Vol.2016-CG-164 No.2
2016/9/21

法線分布関数フィルタリングの実装と評価

徳吉 雄介[1,a)]

概要：本稿は 2016 年に発表された Kaplanyan et al. の鏡面反射のアンチエイリアシング手法の実装と評価について報告する．この手法は視線方向と光源方向の中間ベクトルの微分を用いて法線分布関数をフィルタリングするというものである．しかし実際には微分の推定誤差が大きいため，望ましい結果を得られないことが多い．このことは Kaplanyan et al. も論文中で指摘しており，実用にはフィルタリングにバイアスを加えて正確さより安定性を優先することを推奨している．そこで我々はどのようなバイアスを加えるのが実用的なのか調べるため，実験を行った．その結果，ディファードレンダリング向けの近似手法が最も安定していることが分かった．

Evaluation of Normal Distribution Function Filtering for Specular Anti-Aliasing

Tokuyoshi Yusuke[1,a)]

1. 法線分布関数のフィルタリング

Kaplanyan et al. [4] は鏡面反射のアンチエイリアシングを行うために，視線方向と光源方向の中間ベクトルのスロープスペースにおける微分を用いて法線分布関数をフィルタリングする手法を提案した．しかしながら，実際には微分の推定は不安定で誤差が大きくなりやすい．そのため，実用にはフィルタリングにバイアスを加えることが推奨されている．そこで本稿ではどのようなバイアスを加えるのが実用的なのかを検証する．

1.1 手法

リアルタイムレンダリングにおける中間ベクトルの微分の推定は，ピクセルシェーダーで隣接ピクセルとの差分を求める命令（e.g., DirectX® では ddx/ddy）によって行われる．推定されたスロープスペースでの中間ベクトルの微分（ピクセル間の差分）を $\Delta\mathbf{h}_u, \Delta\mathbf{h}_v$ とし，ピクセルのフィルタリングカーネルを分散 $\sigma^2 = 0.25$ のガウス分布と仮定すると，スロープスペースでの中間ベクトルの共分散行列は以下の式で与えられる．

$$\mathbf{\Sigma} = \sigma^2 \begin{pmatrix} \Delta\mathbf{h}_u \\ \Delta\mathbf{h}_v \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \Delta\mathbf{h}_u \\ \Delta\mathbf{h}_v \end{pmatrix}. \tag{1}$$

次に物体表面の法線分布関数を Beckmann 分布 [1] と仮定する．Beckmann 分布はスロープスペースではガウス分布であり，鏡面の粗さパラメータの二乗 α^2 は法線分布の分散を二倍したものである．中間ベクトルの分布はマイクロファセットの法線分布なので，フィルタリングを解析的な畳み込みによって表すことができる．その結果，フィルタリング後の鏡面の粗さは以下の行列となる．

$$\mathbf{A} = \begin{pmatrix} \alpha^2 & 0 \\ 0 & \alpha^2 \end{pmatrix} + 2\mathbf{\Sigma}. \tag{2}$$

近年映画やゲーム等で一般的に使用されている GGX 分布 [7], [8] の粗さパラメータは，Beckmann 分布の粗さパラメータで近似可能である．そのため本稿ではこの粗さ行列 $\mathbf{A}$ を GGX 分布に適用する．

尚，粗さ行列を使って一般化した GGX 双方向反射率分布関数（bidirectional reflectance distribution function, BRDF）については元論文に記載されていない．そのため，報告者が導出を行った式を §A.1 に記述する．

[1] 株式会社スクウェア・エニックス
[a)] tokuyosh@square-enix.com

1.2 バウンディングボックスを使ったフィルタリング

前述の手法は理論的には正しいが，実際には誤差の影響を強く受けるので望ましい結果を得られないことがあると元論文で述べられている．そこでフィルタリングを安定させるために，Kaplanyan et al. はバイアスのある過剰なフィルタリングを使うことを推奨した．これは $\Delta\mathbf{h}_u, \Delta\mathbf{h}_v$ で作られる平行四辺形のバウンディングボックスの縦横の幅を異方性 BRDF の粗さパラメータとして使うというものである．

1.3 実験結果

実験結果を図 1 に示す．このフィルタリング手法によってエイリアシングを抑えることができた一方で，入射角が浅い場合に過剰かつ異方性の強いフィルタリングが行われており，これによって逆にアーティファクトが発生してしまっている．この原因はスロープスペースにおける中間ベクトルの変化量が入射角の正接に比例するためだと考えられる．そのため，入射角が浅く且つ法線がピクセル間で変化する場合，この中間ベクトルの変化量の誤差が極めて大きくなってしまうようである．こうしたアーティファクトは Kaplanyan et al. が主張するようにバウンディングボックスを使ったフィルタリングで軽減可能である．しなしながら，それでも依然として目立つアーティファクトが発生してしまっている．

2. 平均法線ベクトルを用いた近似

中間ベクトルを用いたフィルタリングは光源毎に計算する必要があり，光源数が多い場合に計算量が多くなる．また，ディファードレンダリングに用いることができないという問題がある．そこでディファードレンダリングに対しては，中間ベクトルの代わりに近傍 4 ピクセルの法線の平均を用いる近似が提案されている．そこでこの近似法について品質の評価を行ってみることにした．近傍 4 ピクセルの法線ベクトルの平均の求め方は元論文に記述されていないので，本稿で解説する．

2.1 実装

2.1.1 Pixel quad message passing

ピクセルシェーダーにおいて，隣接ピクセルの値を得る方法として pixel quad message passing 法 [5] がある．DirectX では隣接ピクセルの値との正確な差分を得る命令として，ddx_fine/ddy_fine 命令がある．これを用いることで，以下のように 4 ピクセルの法線の総和を求めることが可能となる．

```
float3 SumNeighborPixels( float2 pixelID, float3 normal )
{
  float2 dir = 1.0 - 4.0 * frac( pixelID * 0.5 );
  float3 horz = normal + ddx_fine( normal ) * dir.x;
  float3 vert = normal + ddy_fine( normal ) * dir.y;
  float3 diag = horz + ddy_fine( horz ) * dir.y;
  return normal + horz + vert + diag;
}
```

ただし，法線分布関数フィルタリングでは縦横のピクセルの変化量を見るので，対角方向のピクセルの法線は無くても良い．そのため本稿では以下のように省略した．

```
float3 SumNeighborPixels( float2 pixelID, float3 normal )
{
  float2 dir = 1.0 - 4.0 * frac( pixelID * 0.5 );
  float3 horz = normal + ddx_fine( normal ) * dir.x;
  float3 vert = normal + ddy_fine( normal ) * dir.y;
  return normal + horz + vert;
}
```

2.1.2 HLSL Shader Model 6.0

DirectX 12 の HLSL Shader Model 6.0 では隣接ピクセルの値を直接参照する命令が追加された．これを用いることで上述の pixel quad message passing 法よりも単純かつ精度の高い平均値を求めることが可能になると考えられる．今後はこの新命令を使った実装を行いたいと考えている．

2.2 実験結果

実験結果を図 2 に示す．驚くべきことに入射角が浅い場合でのアーティファクトの発生を抑えることができた．この理由としては，中間ベクトルと異なり平均法線は物体表面に対して浅くなりにくいためと考えられる．そのため，理論上は中間ベクトルを用いるのが正しいと考えられるが，平均法線ベクトルを用いた方が実用性は高いと言える．

3. 等方性近似

ディファードレンダリングでは，G-buffer [6] のメモリー使用量を抑えるため等方性の粗さパラメータにした方が望ましい．そこで Kaplanyan et al. が主張するように，バウンディングボックスを使ったフィルタリングにおいて粗さの最大値を用いた等方性フィルタリングを行ってみることにした．また追加実験として，共分散行列の最大の固有値を用いて粗さの推定を行ってみた．

3.1 実験結果

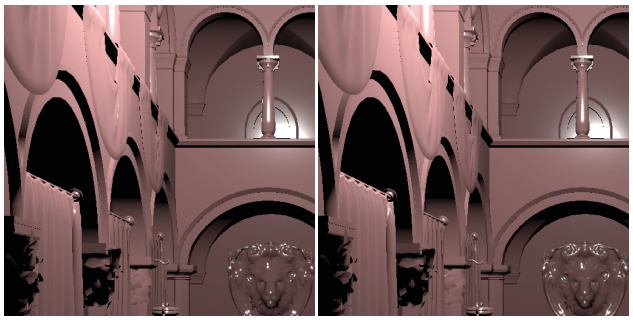
実験結果を図 3 に示す．異方性フィルタリングと異なり，両者とも視覚的に不快なアーティファクトが発生することはなくなった．一方で，全体的に過剰なフィルタリングが行われてしまっている．しかしながら中間ベクトルを用いた異方性のフィルタリングと違い，高周波のアーティファクトではないので視覚的に許容し易いと考えられる．Kaplanyan et al. が提案している粗さの最大値を用いるよりも，最大の固有値を用いて粗さを推定した方が過剰なフィルタリングを若干抑えることができた．

4. まとめ

本研究報告では Kaplanyan et al. の法線分布関数フィルタリングの実装と評価を行った．実験の結果，理論通りの実装では高周波のアーティファクトが発生することが確

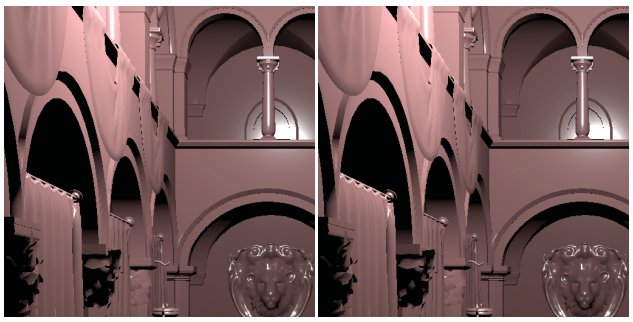
(a) フィルタリング無し　(b) 共分散行列を使ったフィルタリング　(c) バウンディングボックスを使ったフィルタリング

図 1: 中間ベクトルの微分を用いた異方性フィルタリング．入射角が浅い場合に不快なアーティファクトが発生している．



(a) 共分散行列を使ったフィルタリング　(b) バウンディングボックスを使ったフィルタリング

図 2: 平均法線の微分を用いた異方性フィルタリング．入射角が浅い場合のアーティファクトが中間ベクトルを使った場合に比べて大きく減少している．



(a) 共分散行列の最大の固有値を使ったフィルタリング　(b) バウンディングボックスの最大幅を使ったフィルタリング

図 3: 平均法線の微分を用いた等方性フィルタリング．両者とも不快な高周波アーティファクトをほぼ完全に除去できるが，過剰なフィルタリングによって強い低周波のバイアスが発生する．最大の固有値を用いた方が過剰なフィルタリングを抑えることが可能である．

認された．しかしながら，近似手法であるディファードレンダリング向けの等方性フィルタリングでは，副作用としてこれらの問題を回避可能であることが分かった．そのため，フォワードレンダリングであっても，このディファードレンダリング向けの近似を行った方が良い可能性がある．この近似手法は過剰なフィルタリングによって大きなバイアスを生み出すが，このバイアスは低周波である．このような低周波のバイアスは視覚的に不快になりにくいので，ゲーム等の娯楽を目的としたアプリケーションに対しては適している可能性がある．

謝辞　本論文のポリゴンモデルには以下の URL で公開されているデータを使用させて頂いた．データの提供に深謝する．

- F. Meinl and M. Dabrovic,
  URL: http://www.crytek.com/cryengine/cryengine3/downloads

付　　録

A.1　共分散行列ベースの GGX BRDF

マイクロファセット BRDF[2] は以下の式で与えられる．

$$\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) = \frac{G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) D(\boldsymbol{\omega}_m) F(\boldsymbol{\omega}_m \cdot \boldsymbol{\omega}_o)}{4|\boldsymbol{\omega}_i \cdot \mathbf{n}||\boldsymbol{\omega}_o \cdot \mathbf{n}|}.$$

ここで $\mathbf{n}$ は物体表面の法線，$\boldsymbol{\omega}_m = \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{||\boldsymbol{\omega}_i + \boldsymbol{\omega}_o||}$ は中間ベクトル，$D(\boldsymbol{\omega}_m)$ はマイクロファセットの法線分布関数，$G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m)$ は masking-shadowing 関数，そして $F(\boldsymbol{\omega}_m \cdot \boldsymbol{\omega}_o)$ はフレネル反射項である．Smith の masking 関数を $G_1(\boldsymbol{\omega}_o) = \frac{1}{1+\Lambda(\boldsymbol{\omega}_o)}$ とすると，height correlated masking-shadowing 関数は以下の式で与えられる．

$$G_2(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{\omega}_m) = \frac{1}{1 + \Lambda(\boldsymbol{\omega}_i) + \Lambda(\boldsymbol{\omega}_o)}.$$

GGX 分布で表されるマイクロサーフェスは形状／伸縮不

変（shape/stretch invariance）であり，粗さパラメータ α はマイクロサーフェスの伸縮のスケールである．そのため法線分布関数と Smith の masking-shadowing 関数は物体表面と中間ベクトルの伸縮から求めることができる [3]．既存の anisotropic GGX BRDF は接線空間での軸方向にマイクロサーフェスを伸縮させるモデルである．それに対し，共分散行列ベースの BRDF ではマイクロサーフェスの伸縮方向は粗さ行列 $\mathbf{A}$ の固有ベクトルであり，伸縮のスケールの二乗が $\mathbf{A}$ の固有値である．このことから，接線空間において $\boldsymbol{\omega}_m = [x_m, y_m, z_m]$ とすると形状不変より法線分布関数は以下の式となる．

$$D(\boldsymbol{\omega}_m) = \frac{1}{\pi\sqrt{\det(\mathbf{A})}\left([x_m, y_m]\mathbf{A}^{-1}[x_m, y_m]^{\mathrm{T}} + z_m^2\right)^2}.$$

また接線空間において $\boldsymbol{\omega}_o = [x_o, y_o, z_o]$ とすると，伸縮不変より，Smith の masking-shadowing 関数に対して以下の式が導出される．

$$\Lambda(\boldsymbol{\omega}_o) = -0.5 + \frac{\sqrt{[x_o, y_o]\mathbf{A}[x_o, y_o]^{\mathrm{T}} + z_o^2}}{2|z_o|}.$$

参考文献

[1] Beckmann, P. and Spizzichino, A.: *Scattering of Electromagnetic Waves from Rough Surfaces*, MacMillan (1963).
[2] Cook, R. L. and Torrance, K. E.: A Reflectance Model for Computer Graphics, *ACM Trans. Graph.*, Vol. 1, No. 1, pp. 7–24 (1982).
[3] Heitz, E.: Understanding the Masking-Shadowing Function in Microfacet-Based BRDFs, *J. Comput. Graph. Tech.*, Vol. 3, No. 2, pp. 48–107 (2014).
[4] Kaplanyan, A. S., Hill, S., Patney, A. and Lefohn, A.: Filtering Distributions of Normals for Shading Antialiasing, *HPG'16*, pp. 151–162 (2016).
[5] Penner, E.: Shader Amortization using Pixel Quad Message Passing, *GPU Pro 2*, A K Peters, pp. 349–367 (2011).
[6] Saito, T. and Takahashi, T.: Comprehensible Rendering of 3-D Shapes, *SIGGRAPH Comput. Graph.*, Vol. 24, No. 4, pp. 197–206 (1990).
[7] Trowbridge, T. S. and Reitz, K. P.: Average Irregularity Representation of a Rough Surface for Ray Reflection, *J. Opt. Soc. Am*, Vol. 65, No. 5, pp. 531–536 (1975).
[8] Walter, B., Marschner, S. R., Li, H. and Torrance, K. E.: Microfacet Models for Refraction Through Rough Surfaces, *EGSR'07*, pp. 195–206 (2007).