

# SV-BRDF を用いた透明物体表面上の傷のレンダリング

藤 真太郎<sup>1,a)</sup> 佐藤 周平<sup>1,b)</sup> 岩崎 慶<sup>2</sup>

**概要:** 近年, 様々な物体をリアルにレンダリングする手法が提案されており, 物体表面の詳細な情報までリアルに表現できる. これらの手法の一つに, 事前に計算した 2 次元の BRDF から金属のひっかき傷を効率的にレンダリングする手法が提案されている. しかしこの手法は金属のみを対象としており, 他の物体への応用は考慮されていない. そこで本研究では, ガラスなどの透明物体に着目し, 従来手法をベースに透明物体表面の傷をレンダリングする手法を提案する. ガラスについての傷の表面は拡散反射性が増すとされているため, 金属を対象とした従来手法において SV-BRDF を求めるための事前計算を従来の鏡面反射ではなく拡散反射に変更することで手法を拡張した. しかし, そうすると反射するレイの数が増加し, 計算時間が膨大になってしまう. そこで, 1 方向にのみ反射するレイを複数回計算し, その平均をとることで高速化を図る.

**キーワード:** SV-BRDF, 傷のある材質, 透明物体, 拡散反射

## 1. はじめに

近年, コンピュータグラフィックス (CG) 技術は多くの映像作品に用いられている. それに伴い, よりリアルな画像や映像を作り出すためにレンダリング手法の研究が行われており, その中でも物体表面の質感を表現するための Bidirectional Reflectance Distribution Function (BRDF, 双方向反射率分布関数) を用いたレンダリング手法の研究が盛んに行われてきた. BRDF は, 物体表面上のある点において, レイの入射方向と出射方向の関係に応じた反射率を表す. BRDF を用いることで光の反射や物体表面の微小面で発生する物理現象を考慮したレンダリングが効率的に行えるようになった. また, BRDF の研究に伴い様々な BRDF の計算モデルが考案された. 拡散反射を考慮したモデル, 鏡面反射を考慮したモデルや異方性反射したモデルなど, レンダリングする物体や設定に応じて様々な BRDF の計算モデルが存在する. 一般的な BRDF は物体表面上の入射地点に関係なく一定である. 一方で, 物体表面の材質が粗さや汚れなどにより均一でないことから, 入射地点によって BRDF が変化する場合は Spatially Varying BRDF (SV-BRDF) と呼ばれる.

本研究では, SV-BRDF を用いて物体表面上の傷を表現することを目的とする. 傷のレンダリングをする場合, Ward らによる楕円ガウシアン法線分布関数 [3] などの従来手法では, レンダリングする物体とスクリーンとの距離が十分に離れていないと適用できないものが多く, 傷内部での光の相互反射を考慮していないため, 視覚的に尤もらしいレンダリング結果を得るのが難しかった. しかし, 物体表面で発生する光の相互反射を計算するには多くの計算コストを必要とする. それらの課題を解決するために, Raymond らは物体表面の傷を表現するための scratch BRDF を提案した [5]. scratch BRDF は傷を 2 次元の断面上で 2D レイトレーシングをすることによって求められるため, 光の相互反射を 3 次元での計算と比べ少ない計算コストで考慮することが可能である. また, scratch BRDF を事前計算し, レンダリングの際にそれを参照するため, ランタイムにおいて高速に計算が可能である. しかし, この手法は金属への適用のみを考えており, 透明物体など他の材質をもつ物体への適用は考慮されていない.

本研究では, Raymond らの方法をベースにガラスなどの透明物体表面における傷を表現するための scratch BRDF を提案する. Raymond らが提案した従来法では scratch BRDF の反射計算に鏡面反射を仮定していたが, 透明物体への適用を考慮するために拡散反射に変更して計算する. ただし, 拡散反射の場合鏡面反射と比べて, レイが複数方向に反射するために計算コストが高くなる. そのため, 1 方向にのみ反射するレイを複数回計算し, その平均をとる

<sup>1</sup> 富山大学  
University of Toyama

<sup>2</sup> 和歌山大学  
Wakayama University

a) s1870261@ems.u-toyama.ac.jp

b) ssato@eng.u-toyama.ac.jp

ことで高速化する。そして、scratch BRDF の計算後は、設定した傷のパラメータをもとに物体表面に適用するスクラッチレイヤを生成し、入射方向を基に scratch BRDF を参照して傷のレンダリングを行う。

## 2. 関連研究

### 2.1 分布ベースモデルを用いた微小面のレンダリング

ほとんどの BRDF は Torrance-Sparrow のマイクロファセット理論 [1] に基づいており、マイクロファセットと呼ばれる物体の微小面の Normal Distribution Function (NDF, 法線分布関数) として定義されることが多い。マイクロファセット理論の提唱に伴い、物体表面で発生する異方向性を計算して、微小な凹凸を表現するために Ward らによる楕円ガウシアン法線分布関数 [3] や Ashikhmin らによる物体表面で発生するフレネル効果を考慮する分布モデル [2] が提案された。フレネル効果とは、物体表面に光が入射するときに入射角によって光の反射量が変動する現象である。これらの手法により、物体の微小面の BRDF の分布を高速に計算することが可能になった。しかし、微小面の凹凸内部で発生する光の相互反射は無視されているため、物体表面と視点が離れていないと正しいレンダリング結果が得られないという制約があった。

### 2.2 SV-BRDF を用いた金属表面の傷のレンダリング

傷ついた物体のレンダリングを行う手法として、Yan らによる高解像度の法線マップからの鏡面反射面のレンダリング手法が知られている [4]。これは高解像度の法線マップから物体表面の法線分布関数を効率的に求めることにより、傷ついた金属の表面や水面の細かな波を表現することを可能にした。しかし、この手法では傷内部で起きる光の相互反射を考慮していないため、物体と視点の距離が近いと、正しいレンダリング結果が得られなかった。

上記の問題を解決するため、Raymond らは傷内部での相互反射を考慮した scratch BRDF により、金属表面において引っ掻いたような傷を表現するための方法を提案した [5]。この手法では、まず 2次元で定義された傷の断面上で 2D のレイトレーシングを行うことで事前に scratch BRDF のテーブルを計算する。2次元の傷の断面上でレイを入射し、傷から出射するまで鏡面反射させることで scratch BRDF が求められる。そして、物体のレンダリング時に事前に求めた scratch BRDF を参照することで物体表面上の傷を表現できる。この手法では、3次元空間上で計算される BRDF と比べて、低い計算コストで計算が可能である。しかし、この方法は金属物体のみに焦点が当てられており、透明物体など他の物体への適用は考慮されていない。本稿で扱う透明物体の一つであるガラスにおいて、引っ掻いたような傷は拡散反射性が増すといわれる [6]。このことから本研究では、事前計算で実行される 2D レイト

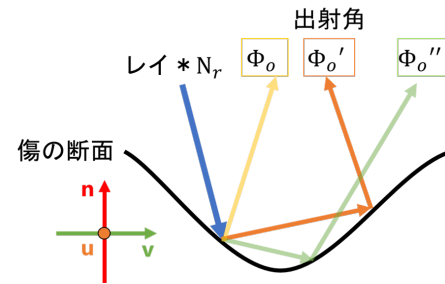


図 1: 2次元の傷の断面における拡散反射の様子

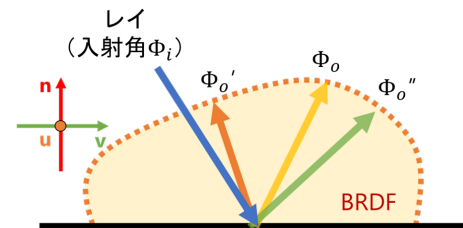


図 2: 2次元における平均化された BRDF

レーシングにおいて、反射の計算を拡散反射に変更することで透明物体表面上の傷においてもリアルなレンダリング結果を得られるかどうか検証する。

## 3. 提案手法

本手法では、まず Raymond らの手法 [5] と同様に、2次元で定義された傷の断面上で 2D のレイトレーシングを行うことで scratch BRDF を事前計算する。その後、あらかじめ設定された傷のパラメータをもとにスクラッチテクスチャを生成し、事前計算された scratch BRDF を参照して物体表面上にある傷のレンダリングを行う。本稿では、事前計算におけるレイの反射計算を鏡面反射から拡散反射に変更することで scratch BRDF を計算する。以下で各処理の詳細を述べる。

### 3.1 scratch BRDF の事前計算

この節では scratch BRDF の事前計算について説明する。2次元で定義された傷の断面上で 2D のレイトレーシングを行うことで 2D の scratch BRDF を計算し、その後、2D の scratch BRDF を用いて 3D の scratch BRDF を近似する。まず、Raymond らの手法と同様に以下の式で 2D の scratch BRDF である  $\rho_m(\phi_o, \phi_i)$  の計算を行う。3次元における計算を 2D レイトレーシングで近似できる理由の詳細については Raymond らの論文 [5] を参照いただきたい。

$$\rho_m(\phi_o, \phi_i) = \frac{M}{N \cos(\phi_o)} \quad (1)$$

ここで、 $\phi_i$  は物体表面への入射角、 $\phi_o$  が出射角である。また、 $M$  は  $\phi_o, \phi_i$  を離散化するパラメータ、 $N$  はサンプリングするレイの本数を表す。2D レイトレーシングを行

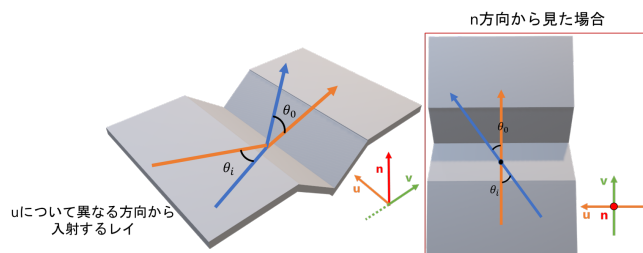


図 3: 3次元における物体表面の傷に入射するレイの様子

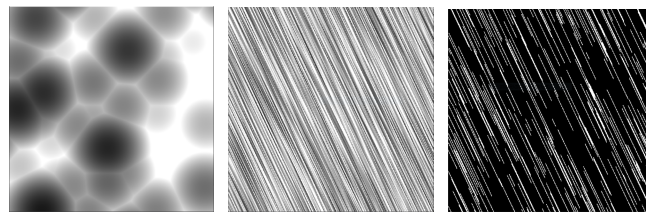
際には、 $N \gg M$  となるようにパラメータを設定する。本稿では、 $N = 10000$ ,  $M = 256$  とする。

本研究では透明物体でのレンダリングを考慮するために、傷に入射されるレイは従来の鏡面反射ではなく、拡散反射するものと仮定する。一般的な拡散反射はレイが物体に衝突するたびに複数本に分かれて反射するという性質がある。しかし、この性質に従って2D レイトレーシングを行うと計算コストが膨大になってしまうため、本研究では図1のように同じ入射方向、入射地点に対して、傷から出射するまでランダムに1方向にのみ反射する、 $N_r$ 本のレイを計算する。本稿では、 $N_r = 100$ とした。そして、それぞれの出射方向ごとに計算したレイの平均をとることで拡散反射の計算を近似する。図2はある入射角 $\phi_i$ に対する平均化された2Dのscratch BRDFを示す。点線で囲まれた黄色の領域がBRDFを表す。入射地点から点線までの距離が長い方向ほど、大きいBRDFの値を持ち、より多くの光を出射することを表している。

これまでの方法により2次元でのscratch BRDFが計算できたが、実際にレイが入射する方向は図3のようにu軸方向によっても異なるため、u, n平面における入射角である $\theta_i$ を考慮する必要がある。しかし、 $\theta_i$ を考慮するには3次元空間上でのレイトレーシングを行うことが必要となり、計算コストが高くなってしまふ。そこで本手法では、2Dレイトレーシングにより計算された2Dのscratch BRDFで3Dのscratch BRDFを近似する。まず傷の断面図をどの角度から切ったとしても同じ断面図と仮定する。これにより、 $\theta_i$ の値によって変化するのは、フレネル効果による光の減衰だけとなり、式(1)により計算されるBRDFに対して、 $\theta_i$ の値に対応したフレネル項をかけることで、 $\theta_i$ を考慮した計算が可能となる。ここでは、フレネル効果を考慮した $\rho_m$ を $\rho_s = \rho_m F$ とする。本稿では、フレネルの計算式は以下のSchlichの計算式を用いる[7]。

$$F(n, \cos \theta_i) = \frac{(n-1)^2 + 4n(1-\cos \theta_i)^5}{(n+1)^2} \quad (2)$$

ここで $n$ は物体固有の屈折率である。ただし、u成分が異なる全ての出射方向 $\theta_o$ についてscratch BRDFを計算すると、計算コストがかかり、占有するメモリ領域が大きくなってしまふ。そこで、 $\theta_o = 0, \frac{\pi}{3}, \frac{\pi}{2}$ のときの $\rho_s(\theta_o, \phi_o, \phi_i)$ のみを事前に計算して、3つのBRDFで3Dのscratch BRDF



(a) セルノイズ (b) 回転伸縮後 (c) 2値化後

図 4: スクラッチテクスチャの生成過程

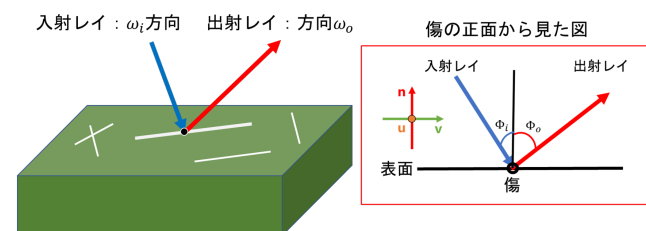


図 5: レイが物体表面の傷に入射する様子

を近似的に求める。3Dのscratch BRDFは以下の式で定義する。

$$\rho_s(\omega_o, \omega_i) \approx R_\gamma(\theta_o) \rho_s\left(\frac{\pi}{2}, \phi_o, \phi_i\right) + (1 - R_\gamma(\theta_o)) \rho_s(0, \phi_o, \phi_i) \quad (3)$$

ここで、 $\omega_i$  および  $\omega_o$  は、 $\phi$  と  $\theta$  で定義される入射および反射方向を表す。 $R_\gamma(\theta_o)$  は反射率を求めるための近似式であり以下のように定義される。

$$R_\gamma(\theta_o) = \left(\frac{2|\theta_o|}{\pi}\right)^\gamma \quad (4)$$

未知数である指数 $\gamma$ は式(3)において、 $\theta_o = \frac{\pi}{3}$ を代入して左辺を $\rho_s(\frac{\pi}{3}, \phi_o, \phi_i)$ とおき、式を変形してlogをとることで求められる。

### 3.2 スクラッチレイヤの生成

物体表面に対して傷を定義するために、傷の大きさや方向の異なる複数のスクラッチテクスチャを持つスクラッチレイヤ作成する。スクラッチテクスチャはセルノイズに対して回転や伸縮操作を行うことで作られる。セルノイズとはWorleyが考案した図4aの細胞のような見た目のノイズである[8]。このセルノイズに対して、テクスチャ画像の中心を軸にランダムな角度に回転し、ユーザにより任意に設定される傷の幅や長さをもとに伸縮を行って、図4bの画像を作成する。この回転操作における回転角の値は、scratch BRDFを参照するために必要な傷の伸びている方向を求めるために用いられる。その後、図4bの画像に任意の閾値によって2値化処理を行うことで、図4cのスクラッチテクスチャが生成される。白い部分が傷を表しており、レンダリングに利用される。また閾値が大きいほど、スクラッチテクスチャ上の傷の数が多くなる。回転操作を行わない





(a) ガラスの板



(b) ガラスのティーポット

図 6: 設定 1 でのレンダリング結果

場合は、傷が横方向に伸びるスクラッチテクスチャが生成される。

本手法では従来手法と同様に、傷の伸びている方向が異なる複数枚のスクラッチテクスチャを生成する。生成されたスクラッチテクスチャをそれぞれ 1 枚のスクラッチレイヤとして扱い、それぞれの回転角を基に傷の伸びている方向ベクトルを求めて保存しておき、scratch BRDF を参照する際に用いる。

### 3.3 scratch BRDF を用いたレンダリング

本節では、事前計算された scratch BRDF を用いてレンダリングを行う方法について述べる。まず生成したスクラッチレイヤの中から各入射地点で用いるスクラッチテクスチャを選択する。本稿では従来手法と同様に傷の占める割合が最も大きいテクスチャを採用する。しかし、複数の傷が交差している箇所の入射地点ではそこに含まれるテクスチャの中からランダムに決定する。計算に使用するスクラッチテクスチャを決定した後は、図 5 の入射地点に対して、傷の伸びている方向を  $v$  軸として入射角  $\phi_i$  を求め (図 5 の赤色矩形内参照)、これを基に事前計算した scratch BRDF を参照して、入射地点における出射輝度値を以下の式で計算する。

$$L_\gamma(\omega_o) = \frac{1}{I} \sum_{i=1}^I \frac{\rho_s(\omega_o, \omega_i)}{\rho_s(\frac{\pi}{2}, \phi_o, \phi_i)} L_i(\omega_i) \quad (5)$$

ここで、 $I$  はサンプル数、 $\rho_s(\omega_o, \omega_i)$  は 3D の scratch BRDF、 $\rho_s(\frac{\pi}{2}, \phi_o, \phi_i)$  は  $\theta_o = \frac{\pi}{2}$  のときの 2D の scratch BRDF、 $L_i(\omega_i)$  は入射輝度値である。

表 1: 実験に用いた傷のパラメータ

	傷の長さ	傷の幅	閾値
設定 1	0.3	0.001	0.1
設定 2	0.6	0.010	0.1

ただし、一般的なレイの入射方向をランダムにサンプリングする一様サンプリングでは、BRDF の小さい入射方向ばかりがサンプリングされる場合があり、効率が悪くなることがある。そのため、BRDF が大きくなる入射方向が優先的にサンプリングされる重点的サンプリングを用いて計算を行った。

## 4. 実験結果

提案法による結果を示す。実験では表 1 に示されているパラメータ設定を用いて、傷のあるガラスとティーポットのレンダリングを行った。レンダリングされる物体の材質はガラスを想定するため、物体の屈折率は 1.50 とする。まず、パラメータ設定 1 を用いてレンダリングを行った結果が図 6 である。ガラスの表面に対して傷をつけることができた。次にパラメータ設定 1 と比べると傷の長さや幅が大きいパラメータ設定 2 を用いてレンダリングを行った結果が図 7 である。

そして図 8 は、パラメータ設定 1 を用いてレンダリングした傷のあるガラスの板を拡大した画像 (8a) と実写のガラス傷 (図 8b) との比較である。比べてみると、細い直線状の傷は実写に近い見た目がえられた。また、光の相互反射を考慮したレンダリングであるため、物体と視点が近い





(a) ガラスの板

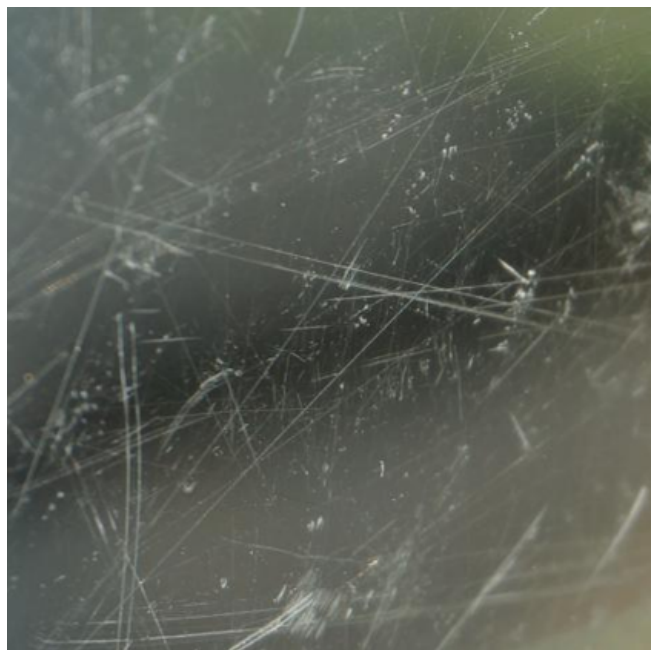


(b) ガラスのティーポット

図 7: 設定 2 でのレンダリング結果



(a) レンダリング結果 (設定 1)



(b) 実写のガラスの傷 (出典: <https://tropica.jp/2017/12/10/post-10050/>)

図 8: 実写のガラスの傷との比較

場合でも傷は不自然な見た目にならなかった。一方で、実写では 1 つの傷において途中でかすれているような部分があるなど、様々な見目が存在する。提案手法によりレンダリングした傷は色の变化している部分が見られるが、実写の傷と比べると一樣な見目であり、今後このような傷も表現できるよう改良していく予定である。

## 5. まとめと今後の課題

本稿では、金属表面の傷を対象としていた従来の方法をベースに、透明物体であるガラスの表面上の傷を表現するための scratch BRDF の事前計算方法を提案した。ガラスなどの透明物体の傷は拡散反射性が増すことに着目し、事

前計算における光の反射を鏡面反射から拡散反射に変更した。その結果、直線状の傷について実写のガラスの傷に近い結果が得られた。

一方で、点や曲線状の傷は現状の方法では表現できない。これは、セルノイズを操作することにより得られるスクラッチテクスチャが直線状の傷しか得られないことが原因である。セルノイズに対する操作の変更や別のノイズを用いるなど、直線状以外の傷の形状を設定できる枠組みを考案する予定である。

また4でも述べたように、本手法では実写の傷に見られるかすれ具合などは表現できず、傷の持つ質感に大きな変化を与えることができない。よって他のBRDF計算モデルに見られる粗さ(Roughness)などを考慮できる方法を考案することも今後の課題である。

今回、鏡面反射から拡散反射に変更したことにより、同じ点に対して入射させるレイの本数にもよるが、事前計算に必要な時間が従来手法に比べ長くなってしまった。今回の実験ではCPUでの実装であるため、GPUでの実装に変更することにより、事前計算時間の課題は改善されることが期待される。

## 参考文献

- [1] K. E. Torrance, E. M. Sparrow: “*Theory for Off-Specular Reflection from Roughened Surfaces*, Journal of the Optical Society of America, Vol. 57, Issue 9, pp. 1105-1114 (1967).
- [2] M. Ashikhmin, P. Shirley: “*An Anisotropic Phong BRDF Model*, Journal of Graphics Tools, Vol. 5, Issue 2, pp. 25-32 (2000).
- [3] G.J. Ward: *Measuring and modeling anisotropic reflection*, 19th annual conference on Computer graphics and interactive techniques, pp. 265-272 (1992).
- [4] L. Q. Yan, M. Hašan, W. Jakob, J. Lawrence, S. Marschner, R. Ramamoorthi: *Rendering Glints on High Resolution Normal Mapped Specular Surfaces*, ACM Transactions on Graphics, Vol. 33, Issue 4, Article No. 116, (2014).
- [5] Boris Raymond, Gael Guennebaud, Pascal Barla: *Multi Scale Rendering of Scratched Materials Using a Structured SV-BRDF Model*, ACM Trans. Graph, Vol. 35 (4), Article No. 57 (2016).
- [6] Yasushi Hasegawa, Seiichiro Kakubo: ガラス面の光学的性質についての研究 第1報ガラス面の変角光度計による反射率測定ならびに耐毀傷性, 分光研究, Vol. 7 (3), pp. 16-25, (1959).
- [7] C. Schlick: *An Inexpensive BRDF Model for Physically Based Rendering*, Computer Graphics Forum, Vol. 13, Issue 3, pp. 233-246 (1994).
- [8] S. Worley: *A Cellular Texture Basis Function*, SIGGRAPH1996, pp. 291-294 (1996).