

Compact and Efficient Generation of Radiance Transfer for Dynamically Articulated Characters

Derek Nowrouzezahrai Patricio Simari Evangelos Kalogerakis Karan Singh Eugene Fiume
University of Toronto - {derek,psimari,kalo,karan,elf}@dgp.toronto.edu



Figure 1: Our system is able to generate PRT coefficients for a dynamically animated character in real-time with a low memory footprint (see Table 2.) Compared to un-shadowed shading using irradiance environment maps (first column), our results clearly improve the visual quality of an animated object. Low memory requirements, fast model fitting and a simple runtime algorithm make our approach suitable for interactive applications, such as games.

Abstract

We present a data-driven technique for generating the precomputed radiance transfer vectors of an animated character as a function of its joint angles. We learn a linear model for generating real-time lighting effects on articulated characters while capturing soft self-shadows caused by dynamic distant lighting. Indirect illumination can also be reproduced using our framework. Previous data-driven techniques have either restricted the type of lighting response (generating only ambient occlusion), the type of animated sequences (response functions to external forces) or have complicated runtime algorithms and incur non-trivial memory costs. We provide insights into the dimensionality reduction of the pose and coefficient spaces. Our model can be fit quickly as a preprocess, is very compact (~ 1 MB) and runtime transfer vectors are generated using a simple algorithm in real-time (> 100 Hz using a CPU-only implementation.) We can reproduce lighting effects on hundreds of trained poses using less memory than required to store a single mesh's PRT coefficients. Moreover, our model extrapolates to produce smooth, believable lighting results on novel poses and our method can be easily integrated into existing interactive content pipelines.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

Keywords: Animated Precomputed Radiance Transfer, Real-time, Dimensionality Reduction, Model Fitting

1 Introduction

Traditional precomputed radiance transfer (PRT) addresses the problem of shading a static scene under dynamic environmental lighting. Real-time performance is achieved by tabulating the visibility in a preprocess and decoupling it from the evaluation of external lighting [Sloan et al. 2002]. Low-frequency PRT can incorporate complex light transport effects, such as indirect illumination and subsurface scattering [Wang et al. 2005; Sun et al. 2005], at no extra runtime cost. By projecting the visibility, BRDF and lighting into a low-frequency basis, soft shading effects can be reconstructed without explicitly sampling all hemispherical directions [Sloan et al. 2002; Kautz et al. 2002]. At the expense of extra storage and reconstruction time, other bases can be used to generate “all-frequency” effects under varying distant illumination [Ng et al. 2003; Ng et al. 2004; Green et al. 2006; Tsai and Shih 2006; Ma et al. 2006] or for the incorporation of glossy and specular materials [Sloan et al. 2002; Kautz et al. 2002; Lehtinen and Kautz 2003; Ng et al. 2004; Tsai and Shih 2006].

Motivation A major limitation of classic PRT techniques is that they only operate on static scenes. The prohibitive cost of sampling the per-vertex visibility every frame precludes the use of regular PRT on animated scenes in real-time. Some works aim at simplifying this calculation in order to facilitate moving geometry [Kautz et al. 2004; Ren et al. 2006]. Alternatively, interactive applications, such as games, may incorporate PRT techniques into the content generation pipeline, but only for scene data that remains fixed. We propose a data-driven technique for real-time generation of PRT coefficients on articulated characters. Instead of simplifying the visibility generation and accumulation, we learn a predictive model trained on standard PRT simulation data. Our technique can be easily incorporated into current game content pipelines, is easy to implement, and has low precomputation times and storage requirements.

Our goal is to capture PRT effects, such as self-shadowing and inter-reflections, on an articulated character. In this work we focus on two areas: low-memory solutions for high-accuracy reproduction of PRT data on trained animation poses and believable generation of shading on novel animation poses.

Overview We present a data-driven technique for learning a linear mapping from a character’s pose described by joint angles to the lighting transfer over the character’s surface; skinned animation models are widely adopted in current art pipelines for interactive applications such as games. Our technique allows for real-time PRT rendering of an articulating character under dynamic low-frequency lighting. We show that a linear function of pose angles can yield radiance transfer coefficients with low reconstruction error on observed poses. Moreover, by applying regularization (see section 4) we show that the model is able to generalize well to novel poses.

In order to handle large meshes while still maintaining low memory costs, we apply dimensionality reduction to both the space of input pose vectors as well as the space of vertex coefficients while avoiding the more complicated clustering and localized model fitting of other approaches (see section 2.)

The benefits of using a reduced model are three-fold: learning requires fewer samples due to the reduction in degrees of freedom, the resulting model can be stored using about the same amount of memory as is required to store a single mesh’s PRT coefficients explicitly, and runtime evaluation of the reduced model becomes more efficient. We explore how the reduced dimensional model trades accuracy for these benefits but can still produce smooth, believable and consistent results on novel test sequences. Figure 1 illustrates the results generated by our system in real-time on two animated sequences.

Contributions

- Our model fitting is fast and uses a simple analytic formula.
- The resulting model is compact yet scalable: the user can trade memory requirements for reconstruction accuracy.
- Runtime evaluation is simple: 1 to 3 matrix multiplications are required per animation frame.
- We investigate the numerical and visual effects of dimensionality reduction on the covariant *joint angle* and *vertex coefficient* spaces.

In our experiments, we are able to produce consistent, believable shadowing for different animation sequences using as little as **1 MB** of storage. This is orders of magnitude less storage than competitive techniques and, in the most compressed cases, the storage required for our model is less than the storage of the transfer vectors for a *single* pose of the animation sequence. Runtime evaluation is also much faster than current techniques, yielding frame-rates above 100 Hz on our CPU-only implementation. This is due to the fact that runtime evaluation does not require any complicated nearest-neighbor searching and extrapolation; we have a single simple runtime algorithm for handling both trained and novel poses.

2 Related Work

The ability to reproduce complicated lighting and transport effects on dynamic scene geometry has been investigated, in different directions, in the recent works we outline below.

2.1 Data-Driven Models for Lighting

We determine an animated character’s response to variable lighting using a data-driven approach. Recent works motivate the problem in a similar fashion and experiment with learning lighting response using different models. James and Fatahalian [2003] precompute a dynamics and illumination model for deformable objects. Their reduced model allows standard PRT lighting vectors to be generated in real-time on a known deformable mesh. We model lighting as a function of pose (as opposed to impulse forces) and, like James and Fatahalian, are able to accurately reproduce shading similar to that in observed poses. Additionally, our system has the capability to generate plausible and consistent lighting response for completely novel pose scenarios in real-time with low memory cost.

Two recent papers address the generation of ambient occlusion values on animated character meshes. Kontkanen and Aila [2006] learn a linear model mapping a character’s pose to the ambient occlusion values over its vertices. Kirk and Arikan [2007] learn a multilinear model over a segmented and reduced pose space and can handle larger datasets. We show how a linear mapping can be found to the entire vector valued transfer function, not only the scalar DC component. Moreover, we apply principled yet simple dimensionality reduction techniques that extend to large datasets while avoiding complicated clustering techniques. Our model assumptions are validated both visually and by the statistics of our results.

Most recently, Wen Feng et al. [2007] take advantage of the covariant joint angle and vertex coefficient spaces to produce glossy transport effects under varying illumination for articulated characters. They introduce a mesh-clustering technique that exploits redundancies in transport data in a similarly motivated fashion as in our work. Their technique can reproduce convincing shading results for articulated characters with parametric BRDF models at the added cost of memory (between 76 and 430 MBs for their animated scenes) and much more complicated training and runtime algorithms. Our technique requires orders of magnitudes less memory storage and precomputation time than Wen Feng et al. and has better runtime performance using an unoptimized CPU-only algorithm, although we only handle diffuse shading effects. Furthermore, Wen Feng et al. locally interpolate transfer matrices per segment using a clustering technique, whereas our generative model synthesizes transfer leading to a technique that requires significantly less training data. For example, on the Armadillo mesh, we train on 250 frames of animation and Wen Feng et al. use 1024 frames. In section 9 we discuss why our framework is well-suited for potential future work in general BRDF lighting of articulated characters.

2.2 Real-Time Methods for Dynamic Scenes

Kautz et al. [2004] use graphics hardware to rasterize the per-vertex hemispherical visibility function of a dynamic object. They report interactive frame-rates for direct-illumination shading on moderately sized diffuse and glossy meshes. Ren et al. [2006] represent dynamic geometry as a hierarchy of spheres and tabulate logarithmic SH visibility vectors as a function of subtended angle. At runtime, log visibility vectors are accumulated and exponentiated to yield the final visibility vector used for diffuse shading. This technique also only supports direct-illumination and special care must be taken to ensure proper self-shadowing.

The low-resolution visibility maps and two-level hierarchy of Kautz et al., as well as Ren et al.’s spherical mesh approximation are generally of adequate visual quality. These geometrical simplifications are not noticeable after shading with low-frequency lighting.

Zhou et al. [2005] precompute and store visibility coefficients for a

rigid object at discrete samples on concentric shells surrounding the object (*i.e.*, shadow fields.) The final visibility function at receiver points is interpolated from these samples for each moving object and multiplied in the projected space. Shadow fields often consume a prohibitive amount of memory. Tamura et al. [2006] worked on optimizing the sampling schemes used on the shells of a shadow field, reducing memory consumption. Mei et al. [2004] introduce a spherical volumetric storage table based on shadow maps. They are able to generate direct and indirect illumination of rigid dynamic objects under a fixed lighting environment in real-time.

Kontkanen and Laine [2005] combine a bounding box tabulation and spherical cap approximation of local visibility to generate approximate ambient occlusion caused by moving rigid objects. Their technique is tailored for evaluation on the GPU and yields real-time performance.

Sloan et al. [2005] focus on *local* effects, such as wrinkles and bumps, and precompute various diffusion and scattering models in the zonal harmonics basis. This serves to reduce storage costs and enables fast rotations. Real-time results are achieved but global effects, such as self-shadowing and indirect illumination, are not supported.

Wang et al. [2006] determine an analytical matrix formulation for approximating a spherical harmonics operator of angular scaling over the sphere. They demonstrate the utility of their operator for approximating near-field illumination effects and cast shadows due to deforming objects. As with Sloan et al. [2005], this model is not easily applicable to attached shadows for articulating or deforming scene geometry.

Each work above approximates the lighting of dynamic scenes in a different way. However, it is evident that low storage cost and simple runtime evaluation are necessary in order to produce a solution to real-time lighting of dynamic scenes that scales with increasing numbers of dynamic objects.

We focus our attention on articulated characters, which are by far the most common deforming object in interactive animations, such as those used in games. We are able to reproduce lighting very accurately on trained poses as well as generating consistent, believable results on novel poses. Our system has the added advantages of very low storage requirements and a simple runtime algorithm.

2.3 Data Compression

Our datasets combine PRT and animation data and are discussed in more detail in section 4. PRT datasets can grow to unwieldy sizes. To compress these datasets in a manner that facilitates efficient runtime reconstruction, Sloan et al. [2003] propose a clustered principal component analysis of glossy PRT data. Their technique uses a combination of clustering and dimensionality reduction in vertex space. The resulting data is significantly compressed and can be used directly on the GPU to evaluate the final shading computation. Tsai and Shih [2006] use spherical radial basis functions and clustered tensor approximation to represent and compress glossy transfer under all-frequency lighting conditions; a substantial boost in runtime performance when compared to CPCA techniques tailored for all-frequency bases is achieved. Both of these techniques focus on the compression and reconstruction of transfer on static scenes with glossy materials.

Similarly, the analysis and use of reduced dimensional pose subspaces has proven to be an effective technique for compression in data-driven character animation [Safanova et al. 2004; Forbes and Fiume 2005; Chai and Hodgins 2005; Arikian 2006]. We apply similar techniques to analyze pose subspaces in order to reduce the

dimensionality of our dataset. We separately reduce the dimensionality of both the input pose space and the output vertex coefficient space (see section 5.) We will contrast the results of a linear model generated using the full dimensionality of the system with the results of applying dimensionality reduction.

3 Precomputed Radiance Transfer for Articulated Characters

Our work makes the same assumptions as PRT for static scenes. All of our materials are diffuse, the external lighting is infinitely distant and we only capture low-frequency effects. The direct-illumination diffuse PRT vector at pose i and vertex j with normal \mathbf{n} is

$$\mathbf{t}^{j,i} = \int_{\Omega_n} \underbrace{V_i^j(\omega) (\mathbf{n} \cdot \omega)}_{T_i^j(\omega)} \mathbf{y}(\omega) d\omega,$$

where Ω_n is the hemispherical domain about the vertex's normal, $V_i^j(\omega)$ is the binary visibility function, $T_i^j(\omega)$ is the transfer function and $\mathbf{y}(\omega)$ is the vector of SH basis functions [Sloan et al. 2002]. The final shading value of a vertex at any pose is simply a dot product of this transfer vector and the SH projected lighting vector, \mathbf{L}_{in} : $L_{out}^{j,i} = \mathbf{L}_{in} \cdot \mathbf{t}^{j,i}$.

Although we perform our analysis on direct-illumination data, our learned models can also be applied to PRT simulations with more complex transport effects (such as indirect illumination [Sloan et al. 2002]); we illustrate examples of these effects in the results section.

We will show that the transfer coefficients at the vertices of an articulated character mesh can be well approximated as a linear function of the pose of the character. Projecting to lower-dimensional input and output spaces reduces the amount of storage required for our system while maintaining accuracy.

3.1 Methods

We learn linear models which, given an input pose vector, output a set of per-vertex diffuse PRT vectors for shading an articulated character. In order to train the system, we perform PRT simulations on the frames of an animating sequence. We can reproduce the lighting response on these training poses with little to no visual difference and at a small fraction of the storage required for the PRT vectors at each frame. On novel poses, we can generate consistently smooth, believable shading.

4 Linear model

The input data to our system is a sequence of meshes that are the poses of one or more animations. The number and relative order of the vertices remains fixed, as is typically the case for animated meshes. For each such pose, we have the set of joint angle values that describe it. Assuming p poses and a angles, we represent this data in a $p \times a$ matrix \mathbf{A} where $\mathbf{A}_{i,j}$ is the value of the j^{th} joint angle at the i^{th} pose.

For each pose, we precompute a 6^{th} order SH transfer function resulting in 36 transfer coefficients and store them in a set of matrices \mathbf{B}^c , $1 \leq c \leq 36$. These matrices are of size $p \times v$ where v is the number of vertices in the mesh and $\mathbf{B}_{i,j}^c$ represents the c^{th} transfer coefficient for the j^{th} mesh vertex at the i^{th} pose.

Given this data, we may now learn a set of linear mappings \mathbf{X}^c for each coefficient c that solves the over-constrained system $\mathbf{AX}^c = \mathbf{B}^c$.

For clarity of notation we may avoid the c superscript, but it should be kept in mind that there is one such linear mapping for each coefficient whose approximation is desired.

4.1 Motivation and Validation of a Linear Model

Using a linear model to predict the lighting transfer on an articulated character has many benefits. A linear model can be fit to our data using a simple and efficient closed-form equation and our runtime reconstruction algorithm is straightforward. Of particular importance to our application, the transfer vectors generated using a linear model are guaranteed to vary smoothly with respect to perturbations in the input (joint angle vectors) by construction. Therefore, given a smoothly varying animation sequence, the lighting response will also vary smoothly. As we will illustrate below, a linear model also happens to be well-suited to the structure of the data.

The previous works of Kontkanen and Aila [2006] and Kirk and Arikan [2007] both use (locally) linear models to fit the ambient lighting response over character poses, however neither provides any direct justification of the linear model for modeling lighting response. We have investigated the validity of a linear model for not only modeling the ambient (DC spherical harmonics term) lighting response, but also the directional lighting response up to a 6th order spherical harmonics expansion.

The large size of the transfer vector datasets precludes the feasibility of validating the linear model via direct visualization of linear patterns in the data. Instead, we use a statistical technique of analyzing the density and cumulative probability distribution of the residuals of our regularized (see section 4.2) linear fit.

Figure 2 plots the distribution and empirical cumulative distribution of the signed residuals of the regularized linear fitting for a representative experimental case. Note the single mode, zero mean and median and low variance that further validate the linear model. We have observed similar distributions in all of our datasets, spanning across different skinned meshes and training animation sequences.

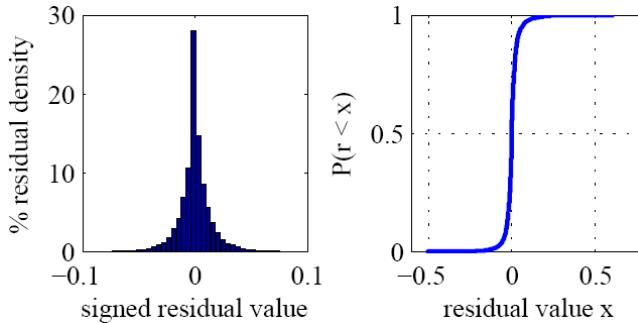


Figure 2: **Left:** Distribution of signed residuals of regularized linear fit: $\mu = 5.1 \times 10^{-5}$, $\sigma^2 = 3.25 \times 10^{-4}$. **Right:** Empirical cumulative distribution of residuals. All of our datasets exhibit the behavior above validating the use of a linear model.

As far as we know, we are the first work to provide insight into the validity of choosing a linear system for fitting lighting response to articulated poses.

4.2 Regularization

The equation $\mathbf{AX} = \mathbf{B}$ can be solved in closed form by using the classical formulation $\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{B}$.

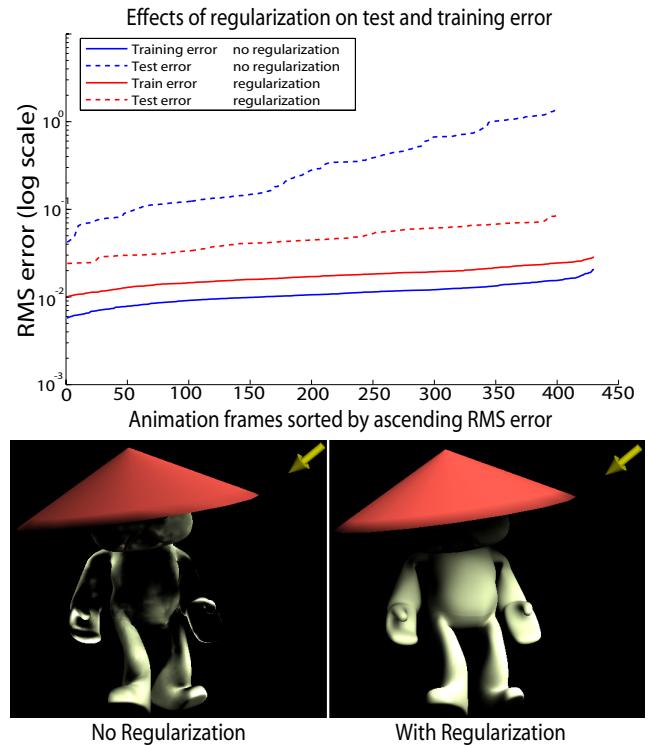


Figure 3: **Top:** The training and test errors of the regularized and unregularized linear systems for a representative set of animation sequences. **Bottom:** Approximation from linear mapping on a novel test pose without (**left**) and with (**right**) regularization. For these results we use $\alpha = 231$. Arrows denote the incoming lighting direction.

While this solution is guaranteed to be the least-squares solution to our training data, it may not be well behaved. In essence we have a prior that for small change in pose, the resulting change in PRT coefficients should tend to be equally small. This can be achieved by penalizing large values in the solution matrix \mathbf{X} . To this end we use Tikhonov regularization which minimizes the sum of squared residuals and the squared Euclidean norm of \mathbf{X} . This objective function can be expressed as $\|\mathbf{AX} - \mathbf{B}\|^2 + \alpha^2 \|\mathbf{X}\|^2$. The parameter $\alpha \geq 0$ controls bias and a suitable value can be chosen using cross validation (for example, we minimize leave-one-out error in order to obtain our α values.) The closed form solution to this expanded system is given by $\mathbf{X} = (\mathbf{A}^T \mathbf{A} + \alpha^2 \mathbf{I})^{-1} \mathbf{A}^T \mathbf{B}$.

The importance of regularization is illustrated in Figure 3: using regularization only slightly increases error on the training poses, but substantially reduces error on the unseen test poses. Unless the system will be used strictly for compression (no novel poses during runtime), using regularization is mandatory in order to allow the model to generalize. An example of the visual artifacts present after applying our linear system on a *novel test pose* with and without regularization is shown in Figure 3 (bottom.) Keep in mind that no such artifacts appear if the system is applied to *trained poses*, with or without regularization; in fact, the numerical accuracy of the model's behavior against trained poses is traded for stability against novel poses (Figure 3, top.) Note that regularization becomes less necessary as the degrees of freedom in the model decrease through the application of dimensionality reduction to inputs and outputs as described in section 5.

5 Dimensionality Analysis and Reduction

The cost of storing the matrices \mathbf{X}^c is determined by the number of joint angles a active in the character, the number of vertices v in the mesh and the number of PRT coefficients approximated (in our case 36.) For typical meshes and character rigs the cost of storing our linear mapping is quite manageable (see Table 2.) However, smaller matrices may be desired to both reduce storage and evaluation costs at runtime. To this end we show that dimensionality reduction is possible on both the inputs and outputs of the linear mapping.

Space of poses: Consider the set of joint angle vectors $\{\mathbf{a}_i\}$ that make up the rows of the \mathbf{A} matrix. Performing principal component analysis (PCA), we note that the variance along each principal direction decreases drastically with increasing dimensions. Figure 4 illustrates this for our training animation. For the example we present, there are 54 angles in each pose vector, however, the first 8 values account for 90% of the variance.

Since the type of natural motion present in character animations exhibits such high correlation in angle values [Safanova et al. 2004; Chai and Hodgins 2005; Arikan 2006; wen Feng et al. 2007] it is possible to project these pose vectors to proportionately few dimensions while incurring little reconstruction error (see Figure 4.)

Space of PRT coefficients: Analogously, consider the set of vectors $\{\mathbf{b}_i\}$ that make up the rows of the \mathbf{B}^c matrices (there is one such vector for each training pose and each PRT coefficient we wish to approximate.) Performing PCA on this set shows a high degree of correlation across vertices regarding associated PRT coefficients, as illustrated in Figure 4. In the case of our training animation with a mesh of approximately 11k vertices, the first 22 values account for 90% of the variance. The high correlation is further validated by the visually pleasing reconstruction results obtained after having projected the data to relatively low dimensions.

At the highest compression rate (22 vertex coefficient eigenvalues), much of the directional shadowing is captured but the shadowing of the right leg onto the left is lacking. At 99% variance (500 vertex coefficient eigenvalues) the leg shadow is captured and at 99.9% variance (1000 vertex coefficient eigenvalues) the results visually converge to the ground truth (although numerically there is still a small error.) The dataset used in Figure 4 is a representative case and we have observed similar behavior in all of our datasets.

Combined Reduced Linear Model: Having selected the number of dimensions desired to represent pose vectors and coefficient vectors across vertices, the pose projection Π_a and per-vertex coefficient projection Π_v can be obtained. Then, we consider the matrices $\mathbf{A}_\pi = \Pi_a(\mathbf{A})$ and $\mathbf{B}_\pi^c = \Pi_v(\mathbf{B}^c)$ and solve for \mathbf{X}^c such that $\mathbf{A}_\pi \mathbf{X}^c = \mathbf{B}_\pi^c$ in the same manner as described above.

Now, given a pose vector \mathbf{a} , the vector \mathbf{b} containing the c^{th} PRT coefficient for all vertices is given by

$$\mathbf{b} = \Pi_v^{-1}(\Pi_a(\mathbf{a})\mathbf{X}^c).$$

This result can be obtained using simple vector-matrix multiplication during runtime.

6 Additional Error Measurements

Our linear model minimizes the sum squared differences between generated and simulated PRT coefficients. However, visual error is also an important measure of our system's accuracy. To visualize the error on a frame-by-frame basis, we integrate the per-vertex

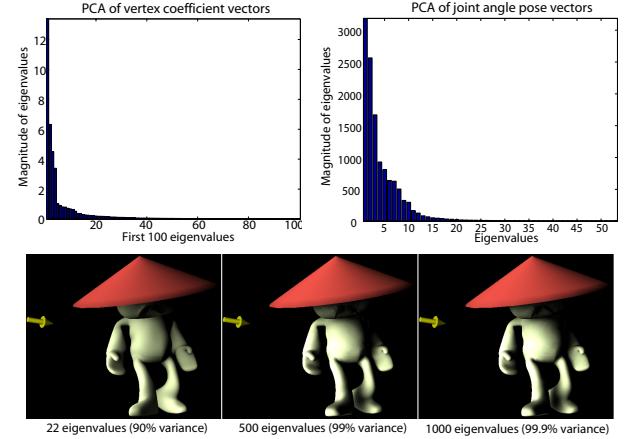


Figure 4: Eigenvalues resulting from PCA of the set of per-vertex coefficient (top left) and angle vectors (top right.) The renderings illustrate the effects of decreasing vertex coefficient bases on the reconstruction of shadows. The number of coefficient bases and percentage of captured variance are listed. Using 1000 vertex coefficient bases yields results that are visually indistinguishable from the ground truth.

squared difference between our generated transfer functions and the simulated results over the visible hemisphere. This transfer error is independent of lighting. At a vertex j with normal \mathbf{n} it is defined as

$$E_{\text{transfer}}^j = \int_{\Omega_n} (T_j^g(\omega) - T_j^s(\omega))^2 d\omega,$$

where Ω_n is the hemispherical cap about the normal, T_j^g and T_j^s are the generated and simulated transfer functions. We also use the lighting dependent shading values for visual feedback. Figure 5 compares our three models with a ground truth rendering and also illustrates scaled transfer differences over the meshes.

7 Results

We tested our data on 7 animation sequences using two rigged character meshes. The Master Pai mesh has 11,534 vertices, 54 joint angles and three animation sequences (1 training sequence at 250 frames and 2 testing sequences at 250 and 450 frames) generated using motion capture data from the CMU motion capture database. The Armadillo mesh has 24,893 vertices, 25 joint angles and four animation sequences (1 training and 3 testing sequences at 250 frames each.)

The two skinned meshes we used exhibit significantly different shadowing patterns: the Armadillo mesh has intricate small-scale shadowing geometry, such as the scales, ears, nose, limbs and digits. The Master Pai mesh has a large hat that can cast wide shadows. We do not alter our model fitting in any way to account for different meshes.

7.1 Model Storage Comparison

The linear model storage requirements depend solely on the size of the mesh and the number of joint angles used in skinning. The number of animation frames used to fit the model only affects training time, not size. Table 2 summarizes the storage costs of our model (in absolute memory, as a percentage of the training dataset size and as a percentage of the size of a single mesh's PRT storage.) The PCA models are identified with the number of retained joint angle dimensions and vertex coefficient angles in a bracketed two-tuple.



Figure 5: Ground truth PRT rendering for a single frame from an animation under distant environmental lighting (**top left**). Results of our full dimensional linear system (**top right**), our reduced model capturing 99% (**bottom left**) and 90% (**bottom right**) of the variance. The shading results of our system are generated on-the-fly for any pose of the character. Inset images show 4X scaled visualization of transfer function differences with the ground truth. Note how shadowing caused by the thumb is progressively smoothed.

Note, for example, that the PCA (8,22) system only uses 61.93% and 62.9% of the storage required for the PRT values of a *single mesh*, yet it is able to reproduce the training data with reasonable accuracy and generate believable lighting effects on novel poses.

In comparison, for meshes of approximately equivalent size, the technique of Kontkanen and Aila requires 3.2MB of storage¹ [2006] where our two compressed systems require 1.04 and 2.22 MBs. Furthermore, Kirk and Arikan [2007] report storage sizes in the same range as our own, however both of these approaches only reproduce ambient occlusion effects. Thus, using approximately half as much data as Kontkanen and Aila [2006] use for generating ambient occlusion, we can generate believable, *full directional* self-shadowing under varying distant illumination with our system. Wen Feng et al. capture directional shadowing and directional reflectance effects at the cost of significantly more storage: they report using 76 and 430 MB of storage for their system, and their runtime reconstruction algorithm is more complex.

7.2 Timing Analysis

Our model fitting, when compared to the amount of time required to generate PRT coefficients for lighting the training poses, incurs a negligible overhead. The linear model fitting takes less than 2 minutes for both our examples, and the PCA model reduction took a handful of minutes for each example. Model fitting and PCA model reduction is performed in MATLAB and 500 spherical sampling directions are used during PRT data generation. Table 1 summarizes the pre-computation times, runtime computational cost and performance of our system. Our system can be used to synthesize indirect illumination data (see Figure 7); PRT data-generation took approx-

imately 3 times as long as direct-illumination data generation and runtime performance was nearly identical to direct-illumination re-lighting performance in these cases.

| Scene | Master Pai | | | Armadillo | | |
|---|--------------------------|----------------------------|----------|--------------------------|----------------------------|----------|
| | P _{8,22} | P _{15,500} | L | P _{8,22} | P _{15,500} | L |
| Runtime Cost ² [megaops/frame] | 0.76 | 17.3 | 1.88 | 1.64 | 37.3 | 1.89 |
| Frame-rate [Hz] | 105 | 34 | 65 | 63 | 24 | 51 |
| PRT Data Generation [sec/frame] | < 25 | | | < 45 | | |
| Model Fitting [sec] | 73 | | | 68 | | |
| PCA [sec] | 324 | | N/A | 349 | | N/A |

Table 1: PRT dataset generation, model fitting, PCA reduction, runtime costs and times for the PCA(8,22) model (**P**_{8,22}), the PCA(15,500) model (**P**_{15,500}), and the full linear model (**L**).

Kirk and Arikan [2007] report precomputation times between 15 and 20 minutes for a mesh of approximately 3 times the size of Armadillo. Kontkanen and Aila [2006] report a precomputation time of a few seconds. James and Fatahalian [2003] report precomputation times of almost 12 hours to 74 hours for more complicated mesh geometry. Wen Feng et al. [2007] report combined precomputation times between 2.2 and 153 hours.

Kirk and Arikan, and Kontkanen and Aila report values that are on par or higher than our reported precomputation times for their three combined meshes. As with our storage requirements, we essentially outperform the previous works using ambient occlusion and the more involved work of Wen Feng et al. takes orders of magnitude more resources than our work. Our runtime performance is slightly higher than the previous ambient occlusion works and, as expected, significantly higher than the work of Wen Feng et al.

We illustrate a variety of results, including indirect illumination effects in Figures 6 and 7. In order to visualize the indirect lighting contribution against the black background, we visualize indirect bounces on top of a white albedo in Figure 7.

We perform all shading calculations and coefficient generation on the CPU every frame with real-time performance on a Pentium 4 3GHz PC. The evaluation of our linear (and reduced linear) system for generating the transfer coefficients each frame can also be accelerated using the GPU; we leave GPU acceleration as an area of future work. Our system is implemented using the DirectX framework.

8 Limitations

We are able to compress and accurately reconstruct the lighting response over an observed animation sequence, as well as generating visually pleasing and consistent lighting on novel poses. The main limitation of this approach is that it requires animation data representative of characteristic motion. Moreover, the learning applies only to the given character and does not generalize to others. However, this is a natural limitation given that the PRT coefficients depend on the character’s geometry.

¹Numbers are adjusted to match our 32-bit floating point accuracy.

²Floating point additions are weighted as 1 op and floating point multiplications as 2.

| | Master Pai (11.5k vertices & 54 joint angles) Trained on a 430 frame animation. | | | Armadillo (24.9k vertices & 25 joint angles) Trained on a 250 frame animation. | | |
|------------------------------------|--|----------------------------|-------------------------------|---|----------------------------|-------------------------------|
| | Memory: Absolute | Memory: % of total data | Memory: % of 1 mesh's data | Memory: Absolute | Memory: % of total data | Memory: % of 1 mesh's data |
| Linear Model (No PCA reduction) | 87.5MB | 12.56% | 5400% | 85.4MB | 10.0% | 2500% |
| PCA (15,500) | 23.6MB | 3.39% | 1457% | 48.6MB | 5.68% | 1421.13% |
| PCA (8,22) | 1.0MB | 0.146% | 62.9% | 2.1MB | 0.248% | 61.93% |

Table 2: Storage requirements for the linear model with and without PCA reduction (see section 7.) Note that the PCA(8,22) system requires significantly less storage than a single mesh's PRT coefficients, yet is able to reproduce believable and consistent directional shadowing for animated mesh sequences.

Our system can only reproduce inter-object effects; shadows from the animated characters onto external receivers can only be modeled in an ad-hoc manner if we bind a receiver object to our articulated character. Since our data is based on diffuse PRT, we currently do not model complex BRDF material properties nor do we handle local lighting effects.

9 Conclusion and Future Work

Our method allows for the accurate, consistent and smooth reproduction of observed and novel low-frequency lighting responses on an articulated character. We observe that lighting response over a mesh is well approximated as a linear function of joint angles and we fit a simple, appropriate model to this data. Dimensionality reduction allows us to compress the data while still maintaining accuracy on observed data and consistent, believable results with novel test poses. Our system can generate PRT coefficients over a mesh using a small fraction of the original data storage requirements and is well suited for interactive applications, such as games.

In future work we will consider modeling more complex material properties and the effects of environmental shadowing. In the direct-lighting scenario, our technique fits a model to the cosine-weighted visibility over mesh vertices. Since visibility is the only component of the direct-lighting rendering equation (see section 3) that varies as a function of articulation/animation, we believe that the method we introduce can be extended to handle arbitrary BRDF shadowed lighting of articulated characters without incurring the high memory requirements or complicated precomputation and runtime algorithms of current techniques.

Acknowledgments

The authors would like to thank Alexis Angelidis for the Master Pai model, Gonzalo Ramos, Paul Debevec for his environment lighting probes and the anonymous reviewers for their helpful comments. The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217. This work is funded in part by an Ontario Graduate Scholarship, a Robert E. Lansdale/Okinoko Graphics Fellowship and MITACS.

References

- ARIKAN, O. 2006. Compression of motion capture databases. In *ACM SIGGRAPH 2006 Papers*.
- CHAI, J., AND HODGINS, J. K. 2005. Performance animation from low-dimensional control signals. *ACM SIGGRAPH 2005 Papers* 24, 3 (August).
- FORBES, K., AND FIUME, E. 2005. An efficient search algorithm for motion data using weighted pca. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*.
- GREEN, P., KAUTZ, J., MATUSIK, W., AND DURAND, F. 2006. View-dependent precomputed light transport using nonlinear

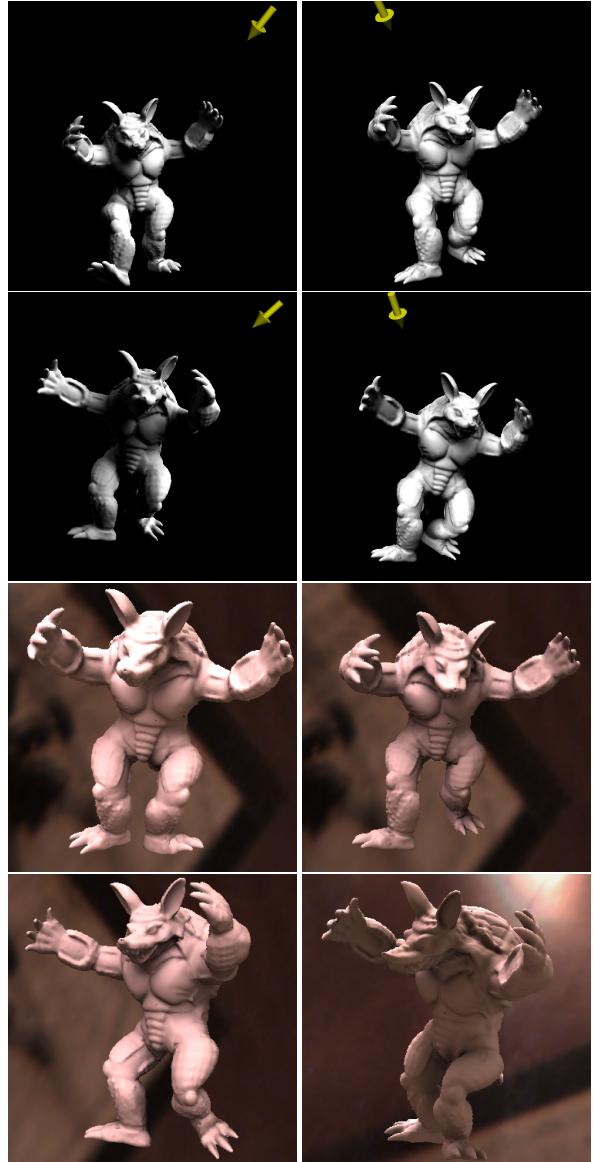


Figure 6: Shading various novel test poses of the armadillo mesh under varying lighting. Our model is able to capture and reproduce realistic directional self-shadowing effects.



Figure 7: Ground truth PRT shading for a three-bounce indirect illumination simulation (**top left**); our system’s shading results (**top right**); ground truth showing only indirect illumination (**bottom left**); our system’s indirect-only results (**bottom right**). The linear model is well suited for capturing the added smoothness of the indirect illumination. In the bottom row, we visualize indirect lighting on top of a white albedo to compensate for the black background.

gaussian function approximations. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*.

JAMES, D. L., AND FATAHALIAN, K. 2003. Precomputing interactive dynamic deformable scenes. In *ACM SIGGRAPH 2003 Papers*.

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*.

KAUTZ, J., LEHTINEN, J., AND AILA, T. 2004. Hemispherical rasterization for self-shadowing of dynamic objects. In *Proceedings of Eurographics Symposium on Rendering 2004*, Eurographics Association.

KIRK, A. G., AND ARIKAN, O. 2007. Real-time ambient occlusion for dynamic character skins. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.

KONTKANEN, J., AND AILA, T. 2006. Ambient occlusion for animated characters. In *Proceedings of Eurographics Symposium on Rendering 2006*, Eurographics Association.

KONTKANEN, J., AND LAINE, S. 2005. Ambient occlusion fields. In *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games*, ACM Press.

LEHTINEN, J., AND KAUTZ, J. 2003. Matrix radiance transfer. In *Proceedings of the 2003 symposium on Interactive 3D graphics*.

MA, W.-C., HSIAO, C.-T., LEE, K.-Y., CHUANG, Y.-Y., AND CHEN, B.-Y. 2006. Real-time triple product relighting using spherical local-frame parameterization. *Vis. Comput.* 22, 9.

MEI, C., SHI, J., AND WU, F. 2004. Rendering with spherical radiance transport maps. In *Computer Graphics Forum*, Eurographics Association.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. In *ACM SIGGRAPH 2003 Papers*.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple product wavelet integrals for all-frequency relighting. In *ACM SIGGRAPH 2004 Papers*.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *ACM SIGGRAPH 2001 Papers*.

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 SIGGRAPH 2006 Papers* 25, 3.

SAFONOV, A., HODGINS, J. K., AND POLLARD, N. S. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM SIGGRAPH 2004 Papers*.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM SIGGRAPH 2002 Papers* 21, 3.

SLOAN, P.-P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. In *ACM SIGGRAPH 2003 Papers*.

SLOAN, P.-P., LUNA, B., AND SNYDER, J. 2005. Local, deformable precomputed radiance transfer. *ACM SIGGRAPH 2005 Papers* 24, 3.

SUN, B., RAMAMOORTHI, R., NARASIMHAN, S. G., AND NAYAR, S. K. 2005. A practical analytic single scattering model for real time rendering. *ACM SIGGRAPH 2005 Papers* 24, 3.

TAMURA, N., JOHAN, H., CHEN, B.-Y., AND NISHITA, T. 2006. A practical and fast rendering algorithm for dynamic scenes using adaptive shadow fields. *Vis. Comput.* 22, 9.

TSAI, Y.-T., AND SHIH, Z.-C. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *ACM SIGGRAPH 2006 Papers*.

WANG, R., TRAN, J., AND LUEBKE, D. 2005. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM SIGGRAPH 2005 Papers* 24, 3.

WANG, J., XU, K., ZHOU, K., LIN, S., HU, S., AND GUO, B. 2006. Spherical harmonics scaling. *Vis. Comput.* 22, 9.

WEN FENG, W., PENG, L., JIA, Y., AND YU, Y. 2007. Large-scale data management for prt-based real-time rendering of dynamically skinned models. In *EGRW ’07: Proceedings of the 17th Eurographics workshop on Rendering*, Eurographics Association, Switzerland.

ZHOU, K., HU, Y., LIN, S., GUO, B., AND SHUM, H.-Y. 2005. Precomputed shadow fields for dynamic scenes. In *ACM SIGGRAPH 2005 Papers*.