

Rui Wang  
Ewen Cheslack-Postava  
Rui Wang  
David Luebke  
Qianyong Chen  
Wei Hua  
Qunsheng Peng  
Hujun Bao

## Real-time editing and relighting of homogeneous translucent materials

Published online: 30 May 2008  
© Springer-Verlag 2008

**Electronic supplementary material**  
The online version of this article  
(doi:10.1007/s00371-008-0237-9) contains  
supplementary material, which is available  
to authorized users.

R. Wang (✉) · Q. Chen  
UMass Amherst,  
Amherst, MA 01003, USA  
ruiwang@cs.umass.edu

R. Wang · W. Hua · Q. Peng · H. Bao  
Zhejiang University,  
Hangzhou 310027, P.R. China  
rwang@cad.zju.edu.cn

E. Cheslack-Postava  
Stanford University,  
Stanford, CA 94305, USA  
ewencp@cs.stanford.edu

D. Luebke  
NVIDIA Corporation, Santa Clara,  
CA 95050, USA  
dave@luebke.us

**Abstract** Existing techniques for fast, high-quality rendering of translucent materials often fix BSSRDF parameters at precomputation time. We present a novel method for accurate rendering and relighting of translucent materials that also enables real-time editing and manipulation of homogeneous diffuse BSSRDFs. We first apply PCA analysis on diffuse multiple scattering to derive a compact basis set, consisting of only twelve 1D functions. We discovered that this small basis set is accurate enough to approximate a general diffuse scattering profile. For each basis, we then precompute light transport data representing the translucent transfer from a set of local illumination samples to each rendered vertex. This local transfer model allows our system to integrate a variety of lighting models in a single framework, including environment lighting, local area

lights, and point lights. To reduce the PRT data size, we compress both the illumination and spatial dimensions using efficient nonlinear wavelets. To edit material properties in real-time, a user-defined diffuse BSSRDF is dynamically projected onto our precomputed basis set, and is then multiplied with the translucent transfer information on the fly. Using our system, we demonstrate realistic, real-time translucent material editing and relighting effects under a variety of complex, dynamic lighting scenarios.

**Keywords** BSSRDF · Precomputed radiance transfer · PCA · Haar wavelets · Spatial compression

### 1 Introduction

Fast and accurate simulation of translucent appearance has become an increasingly important topic in many graphics applications. At different scale levels, most real-world materials exhibit translucency, caused by subsurface scattering of light. Early research in this field focused on developing physically based subsurface scattering models using illumination transport equations for participating media [7, 11]. These models have been widely used in today's photorealistic visualization and cinematic render-

ing systems. Unfortunately, the translucent appearance design process has typically been labor intensive and time consuming, primarily because the high computational cost demanded by these models prevents artists from receiving interactive simulation feedback, even under simplified lighting conditions.

We present a novel technique for accurate rendering and relighting of translucent materials that also enables real-time editing and manipulation of material properties. Our system provides users with real-time feedback on how changes to translucent material parameters would affect



**Fig. 1.** These images demonstrate translucent material editing and are generated by our system in real-time

the appearance of final quality rendering. We assume static scene models and focus on diffuse multiple scattering effects in homogeneous translucent media using the dipole approximation model introduced by Jensen et al. [12]. Figure 1 demonstrates our real-time rendering results with several different materials.

Our system builds on previous research in precomputed radiance transfer (PRT) [16, 19]. PRT precomputes illumination data offline, then applies basis projection methods such as spherical harmonics or wavelets to reduce the datasets, which can then be used at run-time to permit interactive rendering. Early PRT work fixes BRDFs at pre-computation time; recently, several real-time BRDF editing systems have been presented [1, 21] that enable interactive editing of arbitrary BRDFs under complex lighting. While allowing dynamic material changes, they usually require fixing the lighting and viewpoint, or perform at subinteractive rates due to the large data size. Furthermore, they all focus on opaque materials while translucent material editing has received relatively little attention.

We address the translucent material editing problem by incorporating material samples at precomputation time using a data-driven basis set. We focus on diffuse multiple scattering and adopt Jensen et al.’s BSSRDF model described in [12]. We begin by performing principal component analysis (PCA) on this scattering model and derive a compact basis set, consisting of only twelve 1D functions. We discovered that this small basis set is accurate enough to represent a general diffuse scattering profile. We then use PRT to precompute for each scattering basis the translucent transfer from a set of surface illumination samples to every rendered vertex. To efficiently remove data redundancy, we exploit the large spatial coherence present in the precomputed data by using spatial nonlinear wavelet compression. Finally, the precomputation results are linearly combined at run-time to simulate realistic rendering under dynamic BSSRDFs. Our work makes the following two contributions:

- 1) *PCA analysis of diffuse multiple scattering.* We are the first to perform a data-driven analysis on the diffuse multiple scattering model [12], and use PCA to derive a compact basis set.

- 2) *Nonlinear wavelet spatial compression.* We apply nonlinear wavelets to compress the spatial data, achieving superior reconstruction performance. Furthermore, we use the same algorithm on visibility data compression, permitting fast environment lighting with little overhead.

## 2 Related work

*Real-time material editing.* Opaque materials are often described by parametric BRDF functions, and can be easily edited in real-time using standard GPU rendering assuming simple lighting models. A work by [3] developed an intuitive and general BRDF editing system called *BRDF-Shop*. The real-time BRDF editing systems built by Ben-Artzi et al. [1] allow for interactive editing of arbitrary BRDFs under complex environment lighting. However, their system requires fixed lighting and viewpoint. A related system by Sun et al. [21] achieves the same goal but performs at subinteractive rates and computes only a limited range of BRDF samples.

Building a real-time editing system for translucent materials presents a greater challenge, as the simulation involves integrating illumination contributions from the scene model. Several existing techniques [4, 8, 13, 15, 18, 24] render diffuse multiple scattering in real-time, but are either restricted to simple geometry and lighting, or require fixing BSSRDF parameters. A recent work by [5] applies sums of Gaussians for a multi-layered skin model, but does not yet support real-time editing of BSSRDF parameters. Our work is closely related to the work by Xu et al. [25]. However, while they apply a preselected polynomial basis to reduce the translucent transfer function, we perform data-driven reduction using a PCA-derived basis.

*Precomputed radiance transfer.* Our system builds on PRT techniques [16, 19] that precompute light transport effects and compress them using projection onto a basis such as spherical harmonics (SH) or wavelets. With the compressed dataset, the rendering computation at run-

time is significantly reduced and performed at real-time rates. Subsequent work extended PRT to handle changing viewpoint [14, 23], local lighting [26], and deformable models [20]. Wang et al. [24] simulate all-frequency translucency effects including both single and diffuse multiple scattering, but require fixed material parameters.

Our technique is similar in spirit to incorporating surface BRDFs in PRT, using a preselected or derived material basis. However, run-time projection of an arbitrary BRDF onto a basis set is very costly to compute, even for low-order approximations [17]. In contrast, we derive a compact BSSRDF basis set that consists of only twelve 1D functions, thus the editing and projection of BSSRDFs can both perform at real-time rates.

*Spatial compression of PRT.* Previous work in spatial compression schemes for PRT include clustered PCA (CPCA) [14, 18], clustered tensor approximation (CTA) [22], and 4D wavelets [2]. PCA-based approaches are usually very slow to precompute and prone to numerical stability issues. CPCA and CTA reduce the computational cost by partitioning, but require nontrivial algorithms to maximize the partition coherence and reduce boundary artifacts.

Our system uses a simple spatial compression scheme that projects our transport data onto nonlinear wavelets defined in a parameterized spatial domain. Due to the speed and efficiency of the wavelet transform, we achieve fast precomputation at less than 2 h for all test scenes, and 3–4× speed up in relighting performance compared with no spatial compression. In addition, our scheme does not require data partitioning, thus is not subject to compression boundary artifacts.

### 3 Algorithms and implementation

*Review of BSSRDF.* We use the diffuse BSSRDF model introduced by Jensen et al. [12], which assumes multiple scattering within a homogeneous medium. The material properties of such a medium are described by the absorption coefficient  $\sigma_a$ , the scattering coefficient  $\sigma_s$ , and the extinction coefficient  $\sigma_t = \sigma_a + \sigma_s$ . Under spatially varying lighting  $L(x_i, \omega_i)$ , the scattered radiance is computed by:

$$B(x, \omega_o) = \iint S(x_i, \omega_i; x, \omega_o) L(x_i, \omega_i) (\omega_o \cdot n) d\omega_i dA \quad (1)$$

where  $\omega_i$  and  $\omega_o$  are the incident and viewing directions,  $B$  is the scattered radiance at point  $x$  in direction  $\omega_o$ ,  $L$  is the incident radiance at point  $x_i$  in direction  $\omega_i$ ,  $n$  is the surface normal at point  $x$ , and  $S$  is the BSSRDF, defined as the sum of a single scattering term  $S^{(1)}$  and a diffuse multiple scattering term  $S_d$ . The diffuse assumption comes

from the fact that multiple scattering tends to become isotropic due to the large number of scattering events. This is especially true for highly scattering materials such as marble. For these materials, the multiple scattering component also dominates the total scattered radiance [10]. Therefore the single scattering part  $S^{(1)}$  can be ignored; and the multiple scattering  $S_d$  can be further defined as:

$$S_d(x_i, \omega_i; x, \omega_o) = \frac{1}{\pi} F_t(\eta, \omega_i) R_d(\|x_i - x\|) F_t(\eta, \omega_o) \quad (2)$$

where  $F_t$  is the Fresnel transmittance at the interface,  $\eta$  is the relative index of refraction, and  $R_d$  is the diffuse reflectance function approximated using a dipole source model as:

$$R_d(r) = \frac{\alpha'}{4\pi} \left[ z_r \left( \sigma_{tr} + \frac{1}{d_r} \right) \frac{e^{-\sigma_{tr}d_r}}{d_r^2} + z_v \left( \sigma_{tr} + \frac{1}{d_v} \right) \frac{e^{-\sigma_{tr}d_v}}{d_v^2} \right],$$

where  $\sigma'_s = (1 - g)\sigma_s$  and  $\sigma'_t = \sigma_a + \sigma'_s$  are the reduced scattering and extinction coefficients;  $\alpha' = \sigma'_s/\sigma'_t$  is the reduced albedo;  $g$  is the mean cosine of the scattering angle;  $\sigma_{tr} = \sqrt{3\sigma_a\sigma'_t}$  is the effective extinction coefficient;  $d_r = \sqrt{r^2 + z_r^2}$  and  $d_v = \sqrt{r^2 + z_v^2}$  are the distances from an illumination point  $x_i$  to the dipole source, with  $r = \|x - x_i\|$  being the distance between  $x_i$  and  $x$ , and  $z_r = 1/\sigma'_t$  and  $z_v = z_r(1 + 4A/3)$  being the distances from  $x$  to the dipole source. Here  $A = (1 + F_{dr})/(1 - F_{dr})$ , and  $F_{dr}$  is a diffuse Fresnel term approximated by  $F_{dr} = -1.440/\eta^2 + 0.710/\eta + 0.668 + 0.0636\eta$ . For convenience, we denote  $R_d$  in general as

$$R_d(r, \sigma) = R_d(r, \sigma'_s, \sigma_a), \quad (3)$$

where  $\sigma$  represents the collection of all independent BSSRDF parameters.

Equation 1 is expensive to compute as it involves integrating the light transport from all illumination samples. To accelerate this computation, Jensen et al. [10] used a two-pass hierarchical integration approach. The key idea is to compute incident irradiance in a separated pass, thus making it possible to reuse illumination samples.

#### 3.1 A PCA basis set for $R_d$

Note that in Eq. 2 the distance  $r = \|x - x_i\|$  is a 1D variable and is a purely geometric term that can be precomputed assuming static scene models; all other parameters  $\sigma$  are dynamic and cannot be precomputed. With a naive approach, we could attempt to sample all possible BSSRDF parameters at precomputation time, but that would lead to an impractical amount of precomputed data. Instead, our goal is to seek a lower dimensional linear subspace for  $R_d$ , and approximate it using a linear sum

of basis functions:

$$R_d(r, \sigma) \approx \sum_{k=1}^K b_k(r) s_k(\sigma). \quad (4)$$

Here  $b_k(r)$  is a set of orthogonal basis functions, each one of which is a 1D function of  $r$ , and  $K$  is the total number of approximation terms. The basis functions may be preselected, but they can also be derived from standard factorization techniques such as PCA. To do so, we sample and discretize  $R_d$  into a matrix  $\mathbf{R}_d$ , then apply a singular value decomposition (SVD):  $\mathbf{R}_d = \mathbf{U}\mathbf{S}\mathbf{V}'$  which results in an orthogonal left matrix  $\mathbf{U}$ , a corresponding right matrix  $\mathbf{V}$ , and a diagonal matrix  $\mathbf{S}$  that represents the singular values in order of importance. The column vectors of the left matrix  $\mathbf{U}$  naturally form an orthogonal basis set  $b_k(r)$ , and the column vectors of the right matrix  $\mathbf{V}$  represent the projection coefficients of all sampled BSSRDFs.

The idea of seeking a linear basis for  $R_d$  is not new. Several other works have presented the same efforts using preselected basis functions, such as polynomials or Gaussian functions [5, 25]. By using a data-driven approach, however, our method adaptively computes the optimal common basis set for all sampled BSSRDFs in a least squares sense. In Sect. 4, we show that the first 12 PCA basis functions suffice to accurately represent a general diffuse BSSRDF profile. The small size of the basis set allows us to dynamically project any user-specified BSSRDF onto the basis in real-time. Therefore it eliminates the need to store the right matrix  $\mathbf{V}$  of the SVD result, as the basis coefficients  $s_k(\sigma)$  of the actual BSSRDF are being evaluated dynamically on the fly.

*Implementation notes.* To construct the sampled material matrix  $\mathbf{R}_d$ , we discretize both  $r$  and  $\sigma$  parameters within reasonable ranges. For the distance parameter  $r$ , we found through experiments that 650 uniform samples within the range [0.0, 65.0] is sufficient to produce high-quality results. Experiments with more samples or a wider range did not produce significant improvements in our tests. The  $\sigma$  parameters are sampled within a range of [0.0, 5.0] as densely as memory constraints permit. These ranges are representative of values found in measured datasets [12].

### 3.2 Precomputed multiple scattering transfer

We precompute multiple scattering light transport by incorporating the BSSRDF basis derived from above. This step is similar to how a BRDF basis is incorporated in PRT [14, 23]. Substituting Eq. 4 into Eqs. 2 and 1, the rendering equation becomes:

$$B(x, \omega_o) = \frac{1}{\pi} F_t(\omega_o) \sum_k s_k(\sigma) \int_A E(x_i) \cdot b_k(r) dA, \quad (5)$$

where  $E(x_i)$  represents the total incident irradiance received at an illumination point  $x_i$ :

$$E(x_i) = \int_{2\pi} L(x_i, \omega_i) F_t(\omega_i) (\mathbf{n}_i \cdot \omega_i) d\omega_i. \quad (6)$$

This can be computed via any available illumination model, such as GPU-based shadow mapping or PRT-based environment lighting.

For convenience, we express Eq. 5 using matrix form:

$$\mathbf{B} = \sum_k s_k(\sigma) (\mathbf{T}_k \cdot \mathbf{E}) = \sum_k s_k(\sigma) \mathbf{v}_k \quad (7)$$

where  $\mathbf{T}_k$  is a precomputed transport matrix representing  $b_k(\|x - x_i\|)$ , the transfer from  $x_i$  to  $x$  projected onto the  $k$ -th BSSRDF basis,  $\mathbf{E}$  is the illumination vector, and  $\mathbf{v}_k = \mathbf{T}_k \cdot \mathbf{E}$ . The factor  $\frac{1}{\pi} F_t(\omega_o)$  is omitted as it is easily computed in a shader. The matrix-vector multiplication is expensive to compute but can be accelerated by projecting both  $\mathbf{E}$  and the rows of  $\mathbf{T}_k$  onto wavelets:

$$\mathbf{v}_k = (\mathbf{T}_k \cdot w') \cdot (w \cdot \mathbf{E}) = \mathbf{T}_k^w \cdot \mathbf{E}^w \quad (8)$$

where  $w$  is an orthonormal matrix representing wavelet projection, and  $w' \cdot w = \mathbf{I}$ ;  $\mathbf{T}_k^w$  and  $\mathbf{E}^w$  represent the wavelet-transformed transfer matrix and illumination vector. Efficient nonlinear compression of  $\mathbf{T}_k^w$  is achieved by culling insignificant coefficients based on their magnitude. Note that the rows of  $\mathbf{T}_k^w$  represent the illumination transfer dimension, while the columns represent the spatial dimension. Therefore this step applies nonlinear wavelet compression to the transfer dimension (transfer compression), but leaves the spatial dimension intact.

We observe that after the transfer compression, our data still contain considerable coherence in the spatial domain (i.e., vertices that are close to each other have similar data). We therefore apply a further wavelet projection in the spatial domain of  $\mathbf{T}_k^w$ , resulting in:

$$\mathbf{v}_k = w' \cdot (w \cdot \mathbf{T}_k \cdot w') \cdot (w \cdot \mathbf{E}) = w' \cdot (^w \mathbf{T}_k^w \cdot \mathbf{E}^w). \quad (9)$$

Using the linearity of wavelets, Eq. 9 can be combined with Eq. 7, giving:

$$\mathbf{B} = w' \cdot \sum_k s_k(\sigma) (^w \mathbf{T}_k^w \cdot \mathbf{E}^w) \quad (10)$$

where  $^w \mathbf{T}_k^w$  represents the precomputed transfer matrix that has been doubly projected onto wavelets in both the transfer and spatial dimensions. The leftmost multiplication by  $w'$  represents the spatial reconstruction of the color, and is computed by a single inverse wavelet transform as the last step of rendering.

*Transfer compression.* We use a set of uniformly distributed surface sample points as illumination points  $x_i$ . In

order to apply 2D wavelets, we must parameterize  $x_i$  onto a square 2D domain. We create this parameterization using the tree-based hierachal construction scheme presented by Hašan et al. [9]. To do so, we first compute  $m \times 2^n \times 2^n$  uniform samples around the scene, where both  $m$  and  $n$  are integers; we then apply a spatial partition algorithm to split these unstructured points into  $m$  parts. We build a balanced quadtree for each partition using the same method as Hašan et al. and flatten the tree. This maps each partition to a square texture that also maintains good spatial coherence for efficient wavelet approximation.

*Spatial compression.* Due to the size of our precomputed data, a straightforward approach to directly perform the double projection, taken by Cheslack-Postava et al. [2], is not feasible as it requires storage of the full matrix. Instead, we apply nonlinear approximation on the transfer dimension first, then apply the second wavelet projection using compressed row vectors. This two-pass approach has been exploited previously in [24] to significantly reduce the precomputation time while preserving accurate all-frequency effects. Note that the spatial domain wavelet projection requires a parameterized model, such as geometry images [6]. Alternatively, for small to medium size models ( $\sim 20\,000$  vertices), we can also omit the spatial compression step (thus no parametrization is required) and still maintain interactive relighting and editing speed.

*Implementation notes.* To summarize, we perform precomputation in two passes. The first pass loops over each vertex, uses each BSSRDF basis to compute the

transport matrix  $T_k$ , and performs a nonlinear wavelet approximation on  $T_k$ . We maintain a relatively large number of wavelet terms in this stage to retain high accuracy for the second pass of spatial compression. During the second pass, we loop over each BSSRDF basis, read in the compressed data from the first pass, and perform a spatial wavelet transform step on each basis slice. As Haar wavelets generate objectionable spatial reconstruction artifacts, we instead use the smoother CDF 9/7 wavelet. For nonlinear compression, we adaptively choose the transformed vectors with the highest energy in  $L^2$  norm, until a user-specified energy threshold is met. The chosen transport vectors are then truncated to their final size according to a targeted data size, and quantized to 16-bit integers with 16-bit indices. When the number of illumination points exceeds the index range ( $2^{16}$ ), we instead use 14 data bits with 18 index bits.

Much of first pass can be trivially parallelized, so we utilize multiple CPU cores. Furthermore this step requires very little memory beyond an array to hold the current transport vector. The second step could also be parallelized, but would be much more challenging as the memory consumption is substantially higher – each transport matrix  $T_k$  (corresponding to the BSSRDF basis) must fit entirely in memory to be further processed.

### 3.3 Relighting and BSSRDF editing

Relighting from our precomputed data is similar to other PRT systems, with additional steps for editing BSSRDF parameters. Our relighting algorithm is illustrated in Fig. 2 and summarized as follows:

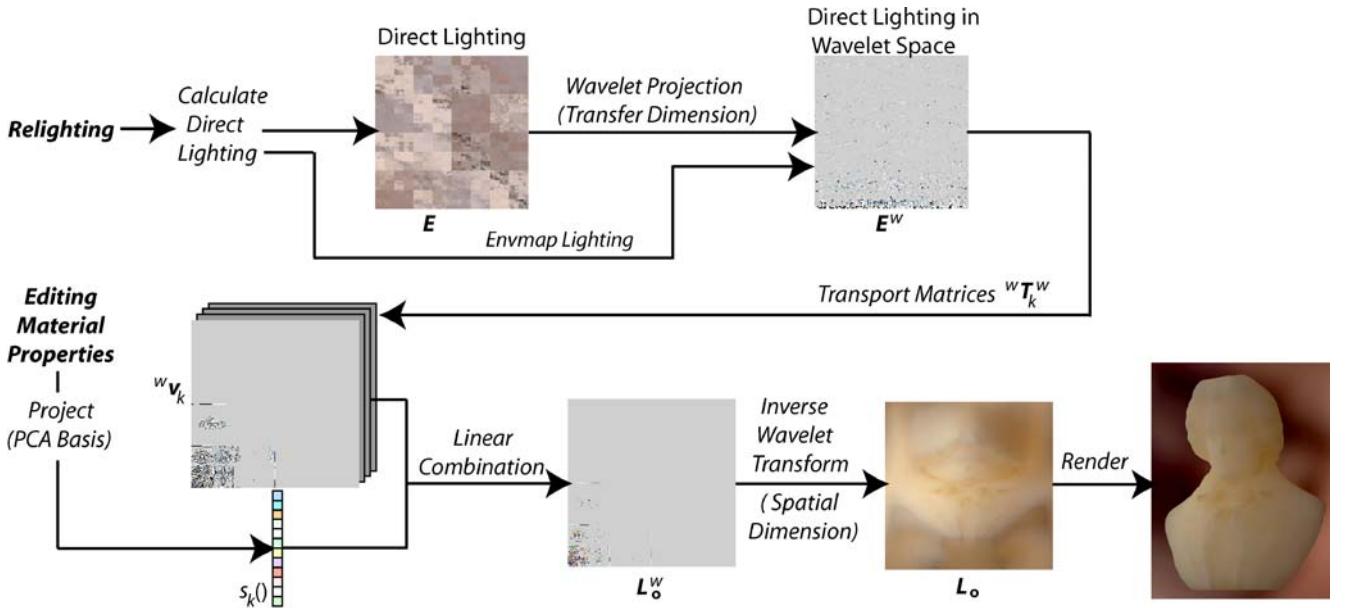
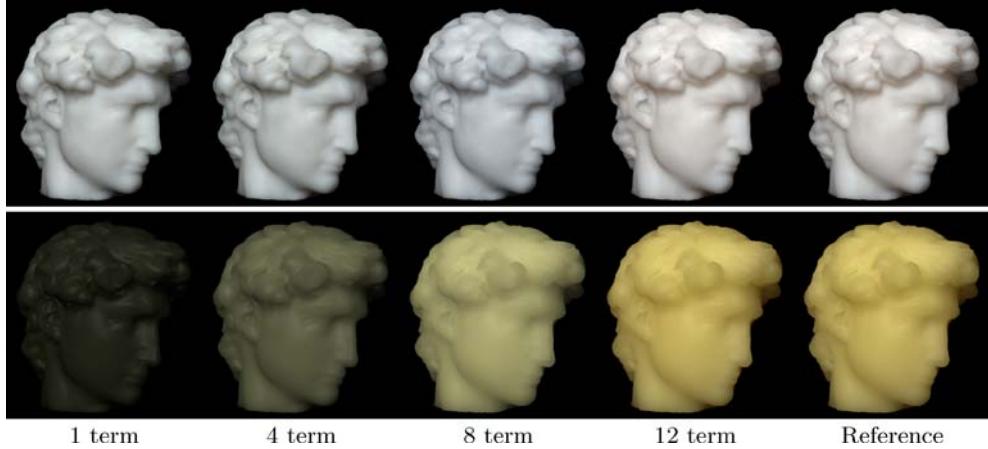


Fig. 2. Overview of our relighting algorithm



**Fig. 3.** Images synthesized from our system using increased number of  $R_d$  basis functions (from *left* to *right*). The two rows show measured marble and potato materials, respectively. At the *far right* are reference images generated without approximation on  $R_d$ . Using only 12  $R_d$  basis vectors, our system produces images indistinguishable from the reference

1. Calculate the direct lighting  $\mathbf{E}$  and project it onto the Haar wavelet basis to find  $\mathbf{E}^w$ .
2. For each BSSRDF basis, calculate the matrix-vector multiplication  ${}^w\mathbf{v}_k = {}^w\mathbf{T}_k^w \cdot \mathbf{E}^w$ .
3. A dynamic BSSRDF is provided by the user by defining its parameters  $\sigma$ ; it is then sampled and projected onto the basis set to obtain coefficients  $s_k(\sigma)$ :

$$s_k(\sigma) = \int_r R_d(r, \sigma) b_k(r) dr.$$

4. Take the linear sum of vectors  ${}^w\mathbf{v}_k$  with  $s_k(\sigma)$  as weights, then perform an inverse wavelet transform on the resulting vector to get a color vector  $\mathbf{L}_o$  representing the color of each vertex.

To compute direct lighting  $\mathbf{E}$  in step 1, we permit any direct lighting model that can be evaluated in real-time, such as shadowed point lights, environment lighting, and local area lighting. Note that the steps involving the dynamic BSSRDF occur after computing  ${}^w\mathbf{v}_k$  which is the major computational bottleneck. Therefore our system provides real-time response to BSSRDF editing, while maintaining interactive relighting speed, even for fairly large models.

*Implementation notes.* Below we describe three types of direct illumination models we support:

1. *Shadowed point light.* This is handled using standard GPU shadow mapping. To do so, we store illumination sample points in textures and use a fragment shader to compute the shadowed irradiance values, which are then read back to the CPU and projected onto the Haar wavelet basis, resulting in  $\mathbf{E}^w$ .

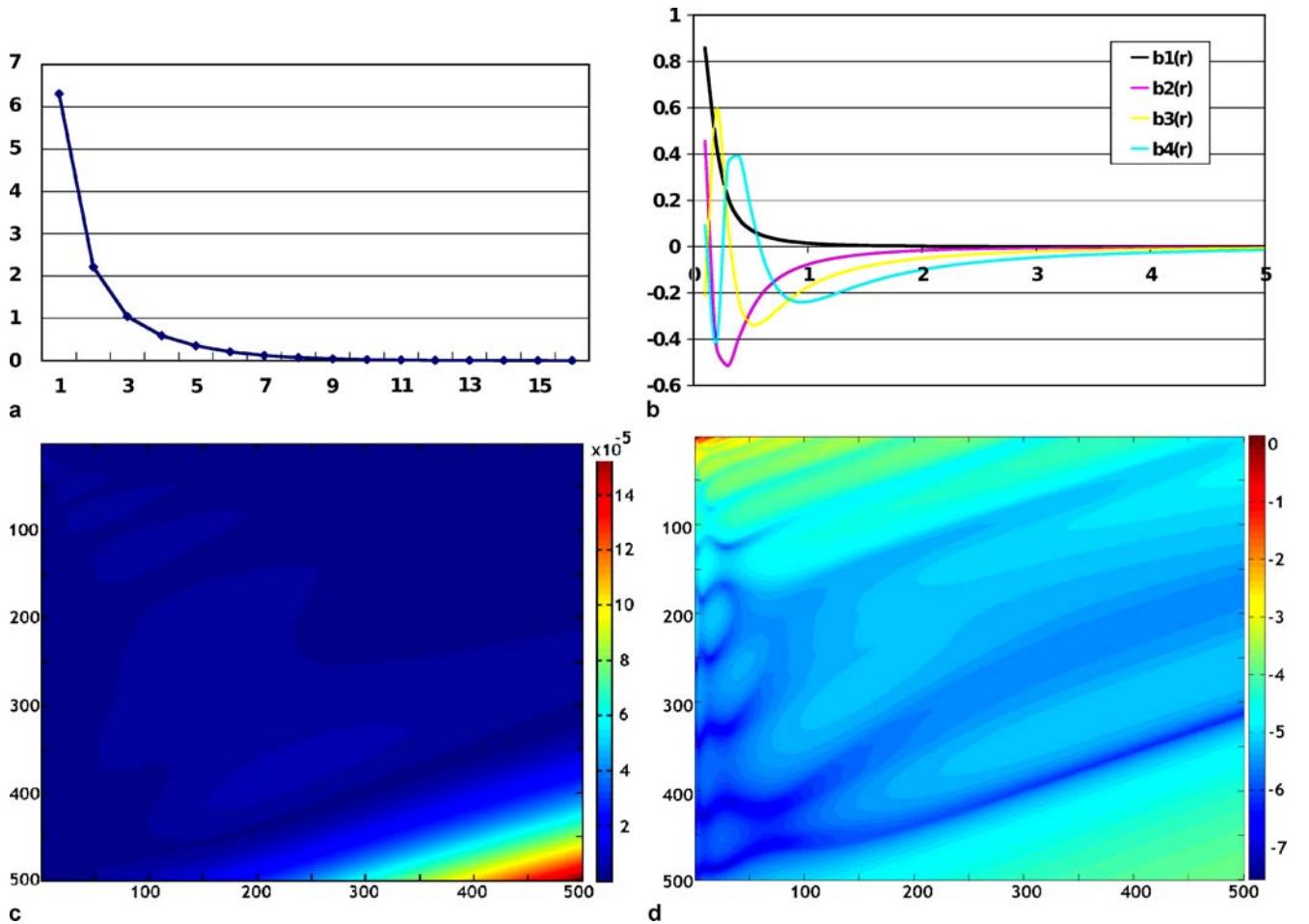
2. *Environment lighting.* We handle environment lighting using standard wavelet-based PRT. Specifically, we precompute a set of visibility vectors for each illumination sample point, apply a nonlinear wavelet approximation, and use them to compute the irradiance  $\mathbf{E}$  at run-time. This requires additional precomputed data; but we discovered that the same spatial compression step can be applied again here to fully exploit the data's spatial coherence and reduce the storage requirement. This method brings an additional advantage that the run-time algorithm can directly obtain wavelet-transformed irradiance  $\mathbf{E}^w$  and eliminate a wavelet transform step from  $\mathbf{E}$  to  $\mathbf{E}^w$ , thus saving time during relighting. Because  $\mathbf{E}^w$  is never visualized directly, it can be heavily compressed with little effect on the resulting quality. This method allows us to include realistic environment lighting effects with minimal overhead.

3. *Local area lights.* To include local area lights, we use the source radiance fields (SBF) technique presented by [26]. This approach samples the radiance field of a local area light in its surrounding 3D space, then uses simple interpolation to quickly evaluate the spatially varying illumination field at an arbitrary point  $x_i$ .

## 4 Results and discussion

Our timings are recorded on an Intel Core 2 Duo 1.8 GHz PC with NVIDIA 7600 GT graphics card. The first pass of precomputation and the relighting algorithms both use threads to utilize both CPU cores.

*Analysis of  $R_d$ .* Figure 4a plots the square roots of the first 15 singular values resulting from the SVD analysis



**Fig. 4.** **a** The decay of square roots of the first 15 singular values of  $R_d$ . **b** Plots of the first four PCA basis functions. **c** Plots of the  $L^2$  approximation error using 12 terms, and **d** plots of  $\log_{10}(\text{relative } L^\infty \text{ error})$ , indicating the digits of accuracy of the  $R_d$  approximation. The axes are for BSSRDF parameters  $\sigma'_s$  (vertical) and  $\sigma_a$  (horizontal)

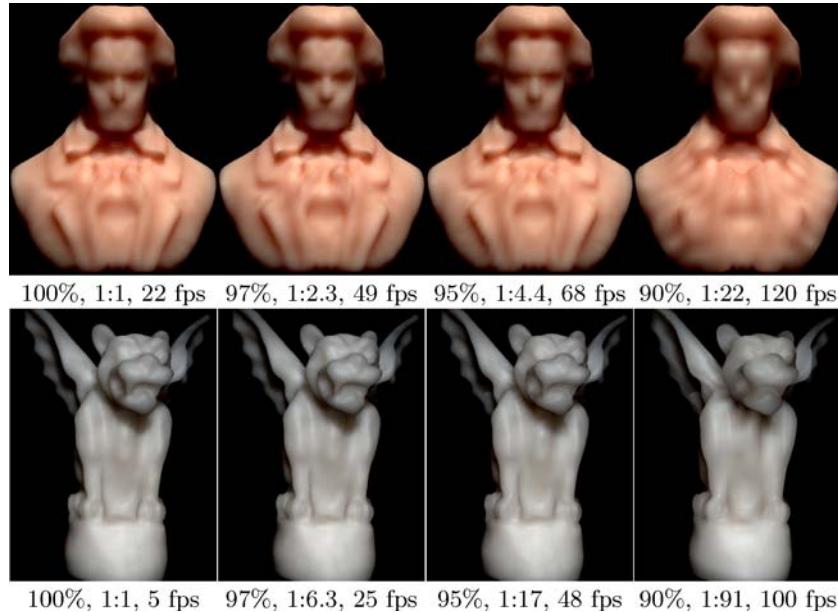
of  $R_d$ . We can see that the singular values decrease very quickly, indicating that only a few terms are needed to accurately approximate  $R_d$ . In Fig. 4b we plot the first four basis functions derived from our analysis. We only plot the range of [0.0, 5.0] as the first four bases represent dominant local transfer effects, thus they decrease to zero very rapidly as the distance  $r$  increases. Figure 3 shows a sequence of renderings with an increasing number of basis terms. A reference rendering is provided at the right. As expected from the SVD analysis, a small number of terms is sufficient to represent these materials. The material shown in the first row (marble) is dominated by near field transfer effects, thus is very well represented only using a few terms; on the contrary, the material shown in the second row (potato) is not as well represented by as few terms. This BSSRDF has stronger scattering effects over longer distances, and therefore cannot be accurately represented with only the first few terms.

To analyze the range of representable BSSRDF parameters using our method, in Fig. 4c we plot of  $L^2$  approximation error of  $R_d$  resulted from using 12 terms for a range of material parameters; in addition, Fig. 4d shows the logarithm of the relative  $L^\infty$  error where the color coding indicates the number of digits of accuracy resulting from the approximation. The axes represent parameter  $\sigma'_s$  along the vertical direction with a range  $[10^{-6}, 5]$  and parameter  $\sigma_a$  along the horizontal direction with the same range. These ranges cover the measured parameter values found in [12] with significant slack. The figure shows that our approximation is accurate for almost all parameters. Those for which it is not are uninteresting cases where multiple scattering is insignificant, thus it is more appropriate to handle these materials using the diffuse BRDF model.

*Precomputation: transfer compression.* Table 1 summarizes our precomputation benchmark for five test scenes. The angel and bird models are unparameterized models

**Table 1.** Precomputation benchmark for our test scenes. The values, from *left* to *right* represent: the number of mesh vertices, illumination points, the total processing time and storage size for precomputation pass 1, the number of terms per stored transport vector after pass 1, and the corresponding compression ratio, the same statistics for pass 2, and the total preserved energy, indicating the  $L^2$  accuracy resulting from the spatial compression

	Verts	Samples	Pass 1: Transport				Pass 2: Spatial			
			Time	Size	NLA	Ratio	Time	Size	NLA	Ratio
Angel	12 k	64 k	3 min	141 MB	256	0.4%	–	–	–	–
Bird	30 k	64 k	8 min	359 MB	256	0.4%	–	–	–	–
Beethoven	16 k	64 k	5 min	1.5 GB	1024	1.5%	13 min	44 MB	256	1:5
Gargoyle	64 k	128 k	40 min	6 GB	1024	0.8%	73 min	125 MB	256	1:6.3
David	96 k	256 k	2 h	9 GB	1024	0.4%	3 h	175 MB	256	1:6.6



**Fig. 5.** Comparison of rendering quality by varying spatial compression ratio. The numbers below each image indicate the percentage of preserved energy, the spatial compression ratio, and the relighting speed

and cannot be compressed spatially by our system. The other three models are all parameterized as geometry images, so we compress them spatially. Raw datasets range from 35 GB to 1.1 TB, but we never have to store them because the first pass immediately compresses most of them to less than 1% of their original size while maintaining a very accurate representation of the light transport. Even for the largest dataset we produce, the intermediate storage is reasonable. Further, common datasets take only minutes to compute.

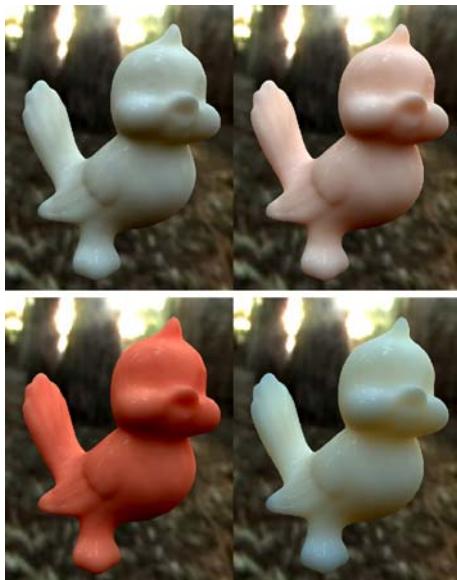
*Precomputation: spatial compression.* The second pass – spatial compression – is more expensive to compute, partly because it is not parallelized. Because spatial compression artifacts will be directly visualized, we use smooth, high-order wavelet filters for this step. We found that the CDF 9/7 filter from the JPEG 2000 image compression standard is most effective for our data even at

very aggressive compression ratios. Figure 5 compares renderings of models computed with progressively higher spatial compression ratios. The figure indicates the percentage of signal energy maintained, the spatial compression ratio, and the relighting speed. In general we find that with 95–98% preserved spatial signal energy (equivalent to normalized  $L^2$  error), we lose very little quality. This corresponds to roughly 1 : 5–1 : 6 spatial compression ratio. Thus after the second pass, our storage never exceeds 0.2% of the original matrix size, and the matrix can now fit comfortably in memory.

Our spatial compression scheme is more efficient to compute compared to previous PCA-based approaches. By using nonlinear wavelets, we achieve excellent compression ratios and our approach is not subject to partition boundary artifacts. This provides a significant performance speed up in relighting while maintaining high accuracy. The artifacts caused by further spatial com-

**Table 2.** Relighting and material editing benchmark for our test scenes

	Verts	Samples	Rendering Relighting	FPS Editing
Angel	12 k	64 k	24	220
Bird	30 k	64 k	10	200
Beethoven	16 k	64 k	68	120
Gargoyle	64 k	64 k	20	50
David	96 k	128 k	17	33



**Fig. 6.** Examples of environment lighting captured in real-time

pression take the form of smoother scattering details, as high-frequency spatial terms are dropped from the approximation. The size of the object greatly affects spatial compression since smaller objects are more influenced by short-distance scattering effects and thus contain more data coherence to exploit.

*Relighting and editing.* Relighting and editing in our system are very efficient (see Table 2). Relighting speed is roughly proportional to the stored matrix size since the algorithm linearly steps through the entire matrix once per relighting frame. Interactive rates are always maintained in all cases. Editing speed is real-time. Editing spatially compressed models is slightly more costly because we must perform an additional inverse wavelet transform at the last step. Figure 6 shows some results captured under environment lighting. Additional demos can be found in the video attached with this paper.

## 5 Conclusions and future work

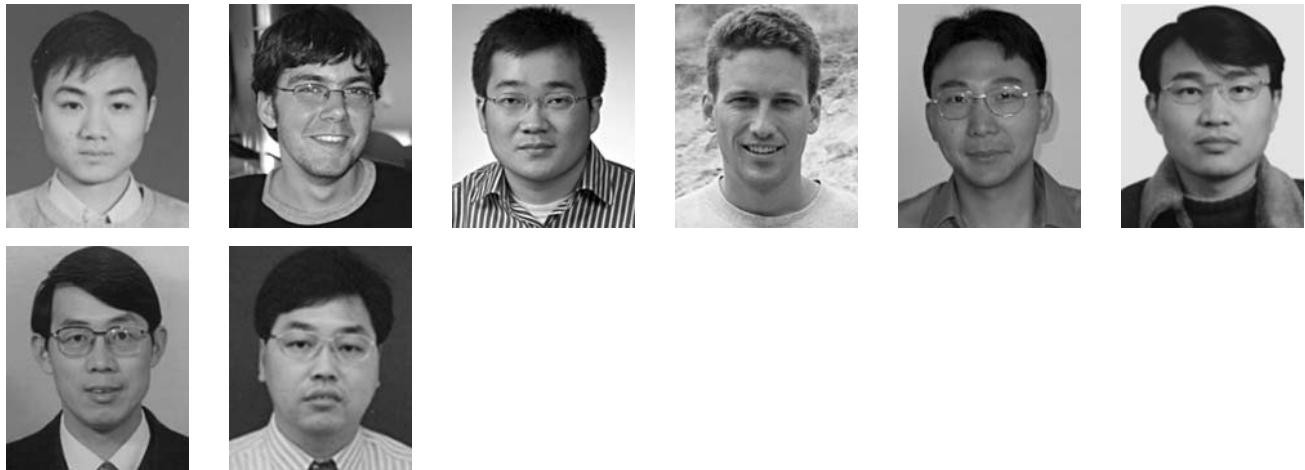
We have presented a novel method for real-time editing and relighting of diffuse translucent BSSRDFs under complex lighting. This permits interactive manipulation of translucent material properties and provides accurate real-time feedback of final-quality rendering. Our system is currently limited in three aspects: first, we are limited to static scene models due to the use of PRT. Second, we only handle diffuse multiple scattering effects and ignore single scattering and directional scattering effects, which are much more challenging to simulate. Third, our spatial compression scheme requires a parameterized scene model, which is not always feasible for complex geometry. In future work, we plan to extend the current editing system by addressing the above issues. First, we would like to incorporate other, more complicated translucency models, such as multi-layer model, measured diffuse scattering model, and more generally, heterogeneous translucent materials. Second, we would like to study extensions of the current system to deformable scene models. Finally, we plan to investigate other spatial compression schemes that do not rely on a parameterized domain.

**Acknowledgement** This work is partly supported by National Natural Science Foundation of China (Grant No. 60633070), the 973 Program of China (Grant No. 2002CB312102).

## References

1. Ben-Artzi, A., Overbeck, R., Ramamoorthi, R.: Real-time BRDF editing in complex lighting. *ACM Trans. Graph.* **25**(3), 945–954 (2006)
2. Cheslack-Postava, E., Goodnight, N., Ng, R., Ramamoorthi, R., Humphreys, G.: 4D compression and relighting with high-resolution light transport matrices. In: *Proceedings of ACM Symposium on Interactive 3D Graphics and Games*, pp. 81–88. ACM, New York, NY (2007)
3. Colbert, M., Pattanaik, S., Krivanek, J.: BRDF-shop: Creating physically correct bidirectional reflectance distribution functions. *IEEE Comput. Graph. Appl.* **26**(1), 30–36 (2006)
4. Dachsbaecher, C., Stamminger, M.: Translucent shadow maps. In: *Proceedings of the 14th Eurographics Symposium on Rendering*, pp. 197–201. Eurographics Association, Aire-la-Ville, Switzerland (2003)
5. D’Eon, E., Luebke, D., Enderton, E.: Efficient rendering of human skin. In: *Proceedings of the 18th Eurographics Symposium on Rendering*, pp. 147–157. Eurographics Association, Aire-la-Ville, Switzerland (2007)
6. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. *ACM Trans. Graph.* **21**(3), 355–361 (2002)
7. Hanrahan, P., Krueger, W.: Reflection from layered surfaces due to subsurface

- scattering. In: Proceedings of SIGGRAPH '93, pp. 165–174. ACM, New York, NY (1993)
8. Hao, X., Varshney, A.: Real-time rendering of translucent meshes. *ACM Trans. Graph.* **23**(2), 120–142 (2004)
  9. Hašan, M., Pellacini, F., Bala, K.: Direct-to-indirect transfer for cinematic relighting. *ACM Trans. Graph.* **25**(3), 1089–1097 (2006)
  10. Jensen, H.W., Buhler, J.: A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.* **21**(3), 576–581 (2002)
  11. Jensen, H.W., Christensen, P.H.: Efficient simulation of light transport in scenes with participating media using photon maps. In: Proceedings of SIGGRAPH '98, pp. 311–320. ACM, New York, NY (1998)
  12. Jensen, H.W., Marschner, S.R., Levoy, M., Hanrahan, P.: A practical model for subsurface light transport. In: Proceedings of SIGGRAPH '01, pp. 511–518. ACM, New York, NY (2001)
  13. Lensch, H.P.A., Goesele, M., Bekaert, P., Kautz, J., Magnor, M.A., Lang, J., Seidel, H.-P.: Interactive rendering of translucent objects. In: Proceedings of the 10th Pacific Graphics, pp. 214–224. IEEE Computer Society, Washington, DC (2002)
  14. Liu, X., Sloan, P.P., Shum, H.-Y., Snyder, J.: All-frequency precomputed radiance transfer for glossy objects. In: Proceedings of the 15th Eurographics Symposium on Rendering, pp. 337–344. Eurographics Association, Aire-la-Ville, Switzerland (2004)
  15. Mertens, T., Kautz, J., Bekaert, P., Seidel, H.-P., Reeth, F.V.: Interactive rendering of translucent deformable objects. In: Proceedings of the 14th Eurographics Workshop on Rendering, pp. 130–140. Eurographics Association, Aire-la-Ville, Switzerland (2003)
  16. Ng, R., Ramamoorthi, R., Hanrahan, P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.* **22**(3), 376–381 (2003)
  17. Ramamoorthi, R., Hanrahan, P.: Frequency space environment map rendering. *ACM Trans. Graph.* **21**, 517–526 (2002)
  18. Sloan, P.-P., Hall, J., Hart, J., Snyder, J.: Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.* **22**(3), 382–391 (2003)
  19. Sloan, P.-P., Kautz, J., Snyder, J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.* **21**, 527–536 (2002)
  20. Sloan, P.-P., Luna, B., Snyder, J.: Local, deformable precomputed radiance transfer. *ACM Trans. Graph.* **24**(3), 1216–1224 (2005)
  21. Sun, X., Zhou, K., Chen, Y., Lin, S., Shi, J., Guo, B.: Interactive relighting with dynamic BRDFs. *ACM Trans. Graph.* **26**(3), 27 (2007)
  22. Tsai, Y.-T., Shih, Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.* **25**(3), 967–976 (2006)
  23. Wang, R., Tran, J., Luebke, D.: All-frequency relighting of non-diffuse objects using separable BRDF approximation. In: Proceedings of the 15th Eurographics Symposium on Rendering, pp. 345–354. Eurographics Association, Aire-la-Ville, Switzerland (2004)
  24. Wang, R., Tran, J., Luebke, D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph.* **24**(3), 1202–1207 (2005)
  25. Xu, K., Gao, Y., Li, Y., Ju, T., Hu, S.-M.: Real-time homogeneous translucent material editing. *Comput. Graph. Forum* **26**(3), 545–552 (2007)
  26. Zhou, K., Hu, Y., Lin, S., Guo, B., Shum, H.-Y.: Precomputed shadow fields for dynamic scenes. *ACM Trans. Graph.* **24**(3), 1196–1201 (2005)



RUI WANG received his Ph.D. degree in applied mathematics from Zhejiang University in 2007. Now He is an assistant researcher in the State Key Laboratory of CAD&CG of Zhejiang University. His research interests include real-time rendering, virtual reality and geometry approximation.

EWEN CHESLACK-POSTAVA is a Ph.D. student in the Graphics Laboratory in the Department of Computer Science at Stanford University. He received is B.S. degree in Computer Science from the University of Virginia. His research interests include interactive photorealistic rendering, rendering techniques for large scale virtual worlds, and computer vision.

RUI WANG is an assistant professor in the Department of Computer Science at Univ. of Massachusetts Amherst. He received his Ph.D. in Computer Science from the University of Virginia in 2006, and B.S. from Zhejiang University in 2001. His research interests are in the field of computer graphics, with focuses on global illumination algorithms, interactive photorealistic rendering, GPU rendering, and appearance modeling.

DAVID LUEBKE is a research scientist at NVIDIA Research, which he helped found in

2006 after eight years on the faculty of the University of Virginia. He received his Ph.D. in Computer Science from the University of North Carolina under Dr. Frederick P. Brooks, Jr., and his Bachelors degree in Chemistry from the Colorado College. Dr. Luebke and his colleagues have produced dozens of papers, articles, chapters, and patents; a short film in the SIGGRAPH 2007 Electronic Theater; the book "Level of Detail for 3D Graphics"; and the Virtual Monticello exhibit seen by over 110,000 visitors to the New Orleans Museum of Art.

QIANYONG CHEN is an Assistant Professor in the Department of Mathematics and Statistics at the University of Massachusetts Amherst. He received his Bachelor degree from the University of Science and Technology of China (USTC) in 1996, and Ph.D. from Brown university in 2004. His research interests include developing stable and highly efficient numerical methods for solving partial differential equations.

WEI HUA received the Ph.D. degree in applied mathematics from Zhejiang University in 2002. He is now an associated professor of the State Key Laboratory of CAD&CG of Zhejiang University. His research interests include real-time simulation and rendering, virtual reality and software engineering.

QUNSHENG PENG is a Professor of computer graphics at Zhejiang University. His research interests include realistic image synthesis, computer animation, scientific data visualization, virtual reality, bio-molecule modeling. Prof. Peng graduated from Beijing Mechanical College in 1970 and received a Ph.D from the Department of Computing Studies, University of East Anglia in 1983. He serves currently as a member of the editorial boards of several international and Chinese journals.

HUJUN BAO received his Bachelor and Ph.D. in applied mathematics from Zhejiang University in 1987 and 1993. His research interests include modeling and rendering techniques for large scale of virtual environments and their applications. He is currently the director of State Key Laboratory of CAD&CG of Zhejiang University. He serves currently as a member of the editorial boards of several international and Chinese journals. He is also the principal investigator of the virtual reality project sponsored by Ministry of Science and Technology of China.