Compression via Sparse Tensor Decomposition. *Comput. Graph. Forum 28,* 4, 1181–1188.

SATTLER, M., SARLETTE, R., AND KLEIN, R. 2003. Efficient and Realistic Visualization of Cloth. In *Proc. of EGSR '03*. 167–178.

SCHÖLKOPF, B., SMOLA, A. J., AND MÜLLER, K.-R. 1998. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Comput. 10,* 5, 1299–1319.

SLOAN, P.-P. J., HALL, J., HART, J. C., AND SNYDER, J. 2003a. Clustered Principal Components for Precomputed Radiance Transfer. *ACM Trans. Graph. 22,* 3, 382–391.

SLOAN, P.-P. J., KAUTZ, J., AND SNYDER, J. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. *ACM Trans. Graph. 21,* 3, 527–536.

SLOAN, P.-P. J., LIU, X., SHUM, H.-Y., AND SNYDER, J. 2003b. Bi-Scale Radiance Transfer. *ACM Trans. Graph. 22,* 3, 370–375.

SMILDE, A., BRO, R., AND GELADI, P. 2004. *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley Press.

STANFORD COMPUTER GRAPHICS LABORATORY. 2011. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/.

SUN, X., HOU, Q., REN, Z., ZHOU, K., AND GUO, B. 2011. Radiance Transfer Biclustering for Real-Time All-Frequency Biscale Rendering. *IEEE Trans. Vis. Comput. Graph. 17,* 1, 64–73.

SUN, X., ZHOU, K., CHEN, Y., LIN, S., SHI, J., AND GUO, B. 2007. Interactive Relighting with Dynamic BRDFs. *ACM Trans. Graph. 26,* 3.

SUYKENS, F., VOM BERGE, K., LAGAE, A., AND DUTRÉ, P. 2003. Interactive Rendering with Bidirectional Texture Functions. *Comput. Graph. Forum 22,* 3, 463–472.

TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science 290,* 5500, 2319–2323.

TSAI, Y.-T. 2009. Parametric Representations and Tensor Approximation Algorithms for Real-Time Data-Driven Rendering. Ph.D. thesis, National Chiao Tung University.

TSAI, Y.-T., FANG, K.-L., LIN, W.-C., AND SHIH, Z.-C. 2011. Modeling Bidirectional Texture Functions with Multivariate Spherical Radial Basis Functions. *IEEE Trans. Pattern Anal. Mach. Intell. 33,* 7, 1356–1369.

TSAI, Y.-T. AND SHIH, Z.-C. 2006. All-Frequency Precomputed Radiance Transfer Using Spherical Radial Basis Functions and Clustered Tensor Approximation. *ACM Trans. Graph. 25,* 3, 967–976.

VASILESCU, M. A. O. AND TERZOPOULOS, D. 2003. Multilinear Subspace Analysis of Image Ensembles. In *Proc. of CVPR '03*. 93–99.

VASILESCU, M. A. O. AND TERZOPOULOS, D. 2004. TensorTextures: Multilinear Image-Based Rendering. *ACM Trans. Graph. 23,* 3, 336–342.

VLASIC, D., BRAND, M., PFISTER, H., AND POPOVIĆ, J. 2005. Face Transfer with Multilinear Models. *ACM Trans. Graph. 24,* 3, 426–433.

WANG, H., WU, Q., SHI, L., YU, Y., AND AHUJA, N. 2005. Out-of-Core Tensor Approximation of Multi-Dimensional Matrices of Visual Data. *ACM Trans. Graph. 24,* 3, 527–535.

WANG, L., WANG, X., TONG, X., LIN, S., HU, S.-M., GUO, B., AND SHUM, H.-Y. 2003. View-Dependent Displacement Mapping. *ACM Trans. Graph. 22,* 3, 334–339.

WRIGHT, J., MA, Y., MAIRAL, J., SPAIRO, G. R., HUANG, T. S., AND YAN, S. 2010. Sparse Representation for Computer Vision and Pattern Recognition. *Proc. of the IEEE 98,* 6, 1031–1044.

WU, Q., XIA, T., CHEN, C., LIN, H.-Y. S., WANG, H., AND YU, Y. 2008. Hierarchical Tensor Approximation of Multi-Dimensional Visual Data. *IEEE Trans. Vis. Comput. Graph. 14,* 1, 186–199.

XU, K., JIA, Y.-T., FU, H., HU, S.-M., AND TAI, C.-L. 2008. Spherical Piecewise Constant Basis Functions for All-Frequency Precomputed Radiance Transfer. *IEEE Trans. Vis. Comput. Graph. 14,* 2, 454–467.

Received April 2011; revised October 2011; accepted December 2011