Fast Extraction of BRDFs and Material Maps from Images

Rafal Jaroszkiewicz
rjaroszk@uwaterloo.ca

Michael D. McCool
mmccool@uwaterloo.ca

Computer Graphics Laboratory
University of Waterloo

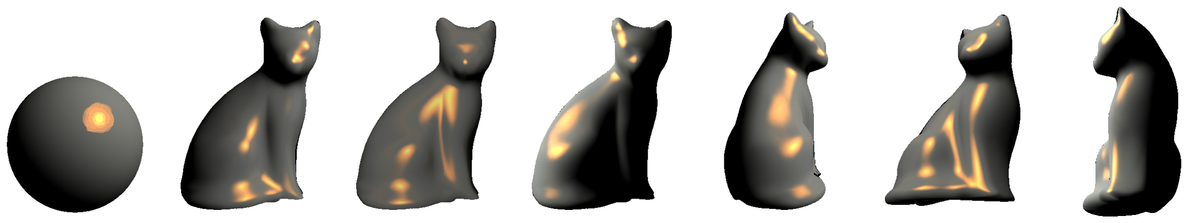

Figure 1: Painted sphere and a cat under changing light and view directions.

Abstract

The high dimensionality of the BRDF makes it difficult to use measured data for hardware rendering. Common solutions to overcome this problem include expressing a BRDF as a sum of basis functions or factorizing it into several functions of smaller dimensions which can be sampled into a texture.

In this paper we will focus on homomorphic factorization, which can be accelerated by preinverting the constraint matrix if the sampling pattern and the layout of the samples in the representation are fixed. Applying the preinverted constraint matrix is very fast and can be used to calculate factorization and material maps at interactive rates. We use this to derive shaders from painted examples. The technique presented in this paper allows interactive definition of materials, and, although based on physical parameters, this method can realize a variety of non-photorealistic effects.

Key words: BRDF, homomorphic factorization, inverse rendering, painting, hardware acceleration

1 Introduction

This paper presents an extension to the homomorphic factorization technique [17]. The extension permits fast computation of approximations to BRDFs. It is possible to reduce factorization time to less than a few seconds (a speedup of a factor of over 1000) by precomputing a pseudo-inverse of the constraint matrix. The precomputation time is still on the order of several minutes, but consecutive calculations of new factors can be performed at interactive rates.

This algorithm allows us to approximate a BRDF quickly from any sample value set with predefined parameters. We use it as a tool for defining an arbitrary spatially variant BRDF by integrating the homomorphic factorization with a painting application.

In such an application, the user is presented with the scene setup (a view of a 3D object and a light position) and is allowed to color the image to define the appearance of the object in this setup. Instead of painting, a photo could also be used after overlaying it on a 3D model. Our new factorization algorithm is invoked to calculate the BRDF approximation from the painted images. Our system is capable of recovering a variety of reflectance models ranging from non-photorealistic to physically based, including multi-material composition and spatially varying BRDFs. Non-homogeneous surfaces are approximated using material maps that combine several BRDFs using texture mapped blending coefficients.

In Section 2 we present previous work related to reflectance models and measuring BRDFs. Section 3 summarizes homomorphic factorization and describes extensions that enable fast calculations. The use of painted images as BRDF samples is described in Section 4 and implementation details are presented in Section 5. We show and discuss our results in Section 6 and conclude the paper in Section 8.

2 Related Work

The idea of representing a BRDF by several functions of smaller dimensions was introduced by Fournier [2], who used SVD factorization, and Heidrich and Seidel [5], who used analytic factorization. Later, Kautz and McCool

[9] reparameterized the SVD approach to better approximate reflectance models. McCool, Ang and Ahmad [17] presented a factorization method that solved some of the drawbacks of the previous methods. This new approach, homomorphic factorization, permits the factorization of functions of arbitrary dimension into products of several functions of smaller dimensions, allows the use of arbitrary parameterizations, and does not require a separate resampling step.

Other methods of BRDF representation rely on the summation of basis functions, for example, spherical harmonics [18]. Malzbender, Gelb and Wolters [14] used a polynomial basis to reconstruct surface color under varying light direction. This method captures spatial variation of reflectance over surfaces [10, 13] and is similar to the use of spherical harmonics. McAllister, Lastra and Heidrich [16] fitted Lafortune lobes [11] to measured data and stored the coefficients in texture maps, allowing real-time rendering of spatially varying BRDFs.

Lensch et al. [13] were successful in recovering spatially variant materials by grouping samples into clusters representing a single material. More recently Hertzmann and Seitz [6] developed an algorithm to recover both materials and normals based on a clustering method. An advantage of these algorithms is the high accuracy obtained. However, they suffer from long computational times.

Measured data was used by Marschner et al. [15] to find reflectance models of human skin [1]. Our work borrows from their approach, but differs in BRDF representation, and, instead of photographing the objects, we use interactive painting. Painting has been explored by Hanrahan and Haeberli [4] and Kalnins et al. [8] to interactively apply material onto a parametrized 3D model. Sloan et al. [19] explored painting as a method to capture shading for non-photorealistic rendering. They used the projection of the surface normal onto the viewing plane as a shading parameter. In contrast, our method incorporates a view vector, a light vector and texture coordinates as function parameters, thus allowing greater flexibility.

3 Homomorphic Factorization and Its Extensions

In order to perform homomorphic factorization [17] of a BRDF (or other high-dimensionality function) it is necessary to solve, in a minimum-residual or least-squares sense, an overconstrained linear system of the form

$$\begin{bmatrix} \tilde{\mathbf{f}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathsf{A}_1 & \mathsf{A}_2 & \cdots & \mathsf{A}_J \\ \lambda\mathsf{L}_1 & & & \\ & \lambda\mathsf{L}_2 & & \\ & & \vdots & \\ & & & \lambda\mathsf{L}_J \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{p}}_1 \\ \tilde{\mathbf{p}}_2 \\ \vdots \\ \tilde{\mathbf{p}}_J \end{bmatrix} \quad (1)$$

Where $\tilde{\mathbf{f}}$ is a vector of data points to be fitted (logarithms of sample values of an N-dimensional function f), the $\tilde{\mathbf{p}}_j$ are logarithms of samples of lower-dimensional functions p_j (which f project onto), the matrices A_j describe the projections of samples onto each factor as a set of constraints, and finally the matrices L_j are Laplacian operators that estimate the curvature of the functions $\tilde{\mathbf{p}}_j$. By setting these equal to zero we damp out ripples in the solution and interpolate over gaps. This overdetermined system of linear constraints can be written more compactly as $\mathbf{g} = \mathsf{B}\mathbf{y}$.

3.1 1D Factors

The application of homomorphic factorization to BRDFs presented by McCool, Ang and Ahmad [17] can handle both isotropic and anisotropic materials. It uses a parabolic map to project the parameters from BRDF's four dimensions onto two dimensional factor functions. However, if a BRDF is isotropic, the factor functions are radially symmetric under their parameterization (see Figure 2) and thus are really one dimensional. In practice the size of the constraint matrix is a linear function of the number of texels. Therefore, changing the dimensionality of factors from 2D to 1D yields a large reduction in matrix size.

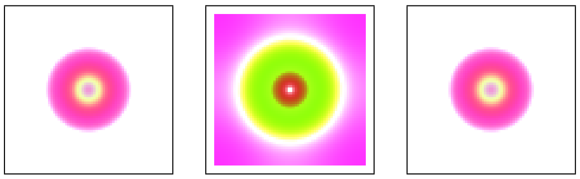

Figure 2: The 2D factors produced by homomorphic factorization of an isotropic material.

Given a set of samples, a constraint matrix is formed. We use three-factor product approximation where each factor is a 1D function. The projection transformations we use for mapping 4D BRDF parameters into a 1D parameter space for each factor are:

$$\pi_p(\hat{\omega}_i, \hat{\omega}_o) = |\hat{\omega}_i - \hat{\mathbf{n}}\,(\hat{\omega}_i \cdot \hat{\mathbf{n}})| \quad (2)$$
$$\pi_q(\hat{\omega}_i, \hat{\omega}_o) = |\hat{\mathbf{h}} - \hat{\mathbf{n}}\,(\hat{\mathbf{h}} \cdot \hat{\mathbf{n}})| \quad (3)$$
$$\pi_r(\hat{\omega}_i, \hat{\omega}_o) = |\hat{\omega}_o - \hat{\mathbf{n}}\,(\hat{\omega}_o \cdot \hat{\mathbf{n}})| \quad (4)$$

where $\hat{\mathbf{h}} = \frac{\hat{\omega}_i + \hat{\omega}_o}{|\hat{\omega}_i + \hat{\omega}_o|}$. These transformations return the length of the vector projected on the plane orthogonal to the surface normal $\hat{\mathbf{n}}$.

3.2 Preinverted Matrix

The minimum-residual solution of the sparse constraint system can be obtained by iterative methods. However, this is slow and only gives one solution for one set of data. The least-squares solution of the system can also be found using the pseudo-inverse B^+ of B. The pseudo-inverse can be computed using an SVD factorization:

$$\mathsf{B} = \mathsf{USV}^T = \sum_{i=0}^{N-1} s_i \mathbf{u}_i \mathbf{v}_i^T \tag{5}$$

$$\mathsf{B}^+ = \mathsf{VS}^{-1}\mathsf{U}^T = \sum_{i=0}^{N-1} \frac{1}{s_i} \mathbf{v}_i \mathbf{u}_i^T \tag{6}$$

Since the singular values can go to zero for singular matrices, this summation is usually truncated for sufficiently small values of s_i. Small singular values can also be damped out for greater stability:

$$\mathsf{B}^+ \approx \sum_{i=0}^{N-1} \frac{s_i}{s_i^2 + \alpha^2} \mathbf{v}_i \mathbf{u}_i^T, \tag{7}$$

but this does not give as much computational advantage as simple truncation. A compromise would be to linearly damp out higher singular values. Let $\operatorname{pos}(x) = \max(x, 0)$. Then compute the linearly damped SVD with

$$\mathsf{B}^+ \approx \sum_{i=0}^{N-1} \frac{\operatorname{pos}(1 - i\gamma)}{s_i} \mathbf{v}_i \mathbf{u}_i^T, \tag{8}$$

where $\gamma = K^{-1}$ for K non-zero weights.

Consider the computation of the approximate solution $\mathbf{y}^*$ in detail:

$$\mathbf{y}^* = \mathsf{B}^+\mathbf{g} = \sum_{i=0}^{N-1} \frac{\operatorname{pos}(1 - i\gamma)}{s_i} \mathbf{v}_i \mathbf{u}_i^T \mathbf{g} \tag{9}$$

Note that $\mathbf{u}_i^T\mathbf{g} = \mathbf{u}_i \cdot \mathbf{g}$, and the result is a scalar. However, in our particular problem, many entries of $\mathbf{g}$ are zero (due to the smoothing constraints), so we can just omit that part of $\mathbf{u}_i$ from the summation involved in the computation of $(\mathbf{u}_i \cdot \mathbf{g})$. Let $\mathbf{u}'_i$ be the appropriately truncated version of $\mathbf{u}_i$ so that $(\mathbf{u}'_i \cdot \mathbf{f}) = (\mathbf{u}_i \cdot \mathbf{g})$.

The vectors $\mathbf{u}'_i$ can be considered to be "analysis" functions and the vectors $\mathbf{v}_i$ can be considered to be "basis" functions:

$$\beta_i = \frac{\operatorname{pos}(1 - i\gamma)}{s_i} (\mathbf{u}'_i \cdot \mathbf{f}), \tag{10}$$

$$\mathbf{y}^* = \sum_{i=0}^{N-1} \beta_i \mathbf{v}_i. \tag{11}$$

The matrix B (and B^+) depends only on the projection functions used and the sampling pattern. It does *not* depend on the data. Therefore, if we perform the factorization once, we can apply it to many different data sets.

In practice, the singular values do not always vanish to zero very quickly, which implies that different data sets can emphasize different basis vectors. Nevertheless, multiplying the pseudo-inverse by the sample vector is still much faster than performing an SVD from scratch and then applying the result to find the BRDF approximation.

Finding the coefficients in Equation 10 is analogous to finding the coefficients for a spherical harmonic representation by summing the products of samples and corresponding values of basis functions. However, for spherical harmonics, or any other basis, it would also be necessary to weigh the products with an inverse of the sample density function.

3.3 Fast Factorization

An interesting way to compute the factorization is to utilize graphics hardware [7, 12]. Suppose $\mathbf{u}'_i$ and $\mathbf{f}$ are stored in texture maps. Then, for each i, we can perform a single dot product by rendering a rectangle containing these texture maps, multiplying them together, and then using recursive reduction operations to add up all the products. Alternatively, we could perform several dot products in parallel by storing the components of all analysis functions together in a single texture map, and using compositing operators to perform the summation over all components.

When done, the necessary coefficients should be stored in a buffer, which can be transferred to a texture map. Once such an approximation is computed, a given BRDF (or other function) can be represented by the β_i coefficients alone (stored in a texture with only K pixels), assuming the basis functions $\mathbf{v}_i$ and scale factors $\operatorname{pos}(1 - i\gamma)/s_i$ are separately known (i.e. are also stored in texture maps). With this approach, the basis functions only need to be stored once regardless of the number of BRDFs stored in the system. For higher performance the summation of basis functions can be performed (again, possibly using hardware acceleration) during a precomputation phase to compute the necessary factors and store them in texture maps.

3.4 Mapping Between Linear and Log Spaces

In order to convert a product into a sum, which is required for setting up a system of linear equations, homomorphic factorization takes a logarithm of the equation $f = \prod p_j$. It was noted in the original paper [17] that sample values close to zero are transformed into large negative values. Those values may dominate the equation, yielding a poor approximation that depicts an extremely absorbing material. At the preinversion stage, there is no general solution to this problem since the described effect stems from the sample values themselves. However, during approximation calculation, the mapping from linear to logarithmic space can be modified to avoid this problem.

For instance, a mapping of samples $f' = a + f$ before taking the logarithm,

$$\tilde{f}' \quad = \quad \log(f') = \log(a + f), \tag{12}$$

assigns more equal significance to all values of samples because the steep portion of the logarithmic function can be avoided. Now, the factors $\mathbf{p}'_j$ found by multiplying the pseudo-inverse by $\tilde{\mathbf{f}}'$ will approximate the values $f' = a + f$ instead of the sample values f. To find the approximation of the BRDF, apply the inverse:

$$f \quad \approx \quad (\prod p'_j) - a. \tag{13}$$

Subtracting a means that f may be negative. This rarely happens providing the samples are consistent.

Although it is usually desirable to assign all samples equal influence on the solution, there may be situations when de-emphasizing or emphasizing certain values is preferred. To emphasize the base color of a surface, we can use the usual identity mapping $f' = f$. However, in situations where relatively few samples capture specular highlights and the approximated material is shiny, it is necessary to use a mapping that puts emphasis on large values. The mapping $f' = a - f$, where a is the largest sample value, leads to the following

$$\tilde{f}' \quad = \quad \log(f') = \log(a - f). \tag{14}$$

This maps sample values close to a in linear space into large negative numbers in logarithmic space. This mapping can be used only if the samples $f \in [0, a)$. As in the previous case, when computing the factorization, a solution obtained by multiplying the pseudo-inverse of the constraint matrix by the vector $\tilde{\mathbf{f}}'$ yields the solution that is an approximation to transformed samples and not to the samples themselves. An inverse mapping is required to obtain an approximation to the original BRDF:

$$f \quad \approx \quad a - \hat{f} = a - \prod p_j \tag{15}$$

Again a problem with negative BRDF approximations arises when $\hat{f} > a$, but if samples are coherent such situation do not occur very often. If they do, they are handled by clamping to zero, as in the previous case (with some additional error).

4 Recovering BRDFs from Images

From an artist's point of view, painting is a much more natural tool for defining reflectance models than mathematical descriptions. Instead of programming a complicated shader, it is conceptually much more appealing to paint an image and let the program compute the textures and reflectance models from it. We will now describe such an application.

4.1 Factorized Approximations from Images

Painted images can be easily used as input to the fast homomorphic factorization if the pixels are thought of as BRDF samples (assuming point source illumination, for now). A pixel can be used as a sample only if there are light, view, and normal vectors associated with it. In order to obtain light and view vectors, we can render a 3D model into an image and calculate the required vectors for each pixel from the scene setup (objects, single light, and camera positions). The rendered image defines the regions where the object is present and we can also store images containing view vectors, normals, and light vectors. A new picture is then painted using the rendered image for guidance. The pixels in the painted picture together with the light and view vectors calculated for the corresponding pixels in the original image can be treated as a BRDF sample set, and then used as an input to the homomorphic factorization (or other approximation scheme).

In this application we do not have to recompute the pseudo-inverse because the light and view vectors have not changed for the scene. To obtain a new BRDF approximation, we would use fast homomorphic factorization to multiply new pixel values by the pseudo-inverse. This multiplication is very fast. The user can see the resulting BRDF right away, and if needed, make corrections to the painting, and recompute the BRDF, thus closing the iterative loop.

4.2 Extensions

Setting up pixels as samples, described above, constitutes the core idea of several techniques we have developed that try to recover more complex, spatially varying reflectance models from paintings.

Spatially Varying BRDF (4-Factor Product)

The homomorphic factorization in the form presented in the original paper [17] assumes spatially invariant BRDFs. Objects rendered using such BRDFs appear to be made of uniform materials (e.g., plastic). In the painting application, the user has freedom to paint objects in any style, and the samples can be, and usually are, inconsistent. That is, there may be two pixels that have the same sample parameters (light and view vectors) but very different color values. Such situations may be unintentional, in which case the pixel with a different color will be treated as an outlier if there are many other similarly parametrized samples with consistent color. On the other hand, the user may intentionally paint spatially varying colors to depict several materials in different regions of one object.

In order to handle such cases we need to use a spatial bidirectional reflectance distribution function (SBRDF)

[16]. It is a six dimensional function, and in addition to the usual parameters of a standard BRDF, it has two variables that parameterize the location on the surface much like texture coordinates. This function describes a relation between incoming and outgoing light at a particular point on the surface. In order to factorize a SBRDF, at least one factor must depend on the surface parameterization. We first tried a straightforward factorization that takes u and v texture coordinates of the surface as input parameters, and so we obtained an approximation consisting of four factors: $p(\pi_p(\hat{\omega}_i, \hat{\omega}_o))$, $q(\pi_q(\hat{\omega}_i, \hat{\omega}_o))$, $r(\pi_r(\hat{\omega}_i, \hat{\omega}_o))$, and $s(\pi_s(u, v))$. These four factors can be calculated in the usual manner using homomorphic factorization from SBRDF samples that have the following projection parameters: light vector, view vector, and texture coordinates.

Faithful approximation of SBRDFs by this four factor approximation depends on the resolution of the fourth factor. The maximum practical resolution is bounded by the limited size of the pseudo-inverse matrix. Since the number of columns is equal to the number of texels in all factors, and the number of rows is equal to the number of samples, the maximum usable factor resolution depends on the number of samples.

Material Mapping Recovery

Another approach to capturing spatially varying BRDFs is material mapping. In this method, the final color at a given point on the surface is assumed to be the result of a linear combination of several simple BRDFs. We assume the sample value can be computed by

$$\sum_{j=1}^{J} c_j(u,\, v) f_j(\hat{\omega}_i,\, \hat{\omega}_o) \tag{16}$$

for unknown BRDFs f_j.

An algorithm that strives to recover these BRDFs and their coefficients can be based on an iterative approximation refinement process. When computing a factorized approximation assuming spatial invariance of a BRDF using the three factor approach, there will be a large residual error in cases when the painted reflectance is spatially varying. We may calculate the residual error and try to fit it with another three factor approximation:

$$\Delta \;=\; f - p(\pi_p(\mathbf{x}))\, q(\pi_q(\mathbf{x}))\, r(\pi_r(\mathbf{x})) \tag{17}$$

where $\mathbf{x} = (\hat{\omega}_i,\, \hat{\omega}_o)$.

Unfortunately, to fit the residual error using homomorphic factorization, the error must be positive. We therefore scale the approximating values so that they are less than any sample value. Then, the residual error will always be positive and lends itself to repeated homomorphic factorization.

$$\begin{aligned} \Delta &= f - c\,\hat{f} >= 0 \qquad (18)\\ \text{where } c &= \begin{cases} f/\hat{f} & \text{if } f < \hat{f} \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

The scaling value can be thought of as a blending factor for this BRDF when evaluating material maps using linear combinations of BRDFs

$$f \;=\; \sum_{j=1}^{J} c_j\, \hat{f}_j \tag{19}$$

Because we are using a 3-factor approximation, the coefficient is not involved in the homomorphic factorization, and therefore we avoid the problem of limited resolution which we encountered previously in the four factor approximation. The resolution of the coefficient can be arbitrarily large, and high frequency variations can be accurately reproduced, as long as the factors p, q, and r can approximate the sample values.

An issue encountered during implementation is coefficient interpolation for material map approximation. When the 2D coefficient for each BRDF is being computed, the samples may fall sparsely on the coefficient domain. If the resolution of the material map is large, many texels will not be set properly. The values of the texels between where the samples fall need to be interpolated (see Figure 3). An easy way to perform interpolation is to render the sample values using a p-buffer: the texels in between the samples will be assigned appropriate values during rasterization.

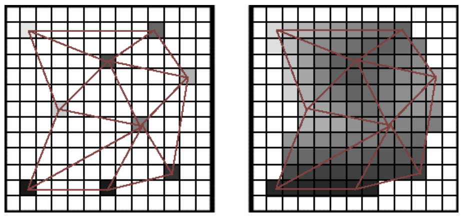

Figure 3: Left: coefficient texels set by sparse sampling. Right: unconstrained coefficient texels interpolated through rasterization in texture space.

Hybrid Method

The material mapping technique presented in the previous section uses the standard 3-factor approximations as representations for subsequent BRDFs that compose a given material. However, the material mapping and the 4-factor approximation methods are not mutually exclusive, and they can be combined. In this case, material mapping would use 4-factor instead of 3-factor approximations. Apart from that change, the procedure of calculating such representations is unaltered. The resulting SBRDF approximation takes slightly more space since 2D factors are used in addition to the usual 1D factors, but the solution tends to converge faster on the desired solution.

Integration of Multiple Views

When defining spatially varying BRDFs on an object, it is important to paint the whole surface in order to define material variation everywhere. Otherwise, during rendering or retargeting, the portions of an object that were lacking samples will not have a defined reflectance behavior.

One painted view is not enough to visually encompass all portions of the object. We need to use several paintings of the illuminated model capturing views from different directions. Integration of the information contained in each view may be challenging, especially if the paintings on different views are inconsistent with one another. However, assuming some degree of coherence, it is possible to find factorized approximations using several views.

In order to find a three or four factor approximation, each valid pixel in each view will participate in setting up the constraint matrix. The views can differ in lighting conditions (e.g., light position) but each painting must be consistent with its own lighting. Setting up the constraint matrix is conducted in the usual way: for each pixel the projected parameters are used to set up entries in the matrix. Finding the factorization using the pseudo-inverse is also the same. The pixel values are multiplied by the pseudo-inverse to obtain the factors.

Recovering a material map from several views is more tricky. If we want to use homomorphic factorization, we need to ensure that the residuals are always positive for each iteration of the algorithm. Thus, during one iteration for each view we calculate the coefficient texture that makes the residual positive for this particular view. Then we merge the coefficient textures, always keeping the smallest value. Keeping the smallest value ensures that the contributing BRDF is scaled down enough to be less than any sample from any view.

5 Testbed

We implemented a simple painting program that allows the user to paint 2D views of a 3D object and incorporates a factorizer. Twelve views are positioned at the vertices of an icosahedron to ensure full visual coverage of an object and for each view a light position is defined. A pixel in any view represents a BRDF sample; it corresponds to a place on an object for which light and view directions are known. So, for each such pixel, sample parameters (i.e., light and view vectors) can be recorded and used to build a constraint matrix.

Once the pseudo-inverse has been calculated the pixel RGB triplet can be used as sample values. The user is allowed to paint spatially varying BRDFs. In these instances, the application will attempt to recover the SBRDF by either four factor factorization or material mapping techniques, or both

If material mapping is chosen, each iteration of homomorphic factorization calculates one BRDF and its coefficient. Minimizing the error means that the residual becomes smaller and smaller. Because residual error is treated as samples for subsequent BRDF calculations, those values will approach zero for samples that are closely approximated by the factorization. If the painting has variation in color, some sample residuals will stay high exactly in the areas of different BRDF domination which will be recovered in the following iterations.

6 Results

Table 1 gives time measurements of SVD computation and factorization using 7884 samples. The calculation times for 4-factor approximation are much larger than for 3-factor factorization because of the number of texels that the 2D factor introduced into the approximation. In all cases, however, the factorization time using preinverted constraint matrices are orders of magnitude smaller than performing the calculations from scratch.

Table 2 shows similar measurements for approximations that used 2D functions. Again, the computation time is much smaller in the fast homomorphic factorization cases. Especially for the 3-factor approximation, three 2D factors result in a huge constraint matrix inversion. The sample count had to be reduced to around 300 in order to bring the constraint matrix down to a manageable size.

res.	SVD computation		fast factorization	
	3-factor	4-factor	3-factor	4-factor
32	4.26	247.36	0.1834	1.2649
64	11.08	330.04	0.2868	1.3207
96	21.58	397.37	0.3842	1.5919
128	38.06	506.11	0.4375	1.6877

Table 1: SVD computation times for preinversion and factorization in seconds for 1D factors using 7884 samples. The resolution of the fourth function in the 4-factor product, which is 2D, is constant and equal to 32x32 texels. Resolution changes only for 1D factors.

Table 3 shows the time measurements for material map recovery using 3-factor BRDF approximation for 7884

res.	SVD computation		fast factorization	
	3-factor	4-factor	3-factor	4-factor
16	1m 43s	3.77s	0.0308s	0.0160s
32	101m 12s	13.76s	0.1048s	0.0418s
48	-	44.88s	-	0.0853s
64	-	121.55s	-	0.1409s

Table 2: SVD computation times for preinversion and factorization in seconds using 317 samples. In 3-factor products all three functions are 2D, and in 4-factor products the first three functions are 1D (kept at 32 texels) and only the fourth factor is 2D. Resolution changes only for 2D factors.

res.	Num. of BRDFs			
	5	10	15	20
32	2.44	5.73	8.91	12.09
128	3.96	8.66	12.64	17.79

Table 3: Material map computation time in seconds

samples. The iterative calculations increased the calculation time, but the speed remained interactive.

The approximation error incurred by the approximation depends on the sample coherency. This is most evident in spatially varying paintings which are approximated using 4-factor factorization and material map recovery techniques. The error between the approximated and painted triceratops model from Figure 8 is shown in Table 4. The material mapping case approximates the samples better, but a 4-factor approximation gives in practice much better visual quality. To match this visual quality for material mapping, paintings from several view directions are required.

	4-factor	Num. of products				
		1	2	3	4	5
RMS	0.15	0.18	0.11	0.08	0.06	0.05
max	4.32	0.83	0.56	0.49	0.46	0.43

Table 4: Error metrics for 4-factor approximation and material map approximations using between 1 and 5 product components. (Sample values are in the interval $[0, 1]$)

Figure 6 shows a rendering of a cow using standard OpenGL (left) and using a factorized approximation of that lighting (right). It is interesting to see that 3-factor approximation captures both diffuse and specular reflection of light, although the Phong model cannot be mathematically factorized into a product.

The standard parameterization of factors is capable of handling non-photorealistic effects as shown in Figure 4. A outline of a duck is painted and then captured by factorization.

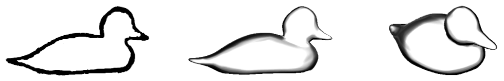

Figure 4: Painted and rendered duck model.

Figure 1 shows a coarse painting of a highlight on a diffuse sphere, and, then a retargeting of this reflection model to render a cat.

Figure 8 compares the reconstruction of the painted picture (a) by material mapping (b) and 4-factor approximation (c). The material mapping uses five BRDFs each approximated by 3-factor factorization. Material mapping performs better if the rendering direction is similar to the painted one, but starts deviating more than 4-factor approximation for arbitrary directions (Figure 8(e) and (f)). The best results, when using only one painting, were achieved by the hybrid method combining material mapping with a 4-factor factorization (Figure 8(d)).

The material mapping approach gives better results if the painting style is close to physically based shading. Figure 7 shows reflectance recovery from multiple views using material mapping. Three images of one object are painted from different view directions. The fourth image shows a new rendered image from another direction.

7 Discussion

This section discusses issues accompanying the methods described in this paper. We start with the fast factorization and then proceed to the recovery of BRDFs.

The gain in factorization speed is achieved by sacrificing flexibility of sampling. Thus, the only applications that can take advantage of this improvement are the ones for which the sampling pattern is fixed. If the light or view direction changes from previous measurement settings, the constraint matrix needs to be rebuilt and its pseudoinverse recomputed. Fortunately, the limitation to a fixed sampling pattern is not always an issue, as, for example, in the painting tool case.

Even when a fixed sampling pattern permits the use of the fast homomorphic factorization, new issues may still arise. The fast factorization uses a preinverted matrix whose size ranges from 10MB to 100MB, depending on the number of samples and factors. For fastest performance, this matrix should be loaded to the main memory, which may degrade performance on computers that have an inadequate amount of RAM. As described so far, the size of the pseudoinverse matrix and the fixed pattern requirement are two major issues associated with fast homomorphic factorization. In addition there are also some issues regarding the painting and BRDF recovery.

One of the issues pertains to the specular highlight, which needs to be painted for correct definition of many reflectance functions. The fast factorization is very sensitive to the exact placement of such highlights. If the highlight is centered incorrectly, the approximation will suffer from ringing artifact showing as an alternating pattern of darker and lighter rings. This can be reduced by applying the new mapping from linear to logarithmic space presented here. However, the only way to fully alleviate the rings is by iterative corrections of the painting itself.

Achieving a desired reflectance model is not always easy. The problem arises from the conflict between the physically based shading parameterization used and the non-photorealisic nature of painting. A best fit solution solves some of the problems caused by this decoupling. If most of the pixels are close to being in agreement with reality, the average is going to approximate the reflectance model well enough. However, if there are many discrepancies in the painting, a less desirable solution is produced. Again, an iterative painting process needs to be employed to converge on a satisfying solution. This can be time consuming and does not even guarantee that the desired shading model can be approximated adequately.

For spatially varying materials, part of the reason why an approximation may not capture the intended shading model is an ambiguity between variation in a BRDF and a material itself. For example, a localized increase in brightness on an object in one view can be attributed either to the specular reflection or a brighter color (albedo) of the material at that location. There is no clear solution other than painting several other images from different views or with different light directions in order to disambiguate the source of the brightness [6].

Our techniques for capturing spatially varying materials are admittedly ad hoc and have some limitations. For example, there is no guarantee that the residual error in material mapping technique can be factorized, since it may not resemble any form of reflectance function. Again, there is a trend that the paintings which diverge from physical plausibility will cause more problems than the paintings having physical reflectance properties. This is illustrated in Figure 7 and Figure 8(e), where marble is adequately captured by material mapping while non-photorealistic painting is not. As before, to pinpoint the correct shading model, more images are required from different views and for different lighting conditions.

A trend can be observed that for an adequate definition of a BRDF, paintings from many views or light directions are required. This is because we are striving to define 4D or 6D functions (for BRDFs and SBRDFs respectively) from 2D images. One image can be thought as a 2D slice through a function of higher dimensions, and so many slices are needed to define it. However, the images have to be consistent, and in particular the spatial variations must "stick" to the surface and their patterns must appear in the same places on the object in all of the images. This task is difficult for hand drawn paintings, but required, because the factorization is sensitive to the mismatched spatial patterns.

Lensch et al. [13], McAllister et al. [16], and Hertzman et al. [6] have accurately recovered spatially varying materials from photographs of real objects by fitting sample data to a Lafortune's reflectance model. Such an approach yields superior results, but requires a solution to a non-linear optimization problem. Available solvers take hours to compute the result, and, thus, are not suited for interactive applications. Our technique, on the other hand, is fast and enables interactive iterations to remove inconsistencies in the paintings.

The number of samples involved in the computations could contribute to the speed. Our system uses tens of thousands sample points, while the other techniques use millions of samples. It would be interesting to investigate the application of the techniques mentioned above to the BRDF recovery from paintings.

8 Conclusions

We have presented a refinement of the homomorphic factorization that improves its performance, and we have also presented a tool that uses this new algorithm for defining BRDFs and SBRDFs by painting examples of objects rendered using the intended reflectance model.

The speed of the new fast homomorphic factorization algorithm stems from the precalculation of the pseudoinverse and ability to apply it to any samples that have similar pattern and layout. An additional speed improvement is gained by replacing 2D factors with 1D factors for the special case of isotropic materials.

The painting tool was shown to reproduce a wide range of BRDFs. The overall accuracy of BRDF approximation depends on the painting style. Since the factorization is based on physical attributes its strength lies in recovering physically plausible reflectance models. Nevertheless, non-photorealistic effects were successfully captured using this model. It would be interesting to investigate other parameterizations that would be more suitable for non-photorealistic imagery.

In our work we focused on the homomorphic approximation technique. However, the same approach of using painted values as BRDF samples could be used with any approximation technique, such as linear PCA factorization (using the SVD results we already have), spherical harmonics, polynomial texture mapping, or Lafortune lobes, or a hybrid of these techniques, if they could be made fast enough.

It would also be straightforward to generalize the approach to approximation of outgoing radiance values rather than just reflectance models under point illumination. The paintings would have to take into account the environment instead of a single light source. The paintings would capture the view from different directions and for different orientations of the object with respect to the surrounding environment.

The clustering of samples that represent the same material has been shown to improve the quality of captured SBRDFs [6, 13]. These methods could potentially be applied to the homomorphic factorization if a sample subset could be selected for participation in factorization.

Other future work could involve following up on previous findings of Lensch [13], McAllister [16], and Hertzmann [6] to improve the performance of these algorithms. Recent advancements presented by Hillesland et al. [7] have reduced the computation time of non-linear optimizers fivefold by exploiting graphics hardware. Despite these results, the performance is still far from being interactive, and more work needs to be done in this area.

9 Acknowledgements

We would like to thank Selina Siu for many valuable comments and for painting input images, Celine Latulipe for revising our writing, and Alla Sheffer and Kevin Moule for providing us with parametrized 3D models.

This research was funded by grants from the National Science and Engineering Research Council of Canada (NSERC), the Centre for Information Technology of Ontario (CITO), the Canadian Foundation for Innovation (CFI), the Ontario Innovation Trust (OIT), the Bell University Labs initiative, and ATI Technologies.

References

[1] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the Reflectance Field of a Human Face. In *SIGGRAPH*, 2000.

[2] Alain Fournier. Separating reflection functions for linear radiosity. In *Eurographics Rendering Workshop*, 1995.

[3] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter S. Shirley, and Rich Riesenfeld. Interactive Technical Illustration. In *Symposium on Interactive 3D Graphics*, 1999.

[4] Pat Hanrahan and Paul E. Haeberli. Direct WYSIWYG Painting and Texturing on 3D Shapes. In *SIGGRAPH*, 1990.

[5] Wolfgang Heidrich and Hans-Peter Seidel. Realistic, Hardware-Accelerated Shading and Lighting. In *SIGGRAPH*, 1999.

[6] Aaron Hertzmann and Steven M. Seitz. Shape and Materials by Example: A Photometric Stereo Approach. In *In Proceedings of IEEE CVPR*, 2003. To appear.

[7] Karl E. Hillesland, Sergey Molinov, and Radek Grzeszczuk. Nonlinear Optimization Framework for Image-Based Modeling on Programmable Graphics Hardware. In *SIGGRAPH*, 2003. To appear.

[8] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. WYSIWYG NPR: Drawing Strokes Directly on 3D Models. In *SIGGRAPH*, 2002.

[9] Jan Kautz and Michael D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *Eurographics Rendering Workshop*, 1999.

[10] Jan Kautz and Hans-Peter Seidel. Towards Interactive Bump Mapping with Anisotropic Shift-Variant BRDFs. In *SIGGRAPH / Eurographics Workshop on Graphics Hardware*, 2000.

[11] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg. Non-Linear Approximation of Reflectance Functions. In *SIGGRAPH*, 1997.

[12] E. Scott Larsen and David McAllister. Fast Matrix Multiplies Using Graphics Hardware. In *ACM/IEEE Conference on Supercomputing*, 2001.

[13] Hendrik P. A. Lensch, Jan Kautz, Michael Goesele, Wolfgang Heidrich, and Hans-Peter Seidel. Image-Based Reconstruction of Spatially Varying Materials. In *Rendering Techniques 2001: Eurographics Rendering Workshop*, 2001.

[14] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial Texture Maps. In *SIGGRAPH*, 2001.

[15] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-based BRDF Measurement Including Human Skin. In *Eurographics Rendering Workshop*, 1999.

[16] David McAllister, Anselmo Lastra, and Wolfgang Heidrich. Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions. In *SIGGRAPH / Eurographics Workshop on Graphics Hardware*, 2002.

[17] Michael D. McCool, Jason Ang, and Anis Ahmad. Homomorphic Factorization of BRDFs for High-Performance Rendering. In *SIGGRAPH*, 2001.

[18] Ravi Ramamoorthi and Pat Hanrahan. A Signal-Processing Framework for Inverse Rendering. In *SIGGRAPH*, 2001.

[19] Peter-Pike Sloan, William Martin, Amy Gooch, and Bruce Gooch. The Lit Sphere: A Model for Capturing NPR Shading from Art. In *Graphics Interface*, 2001.

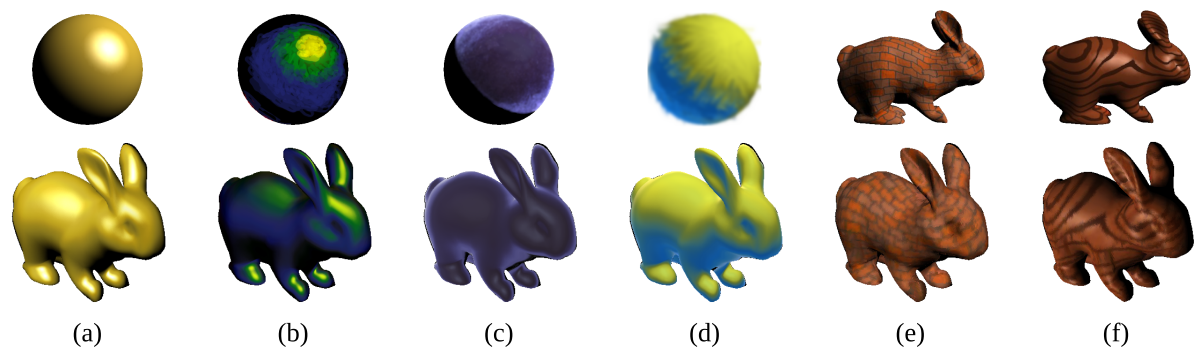

Figure 5: Painted and recovered shading models: (a) golden (b) yellow highlight with green and blue halos (c) velvet (d) warm to cool (e) diffuse brick (f) wood consisting of two BRDFs.

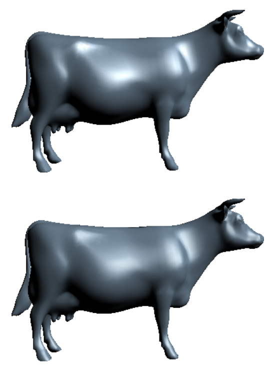

Figure 6: Diffuse and Phong specular reflectance model. Top image was rendered analytically and bottom one was rendered using factorization.

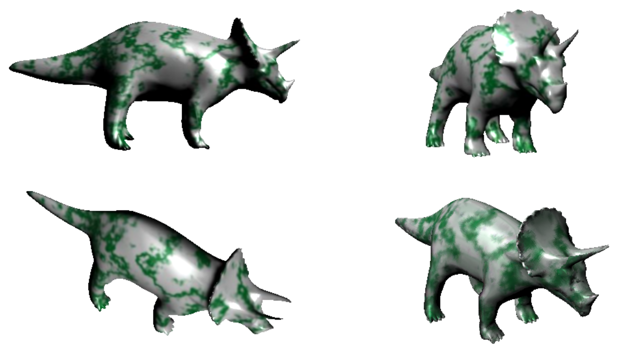

Figure 7: Three views of a triceratops painted with marble BRDFs and in bottom right a triceratops under new lighting direction viewed from new angle.

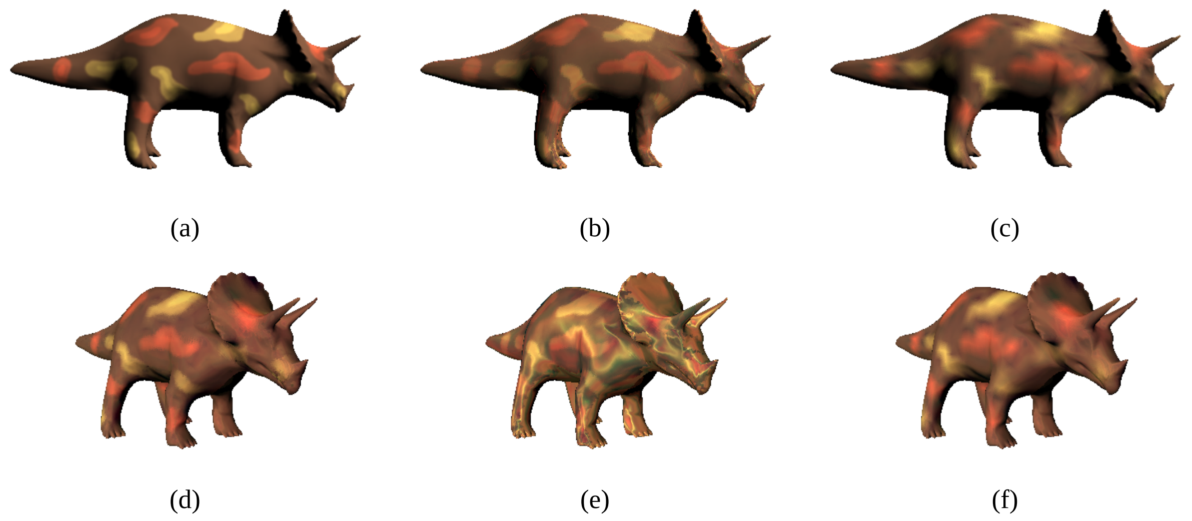

Figure 8: Material captured from one painting: (a) painted triceratops (b) material map (c) 4-factor approximation. Material rendered under new view and lighting directions: (d) hybrid of material map and 4-factor approximation (e) just material map (f) just 4-factor factorization.