

# 代表的な法線に基づいたプレフィルタリング環境マップの 補間による光沢面の効率的なレンダリング

茂木 俊哉<sup>1,a)</sup> 舟富 卓哉<sup>2,b)</sup> 向川 康博<sup>2,c)</sup> 高柳 亜紀<sup>3,d)</sup> 菊田 勇人<sup>3,e)</sup> 久保 尋之<sup>1,f)</sup>

## 概要：

本研究ではプレフィルタリング近似を用いた事前計算によるレンダリングの高速化を行う。プレフィルタリングでは、金属表面の法線分布を予め光源の環境マップに畳み込むことで、必要サンプリング数を減らしレンダリング時間を削減する。本手法では球面上に等間隔に代表的な法線を取り、各法線に応じた環境マップのプレフィルタリングを行うことで、歪みを抑えつつ任意の視点方向、異方性法線分布に対応できるようにした。代表的な法線は等間隔かつ反対方向とペアで取ることで、必要な環境マップのメモリ容量を半分に抑えることができる。さらに複数の環境マップを切り替える際に生じる境界線を、加重平均による補間を用いることで滑らかに接続した。実際のレンダリングはMitsuba2と実装のための独自のレンダラー、及びWebGLを用いて行った。結果的に従来の素朴なプレフィルタリングによる結果と比較し、リアルタイム性に影響なく精度を向上させることができた。

## 1. はじめに

近年デジタルツインを用いた製造業におけるデザイン開発の効率化や資源の節約が注目を集めている。デジタルツインとは、現実世界で困難な状況を仮想空間内のシミュレーションで行う概念である。デジタルツインの方法として物理ベースレンダリングがよく用いられ、シーン情報を入力することであらかじめモデル化された物理現象から、その最終的なシーンの見え方をシミュレートすることができる。可能な限り現実世界の物理法則を用いて計算を行うため、違和感を与えない結果を生成することができる。製造業の一つである金属の切削加工では、このデジタルツインを用いて切削用機械に入力された情報から、切削後の金属製品の審美性を事前に評価することが望まれている。実際にNC加工機に入力されたCADモデルと切削後の金属表面の外観を示す(図1)。

金属表面の微小形状を考慮したレンダリングには、直接微小形状データを用いてレンダリングする方法が考えられる。しかしこの方法では、微小形状の情報を外観に反映す



(a) CAD モデル

(b) 切削後の金属表面

図1: NC 加工機による切削

NC加工機と呼ばれる切削加工機への入力として用いたCADモデルと切削後の金属。切削にかける時間により精度が異なる。

るためにサンプリング数が多く必要であり、比例してレンダリング時間が長くなる。そこで本研究では、切削面の微小形状を考慮しつつ少ないサンプリング数で効率的にレンダリングすることを目的とする。そのためにはプレフィルタリング近似を用いて、レンダリングを効率化する方法を提案する。

プレフィルタリング近似は古くから知られている手法であり、インタラクティブなレンダリングに用いられることが多い[1]。しかし、その多くが長い事前計算時間のためにガウスの重み付き和などを用いた近似や、等方性の法線分布に限定したものであり、精度を大きく損なっている。本手法では、代表的な法線毎に環境マップをプレフィルタリングし、さらに複数の環境マップから補間を行うことで、任意の視点に対応可能な精度の高いプレフィルタリングの方法を提案する。また本手法は多重反射を考慮できないため、

<sup>1</sup> 千葉大学

<sup>2</sup> 奈良先端科学技術大学院大学

<sup>3</sup> 三菱電機株式会社

a) moteshunchiba329@chiba-u.jp

b) funatomi@is.naist.jp

c) mukaigawa@is.naist.jp

d) Takayanagi.Aki@ab.mitsubishi-electric.co.jp

e) Kikuta.Hayato@dp.MitsubishiElectric.co.jp

f) hkubo@chiba-u.jp

レイトレーシングでのレンダリングに用いる際は注意が必要である。

## 2. 関連研究

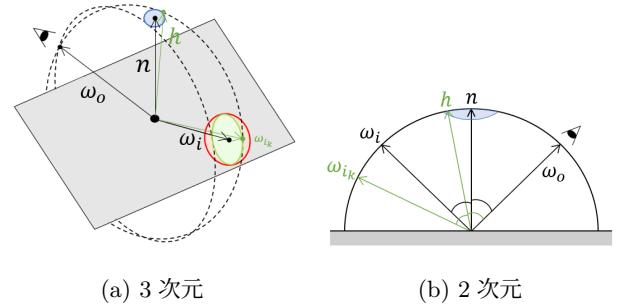
これまでにも、その特徴的な光沢反射や反射率の高さを考慮し、様々なアプローチで金属表面のレンダリングに関する研究が行われてきた。Yan らは視点に依存した光沢反射のピーク、ギラつきを局所的な微小形状の法線分布を用いることで再現した [2]。局所的な法線分布を用いることで、ギラつきのランダム性を再現しつつ、実際の外観から乖離しすぎない工夫がなされている。また、双方向反射率分布関数(BRDF)ではなくレンダラーのサンプリング方法自体を工夫することで、光沢反射を再現する方法も提案されている [3] [4] [5]。一般に光沢反射のレンダリングには重要度サンプリングが用いられる。しかしそれでも一定のランダム性があるため、忠実に微小形状を反映しようと時間がかかる。そこで規則的なサンプリングによりサンプリングする方向を予め固定することで、必要最低限なサンプリングが可能になる。

また、環境マップはインタラクティブなレンダリングにおいて、反射を近似する方法として適している。環境マップは Blinn と Newell によって基本的な仕組みが考案されて以来、球状マップや緯度経度マップ、半球の 2 枚組からなるデュアルパラボロイドマップなど、様々な形式が考案されてきた [6] [7]。本手法では環境マップのプレフィルタリングを用いているが、これは物体の BRDF を予め環境マップに畳み込むもので、Heidrich と Seidel は最も基本的な Phong モデルを近似した [1]。彼らは Phong モデルに則り、鏡面反射成分と拡散反射成分に対応するプレフィルタリングにより 2 枚の環境マップを生成した。また、kautz と McCool はこの Heidrich と Seidel のアイデアを拡張し、視点の変化による歪みを考慮した等方性 BRDF に対応するプレフィルタリングの方法を考案した [8]。

これらの研究に対し、本手法では複数の環境マップをプレフィルタリングすることで視点の変化による歪みを考慮し、また異方性法線分布に対応したプレフィルタリングの方法を提案する。

## 3. 提案手法

プレフィルタリングでは、全ての光源方向と視点方向(法線方向)に対応した事前計算が行えることが理想であるが、実際には計算コストの関係から難しいため、ある程度の間隔でサンプリングをする必要がある。本章では任意の法線分布関数  $D$  を用いて、サンプリングした代表的な法線ベクトルに 1 対 1 で対応するよう環境マップをプレフィルタリングする方法を説明する。この代表的な法線ベクトルは、球面上に等間隔且つ反対方向ベクトルとペアで取るようにする。このプレフィルタリングは計算コストが大きいため、



(a) 3 次元

(b) 2 次元

図 2: 各ベクトルの幾何学的関係  
(a) 青領域は法線分布を表す。実際は赤領域のように法線分布の形状がそのままフィルターカーネルとなることはなく、緑領域のように視点に依存して歪みを見せる。(b) 全てのベクトルが同一平面上に存在する場合の模式図。

本研究のレンダリング実験では代表的な法線は 42 に限定し、それらに対応する 42 枚の環境マップをプレフィルタリングした。次に、複数のプレフィルタリングされた環境マップをレンダリング時に切り替えることで境界が発生してしまう問題を、環境マップの加重平均による補間を用いて解決する方法を示す。本稿で用いる各ベクトルの幾何学的関係を図にまとめる(図 2)。

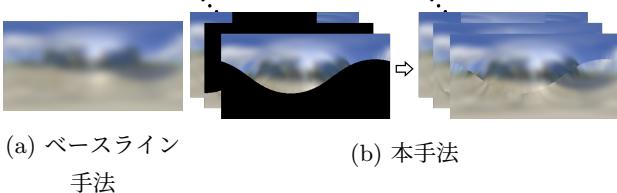
### 3.1 プレフィルタリング

本手法で用いるプレフィルタリングでは、BRDF ではなく法線分布の情報を予め光源の環境マップに畳み込む。これはマイクロファセット理論において、法線分布が BRDF を決定する主な要因であり、各微小面は鏡面反射であるという仮定に基づく。これにより、通常のレンダリングでは微小形状を含む解像度の高い形状データを用いる必要があるところを、解像度の低いマクロな形状データでレンダリングできるようになる。しかしこの方法は、金属表面の NDF が空間的に変化しないという制約を受ける。また、一般に 2 回以上の多重反射は拡散が非常に複雑になり、プレフィルタリングによる事前計算が厳しいため、本手法は単一反射のみを対象としている。以下の式は一般的な BRDF を用いたレンダリング方程式、及び完全鏡面反射の BRDF である。

$$L_r(\omega_o) = \int_H L_i(\omega_i) f_r(x, \omega_i, \omega_o) \omega_i \cdot n d\omega_i \quad (1)$$

$$f_r(x, \omega_i, \omega_o) = \frac{\delta(\omega_i - \omega_{or})}{\omega_i \cdot n} \quad (2)$$

$\omega_{or}$  は視点方向  $\omega_o$  の表面法線ベクトル  $n$  に対する正反射方向ベクトルである。BRDF は一様でありプレフィルタリングでは各マイクロファセットが完全鏡面反射であることを考慮すると、本手法で用いるレンダリング方程式は式(3)のように書ける。



(a) ベースライン  
手法

(b) 本手法

図 3: プレフィルタリングされた環境マップ

(a) は図 2において法線分布が赤領域のように歪まないと仮定した場合の簡易的なプレフィルタリングにより生成された環境マップ。 (b) は複数の代表的な法線方向に対応してプレフィルタリングされた環境マップである。各マップの半分の領域は基本的にレンダリングに使用されないため、代表的な法線を反対方向とペアでサンプルすることで、対応する 2 つのマップを 1 つにまとめることができる。

$$L_r(\omega_o) = \int_H L'_i(\omega_i) \frac{\delta(\omega_i - \omega_{or})}{\omega_i \cdot n} \omega_i \cdot n d\omega_i \quad (3)$$

$$= L'_i(\omega_{or})$$

$$L'_i(\omega_i) = \frac{\int_H L_i(\omega_{ik}) D(\mathbf{h}) d\omega_{ik}}{\int_H D(\mathbf{h}) d\omega_{ik}} \quad (4)$$

ここで、 $L'_i$  はプレフィルタリングされた環境マップ、 $D(\mathbf{h})$  は法線分布関数 (NDF) である。 $\omega_{ik}$  はサンプリングされた光源方向であり、本手法においては環境マップの各画素の方向ベクトルにあたる。 $\mathbf{h}$  は  $\omega_o$  と  $\omega_{ik}$  の正規化されたハーフベクトルである。式 (3) はプレフィルタリングを用いたレンダリング方程式であり、これは結果的にプレフィルタリングされた環境マップの  $\omega_{or}$  にあたる画素をサンプリングすることと等しいことが分かる。式 (4) では環境マップのプレフィルタリングを行っている。本手法では、単純に NDF との加重平均を取り正規化を行うことで NDF を光源に畳み込んでいる。

### 3.1.1 環境マップ

本研究では環境マップは緯度経度の形式を用いており、式 (4) を  $\theta, \phi$  を用いて変数変換したプレフィルタリングの式を以下に示す。環境マップの各画素は球面上における面積比率が異なるため、 $\sin \theta_{ik}$  項で補正を行っている。

$$L'_i(\theta_i, \phi_i) = \frac{\int_0^{2\pi} \int_0^\pi L_i(\theta_{ik}, \phi_{ik}) D(\mathbf{h}) \sin \theta_{ik} d\theta_{ik} d\phi_{ik}}{\int_0^{2\pi} \int_0^\pi D(\mathbf{h}) \sin \theta_{ik} d\theta_{ik} d\phi_{ik}} \quad (5)$$

本手法では 1 つの代表的な法線に対して 1 枚の環境マップのプレフィルタリングを行う。代表的な法線は球面上に等間隔にサンプリングされ、且つ反対方向とペアでサンプルされるものとする。各環境マップにおいてプレフィルタリングされる領域は、その代表的な法線方向を中心とする半球内にあたる部分のみであるため、反対方向の環境マップと組み合わせることで 2 つのマップを 1 つにまとめること

が出来る(図 3)。これによりメモリ容量を約半分に削減することが可能になる。

### 3.1.2 NDF

本手法のプレフィルタリングでは、環境マップに BRDF ではなく法線分布を直接畳み込んでいる。レンダリング実験では、法線分布に Beckmann 分布を用いており以下の式で表される。

$$D(\mathbf{h}) = \frac{\chi(\mathbf{h} \cdot \mathbf{n})}{\pi \alpha_x \alpha_y \cos^4 \theta_h} \exp \left( -\tan^2 \theta_h \left( \frac{\cos^2 \phi_h}{\alpha_x^2} + \frac{\sin^2 \phi_h}{\alpha_y^2} \right) \right) \quad (6)$$

$\chi(\mathbf{h} \cdot \mathbf{n})$  は  $\mathbf{h}$  と  $\mathbf{n}$  の内積が正のとき 1、負のとき 0 を返し、 $D(\mathbf{h})$  が  $\mathbf{n}$  を中心とする半球外に値を持たないようにする。 $\theta, \phi$  は  $\mathbf{n}$  に対する  $\mathbf{h}$  の角度である。また、 $\alpha_x$  と  $\alpha_y$  は直交する 2 方向の粗さパラメータであり、等しいときは等方性分布となる。

### 3.2 環境マップの補間

前節ではプレフィルタリングにより環境マップを複数生成したが、当然ながら生成できる環境マップの数は有限である。そのため、レンダリング時に参照する環境マップを切り替えることにより境界線が発生してしまう。そこで環境マップの補間ににより境界線を滑らかに接続する方法を提案する。

レンダリングする面の法線と最も近い代表的な法線に対応したプレフィルタリング環境マップを用いる、といった単純な切り替えでは、図 4(a) のように 1 対 1 で用いる環境マップが選択されるため、はっきりとした境界線が発生する。本手法の環境マップの補間では、周囲の環境マップの距離に応じた加重平均を取ることで境界線を発生させないようにする。具体的には以下の式に沿って補間を行う。法線  $\mathbf{n}$  を持つ表面のレンダリングに用いる環境マップを  $L_{in}$  とする。

$$L_{in} = \frac{\sum_k L_{in_k} (\mathbf{n} \cdot \mathbf{n}_k)^a}{\sum_k (\mathbf{n} \cdot \mathbf{n}_k)^a} \quad (7)$$

ここで  $n_k$  はサンプリングした代表的な法線であり、 $L_{in_k}$  はそれに対応してプレフィルタリングされた環境マップである。 $a$  は距離に応じて重みを決定するパラメータであり、値を大きく取ることでより近くの環境マップの比重を大きくする。例えば、より多くの環境マップを生成するほど代表的な法線同士の距離が短くなるため、 $a$  の値を大きく取る必要がある。レンダリング実験では経験的に  $a = 50$  として補間を行った。またこの補間は、代表的な法線を球面上に等間隔に取ることを前提としている。

## 4. レンダリング

この章では本手法の有効性を検証するために、レンダリ



(a) 単純な切り替え (b) 補間 ( $a = 10$ ) (c) 補間 ( $a = 50$ )

図 4: 環境マップの切り替えと補間

(a) 最も近い環境マップを用いる単純な切り替え. (b)(c) 環境マップの補間を用いた場合.  $a$  の値が小さいと (b) のように周囲の環境マップが混ざりすぎてしまう.

表 1: 同一シーンにおけるレンダリング時間の比較 (s)

| spp  | Mitsuba2 | 独自のレンダラー |
|------|----------|----------|
| 4    | 0.40     | 3.82     |
| 64   | 7.07     | 23.25    |
| 256  | 26.74    | 108.89   |
| 1024 | 141.49   | 449.86   |
| 4096 | 571.88   | 1762.62  |

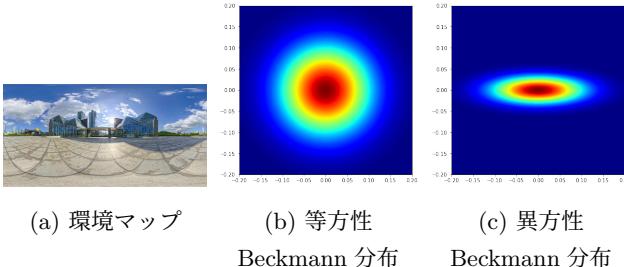


図 5: 環境マップと NDF

(a) は実験に使用した環境マップで解像度は  $200 \times 400$  である. (b) は等方性 Beckmann 分布 ( $\alpha = 0.1$ ) で, (c) は異方性 Beckmann 分布 ( $\alpha_u = 0.1, \alpha_v = 0.03$ ).

ング実験を行った. レンダリング対象の物体には円柱とスタンフォードバニーを用いた. 本手法は環境マップを用いているため, レイトレーシングによるレンダリングが可能で, Mitsuba2 によるレンダリング結果と比較する [9]. しかし本手法は多重反射を考慮できないため, 一般的な凹凸のある物体では誤差が大きい. そこで, プレフィルタリングした環境マップをテクスチャとして利用することを想定し, 独自のレンダラーにおいてテクスチャとしての実装, WebGL でのリアルタイムレンダリングの実装を行った.

#### 4.1 レンダリング環境

レンダリング実験を行った環境を紹介する. 今回レンダリングに使用した PC の CPU は「11th Gen Intel(R) Core(TM) i7-1165G7」, GPU は「Intel(R) Iris(R) Xe Graphics」である. また, Mitsuba2 では本手法の複数の

環境マップを用いる実装ができないため, 本手法を利用したレンダリングには独自のレンダラーを用いた. 両方のレンダラーにおいて GPU ではなく CPU を用いており, 独自のレンダラーのレンダリング時間は Mitsuba2 の約 3 倍である (表 1). 以降独自のレンダラーによるレンダリング時間を記載する場合, \*を設けている. また, 今回レンダリングに使用した環境マップの解像度は  $200 \times 400$  であり, NDF には等方性 Beckmann 分布 ( $\alpha = 0.1$ ) と異方性 Beckmann 分布 ( $\alpha_u = 0.1, \alpha_v = 0.03$ ) を使用している (図 5).

#### 4.2 レンダリング結果

レンダリング結果を図 6 に示す. ジオメトリにはスタンフォードバニーを用い, NDF には等方性 Beckmann 分布 ( $\alpha = 0.1$ ) を用いた. 出力画像解像度は  $1024 \times 1024$  である. Mitsuba2 ではこの BRDF がデフォルトで利用でき, これを用いてレンダリングした結果が図 6(a)である. 図 6(b)は独自のレンダラーにて対象の Beckmann 分布を実装し, 同様にレンダリングをしたものである. しかし多重反射は考慮せず, 単一反射のみを用いている. 図 6(c)は図 3(a)に示した簡易的なプレフィルタリング環境マップを用いてレンダリングをした結果である. 図 6(d), 図 6(e)は本手法のプレフィルタリングを用いたものであり, それぞれ環境マップの補間有り無しの結果である. 今回補間に用いた式 (5) における  $a$  の値は 50 としている.

Mitsuba2 による結果のみが多重反射を考慮しているため, 他の結果より明るく且つボケて見える. そのため, 本手法はレイトレーシングによるレンダリングには向いていないといえる. 多重反射を考慮しない場合であれば, 独自のレンダラーによる単一反射のみの結果を正解値として比較できるため, 定量評価として各手法との MSE を計算した. 本手法は視点の仰角に依存した NDF の歪みを考慮しているため, 全体的に精度が高く出ていることが分かる. 実際に赤枠の領域を比較しても, 補間を用いた本手法が最もよく近似できているといえる. ベースライン手法では反射による映り込みが雑なフィルタリングによって潰れてしまっているが, 本手法ではその点を考慮し映り込みをなるべく正確に保持できている. さらに, 図 6(d)では環境マップの切り替えによる境界線がはっきりと見て取れるが, 図 6(e)では補間ににより境界線が滑らかに接続されており, 環境マップの補間の有効性は明らかである. レンダリング時間に関しては, プレフィルタリングを用いていない図 6(a)と図 6(b)がサンプリング数の多さ故に長くなっている. 他のプレフィルタリング近似を用いた結果では, サンプリング数が少なくてノイズが発生しないため, どれも短い時間でレンダリングが可能である. ベースライン手法と比較して, 本手法は環境マップの切り替えや補間といった計算コストに僅かな差はあるが, 1 枚当たりのレンダリング時間に大きな影響はないといえる.

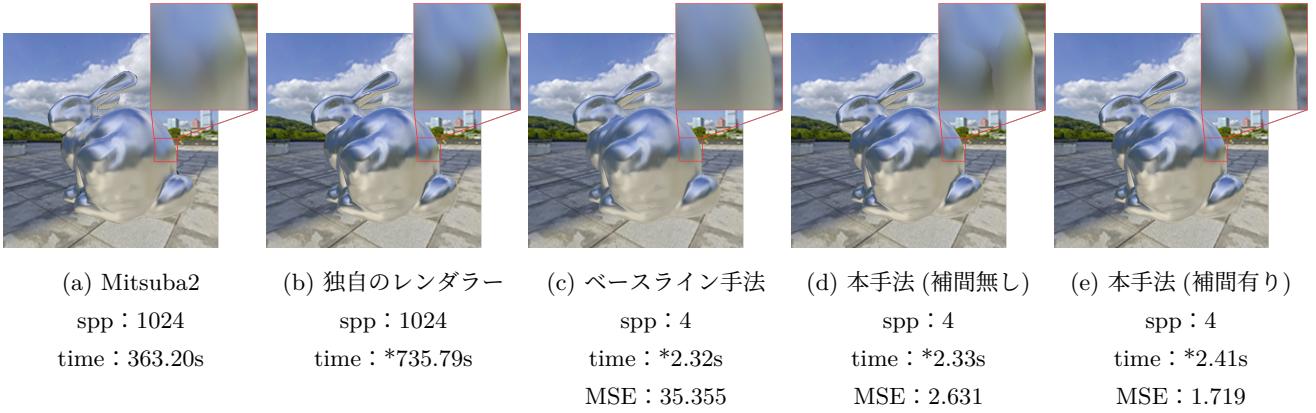


図 6: レンダリング結果

(a) は Mitsuba2 の BRDF の一つである roughconductor を用いたレンダリング結果. (b) は独自のレンダラーにおいて Beckmann 分布を実装した結果で、単一反射のみを計算している. (c), (d), (e) はプレフィルタリングを用いた結果で、各 MSE 値は (b) を基準とした平均二乗誤差である. spp はピクセル当たりのサンプリング数である.

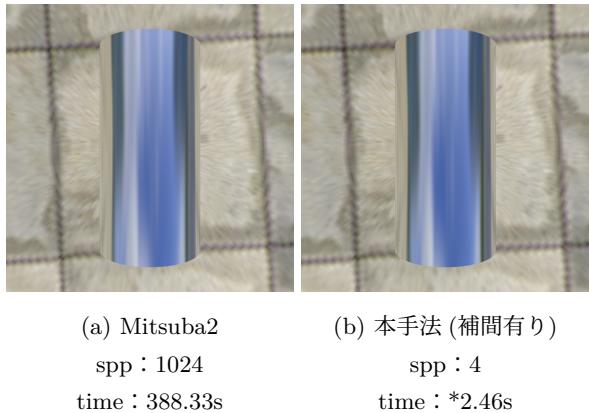


図 7: 異方性 NDF の場合のレンダリング結果

(a) は Mitsuba2 の BSDF の一つである異方性 Beckmann 分布を用いたレンダリング結果. ジオメトリが円柱であるため多重反射は起こらず正解値として比較できる. (b) は本手法のプレフィルタリングと補間を用いた結果. 異方性でも高い精度で比較できている.



図 8: WebGL における本手法の実装

それぞれ等方性 NDF のスタンフォードバニー、異方性 NDF の円柱を WebGL の GLSL にてプレフィルタリングを用いて実装した. どちらも常に 60fps 以上を保っていた.

次に異方性 NDF を持つ物体のレンダリング比較を行った(図 7). NDF には異方性 Beckmann 分布 ( $\alpha_u = 0.1, \alpha_v = 0.03$ ) を用いた. これらは非常に近い結果となり、本手法が異方性に対応していることが確認できる.

最後に WebGL を用いてリアルタイムレンダリングに

本手法を適用した(図 8). GPU は Intel Iris Xe Graphics であるが、視点をインタラクティブに変更してもフレームレートは 60fps 以上を常に保つことが出来ていた.

## 5. 議論

本手法を適用するまでの問題点として、事前計算時間が長い点が挙げられる. 本手法におけるプレフィルタリングでは、各画素において視線ベクトルやハーフベクトルといったベクトルをすべて計算することで、正確に NDF の歪みを考慮するような実装を行った. そのため各マップのフィルタリングにかかる時間は非常に長くなってしまい、大量に環境マップを生成するのは現実的でない. しかし、視点方向と法線方向に依存したフィルタカーネルの歪み方は常に幾何学的に一定であるため、この具体的なプレフィルタリング処理については十分に改善の余地があると考えられる. また、当然より多くの環境マップを生成し用いることで近似精度は高まると考えられるが、これは計算時間とのトレードオフであり、必要なメモリ容量も増えることを考慮すると数十枚程度が適切と考えている.

他にも NDF の空間的変化に対応できないために、リアリティを出すのが難しいといった問題が挙げられる. 実際の金属表面が空間的に一様な NDF を持つ、滑らかな外観になるということはほとんどなく、局所的な反射のピークやざらつき感が視認できることが多い. 本来の審美性評価という点においてはこれらの特徴が重要であるため、特徴を抽出し表現方法を工夫することで、効率的な外観再現が可能になると考えられる.

## 6. 結論

本研究では、金属切削面の微小形状から高速に外観を推定することを目的とし、代表的な法線に基づく複数の環境マップのプレフィルタリングとそれらの補間を用いること

で、効率的に光沢面をレンダリングする方法を提案した。実験の結果から、高速なレンダリング速度を保つつつ、ベースライン手法より高い精度の実現、異方性法線分布への対応が出来たといえる。また、プレフィルタリングで生成する複数の環境マップの必要メモリ容量を半分に抑える工夫も行った。課題として、プレフィルタリングを用いるとNDFの空間的变化に対応できず、リアリティを高めることが難しいという点や、環境マップのプレフィルタリングによる事前計算が長いといった点が挙げられた。

**謝辞** 本研究はJSPS科研費(JP20H05957), JST創発的研究支援事業(JPMJFR206I), JSTさきがけ(JPMJPR2025)の助成を受けたものです。

## 参考文献

- [1] Heidrich, W. and Seidel, H.-P.: Realistic, hardware-accelerated shading and lighting, *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 171–178 (1999).
- [2] Yan, L.-Q., Hašan, M., Jakob, W., Lawrence, J., Marschner, S. and Ramamoorthi, R.: Rendering glints on high-resolution normal-mapped specular surfaces, *ACM Transactions on Graphics (TOG)*, Vol. 33, No. 4, pp. 1–9 (2014).
- [3] Luksch, C., Tobler, R. F., Mühlbacher, T., Schwärzler, M. and Wimmer, M.: Real-time rendering of glossy materials with regular sampling, *The Visual Computer*, Vol. 30, No. 6, pp. 717–727 (2014).
- [4] Heitz, E.: Sampling the GGX distribution of visible normals, *Journal of Computer Graphics Techniques (JCGT)*, Vol. 7, No. 4, pp. 1–13 (2018).
- [5] Křivánek, J. and Colbert, M.: Real-time shading with filtered importance sampling, *Computer Graphics Forum*, Vol. 27, No. 4, Wiley Online Library, pp. 1147–1154 (2008).
- [6] Blinn, J. F. and Newell, M. E.: Texture and Reflection in Computer Generated Images, *SIGGRAPH Comput. Graph.*, Vol. 10, No. 2, p. 266 (online), DOI: 10.1145/965143.563322 (1976).
- [7] Heidrich, W. and Seidel, H.-P.: View-independent environment maps, *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware* (1998).
- [8] Kautz, J. and McCool, M. D.: Approximation of glossy reflection with prefiltered environment maps, *Graphics Interface*, Vol. 2000, pp. 119–126 (2000).
- [9] Nimier-David, M., Vicini, D., Zeltner, T. and Jakob, W.: Mitsuba 2: A Retargetable Forward and Inverse Renderer, *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, Vol. 38, No. 6 (online), DOI: 10.1145/3355089.3356498 (2019).