Eurographics Symposium on Rendering 2021
A. Bousseau and M. McGuire
(Guest Editors)

Volume 40 (2021), Number 4

Moving Basis Decomposition for Precomputed Light Transport

Ari Silvennoinen and Peter-Pike Sloan

Activision Publishing, Inc.

Figure 1: *A production scene lit using our technique from a recent 60Hz AAA-title shipped on Playstation 4, Playstation 5, Windows PC, Xbox One, and Xbox Series X/S platforms. We present a method for efficient representation of precomputed light transport data, that enables compressed rendering and seamless reconstruction in large-scale, real-world applications under hard real-time constraints.*

Abstract

We study the problem of efficient representation of potentially high-dimensional, spatially coherent signals in the context of precomputed light transport. We present a basis decomposition framework, Moving Basis Decomposition (MBD), that generalizes many existing basis expansion methods and enables high-performance, seamless reconstruction of compressed data. We develop an algorithm for solving large-scale MBD problems. We evaluate MBD against state-of-the-art in a series of controlled experiments and describe a real-world application, where MBD serves as the backbone of a scalable global illumination system powering multiple, current and upcoming 60Hz AAA-titles running on a wide range of hardware platforms.

CCS Concepts

• ***Computing methodologies*** → ***Rendering;*** *Image compression;*

1. Introduction

Physically based rendering — one of the key goals in the field of computer graphics — is based on a realistic, global lighting model. Despite recent advances in GPU hardware, real-time global illumination is not feasible for large scenes under the performance constraints of modern video games [SSS*20]. As a result, precomputed lighting techniques are still the most commonly used lighting solution in many applications [Bar17].

Scaling precomputed lighting solutions, e.g., spherical harmonic irradiance volumes [GSHG98], with growing virtual world sizes poses an important challenge. As an example, the raw source data for the indirect illumination in the scene depicted in Figure 1, represented as volumetric irradiance — a continuous, 5D (position × direction) signal per color channel — takes 1.5GB of memory, or 48 bytes for 12 coefficients per voxel, even with only a linear, or first order spherical harmonics encoding. For many target platforms, this would occupy most of the available GPU memory budget, leaving little to no room for any geometry or materials.

We consider the problem of compressing spatially coherent, potentially high-dimensional signals and present a new basis decomposition framework, which we call *moving basis decomposition* (MBD), that enables scalable rendering of compressed precomputed lighting data. In contrast to previous approaches, our solution provides seamless reconstruction with controlled error while keeping high compression ratios. For example, the indirect lighting in Figure 1 can be directly rendered in the MBD representation us-

ing 1.09 bytes per voxel to reconstruct the target linear SH RGB irradiance signal, resulting in a roughly 44:1 compression ratio.

1.1. Contributions and Limitations

The main contributions in this paper are the following:

1. We present a basis decomposition framework, *moving basis decomposition* (MBD), that generalizes many existing linear basis decomposition methods (Section 3.1).
2. We develop an algorithm for approximately solving MBD problems and show that it reliably produces good solutions and scales to very large problems, e.g., problems with $\mathcal{O}(10^8)$ unknown parameters (Sections 3.2 and 3.3).
3. We present an application to precomputed light transport (Section 4) and demonstrate empirically, both qualitatively and quantitatively, improved performance of our method compared to existing methods.

The main limitation of our method is the assumption regarding the spatial coherence of the input signal. We discuss our assumptions and their implications with more detail in Sections 2 and 5.

2. Preliminaries and Related Methods

We begin by introducing a framework that allows us to set the stage as well as compare and contrast our work to existing methods.

2.1. Basis Decomposition Framework

We're studying the following problem: given a (discrete) vector field $\{(\mathbf{x}_i, \mathbf{y}_i)\}$[†], where $\mathbf{x}_i \in R^3$ is a *spatial coordinate* in the *spatial domain* R^3 and $\mathbf{y}_i = f(\mathbf{x}_i) \in R^D$ is a potentially high-dimensional *data vector* in the *data domain* R^D, produced by some input generating function $f : R^3 \to R^D$, our goal is to find a sparse representation of the data set $\{\mathbf{y}_i\}$ in some basis. That is, we seek a basis decomposition, such that every input vector $\mathbf{y} \in \{\mathbf{y}_i\}$ can be approximated by a linear combination of a small number basis vectors $\hat{\mathbf{y}} = \sum_l c_l \mathbf{b}_l$, where $\mathbf{b}_l \in R^D, c_l \in R$, the approximation error $\|\hat{\mathbf{y}} - \mathbf{y}\|$ is small in some norm and the rank L satisfies $L \ll D$. Note that, for convenience, we assume the data vectors $\{\mathbf{y}_i\}$ to have mean zero.

Not all solutions are equal. Given some error threshold, we can compare different basis representations by their *memory efficiency*; that is, the total number of coefficients and basis vectors in the representation such that the approximation error is under the threshold. For compression applications, of course, the fewer coefficients and basis vectors we need in order to stay under a given error threshold, the better. Furthermore, we can compare two equal-error representations by their error distribution, preferring *seamless solutions* — solutions where error is a smooth function of space — to avoid obvious visual artifacts associated with highly non-uniform and discontinuous error distributions (Figure 2).

Our goal is finding a memory efficient and seamless decomposition with minimal error.

[†] For notational convenience, we omit the lower and upper bounds for sums, sets and sequences and implicitly assume that $\sum_l \equiv \sum_{l=1}^{L}$, $\{\cdot_i\} \equiv \{\cdot_i\}_{i=1}^{I}, (\cdot_m) \equiv (\cdot_m)_{m=1}^{M}$, etc.

2.2. Related Methods

We continue by highlighting basis expansion and dimensionality reduction techniques in the context of precomputed light transport data compression.

K-Means and K-SVD. Silvennoinen et al. [ST15] applied K-Means vector quantization [Llo82] — a simple model, where each data vector is represented by a cluster mean vector — to compress precomputed light transport operators. In terms of our basis decomposition framework, the coefficients are constants and we only need to find and store the cluster representatives, i.e., the basis vectors. K-SVD [AEB06] can be seen as a generalization of k-means, where instead of cluster representatives, we build a global dictionary of basis vectors and approximate the given data as a sparse linear combination of these shared basis vectors. However, with both K-means and K-SVD, the choice of global basis vectors with non-zero coefficients, i.e., the cluster, depends only on the data vector $\mathbf{y}$, and there are no guarantees of a seamless reconstruction with respect to the spatial coordinate $\mathbf{x}$.

PCA and SVD. PCA-based approaches come with strong error bounds due to their connection to the SVD [GV13]. In contrast to K-SVD, a global PCA basis is not necessarily sparse and the approximation error depends heavily on the input data distribution. Instead of a global PCA basis, *blockwise PCA* (BPCA) [NNJ05] leverages spatial coherence to form spatially localized clusters while *clustered PCA* (CPCA) [SHHS03] partitions the input data set $\{\mathbf{y}_i\}$ into signal-coherent clusters using a modified K-Means algorithm. Both methods then proceed to compute a local, per-cluster basis for each cluster without considering seamless reconstruction in the spatial domain.

Observation. In the above methods, the clusters are independent; that is, the reconstruction does not allow interaction *between* the clusters. As a consequence of this fundamental limitation, there are no guarantees that a reconstruction across cluster boundaries with respect to the *spatial* coordinate x is continuous (Figure 2). Furthermore, this *cluster discontinuity problem* is more pronounced with highly memory efficient representations.

Spectral Methods. Spectral methods, e.g., methods based on Fourier or wavelet transforms, introduce a change of basis $\mathbf{y} = \Psi\mathbf{z}$ to obtain a sparse representation in terms of Ψ [KTHS06; NRH03; WZH07; LZT*08]. Analogous to PCA, the support of the spectral basis is often global but we can introduce locality by working in smaller windows or clusters, similar to the DCT in JPEG [WZH07]. However, as we noted above, independent clusters lead to the cluster discontinuity problem. Continuing the image compression example, this is often visible in terms of "block"-artifacts between the neighboring compression windows.

Non-linear Dimensionality Reduction. Non-linear dimensionality reduction (NLDR) methods can be roughly divided into embeddings [RS00; TDL00] and mappings [Bra03]. For reconstruction, we need a reversible mapping and some of the methods based on local charts, or clusters, avoid the cluster discontinuity problem by coordinating the local clusters [RSH02; VVK02; TR03]. Methods based on local cluster coordination, however, do not directly consider seamless reconstruction in the spatial domain; that is, they coordinate basis vectors for neighboring clusters only in the data

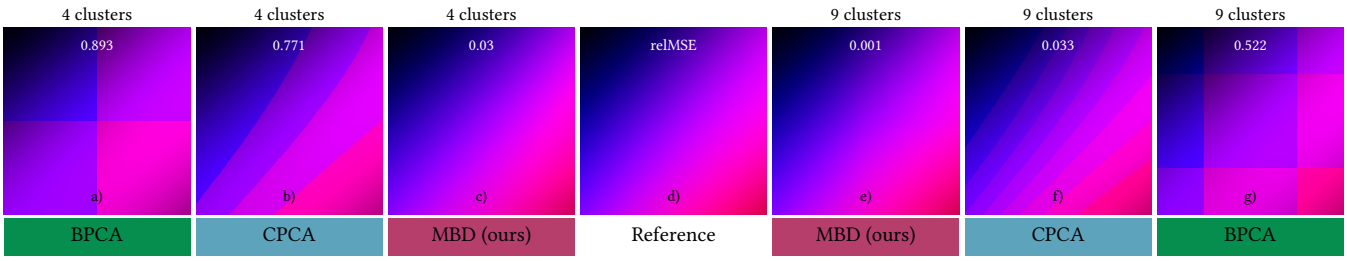

Figure 2: ***Independent clusters and cluster discontinuity problem.*** *A non-linear reference vector field (d) is compressed using block PCA [NNJ05] (a, g), clustered PCA [SHHS03] (b, f), and our method (c, e), using one basis vector per-cluster and either 4 (left) or 9 (right) clusters. Cluster discontinuity artifacts are clearly visible when the reconstruction model does not allow interactions between clusters (a, b, f, g). In contrast, our method (c, e) allows interactions between neighboring clusters via spatial kernels and is visually indistinguishable from the reference (d). Note that figures (c) and (f) have similar error, although (c) is preferable visually, highlighting the fact for roughly equal error solutions, a smooth, uniform error distribution is preferable to a non-uniform, discontinuous one.*

domain, independently of the given spatial coordinates $\{\mathbf{x}_i\}$ using information contained in $\{\mathbf{y}_i\}$ alone.

Neural Methods. Non-linear PCA [BH89] can be seen as a generalization of PCA for capturing non-linear coherence but shares the same limitations as global PCA. A local approach can be derived by considering smaller data windows. For example, Ren et al. [RWG*13] applied neural networks to represent precomputed lighting data in local regions of space utilizing the full data set $\{(\mathbf{x}_i, \mathbf{y}_i)\}$. However, by assuming the windows are independent, their reconstruction is exposed to the cluster discontinuity problem.

In the context of precomputed light transport, previous methods have addressed the cluster discontinuity problem from three main directions:

- **Interpolating the reconstructed values**. To provide visually smooth solution, a common solution is to reconstruct-then-interpolate [LZT*08; ST15; SL17]. In addition to requiring additional memory and compute for the intermediate values, this approach does not provide any control of approximation error due to this additional interpolation step. Furthermore, as we will see in Section 4, reconstruct-then-interpolate approach can be ineffective when dealing with low-frequency cluster artifacts.
- **Increasing the number of basis vectors**. Sloan et al. [SHHS03] proposed to adaptively increase the number of basis vectors until the cluster discontinuity artifacts were unnoticeable. Unfortunately, this solution works against our goal of minimizing the number of basis vectors to achieve high memory efficiency.
- **Overlapping clusters**. To avoid discontinuities between clusters, Ren et al. [RWG*13] expanded the spatial regions to include neighboring samples, similar in spirit to lapped transforms [Cas85]. As we demonstrate in Section 4, this windowing process is not effective in reducing low-frequency artifacts due to cluster discontinuities.

To summarize, methods that assume independent clusters, e.g., K-Means, K-SVD, PCA variants and localized spectral methods, either ignore or apply the above ad-hoc solutions to the cluster discontinuity problem, while methods that allow local cluster coordination in the data domain, e.g., cluster coordination methods in the context of NLDR, do not, by design, consider seamless reconstruction in the spatial domain, leaving a gap which we aim to fill.

Key Idea. Our key idea is to combine interpolation with basis decomposition framework from first principles. In particular, we decouple the spatial frequency of the coefficients and the basis vectors in the basis decomposition model for compression, and apply *spatial kernels* from scattered data interpolation literature to allow coordination and information sharing between neighbors in order to enable seamless reconstruction. Note that in contrast to previous reconstruct-then-interpolate approaches, we seek to jointly minimize the basis expansion approximation and interpolation error by construction.

Next, we formulate our key idea the basis decomposition framework (Section 3.1), consider the related optimization problem (Section 3.2), and develop an algorithm for solving MBD problems (Section 3.3). Finally, we discuss the connections between our method and texture compression, scattered data interpolation and clustering methods in Section 5.

3. Method

3.1. Moving Basis Decomposition

A key component for a moving basis decomposition is the ability to perform interpolation of the basis vectors and coefficients separately in the spatial domain. In particular, we use a kernel formulation of interpolation, where the interpolants for the basis vectors $b : R^3 \to R^D$ and coefficients $c : R^3 \to R$ are written in terms of a spatial kernel expansions:

$$c(\mathbf{x}) = \sum_m \phi_m(\mathbf{x})\mathbf{c}_m \tag{1}$$

$$b(\mathbf{x}) = \sum_n \psi_n(\mathbf{x})\mathbf{B}_n, \tag{2}$$

where $\phi_m : R^3 \to R$ is a *coefficient kernel* and $\psi_n : R^3 \to R$ is a *basis kernel*, $\mathbf{c}_m \in R$ are coefficients and $\mathbf{B}_n \in R^D$ are basis vectors. From this perspective, we can view linear interpolation as a kernel expansion with the usual hat functions, while Shepard's method can be used to construct more general, spatially scattered kernels

with compact support that automatically form a partition-of-unity [She68].

We construct the moving basis decomposition (*MBD*) by combining the kernel formulation of interpolation with the basis decomposition framework (Section 2.1), where the role of the kernel functions ϕ_m and ψ_n is to control the spatial region of influence of the basis vectors and basis coefficients in a decoupled manner. Concretely, a rank L moving basis decomposition is defined as a tuple $((\phi_m),(\psi_n),\mathbf{c},\mathbf{B})$, and the reconstruction $\hat{f}: R^3 \to R^D$ is given by:

$$\hat{f}(\mathbf{x}) = \sum_l c_l(\mathbf{x}) b_l(\mathbf{x}) \tag{3}$$

$$c_l(\mathbf{x}) = \sum_m \phi_m(\mathbf{x}) \mathbf{c}_{m,l} \tag{4}$$

$$b_l(\mathbf{x}) = \sum_n \psi_n(\mathbf{x}) \mathbf{B}_{n,l}, \tag{5}$$

where $\hat{f}(\mathbf{x})$ is the product of two spatial kernel expansions for the moving basis vectors b_l and coefficients c_l, ϕ_m is a coefficient kernel, ψ_n is a basis vector kernel, $\mathbf{c}$ is the *coefficient tensor* and $\mathbf{B}$ is the *basis tensor*. In this context, tensors are an organized collection of vectors or scalars that provide a convenient way of indexing their elements. For example, the lth basis vector over a basis kernel ψ_n is given by $\mathbf{B}_{n,l}$, and the corresponding scalar basis coefficient over a coefficient kernel ϕ_m is given by $\mathbf{c}_{m,l}$.

The coefficients $\mathbf{c}_{m,l}$ are linked to their kernels ϕ_m via index m and the basis vectors $\mathbf{B}_{n,l}$ are linked to their kernels ψ_n via index n, allowing each kernel to be unique. The kernels can all come from a parameterized family, such as linear hat functions, or they can come from different families, allowing flexibility in applications, as we'll see in Section 5. Note that the sums are defined over all kernels, but, depending on the choice of the kernel functions, most terms in this sum will be zero due to the local region of influence, or, compact support of each kernel. For example, in a 3D grid, a trilinear hat kernel is non-zero only in the trilinear footprint of the nearest 8 basis vectors or coefficients for any given query position.

In addition to enabling local information sharing, the spatial kernels ϕ_m and ψ_n allow us to decouple and control the spatial frequency of the low-dimensional coefficients $\mathbf{c}$ and the high-dimensional basis vectors $\mathbf{B}$ separately for compression. Furthermore, the kernels act as glue for seamless reconstruction, allowing us to query the reconstruction $\hat{f}(\mathbf{x})$ at any point $\mathbf{x}$ in the spatial domain. With a suitable choice of kernel functions that form a partition of unity, e.g., linear hat functions, we can think of MBD as interpolating the basis vectors in space and then finding local coefficients that express the input data in terms of this interpolated, moving basis. It is this moving property of the basis that allows the MBD model to continuously adapt to local changes in the spatial domain.

Assumptions. In order for a moving basis decomposition to be feasible, we assume that the high-dimensional input data $\{\mathbf{y}_i\}$ lives on a low-dimensional manifold that is locally linear and piecewise smoothly varying in space as a function of the spatial coordinate $\mathbf{x} \in \mathbf{R}^3$. In the context of our target application, the validity of this assumption is supported by the theory of locally low dimensional light transport [MSRB07].

3.2. Optimization Problem

The previous section introduced the moving basis decomposition, and in this section, we discuss the related optimization problem and develop an algorithm for approximately solving MBD problems. We formulate our objective as a minimization problem in terms of the residual $r: R^3 \to R^D$:

$$r(\mathbf{x}) = \hat{f}(\mathbf{x}) - f(\mathbf{x}), \tag{6}$$

where $\hat{f}$ is the MBD reconstruction (Equation 3) and f is the target function, i.e., either the input generating function f, or, in cases where it is impractical to directly sample the input generating function f, the extension of the discrete input set $\{(\mathbf{x}_i,\mathbf{y}_i)\}$ over R^3 space via interpolation. For example, in our light transport results in Section 4, we use a dense trilinear grid to represent the input data set and evaluate the target function f via interpolation instead of recomputing the light transport operator — expensive operation — for each query point.

Loss. Given fixed sequences of kernels (ϕ_m) and (ψ_n), we define our objective using a loss function $\mathcal{L}(\mathbf{B},\mathbf{c})$ as follows:

$$\mathcal{L}(\mathbf{B},\mathbf{c}) = \frac{1}{2}\int \|r(\mathbf{x})\|_2^2 \mathrm{d}\mathbf{x} = \frac{1}{2}\sum_k \int r_k(\mathbf{x})^2 \mathrm{d}\mathbf{x}, \tag{7}$$

where r_k is the kth component of the residual vector r and the integeral are over R^3. We define the loss function over the full spatial domain R^3 instead of over the discrete input points $\{(\mathbf{x}_i,\mathbf{y}_i)\}$ to prevent the solver from overfitting to the discrete input data.

Gradient and Hessian. The derivatives of the loss function with respect to the unknown parameters $\mathbf{B},\mathbf{c}$ in terms of the kernels ϕ_m,ψ_n and the coefficient and basis expansions c_l and b_l are:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{c}_{m,l}} = \sum_k \int r_k(\mathbf{x}) \phi_m(\mathbf{x}) b_{l,k}(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}_{n,l,k}} = \int r_k(\mathbf{x}) \psi_n(\mathbf{x}) c_l(\mathbf{x}) d\mathbf{x} \tag{9}$$

Second order derivatives, or the diagonal Hessian entries are:

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{c}_{m,l}^2} = \sum_k \int \phi_m(\mathbf{x})^2 b_{l,k}(\mathbf{x})^2 d\mathbf{x} \tag{10}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{B}_{n,l,k}^2} = \int \psi_n(\mathbf{x})^2 c_l(\mathbf{x})^2 d\mathbf{x} \tag{11}$$

Regularization. MBD solutions are not unique, since for any given solution, we can form an equivalent loss solution by scaling the coefficient tensor $\mathbf{c}$ by some $\alpha > 0$ and the basis tensor $\mathbf{B}$ by its inverse. In other words, $\mathcal{L}(\mathbf{B},\mathbf{c}) = \mathcal{L}(\frac{1}{\alpha}\mathbf{B},\alpha C)$. To resolve this *scale ambiguity*, we add a regularization term to our objective that penalizes coefficients far from the origin, pushing the intrinsic scale to the basis vectors[‡]. Our final objective becomes:

$$\mathcal{L}_\lambda(\mathbf{B},\mathbf{c}) = \mathcal{L}(\mathbf{B},\mathbf{c}) + \frac{1}{2}\lambda\|\mathbf{c}\|_F^2, \tag{12}$$

‡ We use $\lambda = 0.0001$ for all our results.

where $\|\cdot\|_F$ is the Frobenius norm. The added penalty term results in slightly altered derivatives with respect to the coefficients $\mathbf{c}$:

$$\frac{\partial \mathcal{L}_\lambda}{\partial \mathbf{c}_{m,l}} = \frac{\partial \mathcal{L}}{\partial \mathbf{c}_{m,l}} + \lambda \mathbf{c}_{m,l} \tag{13}$$

$$\frac{\partial^2 \mathcal{L}_\lambda}{\partial \mathbf{c}^2_{m,l}} = \frac{\partial^2 \mathcal{L}}{\partial \mathbf{c}^2_{m,l}} + \lambda \tag{14}$$

.

Optimization Problem. Finally, the MBD optimization problem can now be stated as follows:

$$\arg\min_{\mathbf{B},\mathbf{c}} \mathcal{L}_\lambda(\mathbf{B},\mathbf{c}). \tag{15}$$

The optimization problem in Equation 15 is non-linear and bi-convex; it is convex in either $\mathbf{B}$ or $\mathbf{c}$ separately. The problem is global but has a sparse structure that comes from the spatial support of the chosen kernels. Spatially local kernels prevent direct long-range interactions between the parameters.

3.3. Solver

To solve the optimization problem in Equation 15, we use stochastic quasi-Newton descent with diagonal Hessian approximation and line-search backtracking [NW06].

In order to evaluate the gradient (Equations 13, 9) and the diagonal Hessian (Equations 14, 11), we use Monte-Carlo integration and apply stratified sampling to generate one spatial sample for each of the coefficient kernels. Note that since the coefficient kernels overlap, each coefficient derivative estimator uses, on average, 8 samples. For further variance reduction, we apply the technique of common random numbers [Owe13] and reuse the coefficient kernel samples to evaluate the basis vector derivatives. Again, since the basis kernels typically have larger support than the coefficient kernels, each basis derivative estimator can use all samples under its support. In summary, each iteration of the optimization loop consists of:

1. Generating spatial samples and evaluating the Monte Carlo estimators for the derivatives, i.e., gradient and diagonal Hessian, using Equations 13, 9, 14, 11.
2. Updating the current values of the unknown parameters $\mathbf{c},\mathbf{B}$ via a quasi-Newton descent step using the gradient and diagonal Hessian estimators from step 1).
3. Performing a back-tracking line search in the gradient direction if the loss increases.

Next, we'll discuss how to initialize the parameter values $\mathbf{c},\mathbf{B}$.

Initialization. A good initial guess for the unknown parameters $\mathbf{c},\mathbf{B}$ has a big impact on convergence. We evaluated three methods for providing a starting point for our solver:

1. **Random**. Each parameter is initialized from a uniform distribution in $(-1,1)$.
2. **Local PCA**. The basis vectors associated with kernel ψ_n are initialized with block PCA vectors, where block contains all data points $(\mathbf{x},\mathbf{y})$ under the support of the kernel ψ_n. After initializing the basis vectors, the coefficients $\mathbf{c}$ are determined by solving a least-squares problem to minimize $\mathcal{L}_\lambda(\mathbf{c};\mathbf{B})$ keeping the basis vectors $\mathbf{B}$ fixed.
3. **Global PCA**. The basis vectors $\mathbf{B}$ are initialized using global PCA and the coefficients $\mathbf{c}$ are determined as with the local PCA method.

The random initialization method converges slowly, while local PCA converges quickly but is prone to getting stuck in a local minima due to overfitting. Out of the methods we evaluted, global PCA provides fast convergence, and, in contrast to local PCA, avoids getting stuck in local minima. Thus, we use global PCA initialization in all our results and based on our experiments, the global PCA initialized solver converges typically in 16-128 iterations.

4. Results

In this section, we present experimental results in an application to precomputed radiance transport and evaluate our method both qualitatively and quantitatively against previous work. For comparison, we implemented the following methods:

- **BPCA** *Blockwise PCA* (BPCA) [NNJ05] uses fixed spatial clusters — or blocks — that match the support of the basis kernels we use for MBD and computes a unique PCA basis for each cluster.
- **WBPCA** *Windowed blockwise PCA* (WBPCA) is a windowed, or lapped, version of BPCA; that is, the blocks are twice as large to incorporate more data points in the neighborhood with the goal of reducing discontinuities at cluster boundaries.
- **CPCA** We implemented the highest quality, distance-to-subspace variant of *clustered PCA* (CPCA), as detailed in [SHHS03].

In our comparisons, we are using equal number of basis vectors and coefficients for all the methods. In particular, this means that CPCA uses more memory, as it needs to store the high-frequency cluster index in addition to basis vectors and coefficients.

4.1. Precomputed Light Transport Experiments

We begin with an application to precomputed light transport (Figures 3, 4). In our application, the input data consists of a 3D light transport volume, where each voxel contains a concatenated, 324-dimensional vector representing both direct-to-indirect and direct transport operators in the quadratic Spherical Harmonic (SH) basis; that is, three 9x9 SH direct-to-indirect transport matrices for RGB color channels and one 9x9 transport matrix for direct illumination, stacked into a single 324-dimensional vector. In this application, the kernels ϕ and ψ are the usual trilinear hat kernels of their corresponding volume textures to enable fast, hardware accelerated texture lookups.

Compression is achieved by representing the input data using a basis expansion and storing less basis vectors than coefficients. Moreover, we can control the number of basis vectors by adjusting the resolution of the 3D basis kernel grid, while keeping the coefficient grid at the input data resolution. For CPCA, which does not rely on any spatial structure, we set the number of clusters to match the resolution of the 3D basis kernel grid. In addition, since the comparison methods are not directly filterable, we resample all reconstructions to a 3D trilinear grid before performing the final lookup for a fair comparison with interpolation enabled for all methods.

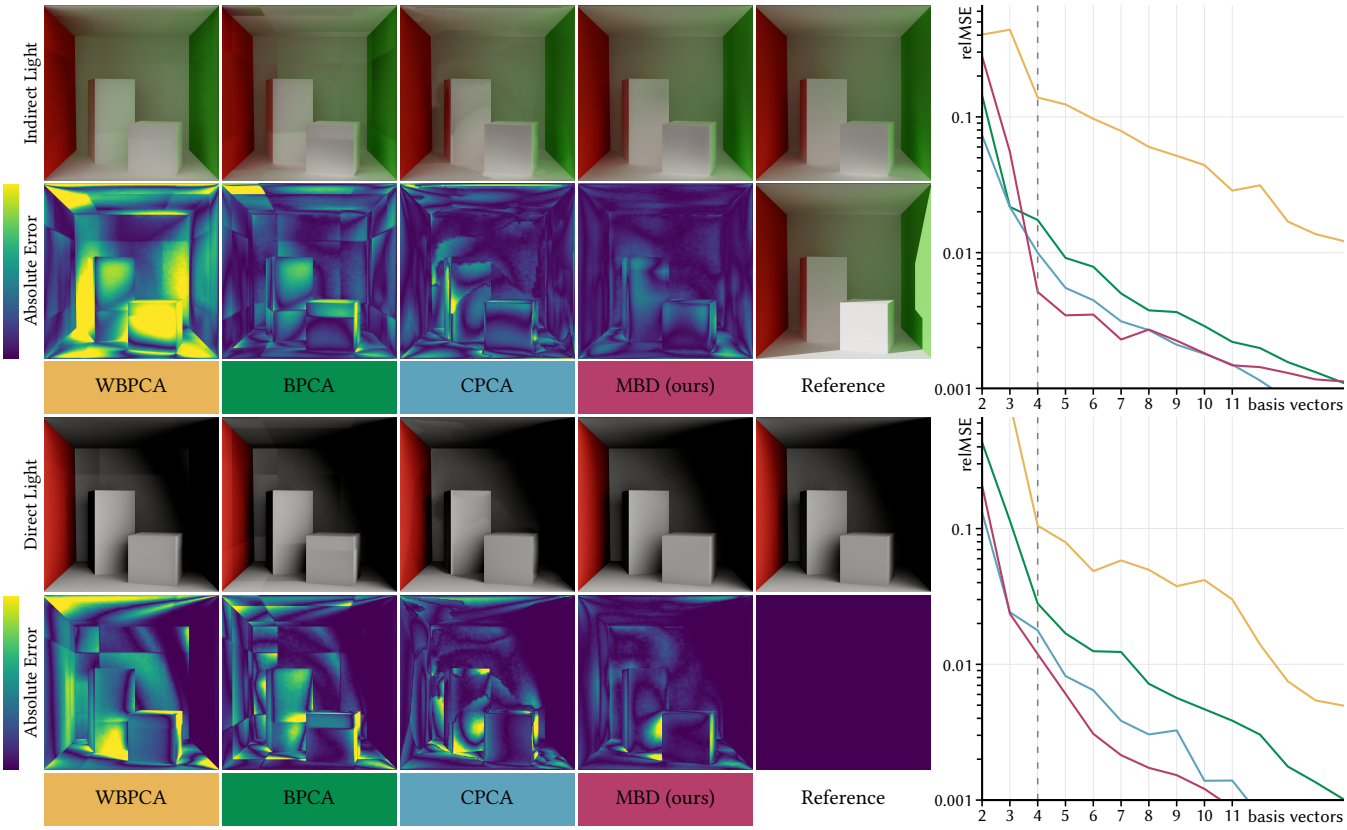

Figure 3: ***Application to precomputed light transport****. We compress a joint, high-dimensional signal consisting of direct-to-indirect diffuse light transport (top) and direct sky visibility (bottom) components using different methods and compare both the reconstruction quality and the resulting approximation error. (The reference column in the top row shows the full, direct and indirect lighting instead a zero error image). The key observation is that MBD provides qualitatively seamless results with quantitatively low error. Furthermore, the convergence in error is consistent as the number of basis vectors increases. Note that MBD has competitive error compared to CPCA, even though MBD uses less memory since the cluster mapping is implicit. (The dashed line in the graph indicates the number of basis vectors used in the images.)*

4.2. Quality and Basis Vector Count

Next, we investigate how the approximation changes with respect to varying the number of basis vectors, i.e., the local rank L of the expansion, both qualitatively and quantitatively (Figures 6 and 5, respectively). For a fixed lighting environment, we compare the convergence visually in Figure 6 and observe that MBD has qualitatively the least artifacts compared to the comparison methods. In order to analyze the convergence dynamics for all possible lighting environments, we study the Frobenius norm of the operator approximation error (Figure 5). We observe that the signal-agnostic MBD is able to match the performance of the signal-specialized CPCA while using less memory.

4.3. Quality and Basis Support Size

The previous section demonstrated that, empirically, MBD provides consistent results with varying number of basis vectors. Now, we investigate how the approximation behaves when adjusting the size of the spatial support of the basis kernels while keeping the number of basis vectors fixed (Figure 7). In contrast to the comparison methods, which are more sensitive to the number of clusters, MBD yields a consistent approximation even with a small number of basis kernel functions with increasing basis kernel support size corresponding to a decreasing basis kernel grid resolution.

4.4. Computation Times

Finally, we compare the computation times for the methods (Figure 8). Although our MBD solver is based on gradient descent — and therefore it is particularly well suited to a GPU implementation — we implemented all methods in multi-threaded C/C++ for a fair comparison. The computation times were measured on a computer equipped with a Intel Core i9-9960X CPU and 128GB RAM.

5. Discussion

We've demonstrated that our method yields qualitatively better and quantitatively similar results with less memory and faster computation times than CPCA, while enabling more efficient direct rendering from the compressed format. Furthermore, our method is robust and consistent with respect to variation in both the number of basis

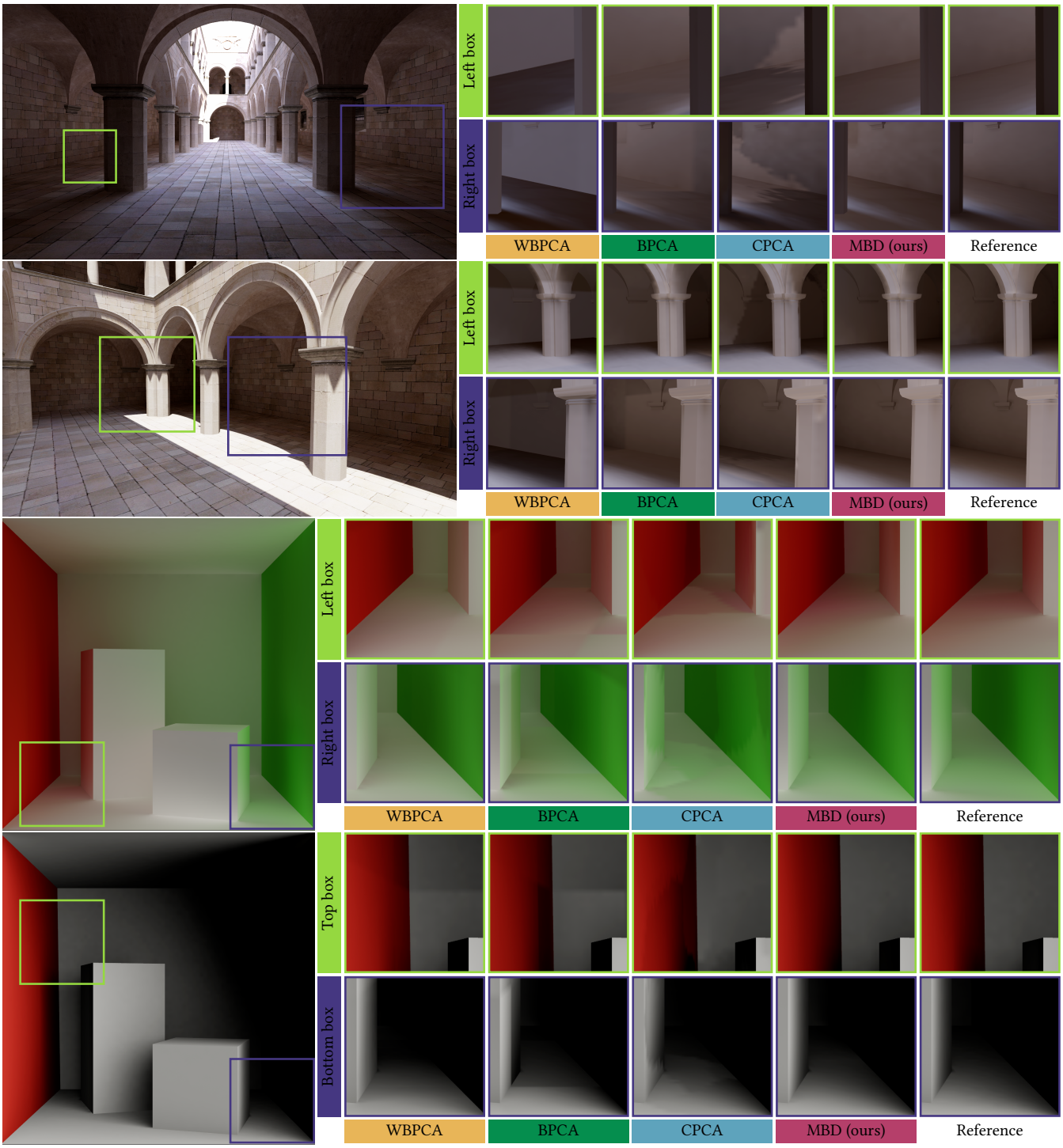

Figure 4: ***Seamless reconstruction comparison***. *Given an equal number of basis vectors, we highlight the differences between the comparison methods in the zoomed-in regions. In contrast to WBPCA, BPCA, and CPCA that suffer from cluster discontinuity problem, MBD provides a seamless reconstruction that is visually close to the reference. The input data consists of light transport operator volume (Section 4.1) with resolution 128x128x128 (Sponza/top) and 64x64x64 (Cornell/bottom). Note that the cluster discontinuity problem is visible even when the reconstruction is resampled to a volume texture for continuous reconstruction. In other words, using a spatial high-frequency, continuous reconstruction is not effective in masking low-frequency cluster discontinuities.*

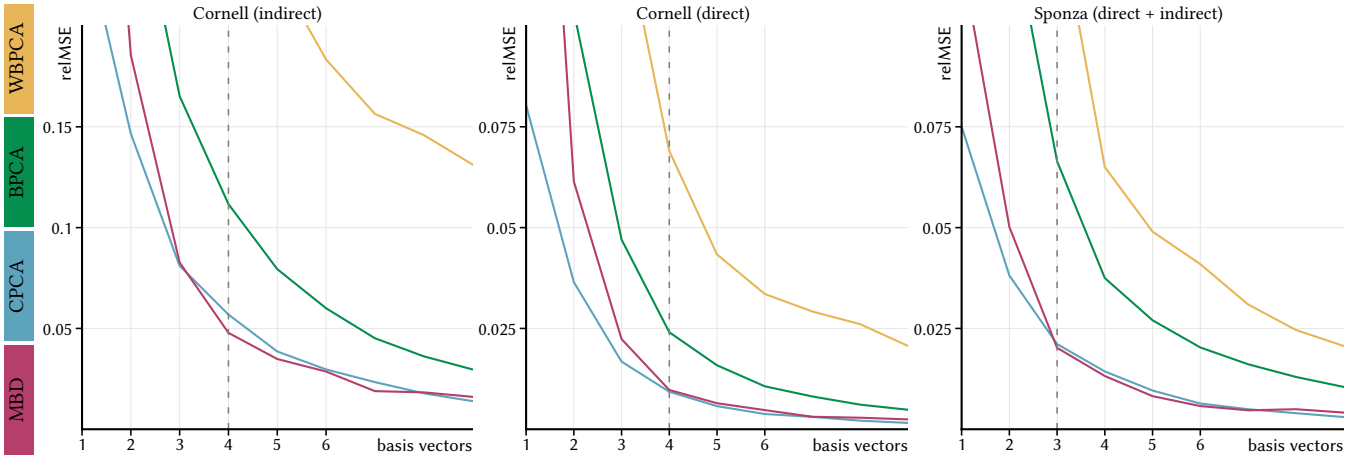

Figure 5: ***Basis vector count and operator approximation error****. As the number of basis vectors increases, the Frobenius norm of the difference between the target light transport operator and the various approximations decreases. MBD and CPCA have similar empirical convergence rates while MBD requires less memory due to its non-adaptive basis support grid. For comparison, both BPCA and WBPCA use identical basis support grid as MBD but suffer from slower convergence rates. (The dashed line indicates the number of basis vectors used for comparisons throughout the results, unless stated otherwise).*

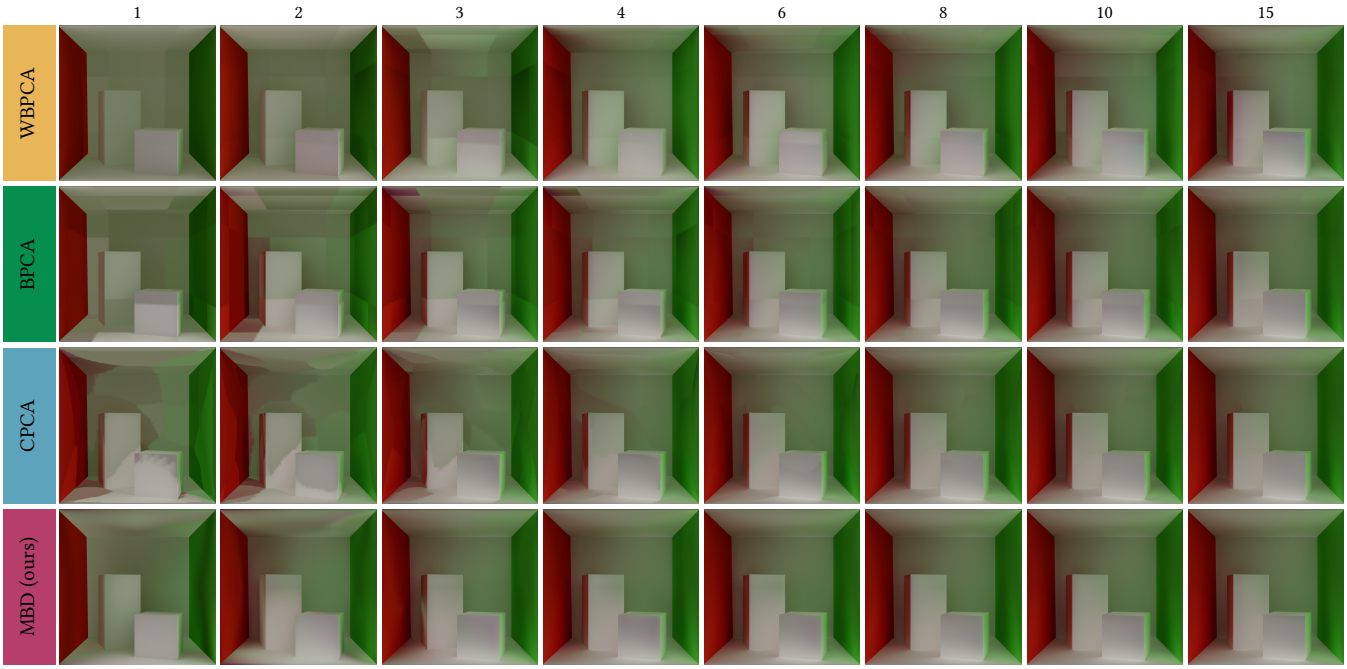

Figure 6: ***Basis vector count and approximation quality.*** *We compare the reconstruction quality as the number of basis vectors, or the local rank, increases. Note that MBD converges fast; with 4 basis vectors, MBD is visually close to the reference solution while WBPCA, BPCA, and CPCA show visible artifacts even with 15 basis vectors (we encourage the reader to zoom in the PDF images).*

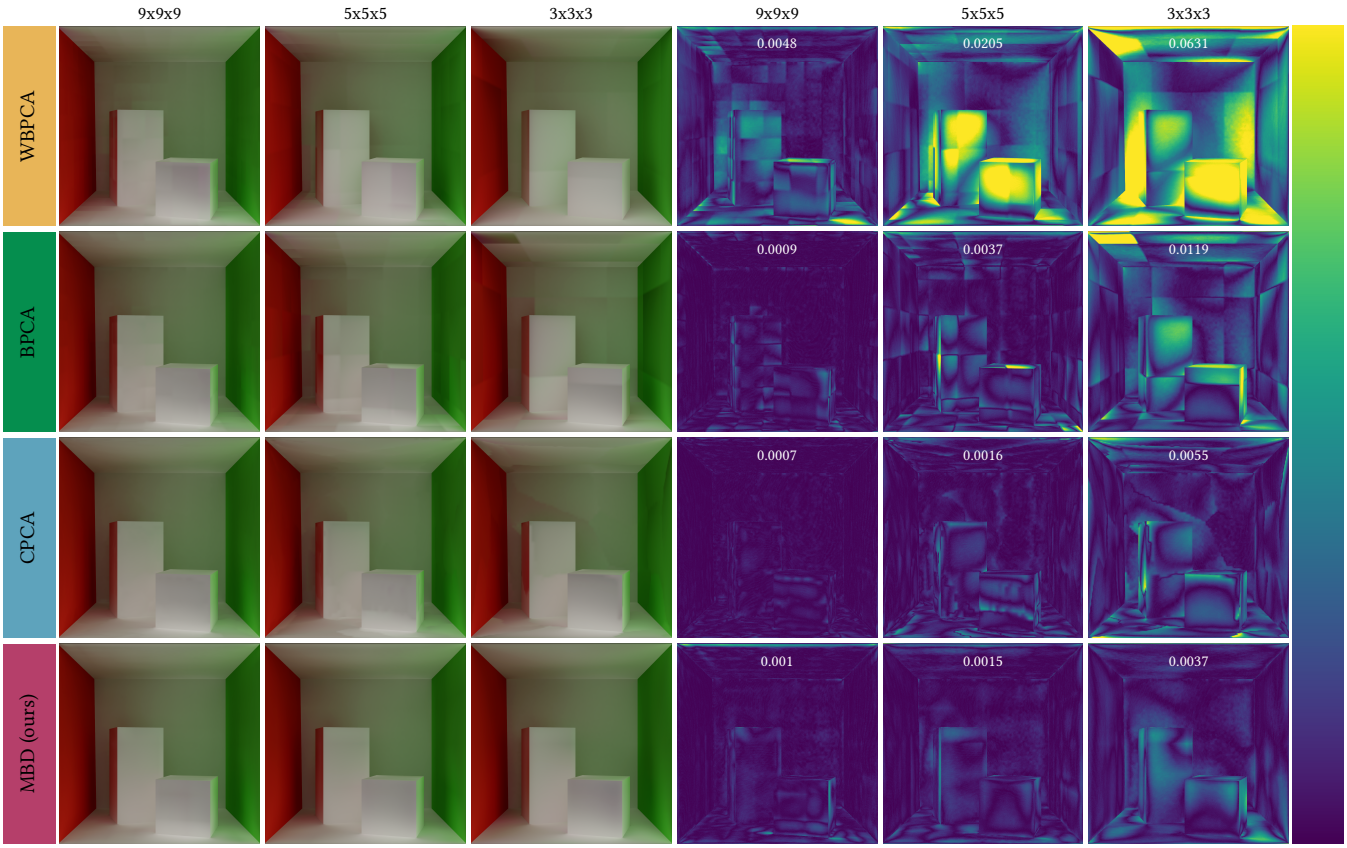

Figure 7: ***Basis vector grid resolution and approximation quality.*** *We compare the reconstruction (left) and the approximation error (right) as the number of clusters decreases from $9^3 = 729$ to $3^3 = 27$ as a function of basis vector grid size. The right panel visualizes the absolute error distribution and shows the relative mean squared error (white labels). WBPCA, BPCA, and CPCA have visible discontinuities at hard cluster boundaries, while MBD is robust with respect to varying basis kernel support size.*

Resolution	$16^3/3^3$	$32^3/5^3$	$64^3/9^3$
WBPCA	4.7s (1.31x)	22.5s (0.95x)	189.5s (0.62x)
BPCA	3.3s (0.92x)	7.8s (0.33x)	52.2s (0.17x)
CPCA	3.6s (1.00x)	23.7s (1.00x)	303.7s (1.00x)
MBD (ours)	2.1s (0.58x)	12.6s (0.53x)	199.3s (0.66x)

Figure 8: ***Computation times.*** *We compare the computation times as a function of increasing target volume/basis resolution. Note that while our method is slightly slower than a purely local BPCA, MBD is faster to compute than CPCA for all resolutions.*

vector and the size of the spatial support of the basis vector kernels, and thus provides a practical and efficient framework for approximation of locally low rank vector fields with a comparatively small number of basis vectors.

5.1. Benefits and limitations

The main benefit of MBD is that it provides *seamless reconstruction* and *efficient random access queries* with *lower error* compared to previous block based methods, such as BPCA. Since the MBD representation is filterable, it enables direct rendering using the compressed representation without decompressing to an intermediate, filterable representation. This additional step uses more memory, compute and it comes without any guarantees on the resulting approximation error. In addition, the choice of the basis and coefficient kernels allow flexibility for applications and makes it possible to mix and match kernels to better suit the problem. We take advantage of this additional freedom in Figure 1 by storing the high-dimensional basis vectors in a low-resolution (256x256) 2D texture over the terrain heightfield while the low-dimensional coefficients are stored in high-resolution (2048x2048x8) 3D volume texture. This hybrid, directly filterable and renderable MBD representation takes 1.09 bytes per voxel to reconstruct the linear SH RGB irradiance in 3D space using a rank-3 MBD.

The main limitation of MBD is the assumption regarding the spatial coherence of the input data. In contrast to CPCA, MBD is unable to capture long-range coherence in the target signal using only locally supported kernels. That is, CPCA is free to leverage any structure in the data domain without being tied to the spatial domain. In addition, MBD requires us to solve a global optimization problem, where block based methods, such as BPCA, are purely

local. However, compared to existing methods, our approximate solver is competitive in terms of computation times and is scalable to large problems. For example, the scene in Figure 1 has on the order of $O(10^8)$ unknowns and the solver takes only a few minutes. In other words, the solver time is a small fraction of the time it takes to generate the input data, making the method suitable for production pipelines.

We conclude this section by drawing connections between our method and texture compression, scattered data interpolation and clustering methods before considering some open questions in Section 5.5.

5.2. Relation to Texture Compression Methods

The idea of utilizing spatial smoothness of the input vector field to aid compression is widely used in texture compression [NLP*12; Fen03] and its application to compression of lighting data [MRP98; KKSM17; XP04]. In particular, Fenney [Fen03] describes a method — PVRTC — for block texture compression that is closely related to our method. PVRTC reconstruction is defined as a convex combination of two linearly interpolated and upsampled color values. In terms of the basis decomposition framework, we can express PVRTC reconstruction as follows:

$$\hat{f}_{PVRTC}(\mathbf{x}) = c(\mathbf{x})b_1(\mathbf{x}) + (1 - c(\mathbf{x}))b_2(\mathbf{x}) \tag{16}$$

$$b_l(x) = \sum_n \psi_n(\mathbf{x})\mathbf{B}_{n,l}, \quad l \in \{1,2\} \tag{17}$$

where ψ_n is the bilinear hat kernel and $0 \leq c(\mathbf{x}) \leq 1$ is a *coupled weight coefficient* that determines how to blend between the two upsampled color values $b_1(\mathbf{x})$ and $b_2(\mathbf{x})$. This line segment lies on an affine rank-1 subspace, and, assuming linear independence of $b_1(\mathbf{x})$ and $b_2(\mathbf{x})$, this affine subspace is a small subset of the plane $\text{span}\{b_1(\mathbf{x}), b_2(\mathbf{x})\}$. Thus, from this perspective, we can think of PVRTC as a special case of rank-2 moving basis decomposition with constrained coefficients.

We note that since MBD is a generalization of PVRTC, MBD has the capacity to represent piecewise smooth signals; that is, signals with discontinuities in space. However, our approximate solver may not necessarily find these solutions before converging to a local minima.

5.3. Relation to Radial Basis Functions and Shepard's Method

Similar to MBD, *radial basis function* (RBF) expansions are built on top of kernel functions that enable information sharing between kernels with overlapping support. The RBF expansion is typically defined in the data domain [Alf89], and thus, it doesn't take seamless reconstruction in the spatial domain into account. In contrast, Shepard's method [She68] allows one to build seamless, potentially high-dimensional, affine rank-0 reconstructions in the spatial domain. Compared to Shepard's method, MBD is suited for compression, due to the explicit decoupling of the spatial frequency of the coefficients $\mathbf{c}$ and the basis vectors $\mathbf{B}$. Furthermore, a rank-1 MBD can be thought of as the product of two Shepard expansions, c_l for the coefficients and b_l for the basis vectors. Finally, both RBF expansion and Shepard's method are instances of function approximation by way of interpolation, while MBD combines interpolation with basis decomposition to enable analysis-by-synthesis.

5.4. Relation to Clustering Methods

Blockwise PCA, as well as the solutions of K-means, K-SVD and clustered PCA — when extended to the spatial domain via the natural point-to-cluster mapping — can be thought of as special cases of MBD where the kernels are simply indicator functions for both the basis vectors and coefficients. In particular, these kernels are piecewise constant functions with non-overlapping support, highlighting the fact that a seamless reconstruction, in general, would require a post-process filtering step, i.e., an example of reconstruct-then-interpolate approach. In comparison, MBD considers interpolation as an essential part of the reconstruction, and takes the resulting interpolation error directly into account when solving for the decomposition.

5.5. Future work

We believe that MBD has potential applications in other problem domains with piecewise smooth structure, e.g., in the context of natural images, and the presented framework opens up some interesting questions for future work:

- **Signal-Adaptive Kernels**. While replacing a fixed grid of kernels with a scattered set of kernels is straightforward, a more interesting question considers the kernels themselves: is it feasible to jointly learn a set of adaptive kernels that are 1) efficient to evaluate and 2) efficient to store?
- **Robust and Adaptive Rank MBD**. Similar to PCA, the local low-rank assumption might be sensitive to outliers. Thus, splitting the input signal into a locally low-rank component and a sparse, high-rank component before computing MBD in addition to adaptively adjusting the local rank seems like a promising avenue to enable even higher compression ratios.
- **Manifold Learning**. What if we are not given position labels $\{\mathbf{x}_i\}$, i.e., the input consists only of the data vectors $\{\mathbf{y}_i\}$? Can we learn a low-dimensional embedding, say $z(\mathbf{y}) : R^D \to R^3$ such that $\{(z(\mathbf{y}_i), \mathbf{y}_i)\}$ has a low rank MBD in $z(\mathbf{y})$ coordinates?

6. Conclusion

Many existing dimensionality reduction and compression methods rely on independent, discrete choices in the data domain without considering the approximation error distribution in the spatial domain. On the other hand, scattered data interpolation methods provide seamless reconstruction in the spatial domain, but do not typically support analysis-by-synthesis. To jointly achieve analysis in the data domain and seamless synthesis in the spatial domain, we constructed a moving, locally adaptive basis decomposition using spatial kernels that transform many discrete decisions, such as choosing a basis for a cluster tangent space, into continuous ones. Furthermore, MBD enables analysis-by-synthesis while decoupling the spatial frequency of the coefficients and the basis vectors for compression tasks. We anticipate that future work will apply the MBD framework to other problem domains along with further investigation into adaptive kernels and robust and adaptive rank solvers.

Acknowledgements

The authors would like to thank Jaakko Lehtinen, Michal Iwanicki and Adrien Dubouchet for their valuable comments.

References

[AEB06] AHARON, M., ELAD, M., and BRUCKSTEIN, A. "K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". *Trans. Sig. Proc.* 54.11 (Nov. 2006), 4311–4322. ISSN: 1053-587X 2.

[Alf89] ALFELD, PETER. "Scattered Data Interpolation in Three or More Variables". *Mathematical Methods in Computer Aided Geometric Design*. Ed. by LYCHE, TOM and SCHUMAKER, LARRY L. Academic Press, 1989, 1–33. ISBN: 978-0-12-460515-2 10.

[Bar17] BARRÉ-BRISEBOIS, COLIN. "A Certain Slant of Light: Past, Present and Future Challenges of Global Illumination in Games". *Open Problems in Real-Time Rendering*. 2017. DOI: 10/ggfk67 1.

[BH89] BALDI, PIERRE and HORNIK, KURT. "Neural networks and principal component analysis: Learning from examples without local minima". *Neural networks* 2.1 (1989), 53–58 3.

[Bra03] BRAND, MATTHEW. "Charting a Manifold". *Advances in Neural Information Processing Systems*. Ed. by BECKER, S., THRUN, S., and OBERMAYER, K. Vol. 15. MIT Press, 2003 2.

[Cas85] CASSEREAU, PHILIPPE MICHAEL. "A New Class of Optimal Unitary Transforms for Image Processing". PhD thesis. Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, May 1985 3.

[Fen03] FENNEY, SIMON. "Texture Compression Using Low-Frequency Signal Modulation". *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*. HWWS '03. San Diego, California: Eurographics Association, 2003, 84–91. ISBN: 1581137397 10.

[GSHG98] GREGER, GENE, SHIRLEY, PETER, HUBBARD, PHILIP M., and GREENBERG, DONALD P. "The Irradiance Volume". 18.2 (March/April 1998). DOI: 10/ckjbb8 1.

[GV13] GOLUB, GENE H and VAN LOAN, CHARLES F. *Matrix computations*. Vol. 3. JHU press, 2013 2.

[KKSM17] KONIARIS, BABIS, KOSEK, MAGGIE, SINCLAIR, DAVID, and MITCHELL, KENNY. "Real-Time Rendering with Compressed Animated Light Fields". *Proceedings of the 43rd Graphics Interface Conference*. GI '17. Edmonton, Alberta, Canada: Canadian Human-Computer Communications Society, 2017, 33–40. ISBN: 9780994786821 10.

[KTHS06] KONTKANEN, JANNE, TURQUIN, EMMANUEL, HOLZSCHUCH, NICOLAS, and SILLION, FRANÇOIS X. "Wavelet Radiance Transport for Interactive Indirect Lighting". 2006. ISBN: 978-3-905673-35-7. DOI: 10/ggfk62 2.

[Llo82] LLOYD, S. "Least squares quantization in PCM". *IEEE Transactions on Information Theory* 28.2 (1982), 129–137. DOI: 10.1109/TIT.1982.1056489 2.

[LZT*08] LEHTINEN, JAAKKO, ZWICKER, MATTHIAS, TURQUIN, EMMANUEL, et al. "A Meshless Hierarchical Representation for Light Transport". 27.3 (Aug. 2008). ISSN: 07300301. DOI: 10/cbpkvx 2, 3.

[MRP98] MILLER, GAVIN, RUBIN, STEVEN, and PONCELEON, DULCE. "Lazy Decompression of Surface Light Fields for Precomputed Global Illumination." Jan. 1998, 281–292. ISBN: 978-3-211-83213-4. DOI: 10.1007/978-3-7091-6453-2_26 10.

[MSRB07] MAHAJAN, DHRUV, SHLIZERMAN, IRA KEMELMACHER, RAMAMOORTHI, RAVI, and BELHUMEUR, PETER. "A Theory of Locally Low Dimensional Light Transport". 26.3 (Aug. 2007). ISSN: 0730-0301. DOI: 10/bv7vzq 4.

[NLP*12] NYSTAD, J, LASSEN, A, POMIANOWSKI, A, et al. "Adaptive Scalable Texture Compression". *High-Performance Graphics 2012, HPG 2012 - ACM SIGGRAPH / Eurographics Symposium Proceedings* (Jan. 2012). DOI: 10.2312/EGGH/HPG12/105-114 10.

[NNJ05] NISHINO, KO, NAYAR, SHREE, and JEBARA, TONY. "Clustered blockwise PCA for representing visual data". *IEEE transactions on pattern analysis and machine intelligence* 27 (Nov. 2005), 1675–9. DOI: 10.1109/TPAMI.2005.193 2, 3, 5.

[NRH03] NG, REN, RAMAMOORTHI, RAVI, and HANRAHAN, PAT. "All-frequency shadows using non-linear wavelet lighting approximation". *ACM SIGGRAPH 2003 Papers*. 2003, 376–381 2.

[NW06] NOCEDAL, JORGE and WRIGHT, STEPHEN J. *Numerical Optimization*. second. New York, NY, USA: Springer, 2006 5.

[Owe13] OWEN, ART B. *Monte Carlo theory, methods and examples*. 2013 5.

[RS00] ROWEIS, SAM T and SAUL, LAWRENCE K. "Nonlinear dimensionality reduction by locally linear embedding". *science* 290.5500 (2000), 2323–2326 2.

[RSH02] ROWEIS, SAM, SAUL, LAWRENCE, and HINTON, GEOFFREY E. "Global Coordination of Local Linear Models". *Advances in Neural Information Processing Systems*. Ed. by DIETTERICH, T., BECKER, S., and GHAHRAMANI, Z. Vol. 14. MIT Press, 2002 2.

[RWG*13] REN, PEIRAN, WANG, JIAPING, GONG, MINMIN, et al. "Global Illumination with Radiance Regression Functions". *ACM Trans. Graph.* 32.4 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2461912.2462009 3.

[She68] SHEPARD, DONALD. "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data". *Proceedings of the 1968 23rd ACM National Conference*. ACM '68. New York, NY, USA: Association for Computing Machinery, 1968, 517–524. ISBN: 9781450374866. DOI: 10.1145/800186.810616 4, 10.

[SHHS03] SLOAN, PETER-PIKE, HALL, JESSE, HART, JOHN, and SNYDER, JOHN. "Clustered Principal Components for Precomputed Radiance Transfer". 22.3 (July 2003). ISSN: 07300301. DOI: 10/dbvt9z 2, 3, 5.

[SL17] SILVENNOINEN, ARI and LEHTINEN, JAAKKO. "Real-Time Global Illumination by Precomputed Local Reconstruction from Sparse Radiance Probes". 36.6 (Nov. 2017). ISSN: 0730-0301. DOI: 10/gcqbvn 3.

[SSS*20] SEYB, DARIO, SLOAN, PETER-PIKE, SILVENNOINEN, ARI, et al. "The design and evolution of the UberBake light baking system". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 39.4 (July 2020). DOI: 10/gg8xc9 1.

[ST15] SILVENNOINEN, ARI and TIMONEN, VILLE. "Multi-Scale Global Illumination in Quantum Break". *Advances in Real-Time Rendering in Games, Part I*. 2015. DOI: 10/gf3s6n 2, 3.

[TDL00] TENENBAUM, JOSHUA B, DE SILVA, VIN, and LANGFORD, JOHN C. "A global geometric framework for nonlinear dimensionality reduction". *science* 290.5500 (2000), 2319–2323 2.

[TR03] TEH, YEE W and ROWEIS, SAM T. "Automatic alignment of local representations". *Advances in neural information processing systems*. Citeseer. 2003, 865–872 2.

[VVK02] VERBEEK, JAKOB J., VLASSIS, NIKOS, and KRÖSE, BEN. "Coordinating Principal Component Analyzers". *Artificial Neural Networks — ICANN 2002*. Ed. by DORRONSORO, JOSÉ R. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, 914–919. ISBN: 978-3-540-46084-8 2.

[WZH07] WANG, RUI, ZHU, JIAJUN, and HUMPHREYS, GREG. "Precomputed radiance transfer for real-time indirect lighting using a spectral mesh basis". *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 2007, 13–21 2.

[XP04] XU, RUIFENG and PATTANAIK, SUMANTA N. "Real-time Rendering of Dynamic Objects in Dynamic, Low-frequency Lighting Environments". *Proc. of Computer Animation and Social Agents04, 2004*. 2004 10.