

携帯端末でも動作する高精細画像生成アルゴリズムの開発と実装

代表研究者

岩崎 慶

和歌山大学システム工学部 准教授

1 はじめに

本研究では、携帯端末でも動作可能な低コスト・省メモリな高精細レンダリング手法の開発を行う。本研究では特に、現実世界の複雑な照明を用いて高精細な画像を生成するイメージベースライティング手法の携帯端末上での開発研究を行う。本研究では、環境照明下の布のレンダリング手法の開発を行う。

CG 分野において、布の質感をリアルに表現することは重要な研究課題の一つである。布のレンダリングでは、現実世界の複雑な照明（環境照明）下において布の質感を表す散乱特性をインタラクティブに変化させて外観を編集することが望まれる。従来の布のレンダリング手法では、単純な光源のみを対象としているか [6]、計測データを用いているため外観をインタラクティブに編集することが難しい [7, 8, 9]。

本稿は、織り合された糸から構成される布の高速レンダリング法および布の外観編集法を提案する。本稿では、布の散乱モデルとして Sadeghi らの提案したマイクロシリンダモデル [6] を使用する。マイクロシリンダモデルにおける布の輝度は、入射光の輝度、布の散乱関数、および布を構成する糸の幾何構造に基づいた重み関数の積を積分することによって計算される。提案法では入射光（環境照明）を球面ガウス関数の線形和で表現する。球面ガウス関数、散乱関数と重み関数の積の積分を、球面ガウス関数と重み関数の畳み込み積分と、球面ガウス関数と散乱関数の積分の積で近似する。畳み込み積分は前計算可能であり、レンダリング時にテーブル参照するだけでよい。球面ガウス関数と散乱関数の積分は解析的に計算可能なため、レンダリング時に布の材質特性をインタラクティブに変更することが可能となる。

2 関連研究

布のレンダリングに関する研究は CG の分野において重要な研究分野の一つであり、25 年以上研究がなされてきている。初期のレンダリング手法では、実験によるシェーディングモデルに基づいている [10, 4]。Weil は、レイトレーシングによってリアルな布をレンダリングする手法を提案した [10]。Daubert らは、編み糸を陰関数表面でモデリングし、BTF に類似したデータ構造を用いて布をレンダリングする手法を提案した [4]。Ashikhmin らは、マイクロファセットモデルをもちいてサテンやベルベットをレンダリングする手法を提案した [2]。Adabala らは、単純な BRDF モデルを用いて効率的に布をレンダリングする手法を提案した [1]。これらの手法では、もっともらしい外観の表現に重きを置いており物理的正確さにかける。布の繊維構造をモデリングすることにより布をレンダリングする手法がいくつか提案されている [13, 3]。これらの手法はリアルな布のレンダリングが可能であるが、反射モデルのコントロールが難しい。安田らは、布の特徴的な光沢反射を再現する手法を提案した [14]。しかしながらこの手法は非常に簡略化したモデルであり、現実の布モデルとの検証・照合を行っていない。Westin らは、繊維レベルの微細構造から反射特性を予測するモデルを提案した [11]。Zhao らは、布の CT 画像からボリュームレンダリングを用いて布をレンダリングする手法を提案した [15, 16]。これらのモデルは非常に高品質な布のレンダリングが可能であるが、特定の布に限られている。Irawan らは、編み込んだ絹の布の外観を表現するレンダリング手法を提案した [5]。この手法は、糸によるシャドウイングおよびマスキングの効果を考慮していない。

本稿では、Sadeghi らの提案したマイクロシリンダモデル [6] を用いて布の輝度を計算する。マイクロシリンダモデルは、糸における散乱モデルを定式化し、実際の布の計測データを用いてこのモデルの有効性を検証している。マイクロシリンダモデルでは、散乱関数のパラメータをコントロールすることにより様々な布の散乱特性を表現することが可能である。しかしながら、この手法では環境照明を考慮していない。そこで提案法では、マイクロシリンダモデルを用いて環境照明下の布をインタラクティブにレンダリング

する手法を提案する。

3 関数の定義

提案法で使用する球面ガウス関数 G ・円形ガウス関数 g^c ・ガウス関数 g ・正規化ガウス関数 g^u は以下の式で表現される。

$$G(\omega, \xi, \sigma) = \exp(2(\omega \cdot \xi - 1)/\sigma^2) \quad (1)$$

$$g^c(x; \mu, \sigma) = \exp(2(\cos(x - \mu) - 1)/\sigma^2) \quad (2)$$

$$g(x, \mu, \sigma) = \exp(-(x - \mu)^2/\sigma^2) \quad (3)$$

$$g^u(x, \mu, \sigma) = \exp(-(x - \mu)^2/\sigma^2)/\sqrt{\pi}\sigma \quad (4)$$

ここで、 ω は方向（単位）ベクトル、 ξ は球面ガウス関数の軸ベクトル、 σ, μ はパラメータで、 $\sigma^2/2, \mu$ は分散、平均に相当する。布の輝度は、糸の接線方向 \mathbf{t} 、法線方向 \mathbf{n} からなるローカル座標系（図 1 参照）で計算される。 \mathbf{t} を法線とする平面とのなす角を θ 、法線 \mathbf{n} との方角を ϕ とし、光の入射方向 ω_i と反射方向 ω_o の角度をそれぞれ $\theta_i, \phi_i, \theta_o, \phi_o$ とする。また、 $\theta_d = (\theta_i - \theta_o)/2, \theta_h = (\theta_i + \theta_o)/2, \phi_d = \phi_i - \phi_o$ とする。 ω_i, ω_o を \mathbf{n} と \mathbf{t} がなす平面へ投影したベクトルと \mathbf{n} とのなす角をそれぞれ ψ_i, ψ_o とし、 $\psi_d = \psi_i - \psi_o$ とする。

4 マイクロシリンダモデルを用いた輝度計算

マイクロシリンダモデルでは、布地を 2 つの直行した方向に沿った円筒からなる糸が織り合わされたメッシュとみなす（図 1(c)）。糸の織り込みパターンのなかで最小な部分を smallest patch と呼ばれる [6]。布地は smallest patch の反復によって構築される。布地における ω_o 方向の輝度 $L_o(\omega_o)$ は、smallest patch の 2 つの糸ごとに計算した輝度 $L_1(\omega_o)$ と $L_2(\omega_o)$ の線形和 $L_o(\omega_o) = a_1 L_1(\omega_o) + a_2 L_2(\omega_o)$ で表現される。ここで、 a_1 および a_2 は、2 種類の糸が smallest patch を占める面積の割合であり、隙間なく織り込まれている場合 $a_1 + a_2 = 1$ となる。smallest patch を構成する各糸 $j (= 1, 2)$ について接線分布を表す tangent curve を定義する（図 1(d) のグラフ）。糸 j の tangent curve を等間隔にサンプリングして得られる接線の集合を C_j とすると、糸 j における反射光の輝度 $L_j(\omega_o)$ は以下の式で計算される。

$$L_j(\omega_o) = \int_{\Omega} L_i(\omega_i) \sum f_s(\mathbf{t}, \omega_i, \omega_o) W(\mathbf{t}, \omega_i, \omega_o) \cos \theta_i d\omega_i$$

ここで、 Ω は半球上の方向の集合、 $f_s(\mathbf{t}, \omega_i, \omega_o)$ は糸の散乱関数であり、以下の式で計算される。

$$\begin{aligned} f_s(\mathbf{t}, \omega_i, \omega_o) &= (f_{r,s}(\mathbf{t}, \omega_i, \omega_o) + f_{r,v}(\mathbf{t}, \omega_i, \omega_o)) / \cos^2 \theta_d \\ f_{r,s}(\mathbf{t}, \omega_i, \omega_o) &= F_r(\eta, \theta_d, \phi_d) \cos\left(\frac{\phi_d}{2}\right) g^u(\theta_h, 0, \gamma_s) \\ f_{r,v}(\mathbf{t}, \omega_i, \omega_o) &= F \frac{(1 - k_d) g^u(\theta_h, 0, \gamma_v) + k_d A}{\cos \theta_i + \cos \theta_o} \end{aligned} \quad (5)$$

F_r はフレネル反射率、 η は糸の屈折率、 F は繊維に入射し再び出射する光のフレネル透過率の積を表している。 k_d は等方散乱パラメータ、 A はアルベド、 γ_s, γ_v はパラメータとする。

関数 $W(\mathbf{t}, \omega_i, \omega_o)$ はシャドウイング・マスキングの関数 $M(\mathbf{t}, \omega_i, \omega_o)$ とスクリーンへ投影される糸の長さを表す関数 $P(\mathbf{t}, \omega_i, \omega_o)$ から以下の式で計算される。

$$W(\mathbf{t}, \omega_i, \omega_o) = M(\mathbf{t}, \omega_i, \omega_o) \frac{P(\mathbf{t}, \omega_i, \omega_o)}{\sum P(\mathbf{t}, \omega_i, \omega_o)}$$

$M(t, \omega_i, \omega_o)$ と $P(t, \omega_i, \omega_o)$ は、表記簡略化のため

$$D(x, y, z) = (1 - x) \cdot y \cdot z + x \cdot \min(y, z)$$

とすると以下の式で表される。

$$M(t, \omega_i, \omega_o) = D(g(\phi_d, 0, \sigma), \max(\cos(\phi_i), 0), \max(\cos(\phi_o), 0))$$

$$P(t, \omega_i, \omega_o) = D(g(\psi_d, 0, \sigma), \max(\cos(\psi_i), 0), \max(\cos(\psi_o), 0))$$

ここで σ はパラメータであり 15 度から 20 度の値とする。

5 提案法

提案法では、環境照明 $L(\omega_i)$ をK個の球面ガウス関数 G の線形和で表す。

$$L(\omega_i) \approx \sum_{k=1}^K L_k G(\omega_i, \xi_k, \sigma_k) \quad (7)$$

ここで、 ξ_k, σ_k, L_k はそれぞれk番目の球面ガウス関数の軸ベクトル、ローブの鋭さ、係数を表し、球面ガウス関数の数 K は $K=10$ としている。以降表記簡略化のため $G_k(\omega_i) = G(\omega_i, \xi_k, \sigma_k)$ とおく。球面ガウス関数の線形和を式(5)に代入すると式(5)は以下の式に変形される。

$$L_j(\omega_o) = \sum_k \sum_{\Omega} \int_{\Omega} G_k(\omega_i) f_s(t, \omega_i, \omega_o) W(t, \omega_i, \omega_o) \cos \theta_i d\omega_i$$

以降、 $L_{j,k}$ を以下の式で定義し、 $L_{j,k}$ の計算方法について説明する。

$$L_{j,k}(t, \omega_o) = \int_{\Omega} G_k(\omega_i) f_s(t, \omega_i, \omega_o) W(t, \omega_i, \omega_o) \cos \theta_i d\omega_i$$

散乱関数 f_s は $\theta_i, \theta_o, \phi_i, \phi_o$ の関数として表現できるため球面ガウス関数との積の積分は解析的に計算可能であることが知られている[12]。一方関数 W は複雑な関数であり、球面調和関数や球面ガウス関数などの基底関数の線形和で表現することも解析的な積分を行うことも難しい(図2参照)。

提案法では、 $L_{j,k}$ を W と G_k の畳み込み積分を用いて以下の式で近似する。

$$L_{j,k}(t, \omega_o) = \frac{\int_{\Omega_i} G_k(\omega_i) W(t, \omega_i, \omega_o)}{\int_{\Omega} G_k(\omega_i) d\omega_i} \int_{\Omega} G_k(\omega_i) f_s(t, \omega_i, \omega_o) \cos \theta_i d\omega_i$$

ここで、 $T(t, \xi_k, \omega_o)$ を以下の式で定義する。

$$T(t, \xi_k, \omega_o) = \frac{\int_{\Omega_i} G_k(\omega_i) W(t, \omega_i, \omega_o)}{\int_{\Omega} G_k(\omega_i) d\omega_i}$$

T は糸の幾何形状にのみ依存するため、前計算しておきレンダリング時に参照する。しかしながら T は6次元の関数のため、前計算データが膨大となり、メモリ量が比較的少ない携帯端末上にデータを保存することが難しい。提案法では、 T を t と σ_t について離散化したテーブルを $T_{t, \sigma_t}(\xi_k, \omega_o)$ とし、特異値分解し圧縮する。実験により少ない数の特異値で元データを精度よく近似できることがわかった。

5.1 円形ガウス関数を用いた輝度計算

次に、球面ガウス関数 G と散乱関数 f の積の積分について述べる。本稿では説明簡略化のため表面散乱成分の計算についてのみ述べるが、体積散乱成分についても同様に計算することができる。 $L_{j,k}$ の後半の積分項に式(5)を代入したものを I_s とおく。

$$I_s = \int_{\Omega} G_k(\omega_i) F_r(\eta, \theta_d, \phi_d) \cos\left(\frac{\phi_d}{2}\right) g^u(\theta_h, 0, \gamma_s) \frac{\cos \theta_i}{\cos^2 \theta_d} d\omega_i$$

提案法は I_s を円形ガウス関数 g^c を用いて計算する。まず、 G_k は θ_i, ϕ_i についての円形ガウス関数の積で表すことができる。

$$G_k(\omega_i) = g^c(\theta_i, \theta_k, \sigma_k) g^c(\phi_i, \phi_k, \sigma'_k) \quad (10)$$

ここで、 θ_k, ϕ_k は ξ_k の角度、 $\sigma'_k = \sigma_k / \sqrt{\cos \theta_i \cos \theta_k}$ とする。さらに、ガウス関数 $g^u(\theta_h, 0, \gamma_s)$ を以下のように円形ガウス関数で近似する。

$$g^u(\theta_h, 0, \gamma_s) = \frac{\exp\left(-\frac{\theta_h^2}{\gamma_s^2}\right)}{\sqrt{\pi}\gamma_s} = \frac{\exp\left(-\frac{(\theta_i + \theta_o)^2}{(2\gamma_s)^2}\right)}{\sqrt{\pi}\gamma_s} = \frac{g(\theta_i, -\theta_o, 2\gamma_s)}{\sqrt{\pi}\gamma_s} \approx \frac{g^c(\theta_i, -\theta_o, 2\gamma_s)}{\sqrt{\pi}\gamma_s} \quad (10)$$

さらに、円形ガウス関数同士の積は定数倍の円形ガウス関数で表現されるため、

$$\frac{g^c(\theta_i, \theta_k, \sigma_k)g^c(\theta_i, -\theta_o, 2\gamma_s)}{\sqrt{\pi}\gamma_s} = \alpha \cdot g_i^c(\theta_i) \quad (11)$$

とおくと、 I_s は以下の式で計算される。

$$I_s \approx \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{g_i^c(\theta_i) \cos^2 \theta_i}{\cos^2 \theta_d} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g^c(\phi_i, \phi_k, \sigma_k') F_r(\eta, \theta_d, \phi_d) \cos \frac{\phi_d}{2} d\phi_i d\theta_i \quad (12)$$

まず ϕ_i についての積分を計算する。フレネル項 $F_r(\eta, \theta_d, \phi_d)$ をSchlickの近似を用いて以下の式で表現する。

$$F_r(\eta, \theta_d, \phi_d) \approx F_0 + (1 - F_0)(1 - \cos \theta_d \cos \phi_d)^5$$

ここで $F_0 = (1 - \eta)^2 / (1 + \eta)^2$ である。この近似式を展開すると、 $\cos\left(\frac{\phi_d}{2}\right)$ の5次の多項式として表現される。

$$F_r(\eta, \theta_d, \phi_d) \approx \sum_{l=0}^5 C_l(\theta_d, \eta) \cos^l\left(\frac{\phi_d}{2}\right) \quad (13)$$

ここで、 $C_l(\theta_d, \eta)$ は $\cos^l\left(\frac{\phi_d}{2}\right)$ の係数とする。式(13)を I_s に代入すると ϕ_i についての積分は以下の式で表現される。

$$\sum_{l=0}^5 C_l(\theta_d, \eta) \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} g^c(\phi_i, \phi_k, \sigma_k') \cos^{l+1}\left(\frac{\phi_i - \phi_o}{2}\right) d\phi_i \quad (14)$$

式(14)の積分は $\phi_o, \phi_k, \sigma_k'$ の3次元テーブル $T_1(\phi_o, \phi_k, \sigma_k')$ として前計算しておくことができる。次に、 θ_i についての積分を計算する。式(14)を I_s に代入すると以下の式に変形される。

$$I_s \approx \alpha \sum_{l=0}^5 \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} C_l(\theta_d, \eta) g_i^c(\theta_i) T_1(\phi_o, \phi_k, \sigma_k') \frac{\cos^2 \theta_i}{\cos^2 \theta_d} d\theta_i \quad (15)$$

ここで、 θ_i についての積分を計算するために、円形ガウス関数 $g^c(\theta_i)$ をガウス関数 $g(\theta_i)$ で近似する。次に、 $h(\theta_i)$ を以下の式で定義する。

$$h(\theta_i) = \frac{\cos^2 \theta_i}{\cos^2 \theta_d} C_l(\theta_d, \eta) T_1(\phi_o, \phi_k, \sigma_k') \quad (16)$$

関数 $h(\theta_i)$ は θ_i に関して滑らかに変化する関数であるため、ガウシアンとの積の積分は[12]の手法を用いて計算することができる。積分区間 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ を等間隔に分割し、分割区間 $[\theta_s, \theta_{s+1}]$ において $h(\theta_i) \approx a_s \theta_i + b_s$ と線分で近似する。線分の係数 a_s, b_s は $h(\theta_s), h(\theta_{s+1})$ から計算される。分割区間 $[\theta_s, \theta_{s+1}]$ における積分は以下の式で近似される。

$$\int_{\theta_s}^{\theta_{s+1}} h(\theta_i) g(\theta_i) d\theta_i \approx a_s \int_{\theta_s}^{\theta_{s+1}} \theta_i g(\theta_i) d\theta_i + b_s \int_{\theta_s}^{\theta_{s+1}} g(\theta_i) d\theta_i$$

これらの積分は解析的に計算することが可能となる。提案法では6分割で十分なレンダリング結果を得ている。

6 結果

提案法でレンダリングされた布の画像を図 3, 4, 5, 6, 7 に示す。実行環境はCPUがCore i7-2700K, GPUがGeForce 580GTXのPCである。画像サイズは640 × 480で、GLSLのフラグメントプログラムによりピクセル単位で輝度計算を行っている。レンダリング速度は4から8である。前計算時間は約3時間で特異値分解が大部分を占めている。 T_1 のデータ量は1.7MBである。 $T_{t,\sigma}$ のデータ量は布ごとに異なり、30MBから80MBである。 ξ_k, ω_o はともに半球上を48x48だけ離散化して $T_{t,\sigma}$ の特異値分解を行った。 $T_{t,\sigma}$ の元データ量が1つの布ごとに数GBであることを考えると、提案法は容量の少ない携帯端末でも十分格納できるサイズに圧縮できたことがわかる。

図3は関数Wと球面ガウス関数Gとの畳み込み積分による近似(下段)と参照画像(上段)を表している。入射照明として1つの球面ガウス関数 $G(\omega_i, (0,1,0), \sigma)$ を用いた。布を構成する糸はシルクショットファブリックであり、赤い糸と緑色の糸からなる。図3(a)(d)は $\frac{2}{\sigma^2} = 10$, (b)(e)は $\frac{2}{\sigma^2} = 100$, (c)(f)は $\frac{2}{\sigma^2} = 1000$ のときのレンダリング結果である。この図により、提案法による畳み込み積分による近似により参照画像と似た画像を生成することが出来る。

図4は、リネンのレンダリング結果を示している。(a)は提案法によるレンダリング結果、(b)は参照画像であり、 $6 \times 32 \times 32$ の環境マップの各方向からの寄与を累積して計算した結果である。(c)は(a)と(b)の差分画像の輝度を8倍した画像である。図4(a)のレンダリング速度は8fpsであり、図4(b)をレンダリングするのに要した時間は184秒であった。提案法により、参照画像と遜色ない画像をインタラクティブにレンダリングすることが可能である。リネンにおける $T_{t,\sigma}$ のデータ量は30MBである。

図5は、シルクのレンダリング結果を示している。(a)は提案法によるレンダリング結果、(b)は参照画像、(c)は(a)と(b)の差分画像の輝度を4倍にした画像である。シルクにおける $T_{t,\sigma}$ のデータ量は66MBである。

図7に散乱関数のパラメータ γ_s, γ_v を変更することにより、ポリエステル・サテン・シャルムーズで構成された布の外観を編集している(ハイライトが鋭くなっている)レンダリング例およびシルクとベルベットのレンダリング例を示す。

図8,9にシルクおよびポリエステル・サテン・シャルムーズの糸1における特異値の数のグラフを示す。球面ガウス関数のローブの鋭さを表す $2/\sigma_k^2$ が大きくなるに従い寄与率90%となる特異値の数が増加しているが、比較的少ない特異値数で高次元データを表現できている。

7 まとめ

本稿は、携帯端末でも動作するレンダリングアルゴリズムとして、布を対象として高速かつ高精細なレンダリングアルゴリズムを提案した。環境照明下における布の高速レンダリングおよび外観編集法を提案した。環境照明を球面ガウス関数の線形和で近似し、球面ガウス関数、重み関数および散乱関数の三重積積分を、球面ガウス関数と重み関数の畳み込み積分と球面ガウス関数と散乱関数の積分の積で近似することにより、インタラクティブレンダリングを可能にした。球面ガウス関数とガウス関数で表現される散乱関数の積分は、球面ガウス関数を円形ガウス関数の積に分解することで解析的に計算することが可能となった。これにより、散乱関数の各種パラメータをレンダリング時に動的に変更することができ、布の外観編集をインタラクティブな速度で行うことが可能となった。

提案法の欠点としては以下の点が挙げられる。まず、現時点では布による自己遮蔽や別の物体による遮蔽によってできる影を考慮していない点が挙げられる。また、重み関数を前計算するため、布の糸の構造をインタラクティブに編集することが難しい。

今後の課題として、現在PC上で基礎実験を行っており、iPadやandroid tabletといった携帯端末での動作実験があげられる。さらに、影の考慮および重み関数の前計算の高速化があげられる。

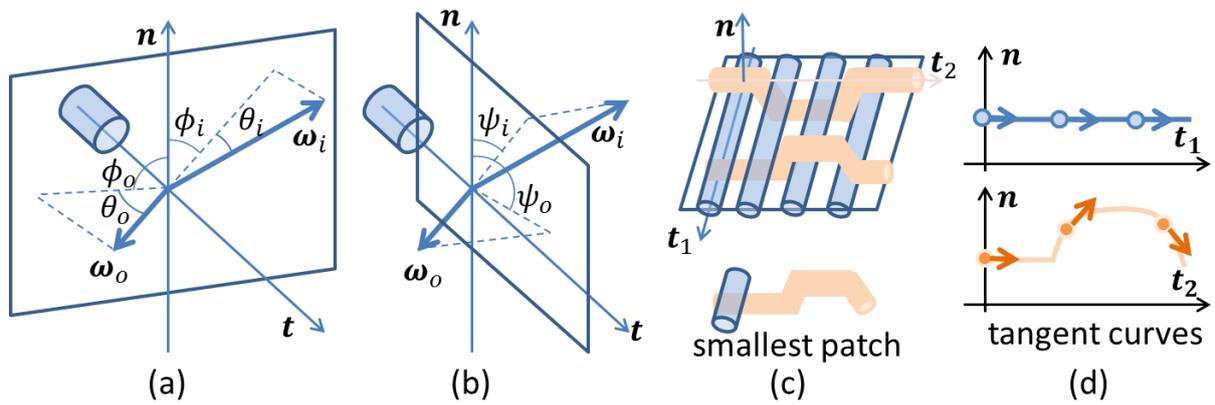


図1. マイクロシリンダモデルにおける角度の表記, smallest patch, tangent curves.

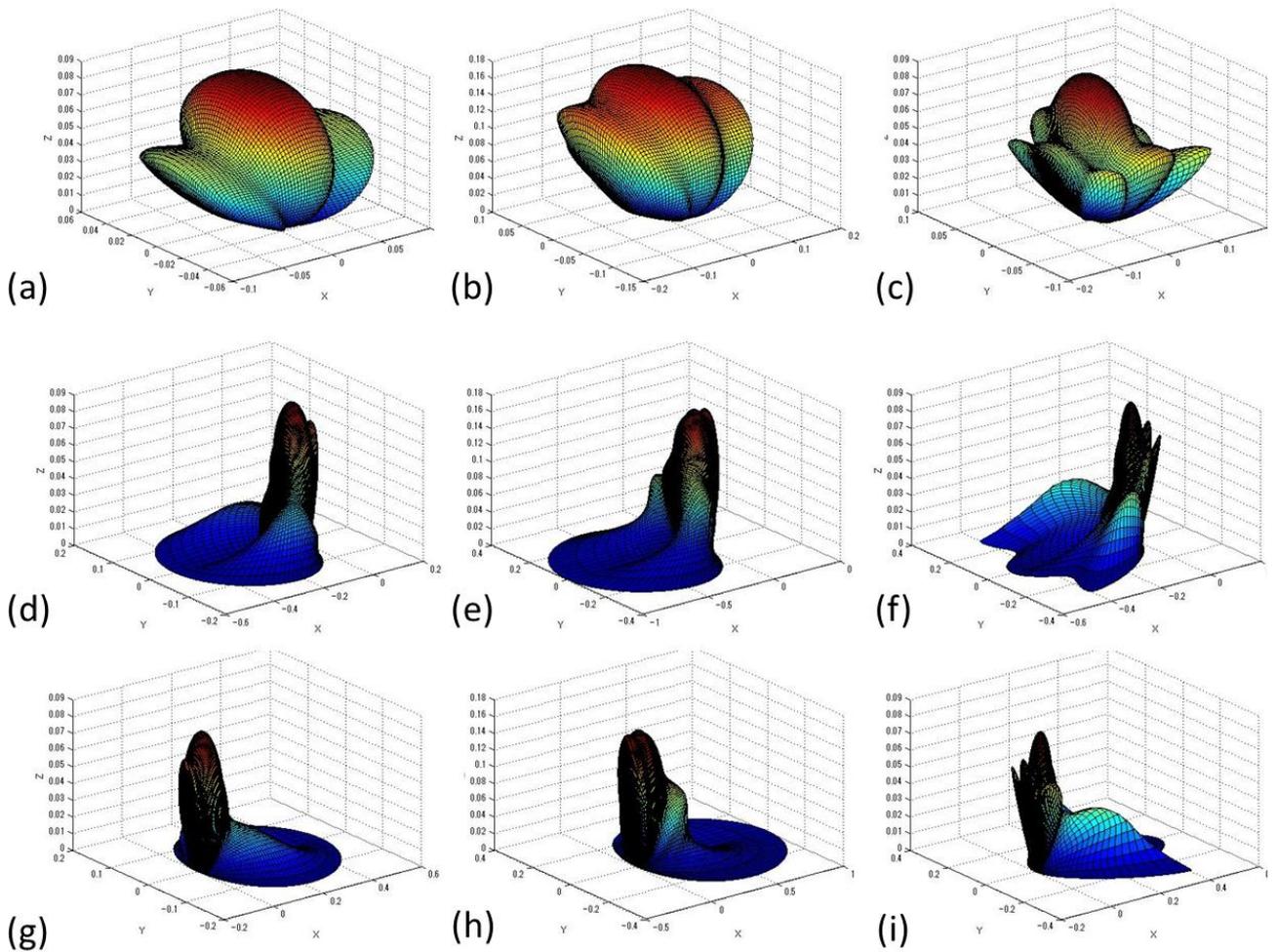


図2. 関数 W の可視化. 上段(a) (b) (c) は $t = (1, 0, 0)^T$, 中段(d) (e) (f) は $t = \left(\cos\left(\frac{25\pi}{180}\right), 0, -\sin\left(\frac{25\pi}{180}\right)\right)^T$, 下段(g) (h) (i) は $t = \left(\cos\left(-\frac{25\pi}{180}\right), 0, -\sin\left(-\frac{25\pi}{180}\right)\right)^T$, 1 列目は $(\theta_o, \phi_o) = \left(-\frac{\pi}{4}, -\frac{\pi}{3}\right)$, 2 列目は $(\theta_o, \phi_o) = \left(-\frac{\pi}{4}, 0\right)$, 3 列目は $(\theta_o, \phi_o) = \left(-\frac{\pi}{4}, \frac{\pi}{3}\right)$ のときの W を可視化したものである.

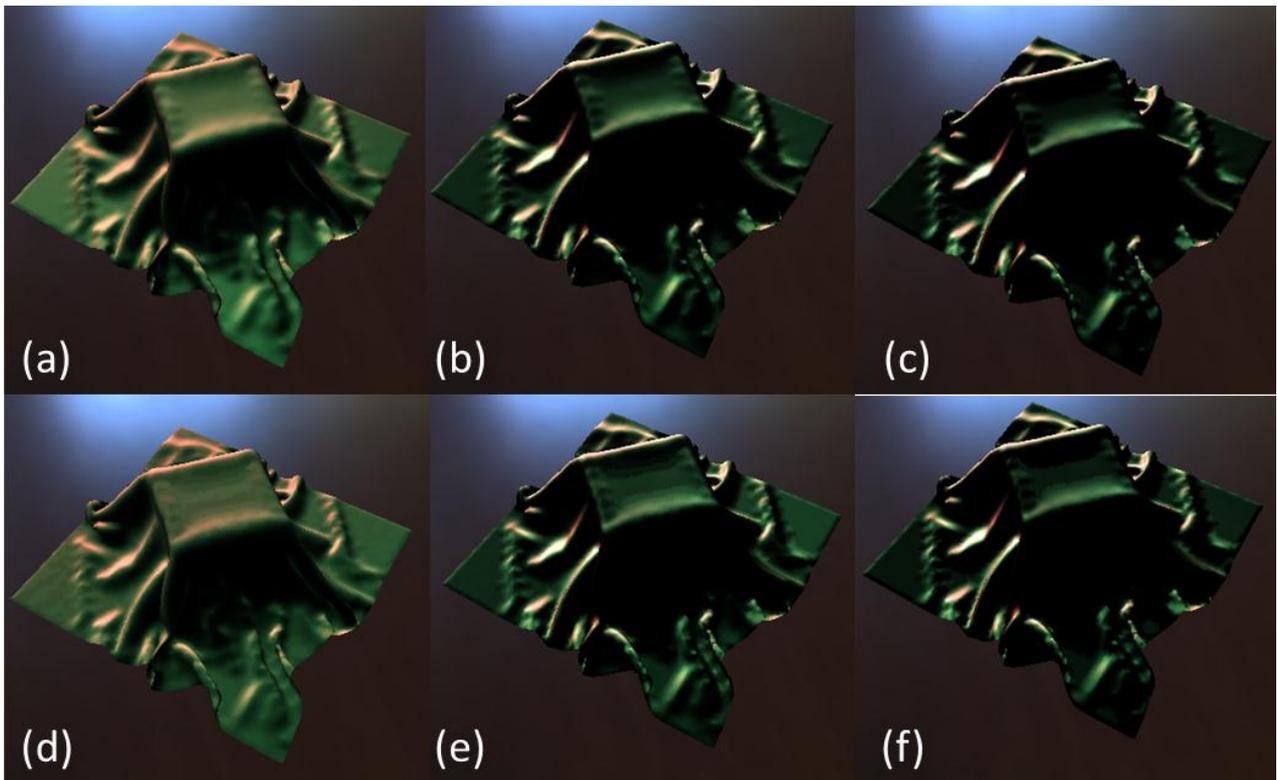


図 3. 赤い糸と緑の糸からなるシルクショットファブリックの例. 球面ガウス関数 $G(\omega_i, (0,1,0), \sigma)$ を入射照明としてレンダリングした例. 上段が参照画像, 下段は畳み込み積分結果を特異値分解して輝度を計算した例.

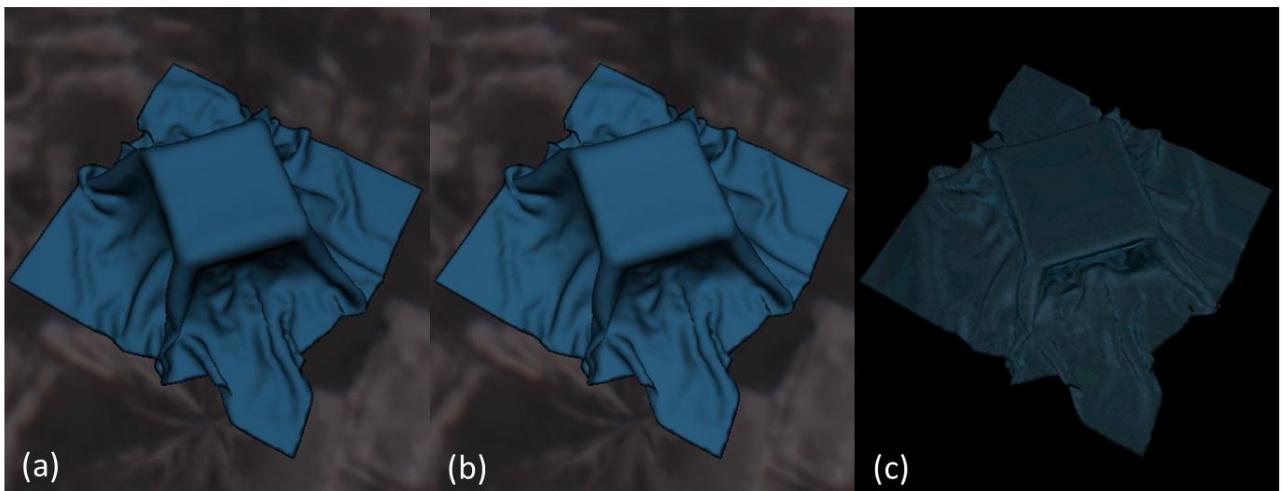


図 4. リネンのレンダリング結果. (a)は提案法によるレンダリング結果, (b)は参照画像, (c)は差分画像(輝度を 8 倍で表示).

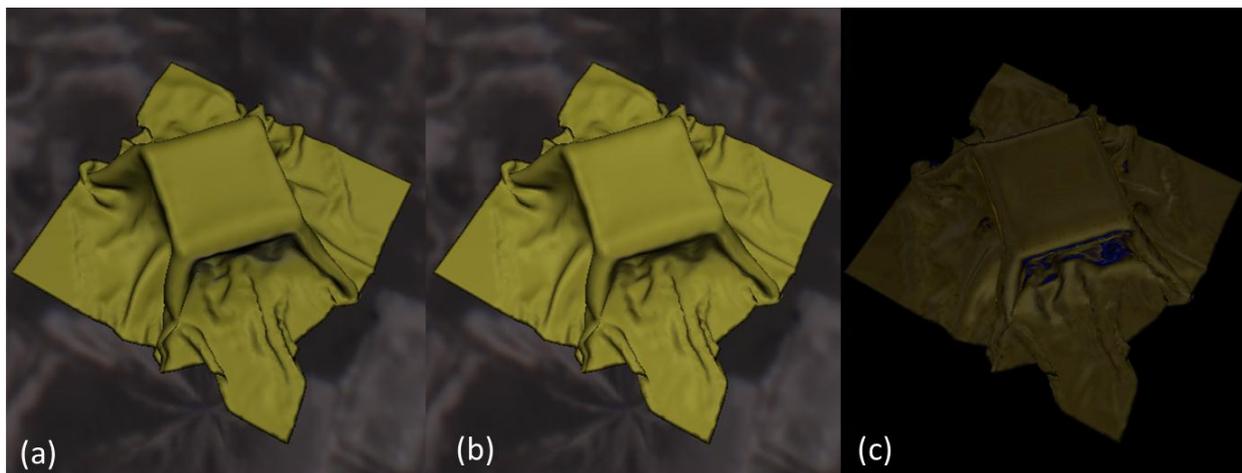


図5. シルクのレンダリング結果. (a)は提案法によるレンダリング結果, (b)は参照画像, (c)は差分画像(輝度を4倍で表示).

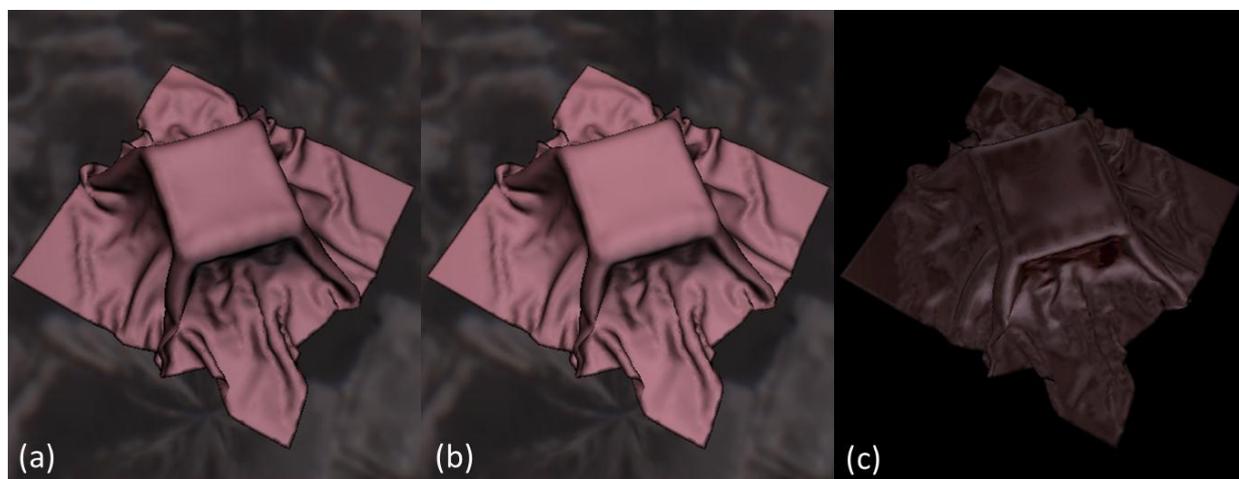


図6. ポリエステル・サテン・シャルムーズのレンダリング結果. (a)は提案法によるレンダリング結果, (b)は参照画像, (c)は差分画像(輝度を4倍で表示).

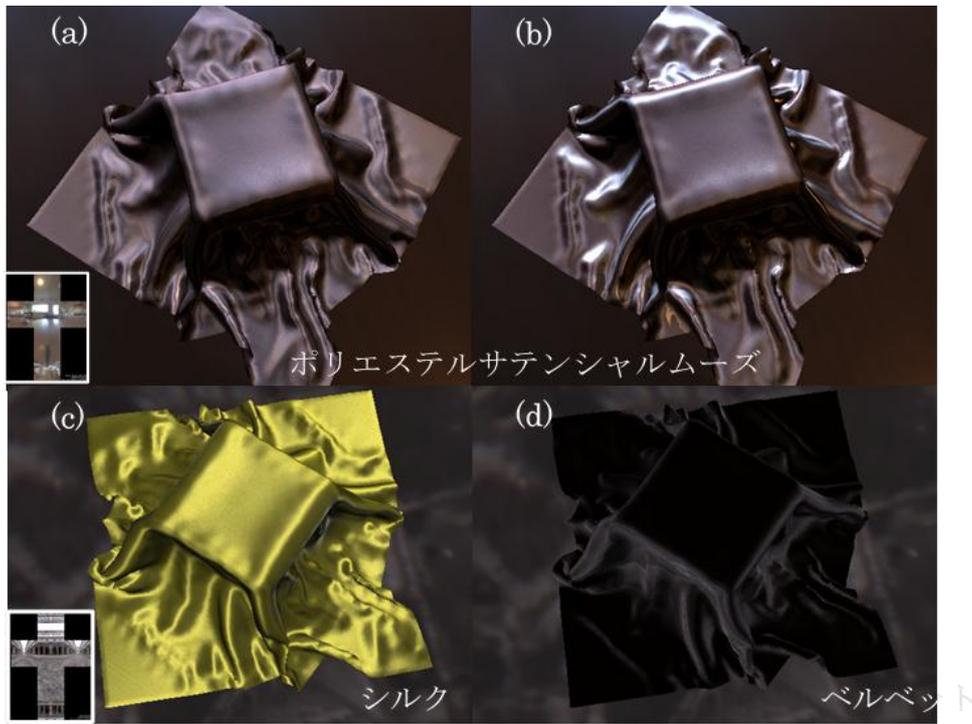
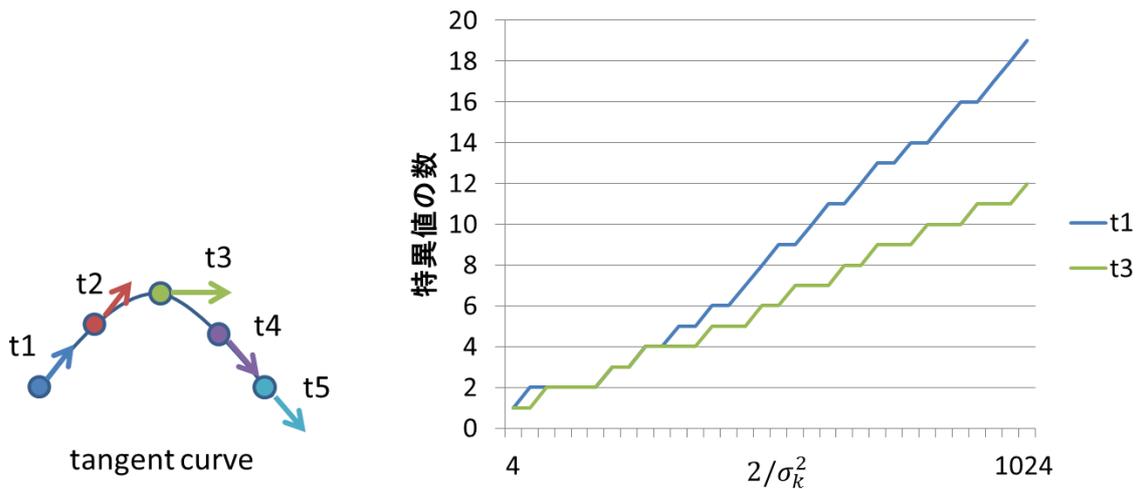


図7. レンダリング結果(上図：鋭さのパラメータを編集した例，左下に環境照明を表示).



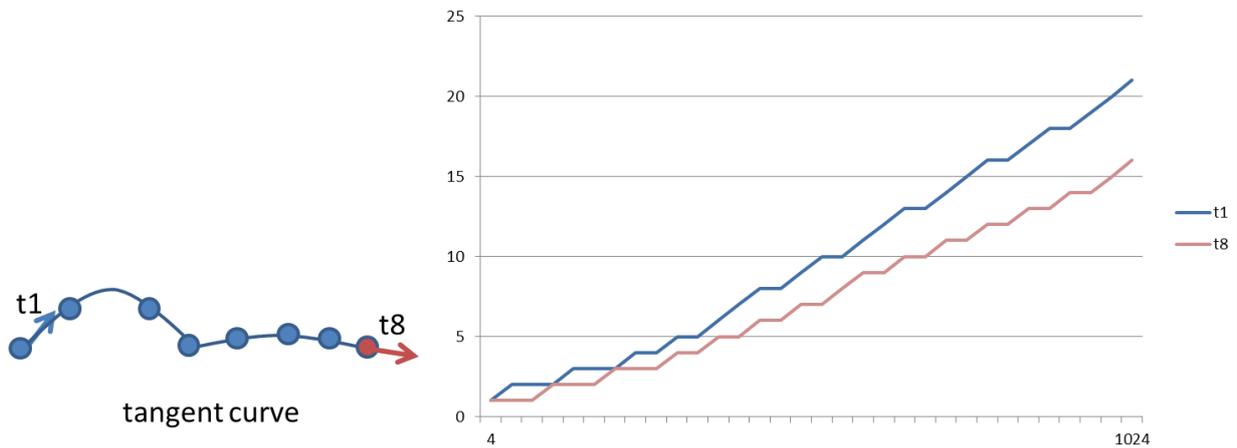


図 9. ポリエステルサテンシャルムーズの糸 1 における寄与率 90%以上となる特異値の数の片対数グラフ。横軸は $2/\sigma^2$ ，縦軸は特異値の数を表す。 $2/\sigma^2$ が大きい，すなわち球面ガウス関数のローブが鋭くなるに従い必要となる特異値の数が増加する。寄与率 90%となる特異値の数は t1, t2, t3, t4 および t5, t6, t7, t8 で等しいため，t1 および t8 を図示している。

7 参考文献

- [1] Neeharika Adabala, Nadia Magnenat-Thalmann, and Guangzheng Fei. Visualization of woven cloth. In Proc. of Eurographics Symposium on Rendering, pages 178–185, 2003.
- [2] Michael Ashikmin, Simon Premořze, and Peter Shirley. A microfacet-based brdf generator. In Proc. of SIGGRAPH'00, pages 65–74, 2000.
- [3] Yanyun Chen, Stephen Lin, Hua Zhong, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Realistic rendering and animation of knitwear. IEEE Transactions on Visualization and Computer Graphics, 9(1):43–55, 2003.
- [4] Katja Daubert, Hendrik P. A. Lensch, Wolfgang Heidrich, and Hans peter Seidel. Efficient cloth modeling and rendering. In Eurographics Workshop on Rendering, pages 63–70, 2001.
- [5] Piti Irawan and Steve Marschner. Specular reflection from woven cloth. ACM Transactions on Graphics, 31(1):11:1–11:20, 2012.
- [6] Iman Sadeghi. Controlling the Appearance of Specular Microstructures. PhD thesis, UC San Diego, 2011.
- [7] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Efficient and realistic visualization of cloth. In Eurographics Symposium on Rendering, pages 167–177, 2003.
- [8] Yuki Takeda, Huynh Quang Huy Viet, and Hiromi T. Tanaka, Image-based rendering of the anisotropic brdf of woven fabrics. In Eurographics Multimedia Workshop, pages 135–143, 2004.
- [9] Jiaping Wang, Peiran Ren, Minmin Gong, John Snyder, and Baining Guo. All-frequency rendering of dynamic, spatially-varying reflectance. ACM Transactions on Graphics, 28(5):133:1–133:10, 2009.
- [10] Jerry Weil. The synthesis of cloth objects. SIGGRAPH Computer Graphics, 20(4):49–54, 1986.
- [11] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. In Proc. of SIGGRAPH'92, pages 255–264, 1992.
- [12] Kun Xu, Li-Qian Ma, Bo Ren, Rui Wang, and Shi-Min Hu. Interactive hair rendering and appearance editing under environment lighting. ACM Transactions on Graphics, 30(6):173:1–173:10, 2011.
- [13] Ying-Qing Xu, Yanyun Chen, Stephen Lin, Hua Zhong, Enhua Wu, Baining Guo, and Heung-Yeung Shum. Photorealistic rendering of knitwear using the lumislice. In Proc. of SIGGRAPH2001, pages 391–398, 2001.
- [14] Takami Yasuda, Shigeki Yokoi, Jun-ichiro Toriwaki, and Katsuhiko Inagaki. A shading model

for cloth objects. IEEE Computer Graphics and Applications, 12(6):15–24, 2013.

- [15] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Building volumetric appearance models of fabric using micro ct imaging. ACM Transactions on Graphics, 30(4):44:1–44:10, 2011.
- [16] Shuang Zhao, Wenzel Jakob, Steve Marschner, and Kavita Bala. Structure-aware synthesis for predictive woven fabric appearance. ACM Transactions on Graphics, 31(4):75:1–75:10, 2012.

〈発表資料〉

題名	掲載誌・学会名等	発表年月
環境照明下における布のレンダリング	情報処理学会関西支部	2012年9月
環境照明下の布の高速レンダリングと外観編集	VC/GCAD 合同シンポジウム	2013年6月