Shadowed Relighting of Dynamic Geometry with 1D BRDFs

Derek Nowrouzezahrai Evangelos Kalogerakis Patricio Simari Eugene Fiume

{derek,kalo,psimari,elf}@dgp.toronto.edu - Dynamic Graphics Project, University of Toronto

Abstract

We present a method for synthesizing the dynamic self-occlusion of an articulating character in real-time ($>$ **170*Hz****) while incorporating reflection effects from 1D BRDFs under dynamic lighting and view conditions. We introduce and derive a general operator form for convolving spherical harmonics (SH) occlusion vectors with arbitrary 1D BRDF kernels. This operator, coupled with a compact linear model for predicting SH occlusion over articulating meshes, segments the BRDF and visibility terms of the direct illumination integral. We illustrate our results on a thin-membrane translucency model and the normalized Phong BRDF.*

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

Real-time shading of dynamic geometry with rich materials under complex lighting is an important challenge in computer graphics. We present a high-performance solution for shadowing and shading dynamically articulated character models with 1D parametric BRDF models under varying low-frequency environmental lighting. Unlike previous works which either focus on diffuse shading [ZHL*05, RWS*06,SGNS07,NSK*07] or require a prohibitive amount of precomputation and memory [wFPJY07], we leverage fast visibility synthesis and a novel operator formulation.

Articulation and deformation affect the per-vertex visibility functions, which cannot be sampled on-the-fly; thus, we fit a linear model to the visibility variation and present a technique for decoupling the BRDF from this changing visibility. Motivated by recent work on perceptual shadowing recognition in the presence of glossy reflections [KK07], our predictive shadowing model generates a Zonal Harmonics (ZH) approximation of visibility which we leverage for fast coordinate frame rotations during runtime relighting. We introduce a spherical harmonics (SH) convolution operator for 1D BRDFs and include a specific derivation for the normalized Phong BRDF. We are able to generate real-time results ($>$ **170 Hz**) and illustrate this on several animation sequences. We illustrate our shadowing effects applied to a modified version of the thin-membrane sub-surface model presented in [SLS05] and the normalized Phong shading model.

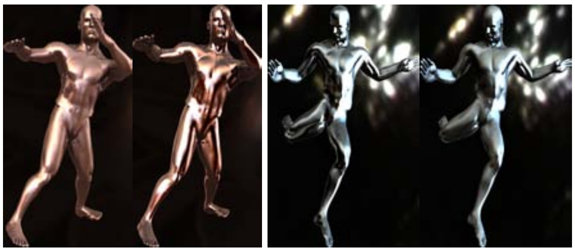

Figure 1: *Real-time results ($>$* ***170 Hz****) with dynamic geometry, lighting, view and Phong exponent.*

2. Previous Work

Precomputed radiance transfer (PRT) techniques segment and project the lighting, visibility [SKS02, NRH03, GKPB04] and BRDF [NRH04, BAOR06] into secondary bases to avoid expensive hemispherical sampling during runtime relighting. These techniques permit real-time manipulation of the lighting, BRDFs [SZC*07], secondary effects such as caustics [BAEDR07] and indirect illumination [HPB06, KTHS06]. However, the scene geometry, and in some cases the camera, must remain fixed. Kontkanen and Laine precompute a per-object field of ambient occlusion (AO) values queried at runtime fast enough to handle rigid objects in motion [KL05]. Bunnell approximates occluding geometry with a set of surface aligned discs in order to accelerate the AO calculation [Bun05]. Linear models of the per-vertex AO efficiently generate contact self-shadows for articulating characters [KA06, KA07].

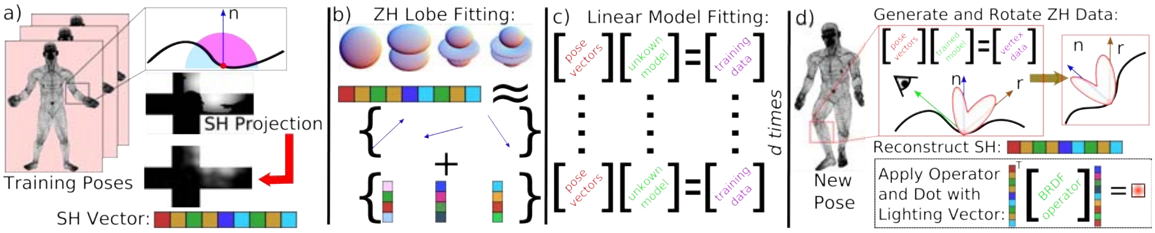

Figure 2: *The various stages of our precomputation (a to c) and runtime (d) algorithms (see sections 3 and 4).*

Zhou et al. tabulate projected visibility at discrete volumetric locations around rigid objects [ZHL*05]. Rigid deformations are supported, but require large amounts of storage and expensive SH multiplications. Ren et al. address these issues by accumulating in SH log-space [RWS*06] and approximating geometry with spheres. Sloan et al. use filtered and up-sampled screen-space accumulation and add an indirect illumination approximation [SGNS07]. Earlier, Sloan et al. modeled local shading effects, not including self-shadows, on deformable geometry using the ZH basis [SLS05]. Kautz et al. perform visibility rasterization and SH projection in real-time for low-resolution animating mesh [KLA04]. Nowrouzezahrai et al. perform dimensionality reduction on the space of SH cosine-weighted visibility across poses and fit a linear model to predict said coefficients in the reduced space as a function of pose [NSK*07]. Feng et al. also perform dimensionality reduction and fit a more complicated RBF model to the BRDF-weighted SH visibility for articulated characters [wFPJY07]. Theirs is the only work to address self-shadowed relighting with arbitrary isotropic BRDFs, however long precomputation times, large precomputation and runtime datasets are major limitations.

Contributions We decouple visibility from the BRDF and quickly fit a compact, predictive linear model to this data. We introduce a novel SH BRDF convolution operator for re-integrating the effects of 1D BRDFs to the synthesized self-shadowing and we include a specific derivation of this operator for the normalized Phong BRDF. We are able to relight animated geometry in real-time ($>$ 170 Hz), with modest memory requirements, dynamic lighting, view and BRDF parameters. We also illustrate our shadowing effects on a thin-membrane sub-surface model.

3. Representing and Synthesizing Visibility

We sythesize per-vertex self-occlusion functions as either SH coefficient vectors or a set of ZH lobe axes and lobe coefficients. We are motivated by recent works that use model fitting to generate ambient occlusion [KA06, KA07] or full SH visibility [NSK*07] as a function of joint angles.

Nowrouzezahrai et al. [NSK*07] fit a linear model to SH projected cosine-weighted visibility data as a function of articulation (joint angle vectors). We differentiate ourselves from this work in three ways. Firstly, prior to linear model fitting, we approximate the SH training data with many ZH lobes. Secondly, we fit a linear model to the SH visibility data, excluding the cosine weighting; we justify this choice in section 3.2. Lastly, our ZH representation (see section 3.1), coupled with a novel analytical shading operator (see section 4) allow us to render arbitrary 1D BRDFs, where [NSK*07] only handle diffuse objects.

3.1. Linear Multi-Lobe ZH Model Fitting

In order to apply our BRDF convolution operator (section 4) we must rotate the SH visibility vectors to align with the local reflection direction. We use a multi-lobe ZH representation for visibility which can be rotated efficiently. Choosing an appropriate coordinate frame for visibility, lighting and BRDF representation is still an openly debated issue in the litterature [NRH04, SLS05, MHL*06, GKD07] with the typical trade-offs being between size and performance.

For each frame of the f training frames, we precompute a 4^{th} order SH visibility projection for the v vertices (fig. 2a) and fit a set of c ZH lobes to this data (fig. 2b). Each ZH lobe, aligned in the local-coordinate frame of its vertex, is represented as 6 scalars: 2 for the direction vector of the lobe, and 4 for the ZH coefficients. With dimensionality $d = 6c$, we store the training data in a set of matrices, B_d of size $f \times v$. The joint angles for the training sequence are stored in a matrix, A, with size $f \times j$, with j active joint angles. We solve the system of equations $\mathsf{AX}_d = \mathsf{B}_d$ to obtain a set linear models used to synthesize the ZH visibility (fig. 2c).

3.2. Justification of Unweighted Occlusion

The mean and standard deviation error plots in figure 3 justify our decision: a linear model not only fits the visibility data with lower error than the cosine-weighted visibility fit (green vs. red), but if we analytically convolve the visibility fit with a clamped cosine filter, the error (blue) is still below that of the linear fit to the cosine-weighted data.

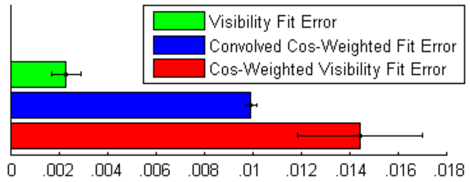

Figure 3: *Fitting error for plain and cos-weighted visibility.*

4. Applying BRDF Transformations

At runtime, we use our linear model to synthesize the ZH lobes and coefficients (4.3 #1), reconstruct the SH vector (see [SLS05], equation 1 and 4.3 #3), and we convolve the visibility with the BRDF (4.3 #4) before dotting with the SH lighting to yield the shade of the vertex (fig. 2d and 4.3 #5). The visibility might have to be rotated into a different frame depending on the selected BRDF (sec. 4.2 and 4.3 #2).

4.1. Transluncency Model

We add real-time self-shadowing to the transluncency model of Sloan et al. (eq. 22 in [SLS05]) by modulating the $s_\zeta \geq 0$ condition by the visibility. After synthesizing the ZH lobes and coefficients with our linear model, we determine the approximate SH coefficient vector as in [SLS05]

$$\boldsymbol{v}_l^m = \sqrt{\frac{4\pi}{2l+l}} \sum_{x=0}^{c} \boldsymbol{v}_l^x \; \boldsymbol{y}_l^m(s^x) \tag{1}$$

with $\boldsymbol{v}_l^x$ and s^x as the x^{th} ZH coefficient vector elements and lobe axis and $\boldsymbol{y}(s)$ as the vector of SH basis functions. A matrix-multiplication with the runtime pose vector of the articulating character is all that is necessary to generate our ZH visibility values. Equation 1 yields an SH visibility vector in the local coordinate frame of the vertex that we efficiently convolve against a diffuse kernel on the GPU (see Appendix A). The backfacing translucency component is a pre-tabulated unshadowed diffuse lobe. Figure 4 compares our shadowed results with the results generated in [SLS05].

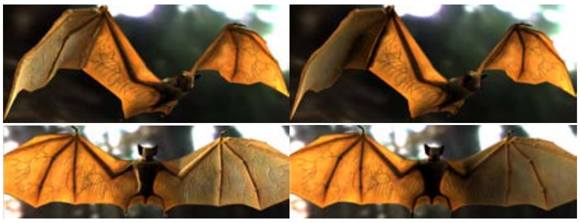

Figure 4: *Unshadowed (**left**) and shadowed (**right**) bat.*

4.2. 1D BRDF Model

The 1D subclass of BRDFs illustrate circular symmetry about view-dependent directions. We derive an analytic expression for the convolution kernel of the normalized Phong BRDF, whose axis of circular symmetry is the reflection vector, however our derivation can generalize to any 1D BRDF kernel. If we align the reflection vector at a vertex along the z-axis, we can derive the form of a *local-coordinate* Phong convolution kernel that convolves an SH visibility vector with the BRDF. The result is dotted with the SH lighting vector, yielding the view-dependent, shadowed BRDF shade. Our system allows any BRDF parameters (eg. the Phong exponent) to be manipulated in real-time, a local viewer model and we only integrate over the upper hemisphere, eliminating any incorrect lighting bleeding from the lower hemisphere, unlike previous derivations [RH02].

The Phong convolution kernel defined in the local-coordinate frame of a vertex, which convolves an SH visibility vector ($\boldsymbol{v}$) with the Phong BRDF with exponent α is

$$\begin{aligned} \boldsymbol{t}_l^m &= \frac{(\alpha+1)}{2\pi} \int_{S^2} \left[\sum_{l',m'} \boldsymbol{v}_{l'}^{m'} \; \boldsymbol{y}_{l'}^{m'}(s) \right] \max(0, (z \cdot s)^\alpha) \; \boldsymbol{y}_l^m(s) \, ds \\ &= \frac{(\alpha+1)}{2\pi} \sum_{l',m'} \boldsymbol{v}_{l'}^{m'} \underbrace{\left[\int_{H^2} \cos^\alpha(s_\theta) \; \boldsymbol{y}_{l'}^{m'}(s) \; \boldsymbol{y}_l^m(s) \, ds \right]}_{\mathsf{M}^\alpha[l'm',lm]} \end{aligned} \tag{2}$$

where $\boldsymbol{t}_l^m$ is the resulting transfer vector, with S^2 and H^2 as the spherical and hemispherical domains. The matrix form of the operator (see Appendix A) is sparse and well-suited for GPU implementation. In order to use this operator, we must rotate our SH visibility into the local reflection coordinate frame; we leverage the ZH representation to do so.

4.3. Runtime Calculations

1. Synthesize ZH visibility (CPU): $\forall c \quad [\boldsymbol{v}_l^c, s^c] = \mathsf{AX}^c$
2. (optional) Rotate ZH lobes (GPU): $\forall c \quad s_*^c = \text{rotate}(s^c, \vec{r})$
3. Convert ZH to SH (GPU): $\boldsymbol{v}_l^m = \sqrt{\frac{4\pi}{2l+l}} \sum_{x=0}^{c} \boldsymbol{v}_l^x \; \boldsymbol{y}_l^m(s_*^x)$
4. Apply BRDF convolution operator (GPU): $\boldsymbol{t}_l^m = \mathsf{M}^\alpha \cdot \boldsymbol{v}_l^m$
5. Shade Point (GPU): $\boldsymbol{l} = \text{rotate}(\boldsymbol{l}_{zh}, \vec{r})$, shade $= \boldsymbol{t} \cdot \boldsymbol{l}$

In the sub-surface case, the ZH lobes need not be rotated (ie: $s_*^x = s^x$), the vertex normal replaces the reflection vector ($\vec{r}$) for the ZH local-lighting rotation, and the diffuse convolution operator is used ($\alpha = 1$; see Appendix A).

5. Results

Generating the ZH training data takes anywhere between 1 and 10 hours, depending on the animation length and mesh size. Fitting the linear model takes less than 5 minutes in all cases (11 animations over 4 meshes). The cost of storing the linear models is $O(djv)$; for our animations, the models require between 4.6 MB and 11.2 MB. In contrast, the method of Feng et al. [wFPJY07] requires between 76 and 430 MBs since the BRDF is also fit; their method does not achieve as high a framerate and also has much longer precomputation times. The bottom of figure 5 illustrates the fitting error of our 1 and 2 ZH lobe linear models and the top shows some real-time results (> 170 Hz) with dynamic self-shadowing, lighting, α and view.

6. Conclusions and Future Work

We present a method that uses a linear model for predicting ZH visibility, as well as a SH BRDF convolution operator, for relighting articulating characters under dynamic lighting and view with varying BRDF parameters. Unlike previous techniques, we segment visibility and BRDF and yield real-time results with modest memory requirements. Supporting arbitrary BRDFs is left to future work.

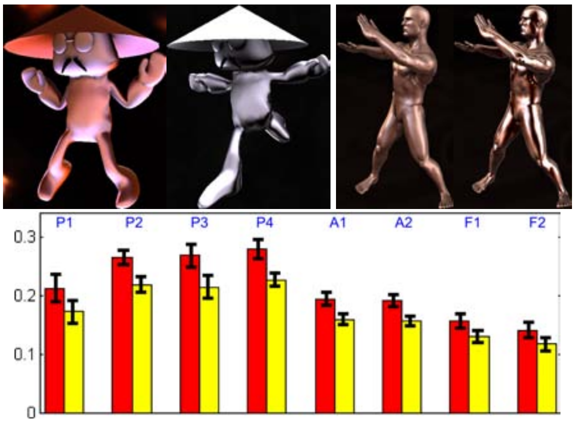

Figure 5: *Top: Rendering results; Bottom: One vs two ZH lobe linear mean errors with +/- standard deviation bars.*

Acknowledgements

The authors thank Alexis Angelidis for the Master Pai model and Robert E. Lansdale of Okino Graphics for the Polytrans software. The bat model is distributed with the DirectX SDK. The motion capture data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217. This work is funded in part by an NSERC PGS-D3 scholarship.

References

[BAEDR07] Ben-Artzi A., Egan K., Durand F., Ramamoorthi R.: A precomputed polynomial representation for interactive brdf editing with global illumination. In *Transactions On Graphics* (New York, 2007), ACM.

[BAOR06] Ben-Artzi A., Overbeck R., Ramamoorthi R.: Real-time brdf editing in complex lighting. In *SIGGRAPH '06* (New York, 2006), ACM, pp. 945–954.

[Bun05] Bunnell M.: Dynamic ambient occlusion and indirect lighting. In *GPU Gems 2* (2005), Addison-Wesley.

[GKD07] Green P., Kautz J., Durand F.: Efficient reflectance and visibility approximations for environment map rendering. *Computer Graphics Forum 26*, 3 (Sept 2007), 495–502.

[GKPB04] Gautron P., Křivánek J., Pattanaik S. N., Bouatouch K.: A novel hemispherical basis for accurate and efficient rendering. In *Eurographics Symposium on Rendering* (June 2004), pp. 321–330.

[HPB06] Hašan M., Pellacini F., Bala K.: Direct-to-indirect transfer for cinematic relighting. In *SIGGRAPH '06* (New York, 2006), ACM, pp. 1089–1097.

[KA06] Kontkanen J., Aila T.: Ambient occlusion for animated characters. In *Proceedings of Eurographics Symposium on Rendering 2006* (2006), Eurographics Association, pp. 343–348.

[KA07] Kirk A. G., Arikan O.: Real-time ambient occlusion for dynamic character skins. In *I3D '07* (New York, 2007), ACM, pp. 47–52.

[KK07] Kozlowski O., Kautz J.: Is accurate occlusion of glossy reflections necessary? In *APGV '07* (New York, 2007), ACM, pp. 91–98.

[KL05] Kontkanen J., Laine S.: Ambient occlusion fields. In *I3D '05* (New York, 2005), ACM, pp. 41–48.

[KLA04] Kautz J., Lehtinen J., Aila T.: Hemispherical rasterization for self-shadowing of dynamic objects. In *Eurographics Symposium on Rendering* (Sweden, 2004), Eurographics Association, pp. 179–184.

[KTHS06] Kontkanen J., Turquin E., Holzschuch N., Sillion F.: Wavelet radiance transport for interactive indirect lighting. In *Eurographics Symposium on Rendering* (2006), Heidrich W., Akenine-Möller T., (Eds.), Eurographics.

[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.

[NRH03] Ng R., Ramamoorthi R., Hanrahan P.: All-frequency shadows using non-linear wavelet lighting approximation. In *SIGGRAPH '03* (New York, 2003), ACM, pp. 376–381.

[NRH04] Ng R., Ramamoorthi R., Hanrahan P.: Triple product wavelet integrals for all-frequency relighting. In *SIGGRAPH '04* (New York, 2004), ACM, pp. 477–487.

[NSK*07] Nowrouzezahrai D., Simari P., Kalogerakis E., Singh K., Fiume E.: Compact and efficient generation of radiance transfer for dynamically articulated characters. In *GRAPHITE* (2007), ACM.

[RH02] Ramamoorthi R., Hanrahan P.: Frequency space environment map rendering. In *SIGGRAPH '02* (New York, 2002), ACM, pp. 517–526.

[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. In *SIGGRAPH '06* (New York, 2006), ACM, pp. 977–986.

[SGNS07] Sloan P.-P., Govindaraju N., Nowrouzezahrai D., Snyder J.: Image-based proxy accumulation for real-time soft global illumination. In *Pacific Graphics* (2007), IEEE.

[SKS02] Sloan P.-P., Kautz J., Snyder J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02* (New York, 2002), ACM, pp. 527–536.

[SLS05] Sloan P.-P., Luna B., Snyder J.: Local, deformable precomputed radiance transfer. In *SIGGRAPH '05* (New York, 2005), ACM, pp. 1216–1224.

[SZC*07] Sun X., Zhou K., Chen Y., Lin S., Shi J., Guo B.: Interactive relighting with dynamic brdfs. In *SIGGRAPH '07* (New York, 2007), ACM, p. 27.

[wFPJY07] wen Feng W., Peng L., Jia Y., Yu Y.: Large-scale data management for prt-based real-time rendering of dynamically skinned models. In *Eurographics Symposium on Rendering* (2007), Eurographics Association.

[ZHL*05] Zhou K., Hu Y., Lin S., Guo B., Shum H.-Y.: Precomputed shadow fields for dynamic scenes. In *SIGGRAPH '05* (New York, 2005), ACM, pp. 1196–1201.

Appendix A: Phong Convolution Matrix

Given the SH coefficients of a visibility function, expressed in a local co-ordinate frame (with the normal pointing along the z axis), we derive an analytic operator that convolves the signal with the normalized Phong BRDF: $\mathsf{M}^{\alpha} =$

$$\begin{bmatrix}
M_{00} & 0 & M_{02} & 0 & 0 & 0 & M_{06} & 0 & 0 & 0 & 0 & M_{0B} & 0 & 0 & 0 & 0 \\
0 & M_{11} & 0 & 0 & 0 & M_{15} & 0 & 0 & 0 & 0 & 0 & M_{1B} & 0 & 0 & 0 & 0 \\
M_{20} & 0 & M_{22} & 0 & 0 & 0 & M_{26} & 0 & 0 & 0 & 0 & 0 & M_{2C} & 0 & 0 & 0 \\
0 & 0 & 0 & M_{11} & 0 & 0 & 0 & M_{15} & 0 & 0 & 0 & 0 & 0 & M_{1B} & 0 & 0 \\
0 & 0 & 0 & 0 & M_{44} & 0 & 0 & 0 & 0 & 0 & M_{4A} & 0 & 0 & 0 & 0 & 0 \\
0 & M_{51} & 0 & 0 & 0 & M_{55} & 0 & 0 & 0 & 0 & 0 & M_{5B} & 0 & 0 & 0 & 0 \\
M_{60} & 0 & M_{62} & 0 & 0 & 0 & M_{55} & 0 & 0 & 0 & 0 & 0 & M_{6C} & 0 & 0 & 0 \\
0 & 0 & 0 & M_{51} & 0 & 0 & 0 & M_{55} & 0 & 0 & 0 & 0 & 0 & M_{5B} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M_{44} & 0 & 0 & 0 & 0 & 0 & M_{4A} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M_{99} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & M_{A4} & 0 & 0 & 0 & 0 & 0 & M_{AA} & 0 & 0 & 0 & 0 & 0 \\
M_{0B} & M_{B1} & 0 & 0 & 0 & M_{B5} & 0 & 0 & 0 & 0 & 0 & M_{BB} & 0 & 0 & 0 & 0 \\
0 & 0 & M_{C2} & 0 & 0 & 0 & M_{C6} & 0 & 0 & 0 & 0 & 0 & M_{CC} & 0 & 0 & 0 \\
0 & 0 & 0 & M_{B1} & 0 & 0 & 0 & M_{B5} & 0 & 0 & 0 & 0 & 0 & M_{BB} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M_{A4} & 0 & 0 & 0 & 0 & 0 & M_{AA} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & M_{99}
\end{bmatrix}$$

$$M_{00} = \frac{0.5}{\alpha_1} \qquad M_{02} = \frac{\sqrt{3}}{2\alpha_2} \qquad M_{06} = \frac{\sqrt{5}\alpha}{2\alpha_1\alpha_3} \qquad M_{11} = \frac{1.5}{\alpha_1\alpha_3}$$

$$M_{15} = \frac{3\sqrt{5}}{2\alpha_2\alpha_4} \qquad M_{1B} = \frac{3\sqrt{7}\alpha}{\sqrt{2}\alpha_1\alpha_3\alpha_5} \qquad M_{2C} = \frac{\sqrt{21}\alpha}{2\alpha_3\alpha_5} \qquad M_{44} = \frac{7.5}{\alpha_1\alpha_3\alpha_5}$$

$$M_{4A} = \frac{7.5\sqrt{7}}{\alpha_2\alpha_4\alpha_6} \qquad M_{5B} = \frac{3\sqrt{35}\alpha_1}{\sqrt{2}\alpha_2\alpha_4\alpha_6} \qquad M_{99} = \frac{52.5}{\alpha_1\alpha_3\alpha_5\alpha_7} \qquad M_{0B} = \frac{\sqrt{7}\alpha_{-1}}{2\alpha_2\alpha_4}$$

$$M_{22} = \frac{1.5}{\alpha_3} \qquad M_{26} = \frac{\sqrt{15}\alpha_1}{2\alpha_2\alpha_4} \qquad M_{55} = \frac{7.5}{\alpha_3\alpha_5} \qquad M_{AA} = \frac{52.5}{\alpha_3\alpha_5\alpha_7}$$

$$M_{BB} = \frac{21\left(2\alpha^2+4\alpha+5\right)}{2\alpha_1\alpha_3\alpha_5\alpha_7} \qquad M_{CC} = \frac{7\left(\alpha^2+2\alpha+15\right)}{2\alpha_3\alpha_5\alpha_7} \qquad M_{6C} = \frac{\sqrt{35}\left(\alpha^2+2\alpha+6\right)}{2\alpha_2\alpha_4\alpha_6}$$

where $\alpha_x = (\alpha + x)$. Note that setting $\alpha = 1$ yields the canonical diffuse convolution operator used with the translucency model. The resulting operator is sparse and symmetric (17.1% non-zero/5.9% unique elements). Red elements are unique entries, green elements are duplicate entries, blue elements are simple multiples of a unique element.