Learning Radiance Transfer for Articulated Characters

Derek Nowrouzezahrai   Patricio Simari   Evangelos Kalogerakis   Karan Singh   Eugene Fiume

CSRG-560 Technical Report, May 2007
Department of Computer Science, University of Toronto

Abstract

We present a data-driven technique for generating the precomputed radiance transfer (PRT) 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. More complex transport effects, such as indirect illumination and subsurface scattering, can also be reproduced by our system. Previous data-driven techniques have either restricted the type of lighting response (generating only ambient occlusion) or the type of animated sequences (response functions to external forces.) We provide guidelines for dimensionality reduction of the pose and coefficient spaces, and detailed insight into training procedures. Our model can be fit quickly as a pre-process and run-time transfer vectors are generated with a simple algorithm in real-time. We can reproduce lighting effects on hundreds of trained poses using approximately the same amount of memory required to store a single mesh's PRT coefficients. Moreover, the learned models extrapolate to produce smooth, believable lighting results on novel poses.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism — Color, shading, shadowing, and texture

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 [SKS02]. Low-frequency PRT can incorporate complex light transport effects, such as indirect illumination and subsurface scattering [WTL05, SRNN05], at no extra run-time 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 [SKS02, KSS02]. At the expense of extra storage and reconstruction time, other bases can be used to generate "all-frequency" effects under varying distant illumination [NRH03, NRH04, GKMD06, TS06, MHL*06] or for the incorporation of glossy and specular materials [SKS02, KSS02, LK03, NRH04, TS06].

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 [KLA04, RWS*06]. 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.

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.

Contributions: 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. Our technique allows for real-time PRT rendering of an articulating character under dynamic low-frequency

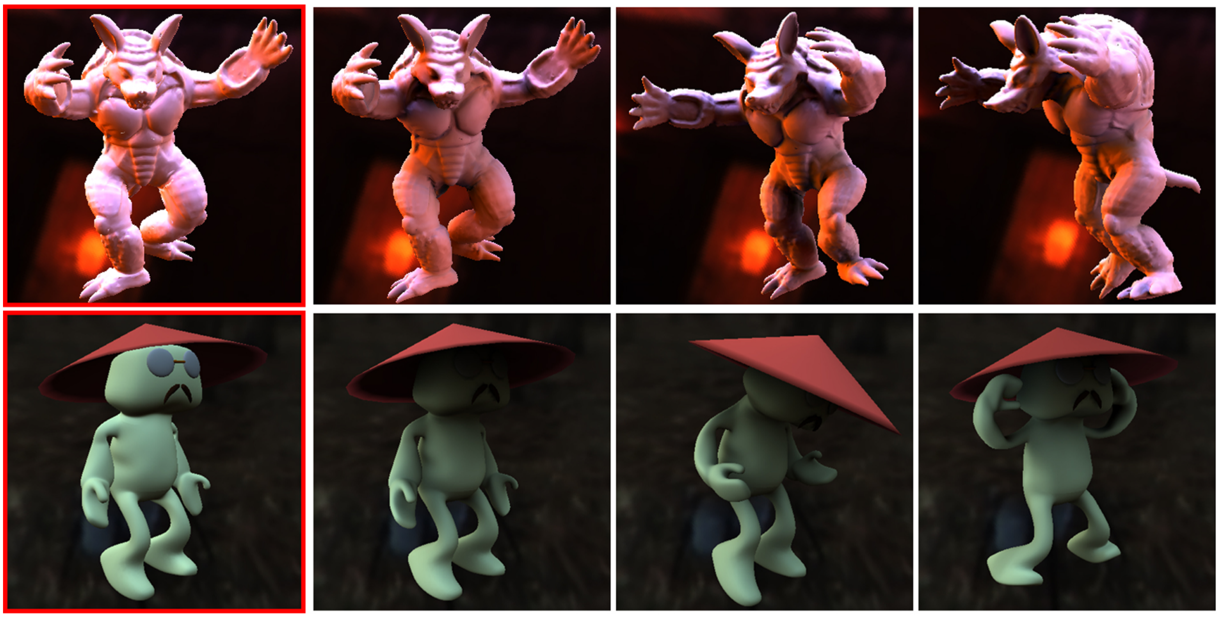

Figure 1: *Our system is able to generate PRT coefficients for a dynamically animated character in real-time and using only a few MBs of memory (see Table 1.) Compared to un-shadowed shading using irradiance environment maps [RH01] (**first column**), our results clearly increase the realism of an animated object.*

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 two-fold: learning requires fewer samples due to the reduction in degrees of freedom, and 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. 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.

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. Three recent works motivate the problem in a similar fashion and experiment with learning lighting response using different models. James and Fatahalian [JF03] 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 motions similar to those learned. Additionally, our system has the capability of being able 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 [KA06] learn a linear model mapping a character's pose to the ambient occlusion values over its vertices. Kirk and Arikan [KA07] learn a multilinear model over a segmented and reduced pose space and can handle larger data sets. 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 data sets while avoiding complicated clustering techniques. Our model assumptions are validated both visually and by the statistics of our results.

2.2. Real-Time Methods for Dynamic Scenes

Kautz *et al.* [KLA04] 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.* [RWS*06] represent dynamic geometry as a hierarchy of spheres and tabulate logarithmic SH visibility vectors as a function of subtended angle. At run-time, 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 do not drastically affect the quality of the results. These geometrical simplifications are not noticeable after shading with low-frequency lighting.

Zhou *et al.* [ZHL*05] 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.* [TJCN06] worked on optimizing the sampling schemes used on the shells of a shadow field, reducing memory consumption. Mei *et al.* [MSW04] 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 [KL05] 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.* [SLS05] 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.

Each work above approximates the lighting of dynamic scenes in a different way. However, it is evident that low storage cost and simple run-time 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 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 run-time 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 run-time reconstruction, Sloan *et al.* [SHHS03] 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 [TS06] use spherical radial basis functions and clustered tensor approximation to represent and compress glossy transfer under all-frequency lighting conditions; a substantial boost in run-time 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.

The analysis and use of reduced dimensional pose subspaces has proven to be an effective technique for compression in data-driven character animation [SHP04, FF05, CH05, Ari06]. We apply similar techniques to analyze pose subspaces in order to reduce the dimensionality of our data set. 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 some of 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 at vertex j with normal $\mathbf{n}$ is

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

where $\Omega_{\mathbf{n}}$ 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 [SKS02]. 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, $\mathbf{L}_{\text{in}}$: $L_{\text{out}}^{j,i} = \mathbf{L}_{\text{in}} \cdot \mathbf{t}^{j,i}$.

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 [SKS02] and subsurface scattering [WTL05]);

we illustrate examples of these effects in the results section. All results use a 6^{th} order SH projection and reconstruction.

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 $\mathbf{B}^c_{i,j}$ 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. Regularization

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

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 usually 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, by minimizing leave-one-out error.) 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. Figure 2 plots the distribution and empirical cumulative distribution of the signed residuals of the regularized linear fitting. Note the single mode, zero mean and median and low variance that further validate the linear model.

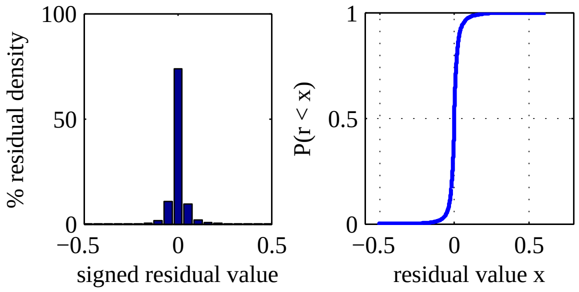

Figure 2: ***Left:*** *Distribution of signed residuals of regularized linear fit:* $\mu = 5.1 \times 10^{-5}$, $\sigma^2 = 3.25 \times 10^{-4}$. ***Right:*** *Empirical cumulative distribution of residuals.*

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 1). 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. There are 53 angles in each pose vector, however, the first 8 values account for 90% of the variance.

The fact that the type of natural motion present in character animations exhibits such high correlation in angle values [SHP04, CH05, Ari06] makes it possible to project these pose vectors to proportionately few dimensions while incurring little reconstruction error (once again, see Figure 4.)

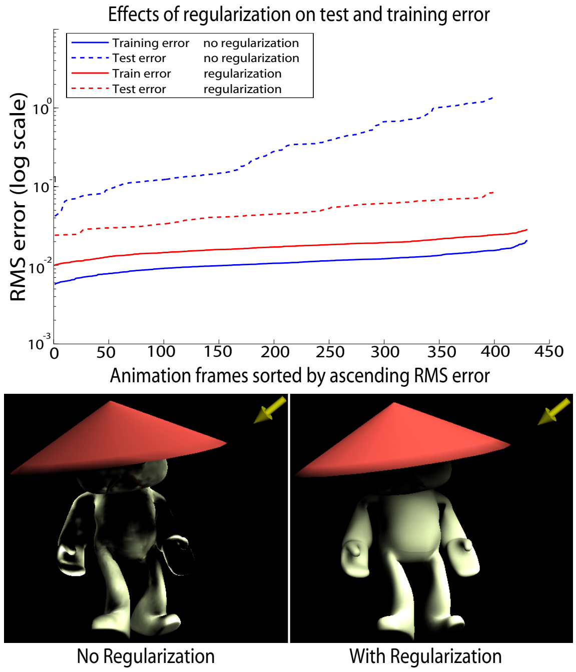

Figure 3: *Top: The training and test errors of the regularized and unregularized linear systems.* ***Bottom:*** *Approximation from linear mapping on a novel test pose without (**left**) and with (**right**) regularization. For our results we use* $\alpha = 231$. *Arrows denote a distant delta lighting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

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.) Once again 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.

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 $\mathbf{B}^c_\pi = \Pi_v(\mathbf{B}^c)$ and solve for $\mathbf{X^c}$ such that $\mathbf{A}_\pi \mathbf{X}^c = \mathbf{B}^c_\pi$ 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 run-time.

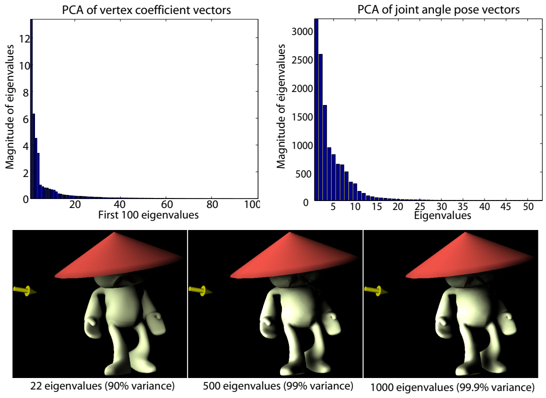

Figure 4: *Eigenvalues resulting from PCA of the set of per-vertex coefficient (**top left**) and angle vectors (**top right**). The renderings illustrate the effects of decreasing vertex coefficient bases on the reconstruction of shadows. The number of coefficient bases and percentage of captured variance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

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

$$E^{j}_{\text{transfer}} = \int_{\Omega_{\mathbf{n}}} \left(T^g_j(\omega) - T^s_j(\omega)\right)^2 d\omega\,,$$

where $\Omega_{\mathbf{n}}$ is the hemispherical cap about the normal, T^g_j and T^s_j 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 4 animation sequences using two rigged character meshes. The Master Pai mesh has 11,534 vertices, 54 joint angles and 430 frame animation sequences (1 training and 1 testing) generated using motion capture data. The Armadillo mesh has 24,893 vertices, 25 joint angles and 250 frame animation sequences (1 training and 1 testing.) PRT simulations on each of the Master Pai animations were performed using 500 sampling directions per vertex and took less than 25 seconds per frame for direct-illumination simulation; three-bounce indirect simulations took approximately 75 seconds per frame. Each Armadillo frame took around 45 seconds to precompute for direct-only

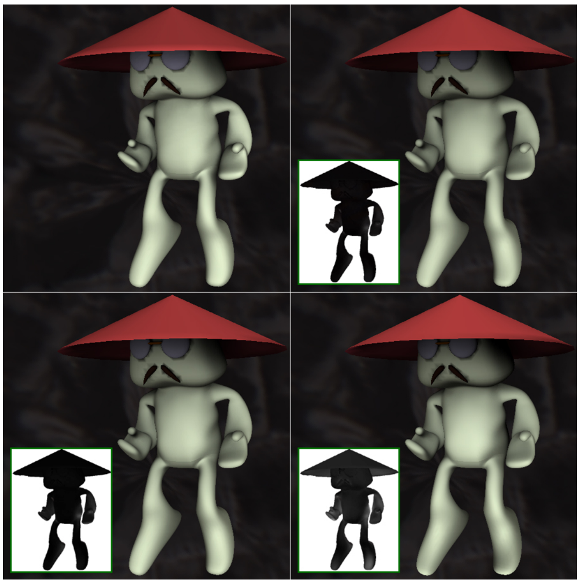

Figure 5: *Ground truth PRT rendering for a single frame from an animation under distant environmental lighting (**top left**). Results of our full dimensional linear system (**top right**), our reduced model capturing 99% (**bottom left**) and 90% (**bottom right**) of the variance. The shading results 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. Shadowing caused by the thumb is progressively smoothed.*

simulation. Running time for training is approximately 1 minute without counting PCA, which takes approximately 5 minutes in MATLAB. We illustrate a variety of results, including indirect-illumination and subsurface scattering effects in Figures 6, 7 and 8.

We learn our models using only the two training animation sequences; the rest of the sequences are simulated for error comparison against our generated results. Table 1 summarizes the model's storage requirements (in absolute memory, as a percentage of the training data set size (PTD) and as a percentage of the size of a single mesh's PRT storage (PSM).) All values in our system are stored as 32-bit floats.

Note, for example, that the PCA (8,22) system only uses 61.93% 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 believable lighting effects on novel poses of the Armadillo mesh. For meshes of approximately equivalent size, the technique of Kontkanen and Aila requires 1.6MB of storage (with 16-bit floats) [KA06] where our two compressed systems require 1.04 and 2.22 MBs (again, using twice as many bits for floats as in [KA06].) Thus, using approximately half as much data as [KA06] use for generating ambient occlusion, we can generate believable, *full directional* shading with our system. We should also note that many current low-frequency PRT techniques only use 4^{th} order SH expansions, where we use 6^{th} order. We can reduce the storage requirements of our system significantly if we were to only use a 4^{th} order SH expansion.

We perform all shading calculations and coefficient generation on the CPU every frame with real-time performance, often greater than 80 FPS, on a Pentium 4 3GHz PC. We leave GPU acceleration as an area of future work; the evaluation of our linear (and reduced linear) system for generating the transfer coefficients each frame can also be accelerated using the GPU. 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.

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 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.

In future work we will consider modeling more complex material properties and the effects of environmental shadowing. We may be able to leverage previous work in compression of PRT datasets for glossy rendering [SHHS03, TS06] to offset the increased memory requirements necessary for incorporating more complex materials into PRT datasets.

10. Acknowledgements

We would like to thank Alexis Angelidis for the Master Pai mesh.

	Master Pai			Armadillo		
	Memory (Abs.)	Memory (PTD)	Memory (PSM)	Memory (Abs.)	Memory (PTD)	Memory (PSM)
No 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 1: *Storage requirements for the linear model with and without reduction.*

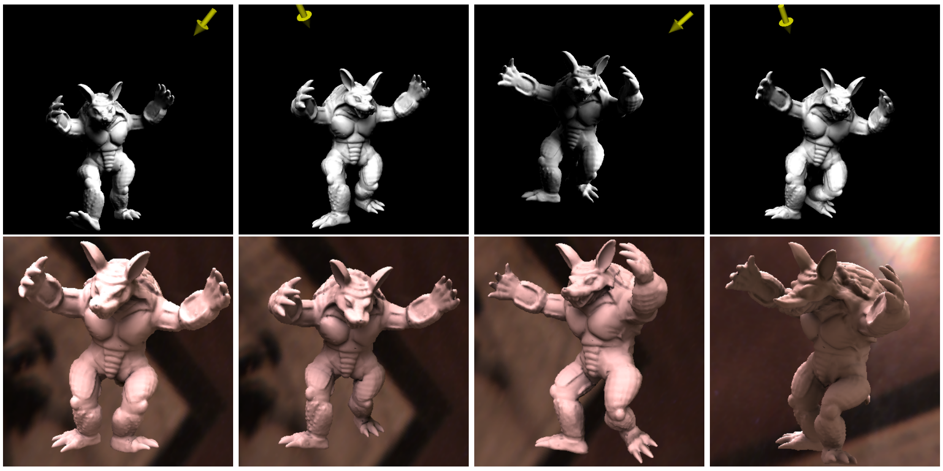

Figure 6: *Shading various novel test poses of the armadillo mesh under varying lighting.*

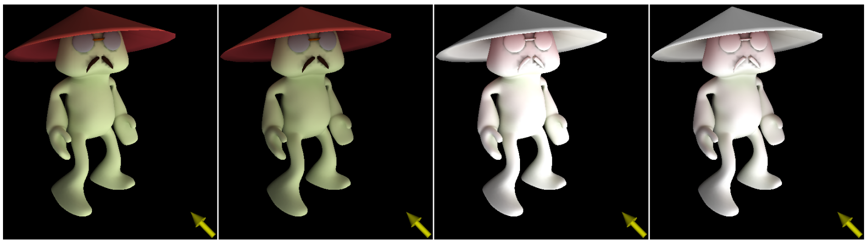

Figure 7: ***Left to right:** Ground truth PRT shading for a three-bounce indirect illumination simulation; our system's shading results; ground truth showing only indirect illumination; our system's indirect-only results. The linear model is well suited for capturing the added smoothness of the indirect illumination.*

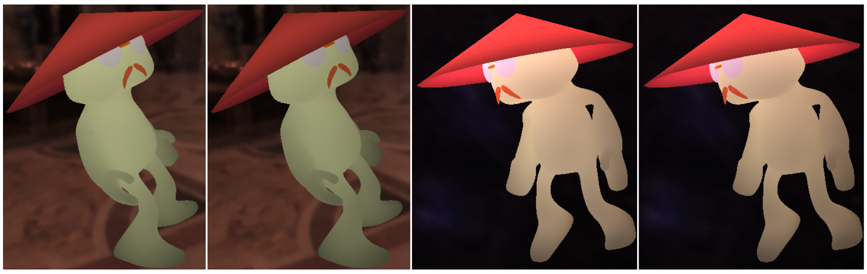

Figure 8: ***Left to right:** A pair of PRT subsurface scattering results with ground truth shading followed by the results of our system. As with indirect illumination, subsurface scattering smooths the shading and this smoothing is captured by our model.*

References

[Ari06] ARIKAN O.: Compression of motion capture databases. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 890–897.

[CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. *ACM Transactions on Graphics (SIGGRAPH 2005) 24*, 3 (August 2005).

[FF05] FORBES K., FIUME E.: An efficient search algorithm for motion data using weighted pca. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 67–76.

[GKMD06] GREEN P., KAUTZ J., MATUSIK W., DURAND F.: View-dependent precomputed light transport using nonlinear gaussian function approximations. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), ACM Press, pp. 7–14.

[JF03] JAMES D. L., FATAHALIAN K.: Precomputing interactive dynamic deformable scenes. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 879–887.

[KA06] KONTKANEN J., AILA T.: Ambient occlusion for animated characters. In *Proceedings of Eurographics Symposium on Rendering 2006* (2006), Eurographics Association.

[KA07] KIRK A. G., ARIKAN O.: Real-time ambient occlusion for dynamic character skins. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (April 2007).

[KL05] KONTKANEN J., LAINE S.: Ambient occlusion fields. In *Proceedings of ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games* (2005), ACM Press, pp. 41–48.

[KLA04] KAUTZ J., LEHTINEN J., AILA T.: Hemispherical rasterization for self-shadowing of dynamic objects. In *Proceedings of Eurographics Symposium on Rendering 2004* (2004), Eurographics Association, pp. 179–184.

[KSS02] KAUTZ J., SLOAN P.-P., SNYDER J.: Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2002), Eurographics Association, pp. 291–296.

[LK03] LEHTINEN J., KAUTZ J.: Matrix radiance transfer. In *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics* (New York, NY, USA, 2003), ACM Press, pp. 59–64.

[MHL*06] MA W.-C., HSIAO C.-T., LEE K.-Y., CHUANG Y.-Y., CHEN B.-Y.: Real-time triple product relighting using spherical local-frame parameterization. *Vis. Comput. 22*, 9 (2006), 682–692.

[MSW04] MEI C., SHI J., WU F.: Rendering with spherical radiance transport maps. In *Computer Graphics Forum* (2004), Eurographics Association.

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 376–381.

[NRH04] NG R., RAMAMOORTHI R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM Press, pp. 477–487.

[RH01] RAMAMOORTHI R., HANRAHAN P.: An efficient representation for irradiance environment maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 497–500.

[RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph. 25*, 3 (2006), 977–986.

[SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 382–391.

[SHP04] SAFONOVA A., HODGINS J. K., POLLARD N. S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM Press, pp. 514–521.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. 21*, 3 (2002), 527–536.

[SLS05] SLOAN P.-P., LUNA B., SNYDER J.: Local, deformable precomputed radiance transfer. *ACM Trans. Graph. 24*, 3 (2005), 1216–1224.

[SRNN05] SUN B., RAMAMOORTHI R., NARASIMHAN S. G., NAYAR S. K.: A practical analytic single scattering model for real time rendering. *ACM Trans. Graph. 24*, 3 (2005), 1040–1049.

[TJCN06] TAMURA N., JOHAN H., CHEN B.-Y., NISHITA T.: A practical and fast rendering algorithm for dynamic scenes using adaptive shadow fields. *Vis. Comput. 22*, 9 (2006), 702–712.

[TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *SIGGRAPH*

'06: ACM SIGGRAPH 2006 Papers (New York, NY, USA, 2006), ACM Press, pp. 967–976.

[WTL05] WANG R., TRAN J., LUEBKE D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph. 24*, 3 (2005), 1202–1207.

[ZHL*05] ZHOU K., HU Y., LIN S., GUO B., SHUM H.-Y.: Precomputed shadow fields for dynamic scenes. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM Press, pp. 1196–1201.