

A Theory of Locally Low Dimensional Light Transport

Dhruv Mahajan
Columbia University*

Ira Kemelmacher Shlizerman
Weizmann Institute of Science†

Ravi Ramamoorthi
Columbia University

Peter Belhumeur
Columbia University

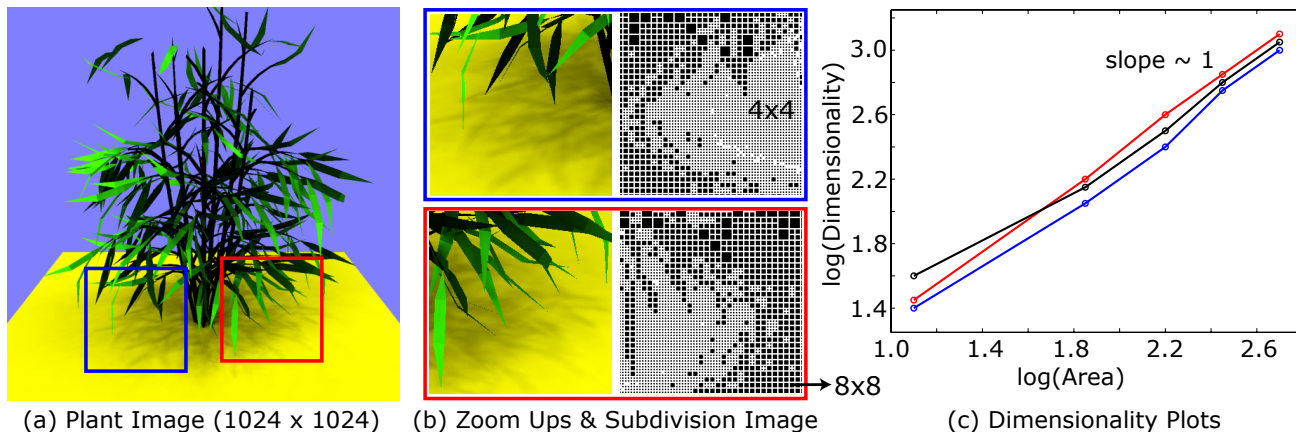


Figure 1: Complex lighting effects like soft shadows require transport matrices that have a very high rank or dimensionality. However, within local blocks, the dimensionality is much lower. This paper analyzes these effects theoretically. One practical application is to all-frequency relighting of high-resolution images; (a) shows a 1024×1024 image, lit by a $6 \times 32 \times 32$ cubemap, rendered interactively. We achieve this result by using our framework to adaptively subdivide the image into local patches. The adaptive subdivision, and zoom-ups to show the full resolution are in (b). Our main contribution is a theoretical analysis of how light transport dimensionality varies with patch size. For complex shadows like these, dimensionality is essentially linear in patch area (c).

Abstract

Blockwise or Clustered Principal Component Analysis (CPCA) is commonly used to achieve real-time rendering of shadows and glossy reflections with precomputed radiance transfer (PRT). The vertices or pixels are partitioned into smaller coherent regions, and light transport in each region is approximated by a *locally low-dimensional subspace* using PCA. Many earlier techniques such as surface light field and reflectance field compression use a similar paradigm. However, there has been no clear theoretical understanding of how light transport dimensionality increases with local patch size, nor of the optimal block size or number of clusters.

In this paper, we develop a theory of locally low dimensional light transport, by using Szego’s eigenvalue theorem to analytically derive the eigenvalues of the covariance matrix for canonical cases. We show mathematically that for symmetric patches of area A , the number of basis functions for glossy reflections increases linearly with A , while for simple cast shadows, it often increases as \sqrt{A} . These results are confirmed numerically on a number of test scenes. Next, we carry out an analysis of the cost of rendering, trading off local dimensionality and the number of patches, deriving an optimal block size. Based on this analysis, we provide useful practical insights for setting parameters in CPCA and also derive a new adaptive subdivision algorithm. Moreover, we show that rendering time scales sub-linearly with the resolution of the image, allowing for interactive all-frequency relighting of 1024×1024 images.

*e-mail: {dhruv,ravi,behumeur}@cs.columbia.edu

†e-mail: ira.kemelmacher@weizmann.ac.il

1 Introduction

Real-time rendering of light transport with glossy reflections, shadows and high frequency lighting is a challenging problem. In recent years, precomputed radiance transfer (PRT) [Sloan et al. 2002] has been widely adopted to achieve these goals. The essence of the approach¹, can be expressed simply as [Ng et al. 2003],

$$B = TL, \quad (1)$$

where B is a vector of outgoing intensities (at each image pixel or object vertex), L is a vector of lighting intensities from each direction, and T is a precomputed light transport matrix. For simplicity, most of this paper uses the canonical problem above of all-frequency relighting of images, with fixed view [Ng et al. 2003]. However, a factorization of the BRDF enables a generalization to changing both dynamic lighting and view [Liu et al. 2004; Wang et al. 2006], and we demonstrate results for this case in Fig. 11.

Note that L may be a 6×32^2 texel or higher resolution cubemap, and B may be a 512^2 pixel or higher-resolution image. Hence, T can have in excess of 10^9 elements, precluding real-time performance and storage. Modern PRT methods address this by clustering into smaller patches, in each of which the transport T is compressed using PCA. These methods are referred to as clustered PCA or CPCA [Sloan et al. 2003; Liu et al. 2004]. A faster to precompute method (but possibly suboptimal for rendering) is to simply divide an image into regular smaller blocks [Nayar et al. 2004].

While our main focus is on PRT, very similar ideas are also used in many other applications. For surface light fields, T represents variation with view, for fixed lighting, rather than vice versa. (Figure 10b shows an example of applying our method to surface light fields.) Previous work [Nishino et al. 2001; Chen et al. 2002] has used PCA on local triangular patches for compression. Block-based PCA for more general reflectance fields is employed by [Matusik et al. 2002]. Nishino et al. [2005] considers more general visual data as well, merging PCA blocks from finer levels. Locally low-rank approximations of sub-regions are used even for adaptive acquisition of reflectance fields [Garg et al. 2006]. Indeed, basic im-

¹Relighting, and equation 1, also has a long earlier history in graphics, including early work by [Nimeroff et al. 1994; Dorsey et al. 1995].

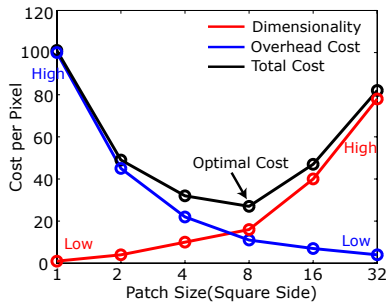


Figure 2: Schematic of how rendering costs vary with patch size. With increasing size, light transport dimensionality (red) increases, but the overhead cost (blue) decreases. The overall cost (black) is minimum where an increase in dimensionality balances the decrease in overhead cost.

age and video encoding with JPEG and MPEG also employ local blocks.

There is a key insight, underlying all of this large body of current practice—*light transport is locally low-dimensional*. Even if the global dimensionality of T is large, the light transport in each local patch is low-dimensional and can be compressed using simple PCA or SVD.² As shown schematically in Fig. 2 (red curve), transport dimensionality increases with patch area. Surprisingly, there has been no theoretical explanation of why this should be so, how exactly dimensionality varies with patch size, or depends on material properties and global effects (shadows, interreflections).

It may seem that an optimal patch size for rendering is one pixel, where light transport dimensionality is lowest. However, there is an overhead (blue curve in schematic of Fig. 2) in projecting to the local basis functions at each patch, that varies inversely with patch area. The total cost is a sum of these two effects, and finding the optimal cost/patch size involves a tradeoff between these competing factors (black curve in Fig. 2). While this is known qualitatively, there has been no theoretical analysis of what the optimal block size or number of clusters is. Indeed, most current methods either use square blocks of constant size [Matusik et al. 2002; Nayar et al. 2004] or arbitrarily fix the number of clusters/patches used [Sloan et al. 2003; Liu et al. 2004]. In this paper, we address these important theoretical questions, making the following contributions:

Theoretical Analysis of Dimensionality: One of our main contributions is a theory of how the dimensionality of T varies with patch size (red curve in Fig. 2).³ First (Sec. 2.1), we consider canonical cases in 2D. We assume shading behaves like a convolution, as has been shown for diffuse and glossy reflection [Bassi and Jacobs 2001; Ramamoorthi and Hanrahan 2004], and cast shadows [Soler and Sillion 1998; Ramamoorthi et al. 2005]. We derive an analytic formula for the eigenvalues of T based on Fourier analysis, and Szego’s eigenvalue theorem [Grenander and Szego 1958]. Specifically, we demonstrate that dimensionality in 2D increases approximately linearly with patch area A , for both reflections and shadows. We show that the key steps extend to 3D (Sec. 2.2), with similar results for convex specular surfaces, as well as complex shadows with multiple blockers as in Fig. 1. However, in the presence of simple shadows in 3D, the dimensionality for symmetric patches increases sublinearly as \sqrt{A} . We validate our theory (Sec. 3), showing numerical plots with a variety of scenes.

Theoretical Analysis of Rendering Cost: Based on our theory, we do a detailed analysis (Sec. 4) of how the overheads and total rendering cost vary with patch size (blue and black curves in

²We use the terms SVD and PCA interchangeably. While the mean is usually subtracted when applying PCA or computing a covariance matrix, we do not do so explicitly in this paper, for clarity and notational simplicity.

³This analysis significantly extends [Ramamoorthi 2002], who only considered the change of dimensionality for Lambertian objects when the front facing normals alone were visible. We consider general glossy reflection and cast shadows, as well as continuously varying patch areas.

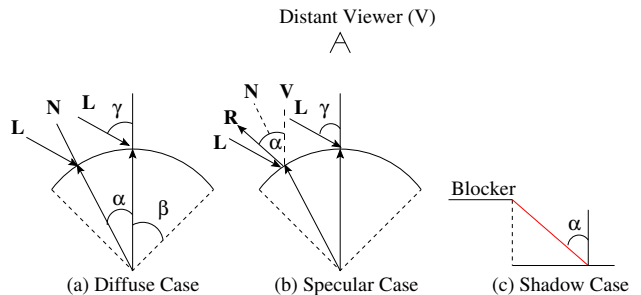


Figure 3: Schematic of canonical 2D setup for (a) Diffuse reflection, (b) Specular or Phong reflection, and (c) Cast Shadows.

Fig. 2). This enables us to derive an optimal patch size, where the increase in local dimensionality balances the decrease in overhead. We also show that the cost increases sublinearly as we increase the resolution of the image, enabling one to scale up to scenes with up to 1 million pixels at interactive rates (see Fig. 1).

Practical Applications: Our analysis provides many useful insights in setting parameters, such as block size or number of clusters in existing PRT methods like CPCA (Sec. 5, Fig. 10). We also present a practical algorithm which adaptively subdivides the object/image into a nearly optimal number of clusters (Fig. 11). We demonstrate results on complex scenes with shadows and glossy objects, where we can change lighting and/or viewpoint at real-time rates, as shown in our results in Figs. 1, 10 and 11.

2 Theoretical Analysis of Dimensionality

Starting with equation 1, assume the object/image is divided into local patches. Consider one such patch i , containing p_i pixels or vertices. Let T_i denote the transport matrix for this patch alone, with dimensions $p_i \times l$, where l is the lighting resolution. We seek to compress T_i using PCA, with $n_i \ll l$ eigenmodes,

$$T_i \approx U_i S_i V_i^T, \quad (2)$$

where U_i is a $p_i \times n_i$ matrix, S_i is a $n_i \times n_i$ diagonal matrix of eigenvalues and V_i is a $l \times n_i$ matrix, in the standard way.

We now conduct a theoretical analysis of how the dimensionality n_i of T_i (or the form of the eigenvalues in S_i) changes with patch size and other properties like the BRDF. For simplicity and clarity, we conduct most of our analysis in the 2D or flatland planar case (Sec. 2.1), deriving an analytic solution for common assumptions. Section 2.2 briefly outlines how the main steps extend to 3D.

2.1 Local Light Transport Analysis — 2D Case

Preliminaries: In 2D, first consider a simple convex arc, as shown in Fig. 3a. We parameterize lighting direction by γ (in the range $[-\pi, +\pi]$.) We parameterize surface location by angle α . For the diffuse case (Fig. 3a), α refers to the normal direction and for the specular case (Fig. 3b), it refers to the reflection of the view about the surface normal. The reflected radiance $B(\alpha)$ is

$$B(\alpha) = \int L(\gamma) f(\alpha - \gamma) d\gamma. \quad (3)$$

Here, we assume the BRDF $f(\cdot)$ is a symmetric function, depending only on the difference between the lighting direction γ and the angle α . This assumption is good for many common BRDFs like diffuse $f(\alpha - \gamma) = \max(\cos(\alpha - \gamma), 0)$ and the Phong BRDF, $f(\alpha - \gamma) = \cos^s(\alpha - \gamma)$. This formulation of the reflection equation is a convolution, as in the 2D setup of [Ramamoorthi and Hanrahan 2004]. Note that we assume the BRDF is homogeneous. This is a very common assumption made in most of the PRT methods, especially for rendering of synthetic objects. It is particularly applicable in our case, since we are considering local regions or patches. It has recently also been shown that even for acquired complex materials,

one can use a linear combination of a small number of basis homogeneous materials [Lawrence et al. 2006] which makes our analysis useful for these materials as well.

The same mathematical form also holds for cast shadows due to a single blocker [Ramamoorthi et al. 2005]. In this case, α is interpreted as the extremal angle induced by the blocker for a given spatial location (Fig. 3c), and f is a step function. Thus, equation 3 is quite general, and we therefore make it our starting point.

The light transport T is a function of spatial location (here α) and light direction (here γ), and is given from equation 3 simply by

$$T(\alpha, \gamma) = f(\alpha - \gamma). \quad (4)$$

Finally, the “image size” or p is simply proportional to the half-angle width β of the patch⁴, so that we can simply study how the dimensionality of $T(\alpha, \gamma)$ varies with β .

Covariance Function: Let $M(\alpha_1, \alpha_2)$ denote the covariance/correlation function for T in PCA. The dimensionality of the arc/patch is essentially the dimensionality of the function M .

$$\begin{aligned} M(\alpha_1, \alpha_2) &= \int_{-\pi}^{\pi} T(\alpha_1, \gamma)T(\alpha_2, \gamma)d\gamma \\ &= \int_{-\pi}^{\pi} f(\alpha_1 - \gamma)f(\gamma - \alpha_2)d\gamma, \end{aligned} \quad (5)$$

where we have made use of the fact that f is symmetric. By expanding in terms of Fourier series, or by substituting $u = \alpha_1 - \gamma$,

$$M(\alpha_1, \alpha_2) = \int_{-\pi}^{\pi} f(u)f(\alpha_1 - \alpha_2 - u)du = g(\alpha_1 - \alpha_2), \quad (6)$$

where g (and M) is a symmetric function, given by the convolution of f with itself, i.e., $g = f \otimes f$.

So far, we have worked in the continuous regime. Now, let us consider a discretization of M for p pixels. Then, the matrix element $M_{ij} = M(\alpha_i, \alpha_j) = g(\alpha_i, \alpha_j) = g(\alpha_i - \alpha_j)$. We can also discretize g so that $g_k = g(\alpha_k)$. In this case,

$$M_{ij} = g_{i-j}, \quad (7)$$

where g is still a symmetric function. Since M_{ij} now depends only on $|i - j|$, it is by definition a *symmetric Toeplitz matrix*.

Eigenvalues: Let $\lambda_r, 0 \leq r < p$ be the eigenvalues of discrete M^p (the superscript indicates a discretization to p pixels). For sufficiently large resolution, i.e., for large values of p , it can be shown using Szego’s eigenvalue distribution theorem [Grenander and Szego 1958] and *Lemma 4.6* in [Gray 2006] that the distribution of eigenvalues of M can be approximated⁵ by those of the truncated Discrete Fourier Transform (DFT) coefficients of g_k (with $I = \sqrt{-1}$),

$$\lambda_r \approx \sum_{k=-(p-1)}^{p-1} g_k e^{2\pi Ikr/p}. \quad (8)$$

In the limiting (continuous) case, as $p \rightarrow \infty$, $M^p \rightarrow M(\alpha_1, \alpha_2)$ and the dimensionality of the patch is the bandwidth⁶ of the truncated Fourier Transform of the continuous function g , now restricted in the range from $-\beta$ to β . Figure 4 shows plots of the DFT coefficients of the truncated g_k function (red curve) and eigenvalues of the Toeplitz matrix M^p (blue curve) computed using SVD for the Phong BRDF with Phong exponent $s = 50$ and two different values of β . The two curves match well, thus validating our analytic result.

⁴In practice, this relation can be somewhat non-linear for specular reflection and cast shadows, where α does not map linearly to spatial location. Our numerical tests do not indicate this is a major factor, except near the edges of objects, which we will discuss separately in Sec. 3.

⁵A technical condition is that g must be *absolutely summable*, or that $\sum_{k=-\infty}^{\infty} |g_k| < \infty$. Hence, we use $g^\pi(\omega)$, which is g when $-\pi \leq \omega \leq \pi$ and 0 otherwise. Since $g_k^\pi = g_k$ for $-p \leq k \leq p$, equation 8 still holds.

⁶In practice, we consider bandwidth to be where the magnitudes of the eigenvalues drop below some tolerance (usually 0.1% of total energy.)

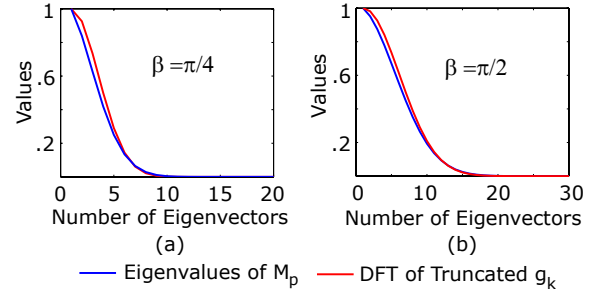


Figure 4: Plots of DFT coefficients of the truncated g_k function (red curve) and eigenvalues of the Toeplitz matrix M^p (blue curve) for a Phong BRDF with Phong exponent $s = 50$. Resolution $p = 3000$ is sufficiently high. Note that the curves match quite well, validating the theoretical analysis.

Dimensionality Analysis: We now have an analytic formula for the eigenvalues, if we know the function f (and hence g). However, we still need to understand how dimensionality varies as a function of patch size or area β . In other words, how does the Fourier transform of g change when restricted in $[-\beta, \beta]$?

To proceed, we first define $\beta' = \beta/\pi$ for simplicity, so that we have $0 \leq \beta' \leq 1$. Then, we define the restriction of g as

$$g(\alpha; \beta') = g(\alpha)h\left(\frac{\alpha}{\beta'}\right), \quad (9)$$

where $h(u)$ is a box function = 1 in $-1 \leq u \leq 1$ and 0 otherwise.

The Fourier transform, $G(\Omega_\alpha; \beta')$ is now given by

$$\begin{aligned} G(\Omega_\alpha; \beta') &= \int_{-\infty}^{\infty} g(\alpha)h\left(\frac{\alpha}{\beta'}\right)e^{2\pi I\left(\frac{\alpha}{\beta'}\right)\Omega_\alpha} d\alpha \\ &= \beta' \int_{-\infty}^{\infty} g(\beta'u)h(u)e^{2\pi Iu\Omega_\alpha} du. \end{aligned} \quad (10)$$

In the first line, the α/β' factor in the exponential is because the DFT effectively “stretches” the region from $[-\beta, +\beta]$ to $[-\pi, +\pi]$. In the second line, we make a simple substitution $u = \alpha/\beta'$.

To simplify further, we note that the Fourier transform of a product is a convolution in the frequency domain. Moreover, by the Fourier scale theorem, the Fourier transform of $g(\beta'u)$ is $(1/\beta')G(\Omega_\alpha/\beta')$. Putting all this together,

$$G(\Omega_\alpha; \beta') = G\left(\frac{\Omega_\alpha}{\beta'}\right) \otimes H(\Omega_\alpha). \quad (11)$$

This is a critical result of the paper, providing a simple but very general analytic formula for the eigenvalues of the covariance matrix, as a function of the patch area β' and the BRDF or shadowing function f (which determines G since $g = f \otimes f$).

Let $\omega_{G\beta'}$, ω_G and ω_H be the effective bandwidth of $G(\Omega_\alpha; \beta')$, $G(\Omega_\alpha)$ and $H(\Omega_\alpha)$ respectively. Since the bandwidth of a convolution of two functions in the frequency domain is the sum of their respective bandwidths, we have

$$\omega_{G\beta'} = \beta' \omega_G + \omega_H. \quad (12)$$

This is a central result of the paper, showing precisely how light transport dimensionality ω varies with patch area β' . The dimensionality varies linearly with β' , having a constant offset ω_H and a linear slope ω_G , that is of the order of the bandwidth of the BRDF or shadowing transfer function ($g(\alpha)$ is the convolution of the BRDF function $f(\alpha)$ with itself). Hence, dimensionality increases more rapidly with β' for highly specular materials or complex cast shadows, and more slowly for diffuse surfaces. This is intuitive since the dimensionality of light transport is of course greater for sharp reflections and shadows. Later in the paper, we will find it more convenient to consider loglog plots where the slope is always close to 1 for linear behavior, irrespective of the material properties, and the absolute linear slope ω_G represents only a constant offset.

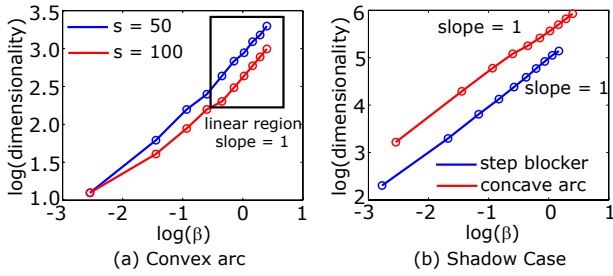


Figure 5: (a) shows the log-log plot of dimensionality vs size for a convex 2D arc with a Phong BRDF and two different Phong exponents s . The graphs are close to linear, with a slope of approximately 1 in most parts, but with a slight decrease early (for small sizes β). (b) shows the plots for the shadow case. The blue curve is for a step blocker, considering only visibility, and the red curve is for a concave arc with diffuse BRDF. Again the curves are linear with slope = 1. Since visibility has very high frequencies ω_G , the curves attain linear slope = 1 early, even for low β .

Note that as we increase β' , the $\beta'\omega_G$ term in equation 12 dominates so that

$$\omega_{G\beta'} \approx \beta'\omega_G \quad (13)$$

Hence, for sufficiently large values of β' , there exists a linear relationship between the dimensionality of the patch and the size of the patch. However, this relationship becomes sub-linear for small patch sizes (small β'), since the ω_H term cannot be neglected. We show later in Sec. 4.3 that the sublinearity for smaller patches plays a critical role in determining the optimal patch size.

Numerical Simulations: Figure 5 shows numerical simulations for a glossy reflector, as well as cast shadows, confirming the analysis above. We use loglog plots throughout this paper, so the slope is the power of dimensionality variation. This enables easy analysis of both linear (slope = 1) and sub-linear (slope < 1) behavior. In this domain, ω_G contributes only to a constant offset between the plots for different BRDFs, and hence the slope is independent of BRDF.

We obtain a nearly linear increase of dimensionality with patch size (with slightly sub-linear behavior for low β and glossy reflections). This behavior is consistent for most scenes, even though the actual value of dimensionality, and the transition point between linear and sub-linear regimes is data dependent. For example, in Fig. 5b, since visibility has very high frequencies (ω_G is large), the curves attain linear slope = 1 early, even for low β . However, the sub-linear effects are clearly visible in Fig. 5a for the Phong BRDF since ω_G is not that high.

Extensions to Changing View: Now consider the case where we fix the lighting and change the viewpoint. We seek the equivalent of the light transport matrix, which is simply the appearance of a surface point α from view direction γ . Assuming a parameterization by the reflected direction, we have,

$$T(\alpha, \gamma) = \int L(\omega) f(\alpha - \gamma - \omega) d\omega, \quad (14)$$

which is only a function of $\alpha - \gamma$, i.e. $T(\alpha, \gamma) = \tilde{f}(\alpha - \gamma)$, with $\tilde{f} = L \otimes f$. This has exactly the same form as equation 4. Hence, our framework is also immediately applicable to patch-based surface light field methods [Nishino et al. 2001; Chen et al. 2002]. A practical result for our method on surface light fields is in Fig. 10b.

2.2 Extension of Key Steps to 3D

We now extend our analysis to 3D, following the 2D framework. We will outline the extensions of the relevant steps, and provide numerical verification where appropriate. Section 3 shows numerical plots with a number of 3D examples, to validate our theory.

Preliminaries: Instead of a circular arc, in 3D we consider a patch on the sphere with $0 \leq \theta \leq \beta, 0 \leq \phi \leq 2\pi$. We parameterize using standard spherical coordinates $\alpha = (\theta, \phi)$. For the diffuse

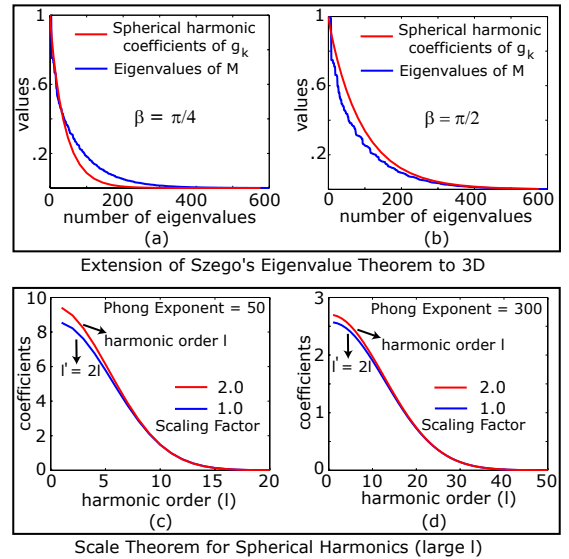


Figure 6: Top: Comparison of spherical harmonic coefficients of the function g_k (red) and the eigenvalues of covariance matrix M (blue) for $\beta = \pi/4$ and $\pi/2$. The two curves match well. (We show a Phong BRDF with exponent 100.) **Bottom:** Verification of the (approximate) scale theorem in spherical harmonics for large l (We show two different Phong exponents.) The spherical harmonic coefficient at G_l^1 for the red curve (with $g' = 2g(2\theta)$) matches G_{2l} for the blue curve (the original function $g(\theta)$).

case, α is the normal direction and for the specular case, it refers to the reflection direction, as in flatland. Let γ be the light direction. In 3D, the light transport function $T(\alpha, \gamma)$ is given by

$$T(\alpha, \gamma) = f(\alpha \cdot \gamma), \quad (15)$$

where $f(\cdot)$ is the radially symmetric BRDF. Equation 15 has a similar form as equation 4, with the difference between the angles in 2D replaced by the dot product or cosine of the angle in 3D.

Covariance Function: The covariance function $M(\alpha_1, \alpha_2)$ is

$$\begin{aligned} M(\alpha_1, \alpha_2) &= \int_{S^2} f(\alpha_1 \cdot \gamma) f(\gamma \cdot \alpha_2) d\gamma \\ &= g(\alpha_1 \cdot \alpha_2). \end{aligned} \quad (16)$$

The last line is a standard result and can be obtained, for instance, by expanding in spherical harmonics and applying the addition theorem. As in flatland, $g = f \otimes f$, and is radially symmetric.

If we discretize, $M_{ij} = g(\alpha_i, \alpha_j) = g(\alpha_i \cdot \alpha_j)$, which depends only on the angle between α_i and α_j . This closely resembles the Toeplitz matrix form in equation 7 and is the 3D analog.

Eigenvalues: Szego's eigenvalue theorem can be extended to the sphere [Okikioulu 1996], with the 2D Discrete Fourier Transform replaced by the Spherical Harmonic Transform. A recent application in vision is [Shirdhonkar and Jacobs 2005], although in a very different context (they seek to ensure positivity given low-frequency spherical harmonic coefficients). For us, we need simply consider the spherical harmonic transform of g , with $0 \leq \theta \leq \beta$ "stretched" out to the $[0, \pi]$ range. Figure 6a,b compares the eigenvalues of M (blue curve) with the spherical harmonic coefficients of g_k (red curve), for a Phong BRDF with exponent $s = 100$, with two values of $\beta = \frac{\pi}{4}$ and $\beta = \frac{\pi}{2}$. The two curves match well.

Dimensionality Analysis: Hence in 3D, the dimensionality of T for the patch can be approximated by the bandwidth of the Spherical Harmonic Transform⁷ $G_l(\Omega_\alpha; \beta')$ of $g(\alpha)$, $0 \leq \alpha \leq \beta$.

We can now conduct a similar analysis as in 2D, obtaining the 3D analog to equations 11 and 12. The only tricky issue is that the

⁷Since the function is radially symmetric with no azimuthal dependence, only the $m = 0$ term matters for each l .

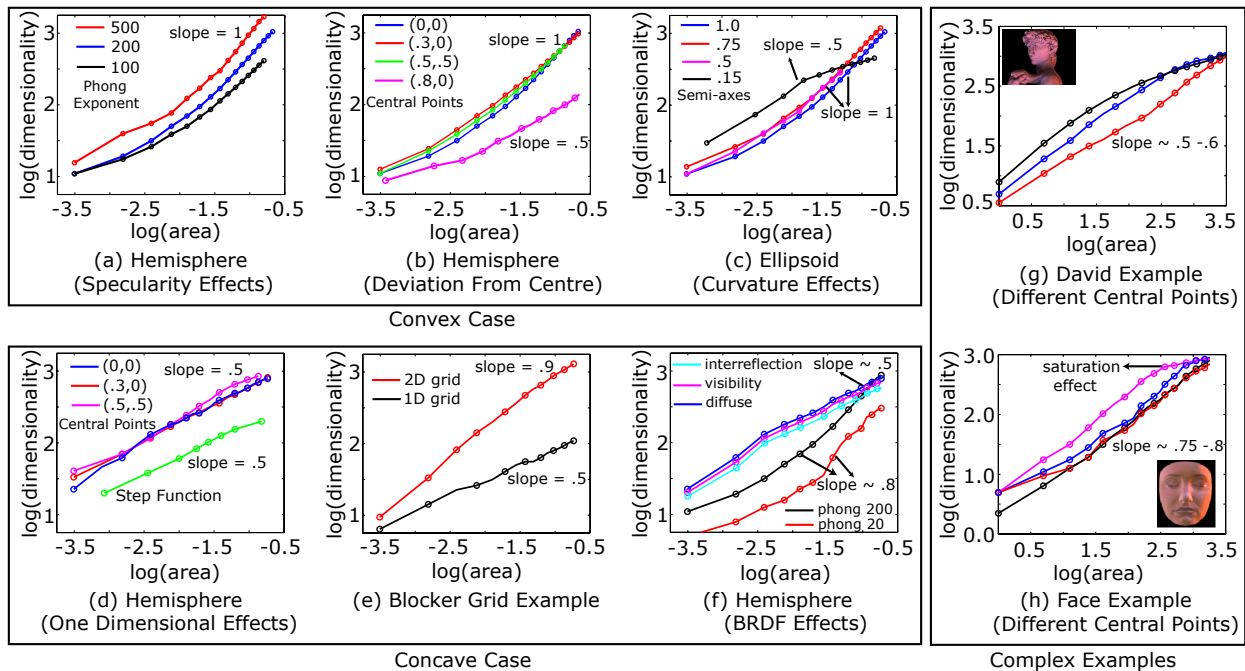


Figure 7: Plots showing validation of a variety of scene behaviors discussed in Sec. 3, including glossy reflections from convex objects, shadows in concave regions, and more complex examples. Dimensionality here means the number of eigenvalues constituting up to 99.95% of the energy.

2D derivation used the Fourier scale theorem, while there is no exact equivalent for spherical harmonics. However, a similar relation can be shown approximately for high-frequency Legendre polynomials, and is verified numerically in Fig. 6c,d (since $m = 0$, Y_{l0} are simple Legendre polynomials). Analytically, the asymptotic form $P_l \sim \cos(l\alpha + \Phi)$, where Φ does not depend on l . Therefore, for high frequencies, Legendre polynomials behave similarly in many respects to high-frequency Fourier modes, leading to the same conclusions in 3D as in equations 12 and 13 in 2D.

3 Experimental Validation and Discussion

In this section, we present several examples to validate our main result—that dimensionality of light transport varies almost linearly with patch size, especially for larger areas (and sub-linearly for smaller areas). We show examples with glossy materials, shadows and interreflections, and discuss several insights and implications.

Convex Glossy Surfaces: Figure 7a plots dimensionality vs size (area) A of the patch on a log-log scale for a convex hemisphere with Phong BRDF and different Phong exponents s . We consider circular patches centered at the center of the hemisphere. As expected, for large values of A , the plot is a straight line approaching slope 1, whereas for small A (left of graph), the curve flattens.

Effect of Specularity: The point at which the plot reaches slope near 1 is a function of Phong exponent s . As specularity s increases, the bandwidth ω_G in equation 12 increases and hence the curve approaches slope 1 earlier, for smaller patch sizes.

Deviation from Centre: Figure 7b shows these plots for some reasonable deviation of the patch center from the sphere center. Nearly all the plots show similar behavior approaching slope 1.

Curvature: In Fig. 7c, we consider ellipsoids with decreasing y and z semi-axes and hence decreasing curvatures. Again, most of the curves show similar behavior, approaching the slope 1.

Extreme Cases: Figures 7b and 7c respectively show one plot each for an extreme point near the edge, and one plot where the surface effectively becomes a cylinder. In these extreme cases, the slope is closer to 0.5 rather than 1, i.e., dimensionality is proportional to \sqrt{A} instead of A . Note that a point near the edge of the hemisphere also corresponds to the important case when a surface

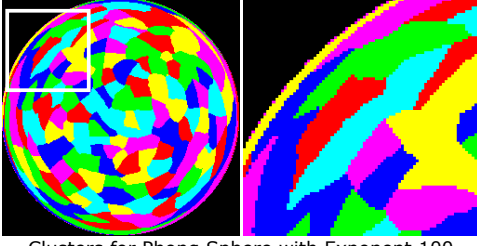
that is tilted away from the camera. These rare deviations are related to the fact that, although the patch is symmetric, variation along one dimension of the surface is much larger than the other, so dimensionality is proportional to patch radius (or \sqrt{A}) rather than area A . We will discuss this effect in more detail next, under cast shadows.

Concave Surfaces with Shadows: Figure 7d shows plots for a concave hemisphere with different centers for the patch. These are the plots of the dimensionality of the visibility function itself (without any BRDF). Note that, unlike for the convex case, there is almost no noticeable sub-linear region in these curves since the bandwidth ω_G for visibility in equation 12 is relatively large. Surprisingly however, the curve has a slope of 0.5 and not 1, so that dimensionality is proportional to \sqrt{A} , rather than A .

There is a simple explanation for this seeming discrepancy. A canonical example is where we have a long step along the Y axis (think of extruding Fig. 3c). The plot for this case is also shown in Fig. 7d, and is similar to the concave hemisphere. For all points lying on a line parallel to the Y axis, the visibility function is the same. We know from the 2D analysis (and numerical plots in flat-land) that transport dimensionality of shadows is proportional to the extent of the patch—but in this case, only extent along the X axis. Therefore, dimensionality is really proportional to the *length* or radius of the patch, and hence to square root of actual area or \sqrt{A} . The same explanation applies to extreme cases of glossy reflection.

Figure 7e shows a particularly interesting example, where we have a grid of blockers over a plane. We get the black curve by varying visibility along the X direction only (a 1D grid of blockers), and the red curve by varying it along both X and Y directions (2D grid). As expected, for the more common case of simple shadows, the black curve has a slope of 0.5. However for the rarer case of the 2D grid (red curve), the visibility function is quite intricate, involving both dimensions, and we get a slope closer to 1.0 ($\approx .9$)—a practical example of very complex shadowing is in Fig. 1.

Combining Glossy Reflections, Shadows and Interreflections: Figure 7f shows the plots for a concave hemisphere with diffuse and Phong BRDF and different Phong exponents s . Since the diffuse BRDF is low frequency, the visibility function dominates and slope



Clusters for Phong Sphere with Exponent 100

Figure 8: CPCA clusters for Phong sphere with exponent 100. Note how the clusters orient themselves along the boundary at the edges.

is 0.5. However, since the Phong BRDF is high frequency, we get a slope of approximately 0.8, between 0.5 and 1.0. We also show the effects of interreflections. Since these are a very low-frequency effect, they do not significantly affect our results.

Complex Examples: Figure 7g shows dimensionality plots for relighting the geometry of the David. Light transport is dominated by shadowing effects for complex geometry, and the slope is consistently 0.5 as expected. Figure 7h shows the dimensionality plots for the light transport of a real face. The face has a large diffuse component, as well as noticeable specularities and shadows (such as from the nose). Hence the slope is around 0.75 – 0.8 as expected.⁸ Similar results are obtained for a number of other scenes we tested, including on view-dependent examples using BRDF factorizations.

Asymmetric Patches: All of our plots use symmetric patches (as for example do the methods of [Matusik et al. 2002; Nayar et al. 2004]). It is also possible with CPCA to obtain asymmetric patches adapted to the light transport. In the canonical case of shadows from a step, the patch area would then increase only along the X axis, while having a fixed width along the Y axis. Another example is a simple Phong sphere in Fig. 8, where CPCA clusters are oriented along the sphere boundary near its edges. In these cases, dimensionality is proportional to A . More generally, for a variety of the effects considered, dimensionality would be proportional to $A^{1-\epsilon}$, where ϵ is small for large patch areas. (ϵ is never exactly 0, because of the convolution with the box function in equation 11).

4 Theoretical Analysis of Rendering Costs

We have studied how dimensionality varies with patch size (conceptually the red curve in Fig. 2). However, this is only one term in the rendering cost. We also need to consider the overhead of projecting onto the local basis functions at each patch, and the overall storage and rendering costs (the blue and black curves in Fig. 2).

4.1 General Framework

We first review the general PCA framework [Sloan et al. 2003]. Consider equation 2. During rendering, V_i^T first transforms l lighting directions to n_i local basis coefficients (this is done only once for the patch). U_i then transforms these local coefficients to p_i sample values (at each pixel separately). The total cost c_i is

$$c_i = n_i l + n_i p_i = n_i(l + p_i). \quad (17)$$

We have verified numerically that equation 17 corresponds closely to wall clock running time, and we therefore use it as a measure directly. Moreover, c_i is just the sum of the sizes of U_i [$n_i \times p_i$] and V_i [$l \times n_i$], and therefore also corresponds to storage size.

High-resolution lighting can easily involve $l = 6000$ or more directions. The size of l can be reduced to $N \ll l$, by a second SVD on

⁸An interesting observation on the face is the magenta (top) curve, with the center of the nose as central point. One can notice the saturation at the far end where the curve flattens. This is because the nose is close to spherical, and hence covers the full range of frontal normals. So, as we increase the patch size and move towards the cheeks, we are not adding any extra information. This is consistent with our theory, which describes only the local (not global) variation of light transport with dimensionality.

stacked versions of the matrix V_i^T , as shown by [Nayar et al. 2004]. An alternative is simply to do an SVD on a reduced representation of the full T directly, by choosing an appropriate (possibly random) subset of the pixels (rows). In particular, the global dimensionality of T , even for complex shadows and specular reflections, will rarely be more than $N = 500$ basis functions. The original lighting is projected into these global basis functions only once per frame, using a wavelet approximation [Ng et al. 2003].

Finally, we actually care about the total cost for the scene, which is the sum for all patches. Alternatively, it will be most useful to actually consider the rendering cost *per pixel*,

$$\frac{c_i}{p_i} = n_i \frac{N}{p_i} + n_i = n_i \left(\frac{N}{p_i} + 1 \right). \quad (18)$$

4.2 Cost-Efficient Method for Patch Subdivision

Equation 18 makes clear there are two opposing influences (the red and blue curves in Fig. 2). The dimensionality n_i increases with patch size p_i , and is optimal for very small patches (in the limit a single pixel). On the other hand, the overhead for converting to local bases decreases with p_i as per the N/p_i term—it is optimal for very large patches (in the limit a single patch for the full image).

Instead of using a fixed number of basis functions, and a fixed block size [Matusik et al. 2002; Nayar et al. 2004] or number of clusters [Sloan et al. 2003], we represent each patch accurately to a desired tolerance, and adaptively or hierarchically subdivide the image (similar in some respects to a kd-tree or quadtree—see Figs. 1b and 11c for examples) to find a near-optimal patch size.

We first derive a cost metric for dividing a patch into a number of sub-patches. Since we deal with a single patch, we omit the subscript i . From equation 17, the cost c is

$$c = nN + np = n(p + N). \quad (19)$$

Assume we divide the patch into r subpatches of equal size. Let n_j denote the number of bases required for the j^{th} subpatch. From equation 19, the cost c' for rendering the subpatches is given by

$$\begin{aligned} c' &= \sum_{j=1}^r n_j \left(\frac{p}{r} + N \right) \\ &= \frac{\sum_{j=1}^r n_j}{r} p + \sum_{j=1}^r n_j N. \end{aligned} \quad (20)$$

Here $\frac{\sum_{j=1}^r n_j}{r}$ is the average number of bases required to represent each subpatch. Let $\frac{\sum_{j=1}^r n_j}{r} = \mu n$,

$$c' = \mu np + r\mu nN. \quad (21)$$

The difference d in the two costs c and c' is

$$d = c - c' = np + nN - (\mu np + r\mu nN) \quad (22)$$

$$= (1 - \mu)np + (1 - r\mu)nN. \quad (23)$$

d acts as a cost metric for subdivision. If $d > 0$, the cost decreases after subdivision and we should subdivide. Otherwise, we have found a good patch size, and cannot easily improve the cost further.

4.3 Analysis of Cost Metric and Implications

First, assume the patch size is reasonably large, so that we are in the range where n varies linearly with p (i.e., equation 13 is a good approximation to equation 12), as shown in Sec. 3. Assuming a convex patch with no shadows, from equation 13, $\mu \approx \frac{1}{r}$. The first term (per-pixel cost) in equation 21 decreases sharply by a factor μ , since smaller patches mean fewer local basis functions. Moreover, the second term $r\mu nN$ in equation 21 remains constant at nN . In other words, although the number of patches has increased, the number

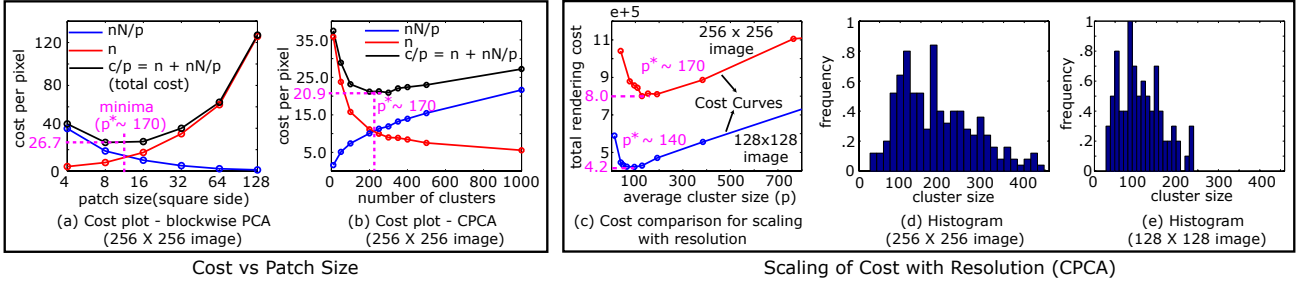


Figure 9: Left (a,b): Showing how cost varies with patch size for images of a 256×256 face with different lighting. The blue curve shows the global basis to local basis conversion cost and decreases as patch size increases. The red curve shows the local basis to pixel conversion cost and increases as patch size increases. The global basis count N is 150. (a) shows the plots for blockwise PCA whereas (b) shows them for CPCA. In both cases, minima occur around $p^* \approx 150$ as predicted by our analysis. Right (c,d,e): How minimum cost changes with resolution of the image. (c) shows the total cost curves for 128×128 and 256×256 face images using CPCA. Note that minimum cost increases only by a sublinear factor of 1.9 as predicted by our analysis. Also the position of minima (in terms of pixel size) essentially remains the same. The distribution of clusters is shown in (d) and (e).

of local bases per patch has proportionately decreased. Considering the cost differential in equation 23, the term $(1 - r\mu)nN \approx 0$ and

$$d \approx (1 - \frac{1}{r})np > 0. \quad (24)$$

Therefore, we should always subdivide the patch if we are in the linear regime, down to a very small size (for an example, see Fig. 1).

Now, consider the opposite case, when we are in the flat region of the dimensionality curve i.e. patch size is very small (so that the constant factor ω_H dominates in equation 12). μ is close to 1, so that the per-pixel cost does not decrease significantly (remains at np), while the local basis conversion cost increases by approximately a factor of r . In terms of d , $(1 - \mu)np \approx 0$, and

$$d \approx (1 - r)nN < 0. \quad (25)$$

So, we should not subdivide the patch further.

In general, the patch may be between linear and flat regions, and/or have parts with shadows. From our analysis, the slope of dimensionality is usually between 0.5 and 1.0. Hence, μ often lies between $\frac{1}{r}$ and 1.0. Assuming constant μ , there exists an optimal patch size p^* , which we can determine by setting $d = 0$. At p^* , the decrease in per-pixel cost when subdividing exactly balances the increase in cost from more patches and more local basis functions.

$$(1 - \mu)np^* + (1 - r\mu)nN = 0 \quad (26)$$

$$p^* = N \frac{r\mu - 1}{1 - \mu}. \quad (27)$$

An important insight is that p^* is proportional to N , i.e., the optimal number of pixels in a patch is roughly the same as the number of global basis functions N . We should subdivide less (larger patches) if the global transport complexity N is large, while simpler scenes (like low-frequency convex objects) should be subdivided more.

This may appear counterintuitive, since it would seem that for low-frequency objects, the global dimensionality is already low and further subdivision will provide limited benefits. However, the overheads for additional subdivision are minimal precisely because the total number of basis functions required is small—on the other hand, for high-frequency materials or shadows, these overhead costs soon become significant.

Evaluation: Figure 9a shows the cost vs size plot for a 256×256 face image, lit from a $6 \times 32 \times 32$, cubemap, with global dimensionality $N = 150$. The image is divided into square blocks of different sizes and cost computed for each size. As expected, the minimum cost is for $p \sim N$, and lies between 8×8 ($p = 64$) and 16×16 ($p = 256$) patches. Somewhat better results are obtained using CPCA [Sloan et al. 2003], which does not place any constraint on the structure of clusters. Figure 9b shows the plot for the face dataset created using different numbers of CPCA clusters. Note that cost is minimum for approximately 220 clusters. Since

the face has 38,000 occupied pixels, this corresponds on average to $p \approx 170$ which is again of the order of N ($\mu \approx 0.68$ and $r = 2$ here).

4.4 Scaling of Cost with Resolution

We now analyze how cost varies if we increase the resolution of an image/mesh. Assume we double the number of pixels p in the patch and that the dimensionality n does not change. From the previous section, since the optimal patch size p^* is dependent only on μ and global basis N , we can subdivide the patch into two smaller patches. Hence increasing resolution allows further subdivision.

In each of the two (equally sized) smaller patches, $n \rightarrow \mu n$, but p and N remain the same. Hence, new cost c_1 after subdividing is,

$$c_1 = 2\mu n(p + N), \quad (28)$$

where the factor of 2 is because we now have two sub-patches. Note that if we are in the linear regime, with $\mu \approx 1/2$, the cost remains the same, even though we have increased resolution. This makes sense, because even though there are now more pixels, the information content is the same. More generally, comparing c_1 and c ,

$$\frac{c_1}{c} = 2\mu. \quad (29)$$

Since $\mu < 1$, we always get a sub-linear increase in the cost. For example, for $\mu = .7$ the cost increases by only 1.4 and not 2. Equivalently, the per-pixel rendering cost decreases by a factor of μ .

Figure 9c shows the cost vs. size plot for the face dataset at 128×128 and 256×256 resolutions. We estimate $\mu \approx 0.68$ by taking a few patches. Since we have increased the resolution by a factor of 4 or 2×2 , our analysis predicts an increase in the total cost by a factor of $2\mu \cdot 2\mu = 4\mu^2 = 1.85$. Equivalently, the per-pixel cost should decrease by a factor of $\mu \cdot \mu = \mu^2 \approx 0.5$. The minimum cost for each size is shown in the plot. Note that the ratio of the total costs comes out to be 1.9 which is consistent with our theoretical estimate (and the per pixel cost decreases to approximately one half, also consistent with our estimate). The optimal average patch size p^* for both the resolutions come out to be approximately the same. Figure 9d,e further show the histograms of the size of the clusters. Note that both show a similar distribution (but the tail of the distribution is reduced for the smaller 128×128 resolution.)

5 Practical Applications

In this section, we discuss some practical applications of our analysis. First, we show how to set the right parameters in existing PRT methods, without any other modification. We then present a simple algorithm that adaptively subdivides the object/image into a nearly optimal number of clusters. Finally, we scale up our resolution to show all-frequency relighting of 1024×1024 images.

5.1 Setting Parameters in Existing Methods

In Sec. 4.3, we derived a relation between optimal patch size p^* and global basis count N (equation 27). Once we know μ and N

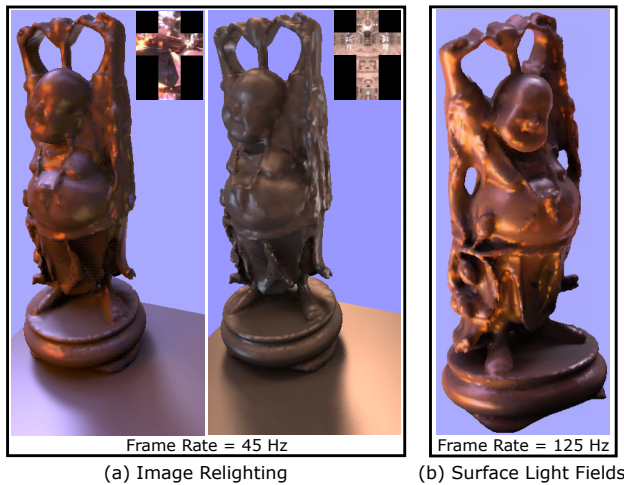


Figure 10: (a) Using our analysis to set parameters in Clustered PCA (for relighting with fixed view) for the Buddha dataset (24,000 vertices, optimal number m of clusters = 220). (b) Application of our adaptive subdivision algorithm to render with changing viewpoint for surface light fields.

for the object/image, we should be able to set the optimal patch size in existing methods like blockwise PCA [Nayar et al. 2004] or CPCA [Sloan et al. 2003], and use them without any modification.

This requires us to estimate μ and N . For any given patch, with a given central point, we determine μ based on the average slope s or growth rate of dimensionality with the number of pixels/vertices. We then use $\mu = r^{-s}$, with $r = 2$ for CPCA and $r = 4$ for block PCA (corresponding to subdivision into 2 clusters or 4 sub-blocks).⁹ Finally, to determine N , we randomly choose a large number of sample points and do an SVD decomposition of the resulting transport matrix (we could use the full image/mesh, but this is too computationally expensive). We choose N as the number of eigenvectors needed to capture most of the total energy (usually 99.9%).

Figure 10a shows the Buddha (24,000 vertices), with complex shadows using CPCA relighting for fixed view, with illumination from $6 \times 32 \times 32$ cube maps. For the estimated $\mu = .57$ (220 clusters), the optimal per-vertex cost is 114.78. (Note that the unit of costs in this section will be the effective number of basis functions required at each vertex—corresponding to the average value of c/p in equation 18.) A range around the optimal μ of [.53, .6] corresponds to 130 – 600 clusters, where the cost is between 114.78 – 130, so our method is not very sensitive to the initial points chosen to estimate μ . However, if we move far outside this range such as 11 clusters, the cost is 310.7 which is not even below the global number of basis functions, and nearly 3 times the cost of our method. Prior to our work, there was no simple way to determine the correct number of clusters (trial and error was effectively the only possibility, but the precomputations take a long time to run, often making that approach infeasible).

5.2 Practical Adaptive Algorithm

In practice, μ may vary non-uniformly along the scene, and we now present a simple algorithm that adaptively and hierarchically subdivides the object/image. We initially run the standard CPCA algorithm with a few (usually 5-10) clusters. We then take each initial cluster, and subdivide it into two using CPCA (see Fig. 11c), comparing the cost of rendering the cluster before and after subdivision. The subdivision stops if the cost increases after subdivision. Otherwise, the process is repeated for each newly formed cluster.

Since the lighting dimensions are large ($6 \times 32 \times 32$), and initially when we have only a few clusters, the cluster size is also large,

⁹In standard CPCA or block PCA, there is no actual adaptive subdivision, or notion of r . However, equation 27 is relatively insensitive to r , and we therefore use values corresponding to our adaptive technique in Sec. 5.2.

PCA becomes computationally difficult. Therefore, for the initial few subdivisions, we do a wavelet decomposition of light transport for each vertex and take the first K wavelet coefficients with highest energy globally.¹⁰ After a few subdivision steps, when cluster size becomes manageable, we shift to the full original light transport. Our algorithm can be applied to changing view as well [Liu et al. 2004], separately considering each view-dependent BRDF factor.

Figure 11 shows the results of our algorithm on a complex scene with specular objects (street lamps). The scene has approximately 40,000 vertices, and we can change lighting and view in real-time. For the ground plane, with adaptive subdivision, the cost (per-vertex) is 64. In this case, the number of global basis functions is $N = 500$, so we provide a speedup of nearly an order of magnitude.

The number of clusters generated are 219. Even if we run standard CPCA with the same fixed (219) number of clusters, the per-vertex cost is 76, which is around 20% more than our adaptive algorithm. This is because running standard CPCA must use a wavelet approximation throughout. More importantly, since the clustering is being done globally for the full scene (instead of reclustering each individual cluster hierarchically), it is more prone to local optima. Again, if we move to extremes such as 10 clusters, the cost increases to 197, which is more than 3 times the cost with our algorithm. At the other extreme, if we make the number of clusters very large (around 5000) the cost becomes 184, which is again nearly thrice the cost of our method. In the intermediate range (50 clusters), the cost is 112 which is around twice the optimum.

Application to Surface Light Fields: While our main focus is PRT, we can also use the adaptive subdivision algorithm to render surface light fields with dynamic view. Instead of the relighting transport matrix, we have the intensity of each vertex from all views (on a $6 \times 32 \times 32$ cube map.) Figure 10b shows a surface light field of the Buddha, rendered at a cost of 31.8 bases per vertex (compare with $N = 300$ global basis functions). As with relighting, the cost is $3 \times -4 \times$ less than CPCA with a sub-optimal number of clusters.

5.3 Scaling to Large Resolutions

Our theory indicates we can increase resolution with only a sub-linear increase in cost. Fig. 1a demonstrates this practically, showing interactive rendering at a 1024×1024 resolution. We use blockwise PCA (similar to [Nayar et al. 2004]), with adaptive subdivision using our method (Fig. 1b). Because the shadows are very complex, the global dimensionality $N = 700$ is large, and the variation of dimensionality with patch size is nearly always linear in Fig. 1c. Hence, as predicted by our cost analysis, we subdivide down to very small patches with areas of 4×4 or 8×8 , which is much less than N . Note that the subdivision becomes finer (Fig. 1b) as we move closer to the plant on the ground plane, since the shadows become more complex. Overall, our adaptive block PCA provides a speedup of approximately an order of magnitude, enabling interactive relighting of this very high-resolution image.

6 Conclusions and Future Work

We have developed a theoretical analysis of the dimensionality of local light transport. First, we show how the dimensionality of a patch changes with its size, observing that dimensionality is proportional to area for glossy reflections, and proportional to length or radius for shadows. Second, we analyze the rendering cost, and derive the optimal patch size. In practical applications, this analysis allows us to fine tune the parameters of existing methods, scale to very large resolutions, and develop adaptive clustering algorithms.

In the future, we would also like to analyze other global illumination effects such as sub-surface scattering or caustics. Another important theoretical question is how this analysis relates to the local frequency-space analysis in [Durand et al. 2005], and whether

¹⁰This is used only for clustering. The full light transport is always used for computing the final bases needed for rendering.

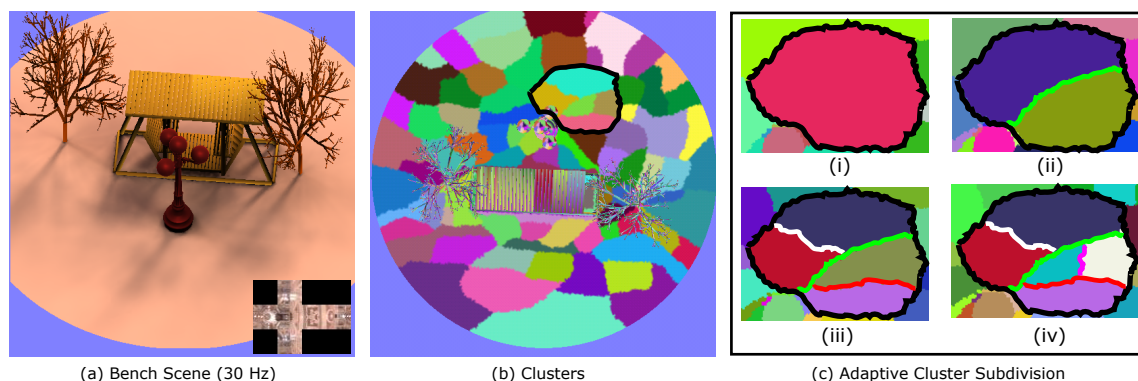


Figure 11: Adaptive algorithm with CPCA for a scene with 40,000 vertices. (a) shows the rendering of the scene with dynamic lighting and view, (b) shows the final clustering of the scene, and (c) shows the steps of our adaptive subdivision algorithm for a small patch on the ground plane—notice the subdivisions into two in steps (i), (ii) and (iii), but how most of the patches do not subdivide further in step (iv).

they can be unified. Moreover, clustering techniques and PCA are also widely used to represent visual appearance and other quantities in different application areas like computer vision.

While we have presented our theory in the context of PRT and rendering, the paper describes a fundamental analysis of light transport that has potential applications in many other domains. For example, locally low-dimensional subspaces provide a robust and efficient computational framework for inverse rendering problems like illumination estimation. In computer vision, they allow us to understand and recover the effects of lighting in the context of applications like lighting-insensitive face recognition (indeed, our numerical validations have included tests on real faces).

With the growing application and maturity of algorithms based on local patches and clustering, we believe that a fundamental theoretical analysis, as in this paper, is critical to explain and make further progress. In the future, we expect a more solid foundation for a variety of methods that compress and represent high-dimensional visual appearance compactly.

Acknowledgements: We thank the anonymous reviewers for their feedback and comments to improve clarity. This research was supported in part by an ONR Young Investigator Award N00014-07-1-0900 (Mathematical Models of Illumination and Reflectance for Image Understanding and Machine Vision), as well as NSF grants # 0430258, #0325867, #0305322 and #0446916, and a Sloan research fellowship.

References

BASRI, R., AND JACOBS, D. 2001. Lambertian reflectance and linear subspaces. In *International Conference on Computer Vision*, 383–390.

CHEN, W., BOUGUET, J., CHU, M., AND GRZESZCZUK, R. 2002. Light field mapping: Efficient representation and hardware rendering of surface light fields. *ACM Transactions on Graphics (SIGGRAPH 2002)* 21, 3, 447–456.

DORSEY, J., ARVO, J., AND GREENBERG, D. 1995. Interactive design of complex time-dependent lighting. *IEEE Computer Graphics and Applications* 15, 2, 26–36.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. 2005. A frequency analysis of light transport. *ACM Transactions on Graphics (SIGGRAPH 2005)* 25, 3, 1115–1126.

GARG, G., TALVALA, E., LEVOY, M., AND LENSCH, H. 2006. Symmetric photography: Exploiting data-sparseness in reflectance fields. In *EuroGraphics Symposium on Rendering*, 251–262.

GRAY, R. 2006. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory* 2, 3, 155–239.

GRENANDER, U., AND SZEGO, G. 1958. *Toeplitz Forms and Their Applications*. University of Calif. Press, Berkeley and Los Angeles.

LAWRENCE, J., BEN-ARTZI, A., DECORO, C., MATUSIK, W., PFISTER, H., RAMAMOORTHY, R., AND RUSINKIEWICZ, S. 2006. Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics (SIGGRAPH 2006)* 25, 3 (jul).

LIU, X., SLOAN, P., SHUM, H., AND SNYDER, J. 2004. All-frequency precomputed radiance transfer for glossy objects. *EuroGraphics Symposium on Rendering 2004*, 337–344.

MATUSIK, W., PFISTER, H., NGAN, A., BEARDSLEY, P., ZIEGLER, R., AND MCMILLAN, L. 2002. Image-based 3D photography using opacity hulls. *ACM Transactions on Graphics (SIGGRAPH 2002)* 21, 3, 427–437.

NAYAR, S., BELHUMEUR, P., AND BOULT, T. 2004. Lighting-sensitive displays. *ACM Transactions on Graphics* 23, 4, 963–979.

NG, R., RAMAMOORTHY, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics (SIGGRAPH 2003)* 22, 3, 376–381.

NIMEROFF, J., SIMONCELLI, E., AND DORSEY, J. 1994. Efficient re-rendering of naturally illuminated environments. In *EuroGraphics Workshop on Rendering*, 359–373.

NISHINO, K., SATO, Y., AND IKEUCHI, K. 2001. Eigen-texture method: Appearance compression and synthesis based on a 3d model. *IEEE PAMI* 23, 11, 1257–1265.

NISHINO, K., NAYAR, S., AND JEBARA, T. 2005. Clustered blockwise pca for representing visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 10, 1675–1679.

OKIKIOULU, K. 1996. The analogue of the strong Szego limit theorem on the 2- and 3- dimensional spheres. *Journal of the American Mathematical Society* 9, 2, 345–372.

RAMAMOORTHY, R., AND HANRAHAN, P. 2004. A signal processing framework for reflection. *ACM TOG* 23, 4, 1004–1042.

RAMAMOORTHY, R., KOUDELKA, M., AND BELHUMEUR, P. 2005. A fourier theory for cast shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 2, 288–295.

RAMAMOORTHY, R. 2002. Analytic PCA construction for theoretical analysis of lighting variability in images of a lambertian object. *IEEE PAMI* 24, 10, 1322–1333.

SHIRDHONKAR, S., AND JACOBS, D. 2005. Non-negative lighting and specular object recognition. In *ICCV 05*, 1323–1330.

SLOAN, P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics (SIGGRAPH 2002)* 21, 3, 527–536.

- SLOAN, P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics (SIGGRAPH 2003)* 22, 3, 382–391.
- SOLER, C., AND SILLION, F. 1998. Fast calculation of soft shadow textures using convolution. In *SIGGRAPH 98*, 321–332.
- WANG, R., TRAN, J., AND LUEBKE, D. 2006. All-frequency relighting of glossy objects. *ACM Transactions on Graphics* 25, 2, 293–318.