4D Compression and Relighting with High-Resolution Light Transport Matrices

Ewen Cheslack-Postava, Stanford University
Nolan Goodnight, NVIDIA Corporation
Ren Ng, Refocus Imaging
Ravi Ramamoorthi, Columbia University
Greg Humphreys, University of Virginia

Abstract

This paper presents a method for efficient compression and relighting with high-resolution, precomputed light transport matrices. We accomplish this using a 4D wavelet transform, transforming the columns of the transport matrix, in addition to the 2D row transform used in previous work. We show that a standard 4D wavelet transform can actually inflate portions of the matrix, because high-frequency lights lead to high-frequency images that cannot easily be compressed. Therefore, we present an adaptive 4D wavelet transform that terminates at a level that avoids inflation and maximizes sparsity in the matrix data. Finally, we present an algorithm for fast relighting from adaptively compressed transport matrices. Combined with a GPU-based precomputation pipeline, this results in an image and geometry relighting system that performs significantly better than 2D compression techniques, on average 2x-3x better in terms of storage cost and rendering speed for equal quality matrices.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; G.1.2 [Numerical Analysis]: Approximation—Wavelets and Fractals, Nonlinear Approximation

Keywords: Relighting, Pre-computed Radiance Transfer, Nonlinear Approximation, Wavelet Compression

1 Introduction

Many realistic image synthesis techniques require the evaluation of complex, multi-dimensional integrals, and are therefore impractical for real-time rendering. Recent research has bridged the gap between image quality and rendering speed by precomputing light transport effects. These techniques precompute functions in the light transport integral, and reuse this data to render a scene under dynamic viewpoint and illumination. Using such precomputation methods, it is possible to render high-quality images in real-time, but there is the burden of significant storage and memory costs. In most cases, approximation techniques are used to compress the light transport data and reduce storage requirements.

The earliest systems offered only low frequency lighting effects, fixed viewpoint or diffuse surfaces, and static scenes. Different approaches have removed various combinations of these limitations. Capturing high-frequency lighting effects requires high-resolution sampling and consequently more basis function coefficients for accurate approximation. However, these high-resolution transport matrices require significant storage space, which can easily exceed hundreds of megabytes. As scene complexity grows, the space necessary to store precomputed results can become prohibitively large. This paper addresses the problem of reducing the storage required for precomputing high-resolution transport matrices.

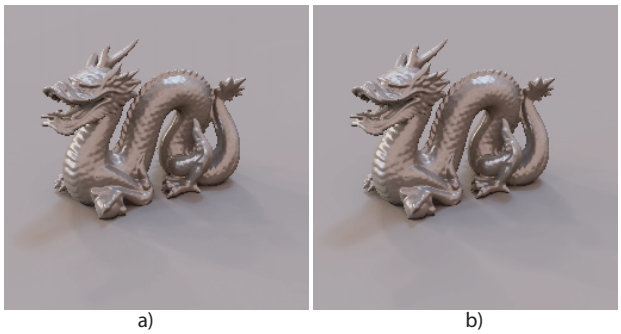

Figure 1: Comparison of relighting from 2D and 4D compressed matrices. The image is relit under illumination from St. Peter's Basilica, **a)** shows the result from a 2D compressed matrix and **b)** the result from an 4D compressed matrix. The results are nearly indistinguishable, but the 4D matrix is one third the size of the 2D matrix.

We present a 4D wavelet decomposition of the light transport matrix that consists of the following: a 2D transform of the light samples (matrix rows), followed by a 2D transform of the image samples (matrix columns). Using this representation, we identify and eliminate coherence in the matrix, which corresponds to unnecessarily high sampling of image and lighting. This is a seemingly natural extension to the 2D wavelet decomposition technique of Ng et al. [2003] in that they compress the matrix by making the rows sparse. However, we found that a naive 4D transform, while designed to create additional sparsity in the matrix columns, can actually inflate the size of the matrix.

To correct for this inflation, we modify the 4D wavelet transform to better account for existing sparsity in the matrix. We do this by transforming the columns adaptively, meaning we terminate the wavelet transform once we reach a level where there is no more coherence in the downsampled image. In this way, we compress the matrix using a minimum set of wavelet basis functions, which is effectively the same as minimizing the light and image sampling rates using non-uniform sampling. We then present a novel algorithm for relighting from adaptively compressed light transport matrices, along with a complete system capable of achieving 2x-3x greater compression and rendering speed compared to 2D techniques. Finally, we present results from our system and discuss future directions for research in light transport matrix compression.

2 Related Work

Precomputed radiance transfer was introduced by Kautz et al. [2002] and Sloan et al. [2002] for dynamic low frequency environments. These systems were very limited. They used either a fixed viewpoint or a diffuse BRDF and a low-dimensional approximation using spherical harmonics, which restricted them low frequency lighting effects in order to keep the transport matrix a managable size. Furthermore, some techniques could only handle static scenes. Since the introduction of PRT, research has mainly focused on expanding the feature sets of the system, sometimes using new compression schemes to achieve this goal.

One branch of improvements focused on adding all-frequency lighting and view dependent effects. Ng et al. [2003] used the Haar wavelet basis to allow for all-frequency shadows, but were still limited to either static scenes or a diffuse BRDF. Liu et al. [2004] and Wang et al. [2004] concurrently extended PRT to allow for glossy materials by factoring the BRDF into view and light direction dependent terms. Wang et al. [2005] introduce a method for precomputing transport vectors for translucent objects, including single and multiple scattering. Ng et al. [2004] introduced the wavelet triple product integral which allows the separation of the BRDF and visibility terms. Their system can relight a scene under varying all-frequency lighting and viewing conditions in a few seconds, but is still limited to static scenes.

Another branch of techniques focused on adding local lighting and dynamic scenes, accepting low-frequency and view independent lighting effects. Sloan et al. [2002] use a neighborhood transfer technique to generate soft shadows from dynamic objects. Sloan et al. [2005] use zonal harmonics to quickly rotate transfer and produce dynamic local shading effects. Kautz et al. [2004] use approximate hemispherical rasterization to render soft shadows with explicit occlusion information. Kontkanen and Laine [2005] precompute ambient occlusion fields to render soft cast shadows in real-time. Ren et al. [2006] model geometry as a set of spheres and use spherical harmonic exponentiation to calculate visibility. Yet another class of techniques computes the direct lighting on a set of sample points in the scene using efficient existing techniques and computes the resulting indirect lighting at view points [Kontkanen et al. 2006; Hašan et al. 2006]. However, these techniques cannot account for high-frequency view-dependent lighting effects.

Recently the gap between these two branches has been bridged. Zhou et al. [2005] precompute shadow fields for each type of object to render dynamic objects illuminated by multiple local light sources. Wang et al. [Wang et al. 2006] present an efficient method for wavelet rotation and apply this to environment rendering. Many of the techniques mentioned could use this wavelet rotation to implement more efficient relighting by rotating a single compressed version of the BRDF into the local frame for each vertex. Sun and Mukherjee [2006] generalized wavelet product integrals, incorporating dynamic occlusions. They also present a just-in-time radiance transfer technique which reduces the shading integral performed each frame to a double product integral.

All precomputed radiance transfer techniques exploit angular coherence by projecting transport vectors onto some basis and approximating the result. Overbeck et al. [2006] present a system which exploits temporal coherence in lighting by using the differences in lighting between consecutive frames to efficiently relight a scene. This coherence is orthogonal to that exploited by other systems and is thus easily integrated with other techniques. Our technique exploits spatial coherence in the image, which is often ignored in other systems. Some techniques have exploited signal coherence by clustering transport vectors [Sloan et al. 2003; Liu et al. 2004; Tsai and Shih 2006]. Our technique instead uses an adaptive wavelet transform to exploit spatial coherence.

3 Matrix Light Transport

The light transport operator $T(\mathrm{x}, \omega_i \rightarrow \omega_o)$ is a transfer function that linearly maps incident radiance along ω_i to exitant radiance along ω_o at a point x on a surface. As is typical of precomputed radiance transport methods, we assume that the illumination is distant, and can therefore be written as $L(\omega)$, a function of direction only. In this case the light transport operator is

$$T(\mathrm{x}, \omega_i \rightarrow \omega_o) = S(\mathrm{x}, \omega_i)\, f_\mathrm{r}(\mathrm{x}, \omega_i \rightarrow \omega_o)(\omega_i \cdot \mathbf{n}(\mathrm{x})),$$

where S is a binary visibility term that tells whether the light incident at x along ω_i is shadowed, f_r is the BRDF, and $\mathbf{n}(\mathrm{x})$ is the surface normal at x. This form allows us to compactly express the exitant radiance $B(\mathrm{x}, \omega_o)$ at a point x on a surface:

$$B(\mathrm{x}, \omega_o) = \int_\Omega T(\mathrm{x}, \omega_i \rightarrow \omega_o) L(\omega_i)\, d\omega_i. \tag{1}$$

If we make the assumption that either the viewpoint is fixed or the geometry is perfectly diffuse, it can be shown that both the light transport operator and the exitant radiance no longer depends on the outgoing direction ω_o [Ng et al. 2003].

We can then discretize both T and L by choosing a fixed set of sample points x_i in the scene and a fixed set of directions ω_j. This yields a lighting vector $\mathbf{L}$ (e.g., a cubemap) and a transport matrix $\mathbf{T}$, and allows us to rewrite Equation 1 in matrix form:

$$\mathbf{B}[\mathrm{x}_i] = \sum_j \mathbf{T}[\mathrm{x}_i, \omega_j]\mathbf{L}[\omega_j]. \tag{2}$$

Each row in $\mathbf{T}$ corresponds to a single sample point x_i, and each column corresponds to a single lighting direction ω_j. The size of the matrix is therefore the product of the sampled lighting resolution and the spatial sampling resolution. For image relighting, each spatial sample in $\mathbf{T}$ is an image pixel; for geometry relighting, the x_i are model vertices. In general, there is significant coherence in this matrix, both in the rows and the columns.

In order to capture high-frequency lighting effects like sharp shadows, we must finely sample both the image and the lighting, leading to an extremely large matrix $\mathbf{T}$. For some examples in this paper, the raw transport matrix has roughly 10^9 elements, requiring many gigabytes of storage. Therefore, compression is necessary to make $\mathbf{T}$ a manageable size and to permit interactive rendering of scenes under dynamic, complex illumination.

4 4D Wavelet Compression and Relighting

A 2D row transform alone can create a sparse matrix [Ng et al. 2003]. However, there is still significant coherence in the matrix. As shown in Figure 2, the coherence in the columns depends on the row wavelet level. Wavelet lights in the highest subbands create sharp lighting effects and are already sparse. The wavelet scaling function light is a large area light, which produces soft lighting effects and coherent images. Therefore we can further increase sparsity by using the 2D column transform to eliminate the remaining coherence in the image space.

A 4D wavelet transform of $\mathbf{T}$ is illustrated in Figure 3. We perform a 2D wavelet transform over each cubemap face of each row of $\mathbf{T}$.

Figure 2: Matrix column images showing different types of light transport. The left image shows transport from the Haar scaling function (large area light), while the other images correspond to increasingly high-frequency lighting effects from smaller wavelet lights. The left image is easy to compress with a full wavelet transform, while the far right image is already sparse and thus more difficult to compress further.

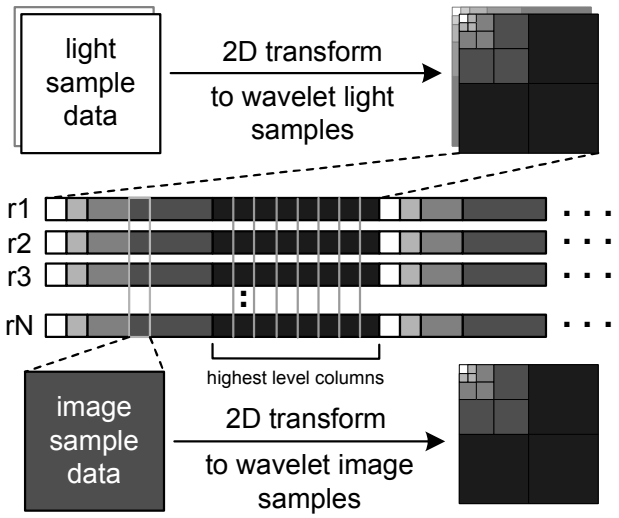

Figure 3: A 4D wavelet decomposition of the light transport matrix **T**. We perform a 2D wavelet transform of the light sample data for each vertex (labeled $r1$ through rN), followed by a 2D wavelet transform of each column in **T**. The matrix layout and example images are colored according to wavelet level, where black is the finest level.

Next, we interpret the columns as 2D images and perform a 2D wavelet transform of each column. Because we use a 2D transform over lighting (row transform) and image (column transform), this can be viewed as a 4D wavelet decomposition of the light transport matrix.

Conceptually, this 4D decomposition partitions **T** according to its frequency content. That is, the row transform turns arbitrary light transport at each point into a well-defined set of wavelet basis functions. This is illustrated in Figure 3, where the matrix elements (shown for rows r1 through rN) are colored according to their wavelet level in the row transform. As a result, each column is composed entirely of coefficients from one level of the row transform, and contains lighting effects (e.g. shadows) that are related to the size and position of a basis function at that level. In general, there are 4^l columns holding light transport from wavelet lights in level l of the row transform.

Figure 4: Column images from T showing light tranport energy from different basis functions. We show illumination from: **a)** A single pixel light, **b)** a single Haar wavelet light at the same angular location as **a**, and **c)** a single Daubechies 6 orthogonal wavelet light. The dark background in the wavelet images illustrates the effectiveness of these filters for compressing rows in T.

4.1 Filters and Parameterization

Previous research has shown that a 2D Haar wavelet filter is effective for compressing light transport vectors and for non-linear lighting approximation [Ng et al. 2003]. Compression errors commonly associated with Haar, such as blocking artifacts, are less a of problem since they are not visualized directly. That is, a rendered image is a linear combination of row elements from **T**, not the elements themselves.

In addition to Haar, we transform the matrix rows using several orthonormal, high-order wavelets including Daubechies 4, 6 and 8 [Daubechies 1988]. These filters are energy-preserving and can, in some cases, more accurately approximate light transport for a given number of terms. Figure 4 shows examples of light transport from pixel and wavelet lights. The shadow information from the wavelet lights is relatively sparse, and thus easier to compress, compared to pixels. Note the significant difference in frequency content between Haar and Daubechies 6.

In contrast to the row transform, it is far more important that we minimize spatial error in the column image transform, since artifacts in this domain are directly visible. Therefore, we perform 2D column transforms with wavelet filters commonly used in the image compression literature. These include the Daubechies filters mentioned above as well as the biorthogonal Legall 5/3 and Daubechies 9/7 filters from the JPEG 2000 image compression standard. In general, we find the 5/3 and 9/7 filters most effective for removing coherence in the image space without introducing unacceptable artifacts.

For any filtering operation (i.e. wavelet transform), there is the need for proper parameterization or there might be severe distortions in the approximation. That is, the data set must be parameterized over the same domain as the filter. For example, if the spatial samples in a scene correspond to model vertices, then the model needs to be parameterized over the plane before we can perform a 2D wavelet transform of the matrix columns. Finding area-preserving planar parameterizations of complex geometry, is a current research topic, with significant recent interest in the graphics community [Lee et al. 1998; Gu et al. 2002]. We use geometry images generated by spherical parameterization [Praun and Hoppe 2003; Gu et al. 2002] to perform geometry relighting. Therefore image and geometry relighting use the same pipeline and column transforms are simply 2D image wavelet transforms for both cases.

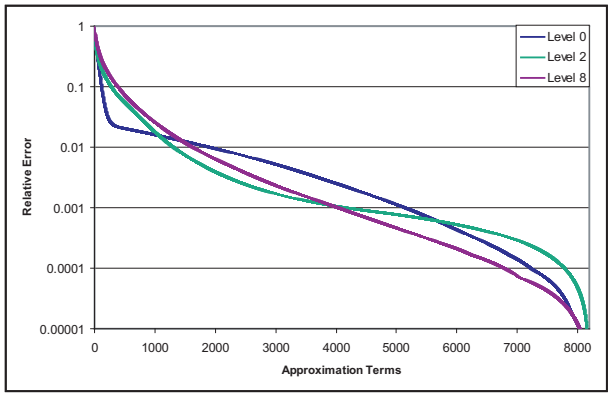

Figure 5: Relative error curves for untransformed, partially transformed, and fully transformed columns. The optimal transform depth is given by the curve with the lowest error for a given number of approximation terms. For some approximations, performing full 4D compression actually increases error compared to 2D compression.

4.2 Adaptive Wavelet Tranforms

Unfortunately, it is possible that a 4D wavelet representation of **T** will capture less energy or require more storage space than the original pixel data. After the row transform, some matrix columns are amenable to wavelet transform while others are not. In particular, there is an *inflation point* for each column, where further wavelet transform steps will in fact increase the number of small, non-zero transport elements. Naively performing a full 4D wavelet transform can inflate the data size by as much as 4x for some columns in our test matrices. This inflation can be explained by the relatively large support of some wavelet bases. If energy is very well localized, then filters with larger support such as those generally used for image compression, actually increase data size by spreading energy among more wavelet coefficients. Figure 5 shows error curves for a few column transform levels of a typical column. This shows that a full 4D transform can inflate data size, requiring more terms to preserve the same amount of energy.

To avoid inflating the size of the transport matrix, we first have to determine, for each column image, at what spatial resolution the inflation begins. We define the *optimal transform depth* for a given column image as the number of transform steps prior to the inflation stage for that image. We can maximize sparsity in **T** using an adaptive 4D wavelet transform. Instead of performing a full 2D wavelet transform of each column, we transform each column image to its optimal transform depth. At step l in the wavelet transform we only need to compare the energy of the non-linear approximation at l and $l+1$, choosing the level which preserves more energy. If both levels contain all the energy in the column image, we choose the level using fewer non-zero terms. Figure 5 shows that the optimal transform depth depends on the number of approximation terms and is not always the 2D or full 4D levels.

The optimal transform depth for a particular column is related to the size of its corresponding wavelet light. Lights in the lower wavelet subbands are large area lights and thus produce low-frequency effects that are amenable to further compression. Lights in higher wavelet subbands create high frequency effects that are less amenable to further compression. This is important because it provides us with a quantitative method for relating the frequency of the lights with that of the image. However, although this relationship follows a general trend, there will likely be data-dependent variations. Further, some high frequency effects, such as silhouettes, are independent of lighting.

4.3 Scaling and Quantization

Our adaptive 4D compression scheme for **T** is especially sensitive to scaling and quantization of the wavelet coefficients. To maintain an accurate light transport approximation, we use non-linear scaling to alter the histogram of the matrix columns. We first transform the matrix rows in full floating-point with the 2D wavelet transform. 2D wavelet compression commonly normalizes each row with a linear scale factor equal to the maximum absolute coefficient value. However, we need a more sophisticated approach to preserve accuracy in the 4D setting. Therefore, following the column transform, we apply a simple non-linear scaling function to lift more terms above the zero quantization threshold. We scale all coefficients at level l by 2^l, since the structure of the wavelet transform ensures that the resulting values are proportional to 2^{-l}, where $l = 0$ is the coarsest level. We note that although we scale the coefficients, the non-linear approximation is performed on the original coefficients. The non-linear scaling is performed only to preserve accuracy which would otherwise be lost due to quantization. Finally, we normalize the column data and dither and quantize the entire matrix to a user specified number of bits. We find 9 bits to be sufficient in general, and in many cases, the parameterization of geometry images hides error and allows us to use fewer bits.

4.4 Precomputation Implementation

Most precomputed light transport methods require significant engineering effort to implement efficiently, and our system is no exception. In fact, our 4D precomputation pipeline is more complicated than typical 2D systems because we evaluate the rows of **T** in one pass and wavelet transform the column images in another. Therefore, we have to transpose **T** between passes, which is a challenge for multi-gigabyte transport matrices and limited memory resources. At a high level, our main precomputation steps include:

1. Light transport sampling.
2. 2D wavelet transform of the matrix rows.
3. Blocked matrix transpose operation on **T**.
4. 2D wavelet transform of the matrix columns.
5. Scaling and scalar quantization of **T**.

For the first step, we sample the light transport operator (visibilty and BRDF) by rasterizing a high-resolution hemicube at each spatial sample and downsampling with bilinear interpolation to the appropriate lighting resolution [Ng et al. 2003]. This sampling process is implemented entirely on programmable graphics hardware. We use pixel shaders to evaluate any analytic reflection function (BRDF) that can be parameterized by the view direction and surface normal. For step 2, we wavelet transform each hemicube face using one of the row filters discussed in Section 4.1.

Step 3 is necessary because we sample **T** in row-major order, but we need **T** in column-major order to perform 2D wavelet transforms on the column images. Transposing the matrix makes data access cache-coherent and further allows us to store the matrix as compressed, wavelet transformed images for efficient relighting. For greater efficiency, we sample the matrix rows for relatively small blocks of image pixels. We then transpose each block independently, and construct column images progressively by storing the

transposed rows into memory mapped files. This is simply a standard, out-of-place matrix transpose and keeps disk activity to a minimum.

In the last precomputation steps, we transform each column image in $\mathbf{T}$ to its *optimal transform depth* for a user specified N term non-linear approximation, as described in Section 4.2. It is necessary to examine all transform depths because in some cases the sparsity can fluctuate. This occurs in only a small fraction of matrix columns but is enough to affect compression.

4.5 Relighting

Our image relighting algorithm is based on the technique of Ng et al. [2003] in that we use a non-linear approximation for the lighting. As a result, we are able to render each frame accurately and efficiently using a very small number of wavelet lights and, in turn, a small fraction of the light transport matrix $\mathbf{T}$. Relighting from a 2D compressed matrix is simple because Equation 2 is valid as long as $\mathbf{T}$ and $\mathbf{L}$ are represented in the same orthonormal basis such as pixels or Haar wavelets. However, relighting from our 4D representation is more complicated because the matrix column images are wavelet transformed, yet we need the rendered image in pixels for display.

We begin by presenting a relighting formula for full 4D wavelet transforms, which we then extend to support adaptive transforms. For an N-term non-linear lighting approximation, we can reformulate Equation 2 as:

$$\mathbf{B} = \sum_{i=1}^{N} w^i \mathbf{S}' \left(\mathbf{C}^i \right) \tag{3}$$

where $\mathbf{B}$ is the rendered image, w^i is a wavelet light, $\mathbf{C}^i$ is the compressed column image corresponding to w^i, and $\mathbf{S}'$ is the inverse wavelet transform operator, modified to accommodate the non-linear scaling from Section 4.3. However, Equation 3 is painfully inefficient since it requires N inverse wavelet transforms to produce a single image. Fortunately, $\mathbf{S}'$ is a linear operator, which means we can factor it out of the summation to get:

$$\mathbf{B} = \mathbf{S}' \left(\sum_{i=1}^{N} w^i \mathbf{C}^i \right) \tag{4}$$

Now, this formulation is quite efficient considering it involves only one inverse transform independent of N. In fact, as we show in Section 5, the computational cost of $\mathbf{S}'$ quickly becomes negligible as N increases. In short, Equation 4 is a fast relighting solution for our full 4D compression scheme.

We cannot use Equation 4 for relighting from an adaptively compressed transport matrix because the columns are not expressed in the same basis. We could go back to using Equation 3 by changing $\mathbf{S}'$ to start at the transform depth of each column image, but this would be prohibitively expensive. Fortunately, we can easily derive an adaptive relighting formula with the same computational cost of Equation 4. Note that the structure of the wavelet transform allows us to rewrite $\mathbf{S}'$ recursively in terms of single-step inverse transforms:

$$\mathbf{S}'(\mathbf{C}) = \mathbf{S}'_{l-1 \to l} \left(\mathbf{S}'_{l-2 \to l-1} \left(\ldots \left(\mathbf{S}'_{0 \to 1} (\mathbf{C}) \ldots \right) \right) \right) \tag{5}$$

where $\mathbf{C}$ is a column image, and $\mathbf{S}'_{l \to l+1}$ denotes a single-step inverse wavelet transform from level l to $l+1$. Substituting Equation 5 into Equation 3, and once again exploiting the linearity of $\mathbf{S}'$, lets us write a simple recursive formula for relighting from adaptively compressed matrix columns:

$$\mathbf{B}_{l+1} = \sum_{i \in \mathbf{N}(l+1)} w^i \mathbf{C}^i + \mathbf{S}'_{l \to l+1} (\mathbf{B}_l) \tag{6}$$

$$\mathbf{B}_0 = \sum_{i \in \mathbf{N}(0)} w^i \mathbf{C}^i \tag{7}$$

where $\mathbf{B}_l$ is the rendered image for wavelet levels up to and including l, and $\mathbf{N}(l)$ is the set of all wavelets lights for columns transformed to level l. The desired cost of only one full inverse transform per frame is explicit in the recursion. Finally, for efficient implementation, we can rewrite this equation in iterative form for all wavelet levels W for a given image resolution:

$$\mathbf{B} = \sum_{l \in W} \mathbf{S}'_{l \to l+1} \left(\sum_{i \in \mathbf{N}(l)} w^i \mathbf{C}^i \right) \tag{8}$$

Figure 6 shows a graphical illustration of the steps we use to directly implement Equation 8. The steps of this algorithm for an N-term non-linear lighting approximation are as follows:

1. Sort the lights according the transform depth of their corresponding columns in $\mathbf{T}$, such that lights for columns compressed with more wavelet transform steps come first.
2. For each set of wavelet lights $\mathbf{N}(l)$, compute the partial relighting approximation using a sparse matrix multiply[Ng et al. 2003].
3. Inverse transform the rendered image from level l to $l+1$, and repeat these steps for all wavelet levels (i.e. for all N lights).

At each frame, we sample the illumination in the raw pixel basis of a cubemap and compute a non-linear wavelet approximation for the light vector. The running time for this relighting algorithm is linear in the number of lights used for the non-linear approximation. The inverse wavelet transform is $O(n)$, where n is the number of elements in a compressed column image, so the cost here only depends on the size of the image, but is constant for any particular scene. Overall, this is significant because it provides us a way to predict an approximate upper bound on the number of lights we can use given a desired framerate. We discuss this in more detail in Section 5.

5 Results

In this section, we present results for the algorithms in Section 4, including compression and rendering times for test scenes. We also analyze and compare error between 2D, 4D and adaptive 4D wavelet approximations of $\mathbf{T}$. Not surprisingly, some scenes are more amenable to 4D wavelet compression than others. In general, however, we achieve around 2x-3x greater compression and relighting speed relative to 2D compression with the same approximation error.

5.1 Precomputation

Our precomputation times are reasonably efficient, ranging from minutes for scenes with roughly 50-100k image samples to hours for scenes with very high image and lighting resolutions. We greatly reduce the cost of visibility sampling by storing the scene geometry in fast video memory and substituting lower resolution

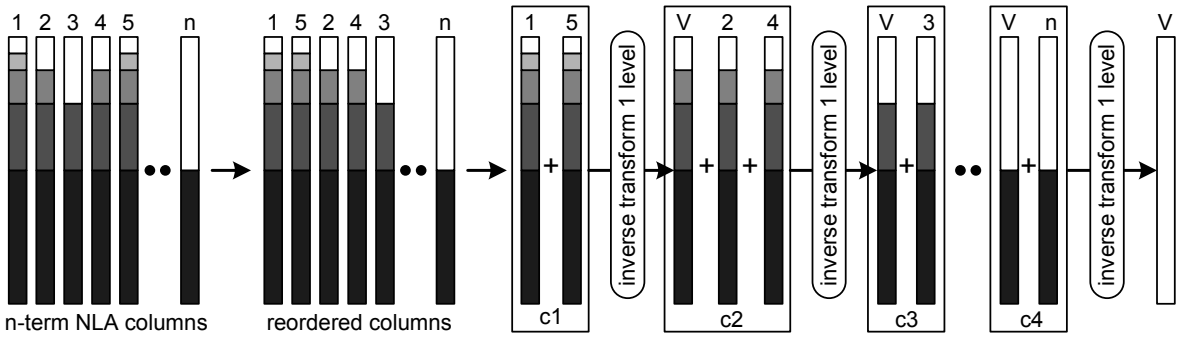

Figure 6: A 1D illustration of our relighting pipeline, as applied to an n-term non-linear approximation. The columns are colored according to a standard wavelet decomposition; the scaling function is shown as white, and the wavelet levels are differentiated with gray scales (black being the finest level). First, we sort the lights based on the transform depth of their corresponding columns. This process forms clusters of columns (labeled c1 through c4), where a cluster is defined as the set of all columns with the same size scaling function. We relight each frame using a sparse matrix multiply within each cluster, followed by a single-level inverse transform. This continues until we have a fully reconstructed column image V.

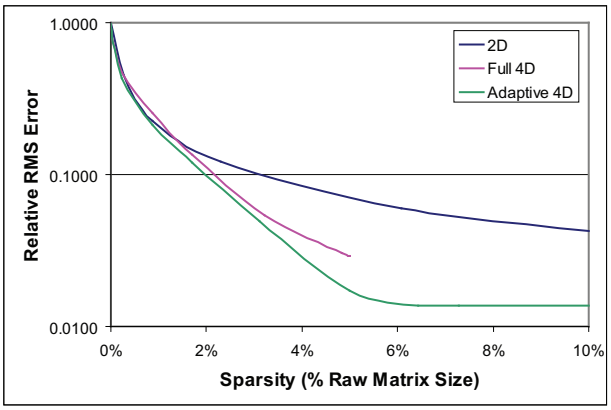

Figure 7: A graph showing sparsity vs. error in the light transport approximation for 2D, full 4D, and adaptive 4D compression schemes for an average scene. The adaptive 4D compression produces more numerically accurate approximations at all levels. The naive 4D compression inflates the matrix at some sizes and compresses at others. Note the y-axis is log scale to focus on matrices with reasonable error, 10% or less.

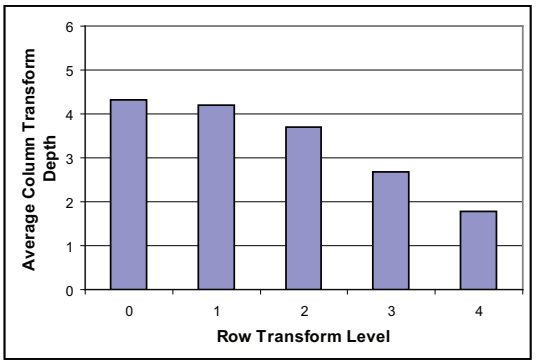

Figure 8: Graphs showing wavelet transform statistics for a sample matrix. The y-axis shows the average column transform depth chosen using the algorithm in Section 4.4, as a function of the row transform level of the associated wavelet lights (x-axis). As expected, low-frequency column images are transformed more on average than high-frequency columns.

models for visibility sampling. Transport sampling makes up a relatively small percentage of the total precomputation time for most scenes. The matrix transpose and search for the optimal transform depth consume the majority of processing time. This pass is expensive because the raw transport matrix is usally many gigabytes, so we cannot load the entire matrix into memory. Instead, we transpose the matrix using an $O(n^2)$ process, where n is the number of blocks. Although the wavelet transform is a relatively fast, $O(n)$ algorithm, finding the optimal transform depth is expensive because we sort the coefficients to compute the non-linear approximation.

5.2 Compression and Error Analysis

Measuring approximation error for precomputed transport matrices is challenging because numeric error does not always translate to perceptual error in the image. However, common error metrics like the L^2 norm are still useful for comparing compression methods. Figure 7 shows graphs of sparsity vs. RMS error for typical light transport matrices. The important point here is that adaptive 4D compression has less error over the entire range of matrix sizes. Also note that full 4D compression produces matrices both larger and smaller than 2D compression. Full 4D compression also cannot generate matrices with as large a range of sizes and error as adaptive 4D compression, reaching its minimum error and maximum size at about 5% for the sample matrix. This occurs because, even with non-linear scaling, most coefficients in the full matrix will quantize to zero.

Figure 8 shows the average transform depth of the columns of a sample matrix by the row transform level for the column. Images corresponding to large wavelet lights are, on average, transformed to a greater depth than those corresponding to small wavelet lights. This confirms our expectation concerning the compressibility of different column images like those in Figure 2.

In some cases, 2D compression schemes for **T** can yield highly non-uniform storage requirements, since high-frequency columns are compressed significantly more than low-frequency columns [Ng et al. 2003]. The 4D adaptive wavelet transform removes coherence in the columns wherever it exists, so the storage requirements

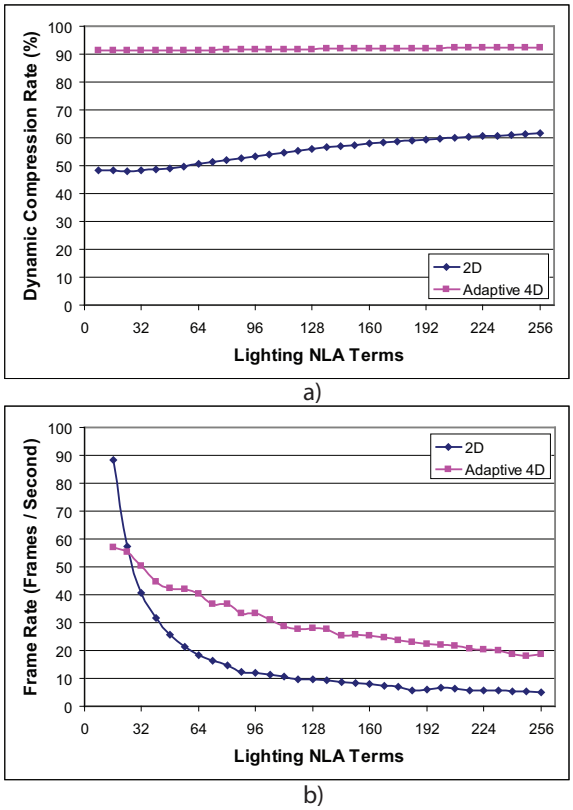

Figure 9: Graphs comparing 2D and adaptive 4D relighting performance for matrices of equal quality over a range of lighting approximations, averaged over 200 frames. **a)** shows the dynamic compression rate for each matrix, which is high and steady for 4D compression and increases slowly with more lighting terms for 2D compression. **b)** shows the frame rate for both matrices. 4D compression quickly surpasses 2D as we increase the number of wavelet lights.

do not depend as heavily on the frequency content of **T**. The result of the 4D wavelet compression is that *columns at all levels are compressed to essentially the same extent*. This is a useful result because we can approximate the working data set for relighting as NX, where N is the number of lights and X is the average number of non-linear approximation terms for a column image.

The wavelet filters used in each of the 2D transforms can greatly affect the compression and error. We have performed a simple analysis of the filters mentioned in Section 4.1, and we found the combination of Daubechies 4 for the row transform and Legall 5/3 for the column transform is best. However, we have not performed this analysis over a wide variety of matrix sizes and sample scenes. The best choice will be data dependent. A full analysis of the interaction of these wavelet transforms would be very valuable.

5.3 Relighting

We achieve interactive relighting using a straightforward implementation of the algorithm described in Section 4.5. Actual framerates for our test scenes range from 5 to 100 frames per second depending on the image and light resolutions, as well as the size of the nonlinear approximation for L. For example, Figure 1 shows an example image generated by our system and the graphs in Figure 9 show relighting results from a sample scene. We use the area weighted method for non-linear approximation of the lighting as described in Ng et al. [2003]. The dynamic compression rate, or the effective compression of the working set of matrix columns for a particular relit frame (graph **a**), is low for the 2D matrix and slowly increases with more lighting terms. Since large wavelet lights usually contain the most energy and are among the first terms in the lighting approximation, this indicates that the 2D matrix is less compressed for large lights than for small lights. Adaptive 4D compression provides significantly better dynamic compression rates that are almost constant over the entire range of lighting approximation sizes, even for matrices which are not limited by the user specified number of column non-linear approximation terms. This indicates that our adaptive 4D technique compresses all columns to approximately the same extent, as we asserted earlier. In graph **b** we see that initially the 2D compression scheme yields higher framerate, which is due to the overhead of the inverse wavelet transform. However, the 4D framerate quickly surpasses the 2D, by 2x-4x for this particular scene, as we increase the number of wavelet lights. The framerate is near linear once the inverse wavelet transform becomes insignificant, as expected given the linear working set described in Section 5.2. We can use this property to approximate the frame rate for a given number of lighting approximation terms.

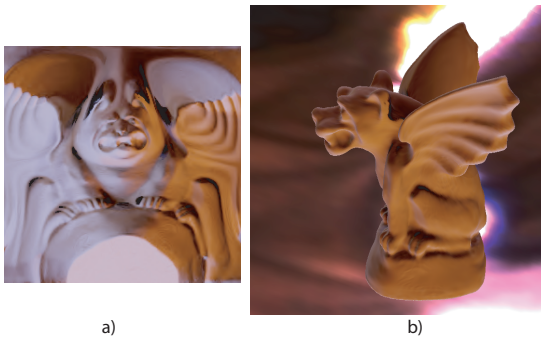

Figure 10: An example of geometry relighting for a 256x256 mesh and 6x64x64 lighting resolution, compressed to about 5.5% the original matrix size. **a)** shows the geometry image, relit at 18 frames per second using a 256 term non-linear lighting approximation with no visible compression artifacts and **b)** shows this texture applied to the mesh.

6 Conclusions and Future Work

We have presented a compression and relighting technique, based on adaptive 4D wavelet transforms, for high-resolution light transport matrices. One of our insights is that the compressibility of images or columns decreases for higher frequency lights. Indeed, the net compression from both row and column transforms is approximately the same across all columns in our examples.

To make PRT more useful in practice we believe there must be a focus on more effective compression as well as additional features. Our technique improves compression and reduces the number of transport terms involved in relighting, thus increasing the overall frame rate. There is significant future work in extending this system to other PRT techniques. In some cases the system should extend easily by applying the additional transform and performing the new relighting procedure, such as in systems which allow view dependent BRDFs [Liu et al. 2004; Wang et al. 2004]. In these systems there are essentially k matrices for a k-term BRDF approximation.

Each of this matrices can be compressed independently using our technique. To perform relighting, view independent color vectors are generated using each matrix. The final colors are calculated as a linear combination of the view-independent color vectors using view-dependent weights. We maintain all the benefits of our system without significant changes to the underlying algorithm. However, the method is not directly applicable to other systems, such as the triple or generalized wavelet product integral based systems [Ng et al. 2004; Sun and Mukherjee 2006].

CPCA has been used in other systems to exploit spatial coherence in PRT data [Sloan et al. 2003; Liu et al. 2004]. Our technique offers a much cheaper alternative to CPCA by using a standard wavelet transform, choosing an optimal basis through our adaptive transform. This becomes increasingly significant as the dimension of the transport vectors increases to better represent lighting effects. This improvement comes at the cost of requiring data to be parameterized as a geometry images [Praun and Hoppe 2003] for geometry relighting. Investigating remeshing procedures which take into account relighting would be useful. Also, a more in depth and principled approach to determine which wavelets interact well and why would be useful not only in our system but in any system which might combine sets of wavelet transforms. A more significant area of future work would be investigating alternative spatial transforms, specifically those which do not require special parameterization and are not data dependent.

While the 4D adaptive wavelet transform presented here is one solution to the problem of non-uniformly sampling light transport matrices, it is by no means the only one. Future work can also focus on direct approaches to non-uniformly sample precomputed matrices and functions on geometry and images. We believe approaches based on sampling the light transport function properly will be very significant in future work in this area.

References

DAUBECHIES, I. 1988. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41, 909–996.

GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 355–361.

HAŠAN, M., PELLACINI, F., AND BALA, K. 2006. Direct-to-indirect transfer for cinematic relighting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, ACM Press, New York, NY, USA, 1089–1097.

KAUTZ, J., SLOAN, P.-P., AND SNYDER, J. 2002. Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 291–296.

KAUTZ, J., LEHTINEN, J., AND AILA, T. 2004. Hemispherical rasterization for self-shadowing of dynamic objects. In *Eurographics Symposium on Rendering*, Eurographics Association, Norrkping, Sweden, A. Keller and H. W. Jensen, Eds., 179–184.

KONTKANEN, J., AND LAINE, S. 2005. Ambient occlusion fields. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, ACM Press, New York, NY, USA, 41–48.

KONTKANEN, J., TURQUIN, E., HOLZSCHUCH, N., AND SILLION, F. 2006. Wavelet radiance transport for interactive indirect lighting. In *Rendering Techniques 2006 (Eurographics Symposium on Rendering)*, W. Heidrich and T. Akenine-Möller, Eds., Eurographics.

LEE, A., SWELDENS, W., SCHRODER, P., COWSAR, L., AND DOBKIN, D. 1998. MAPS: Multiresolution adaptive parameterization of surfaces. In *SIGGRAPH 98*, 95–104.

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

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

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. 23*, 3, 477–487.

OVERBECK, R., BEN-ARTZI, A., RAMAMOORTHI, R., AND GRINSPUN, E. 2006. Exploiting temporal coherence for incremental all-frequency relighting. In *Eurographics Symposium on Rendering*, Eurographics Association, 151–160.

PRAUN, E., AND HOPPE, H. 2003. Spherical parametrization and remeshing. *ACM Trans. Graph. 22*, 3, 340–349.

REN, Z., WANG, R., SNYDER, J., ZHOU, K., LIU, X., SUN, B., SLOAN, P.-P., BAO, H., PENG, Q., AND GUO, B. 2006. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph. 25*, 3, 977–986.

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

SLOAN, P.-P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics 22*, 3 (July), 382–391.

SLOAN, P.-P., LUNA, B., AND SNYDER, J. 2005. Local, deformable precomputed radiance transfer. *ACM Trans. Graph. 24*, 3, 1216–1224.

SUN, W., AND MUKHERJEE, A. 2006. Generalized wavelet product integral for rendering dynamic glossy objects. *ACM Trans. Graph. 25*, 3, 955–966.

TSAI, Y.-T., AND SHIH, Z.-C. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph. 25*, 3, 967–976.

WANG, R., TRAN, J., AND LUEBKE, D. 2004. All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *Proceedings of the Eurographics Symposium on Rendering*, 345–354.

WANG, R., TRAN, J., AND LUEBKE, D. 2005. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph. 24*, 3, 1202–1207.

WANG, R., NG, R., LUEBKE, D., AND HUMPHREYS, G. 2006. Efficient wavelet rotation for environment map rendering. In *Eurographics Symposium on Rendering*, Eurographics Association, 173–182.

ZHOU, K., HU, Y., LIN, S., GUO, B., AND SHUM, H.-Y. 2005. Precomputed shadow fields for dynamic scenes. *ACM Trans. Graph. 24*, 3, 1196–1201.