

実写画像に基づく 毛皮の特徴抽出と実時間描画手法

宇梶弘晃[†] 小坂昂大[†] 服部智仁[†]
久保尋之[†] 森島繁生[†]

CG において、毛皮の質感表現は非常に重要であるが、従来の実時間性に特化した描画手法では、動物が持つような不規則な毛並を再現することは難しかった。そこで、本研究では毛皮の実写画像を入力に用いることにより、毛並のリアリティを再現しながらも高速に動作する描画手法を新たに提案する。提案法では、毛皮を散乱媒質と仮定し、毛皮画像の輝度値と表面からの深度の対応をとったテクスチャを積層する事で、実時間内での毛皮の描画を行う。

Feature Extraction and Real-time Rendering of Fur from Realistic Image

Hiroaki Ukaji[†] Takahiro Kosaka[†] Tomohito Hattori[†]
Hiroyuki Kubo[†] and Shigeo Morishima[†]

Fur shading is an important factor on Computer Graphics. However, it is challenging to represent unsteady fur like animals in the conventional real-time fur shading methods. In this paper, we use a single realistic image of fur, and reproduce fur in real-time. We assume the fur to be a scattering medium, and laminate the texture took correspondence of the brightness of the realistic image and the depth from surface of fur.

1. はじめに

近年の 3DCG コンテンツにおいて、動物がキャラクターとして登場するものは数多く挙げられる。動物キャラクターの表面はその大部分が毛皮で覆われていることが多いため、毛皮の描画の質はキャラクターのリアリティを大きく左右する。既存の毛皮の描画手法は、リアルタイムでのレンダリングが可能なものから、計算時間を必要とするが写実性に特化したものまで多岐に渡る。しかしゲーム中など、実時間での処理が求められる場面においては、依然として毛並表現などの写実性に欠けている物が見られるのが現状である。

現在、ゲーム等で用いられている実時間処理可能な毛皮描画手法の一つとして Shell 法が挙げられる。Shell 法は、Shell Texture と呼ばれる毛皮の断面図の積層描画により毛を表現する手法で、比較的写実的な毛皮をリアルタイムで表現することが可能である。しかし Shell 法においては、目的の毛皮を表現できる Shell Texture の制作にコストがかかることが問題として挙げられる。ゲーム等においては、一種類の Shell Texture を用いることによりその簡略化を行なっているものの、一種類のみの Texture の積層では各層の点の繋がりが直線的になってしまうため写実性の表現に乏しい。積層方向の傾きを調整するなどの工夫でリアリティの向上は図れるものの、依然として毛並表現などにおいて課題が残されている。

そこで本稿では、一枚の毛皮実写画像から Shell Texture を生成することを目的とし、実写画像中に見られる毛並みを 3D オブジェクト上に再現することが可能な描画手法を提案する。本手法では毛皮の構造を簡易的にモデル化することにより、毛皮内部で起こる散乱特性を近似的に表現する。本稿における毛並の再現手法は毛の一本ごとの散乱現象をモデリングし、厳密なレンダリングを行う手法ではないが、実時間描画可能な手法である Shell 法の拡張として与えられるため、十分に高速な描画が可能となっている。

提案手法は実時間での描画を前提とし、その枠組の中で毛並表現による写実性の向上を図るものである。オフラインレンダリングで実現可能なレベルの写実性は比較対象とできないが、ゲームなどの実時間処理が求められる場面において、効率的な写実性の表現が期待できる。

本手法では入力として UV 座標が適切に貼られた 3D オブジェクトとともに毛皮の実写画像一枚を必要とし、入力画像の輝度値情報から毛皮における高低を推定し、このモデルから入力画像中の毛並が生成可能な毛皮のテクスチャを生成する。モデリングの際に輝度値情報を参照するため斑模様のような複数色により構成される毛皮には適応できないが、単色の濃淡のみで表される毛皮においてはリアルタイムでの毛並再

[†] 早稲田大学
Waseda University

現が可能となっている。また提案手法では描画の際に複数枚のテクスチャが用いられるが、フレームごとにシェーダに転送する画像を一枚に留めているため、簡易的な Shell 法と比較しても使用するメモリが大量に増えることなく、描画時間においても短時間での実装となっている。

2. 関連研究

毛皮の描画については過去に多くの手法が提案されており、大別するとリアリティを追求したオフラインレンダリングの手法と描画時間の高速化を図ったリアルタイムレンダリングの手法が挙げられる。

2.1 オフラインレンダリング

毛皮のレンダリングにおいて、正確に一本ごとのモデリングを行い厳密な光の散乱を計算する手法として、Ogaki らによる手法がある¹⁾。この手法は毛の細部構造である表皮質や毛髄質などから詳細なモデリングを行っているため、非常に写実性の高い手法となっている。

オフラインレンダリングの中でも処理計算の高速化・簡略化などの研究がなされているが、写実性を追求した手法である以上オフライン処理の枠を出ず、リアルタイムでの処理が求められるゲーム等への適用は不可能である。

2.2 リアルタイムレンダリング

実際にゲーム内などで毛皮の描画を行う場合は、実時間での処理が求められるシーンが存在する。現在、実際にゲーム等で用いられている実時間処理の代表的な手法として、Lengyel.らによる Shell 法と称される手法が挙げられる²⁾。Shell 法では図 1 に示される通り、あらかじめ Shell Texture と呼ばれる毛皮の断面図に相当するテクスチャを用意し、これを 3D オブジェクト上に法線方向に対して積層させるように描画することにより、各層間の点の連なりによる毛の表現を行なっている。この手法は 3D オブジェクトに対して積層数分のテクスチャマッピングを行うのみの処理となっているため実時間での計算が可能となっている。しかし Shell 法によって写実的な毛を生成する際には、目的の質感をもった毛皮を描画できる Shell Texture を積層数分だけ用意しなければならない、テクスチャの準備の段階でコストを要するという問題が挙げられる。一方で、一種類の Shell Texture のみを使用する単純な実装も可能ではあるが、この場合はオブジェクト表面に対して垂直方向に伸びる直線的な毛皮しか表現することができない。この問題の解決のため、現在のゲームなどでは Shell Texture に傾き情報を持たせ、この情報を元に描画をずらして行うことで毛並の表現を行う等の工夫が施

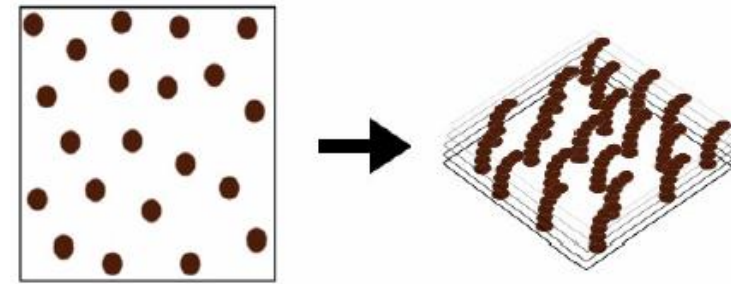


図 1 Shell 法

されている。しかしこれにより表現できる毛並は直線的なものに留まり、依然として実際の動物に表出するようなランダムな毛並は表現することができない。

そこで提案手法では、実際の毛皮の画像から指定した枚数の Shell Texture を自動的に生成し、これを積層状にマッピングすることにより入力画像に見られる毛並を 3D オブジェクト上に再現させる。各層のテクスチャが入力画像から自動的に生成されるため、従来法と比較してテクスチャを準備するコストが削減可能となっている。なお提案手法においては描画の際に複数のテクスチャを用いるが、毎フレームにシェーダに転送する画像は実質上入力画像一枚となっており、使用メモリの削減が可能となっている。

3. 提案手法

本手法ではまず、UV 座標が適切に張られた 3D オブジェクトについて、各頂点から法線方向に一定間隔でずらした点を積層数分だけ確保しておく。ここで用意した各層の頂点群に対し、後の行程で作成するテクスチャをマッピングしていく。ここで、入力として用いる単色の毛皮画像一枚に対して、画像における輝度値と実際の毛皮における高低の対応付けを行い、各層にマッピングするテクスチャを自動的に生成する手法について述べる。

3.1 毛皮における散乱現象の簡易的なモデル化

(1) 毛皮実写画像の特徴

毛皮の実写画像として、毛皮を真上から撮影した画像について考える。図 2 から見て取れる毛皮の特徴として、毛の一本一本が成す反射特性とは別に、表層近くでの散



図2 毛皮実写画像

乱が主な明領域や、底面近くでの遮蔽の影響が強い暗領域が存在することが挙げられる。これは毛皮上方から入射した光が毛皮によって散乱される現象に依るものである。すなわち、毛皮を上方から見た時の輝度値と、観察地点の毛皮内での高低には、相関関係が存在すると考えられる。

(2) 散乱モデルの表現

ここで毛皮の実写画像における輝度値と、実際の毛皮における高低を関連付けるモデルについて考える。毛の集合を毛皮底面からの厚み h_b を持った様な散乱媒質と仮定すると、上方から毛皮に垂直方向に入射した光は、毛皮内部を進む毎に指数減衰する。つまり実写画像における輝度値 I 、毛皮の表面からの深さを h とすれば、

$$I = I_{\max} e^{-ah} \quad (1)$$

とできる。ここで I_{\max} は入力画像における輝度値の最大値であり、本手法のモデルにおいては一番上の層に書き込まれるべき点の一つである。なお式(1)における減衰係数 a については、入力画像中の輝度値の最小値 I_{\min} と、毛皮の厚み h_b を用いることにより、

$$a = \frac{1}{h_b} \log \frac{I_{\max}}{I_{\min}} \quad (2)$$

と表される。ただし、毛皮上方から入射した光が毛皮底面に至るまで指数減衰するという仮定を行なっているため、 $I_{\min} \neq 0$ である。よって、式(1)で与えられる関数により、

入力となる毛皮の実写画像の輝度値を参照すれば、その点が毛皮のどの高さに存在していたか、つまり何番目のテクスチャに書き込まれるべきかを決定することが可能となっている。

3.2 Shell Texture の作成

実際に入力画像の毛皮実写画像から Shell Texture を作成する手法について述べる。まず入力画像をグレースケール化し、輝度値のヒストグラムを作成する。ここで、ヒストグラムから分布する輝度値の最大値 I_{\max} 、及び最小値 I_{\min} を取得するが、この時輝度値ヒストグラムの山から大きく外れた地点に微少なノイズが乗る可能性を考慮し、閾値処理などをしておくことが望ましい。次に、取得した I_{\max} 、 I_{\min} 及び h_b を用い式(1)で与えられる減衰関数を設定する。ここであらかじめ N 層のテクスチャにより毛皮を表現することを決めておけば、毛皮表面から n 番目 ($n=0,1,2,\dots,N$) の層は毛皮表面から距離

$$h = h_b \times \frac{n}{N} \quad (3)$$

に存在するため、この n 層目の深さに存在することが期待される毛皮は、これを式(1)に代入することにより算出される輝度値を持つ点である。なお $n=0$ 、つまり $h=0$ は毛皮表面を表している。

ここで式(1)～式(3)より

$$I(n) = I_{\max} e^{-\log \frac{I_{\max}}{I_{\min}} \cdot \frac{n}{N}} \quad (4)$$

とできる。以上から、各輝度値 $I(n)$ を区分とし、入力画像を N 枚のテクスチャに振り分けることで Shell Texture の生成を行う。

3.3 使用メモリの削減

本手法では、描画に用いるテクスチャが複数種存在する。しかし描画の際にこれらをフレーム毎にシェーダに転送する方法は、メモリを多く消費してしまうため好ましい手法ではない。実際に従来の Shell 法においても、各層のテクスチャを画像としてフレームごとにシェーダへ転送すると、その分使用メモリが増加するという問題点が挙げられる。しかし本手法においては、複数種のテクスチャが入力画像一枚から生成されていることを利用し、フレーム毎に転送する画像を入力画像一枚のみに留めることが可能である。

提案法では入力画像をあらかじめシェーダに転送しておくため、実際フレームごとに転送するものは式(3)の $I(n)$ で与えられる輝度値区分のみである。 n 番目の層の描画を行う際には $I(n-1)$ と $I(n)$ のみが転送され、シェーダ側で $I(n-1)$ から $I(n)$ の範囲に存在する輝度値の点を判定し描画を行う。

これによりフレーム毎に積層数分のテクスチャを転送する分の使用メモリを削減することができ、高速な描画が実現可能となっている。

4. 結果

4.1 正方形ポリゴンへの適用

図3に、入力画像及び正方形ポリゴンに提案手法を適用した結果を示す。結果については、8層の Shell Texture による描画を行なっている。図中に見られるように、入力画像に見られる毛並が描画結果においても再現されていることがわかる。なお今回、入力画像として 512×512 [pixel] の画像を用いた。

4.2 3D オブジェクトへの適用

図4に2種類の3Dオブジェクトに提案手法を適用した結果、表1に2つのモデルの頂点数、Shell Texture の積層数、及び描画の際のフレームレートを示す。図4に見られるように、入力画像に見られる毛並が3Dオブジェクト上に表出している様子がわかる。各モデルにおける処理時間については表1の通りで、実時間での処理が実現している。なお、フレームレートの測定を行った動作環境として、CPU : Inter® Core™2 Duo 2.66GHz, GPU : NVIDIA GeForce GTX 285, DirectX 11 を用いた。

表1 描画結果に対する諸数値

モデル	頂点数	積層数	フレームレート [fps]
Teapot	2082	15	1648.3
Bunny	34832	15	137.0

また、本手法による描画結果を複数の視点から見た結果を、図5に示す。図中の通り、入力画像中に見られる毛並が、3Dオブジェクト上全面に反映されている様子が見取れる。なお視点は Bunny モデル正面を 0° とし、真上方向を軸とし 90° ごとに視点を変更し描画を行ったものである。

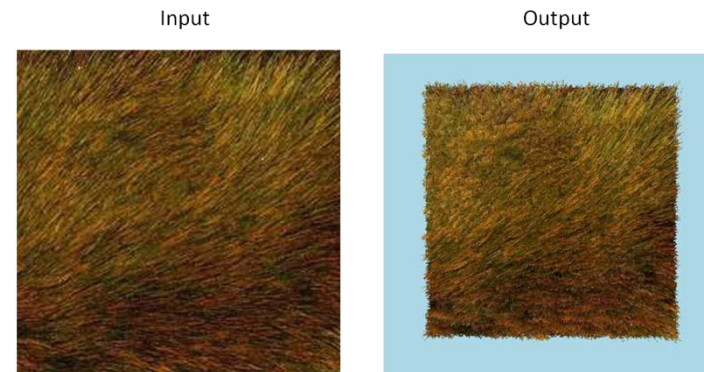


図3 正方形ポリゴンへの適用

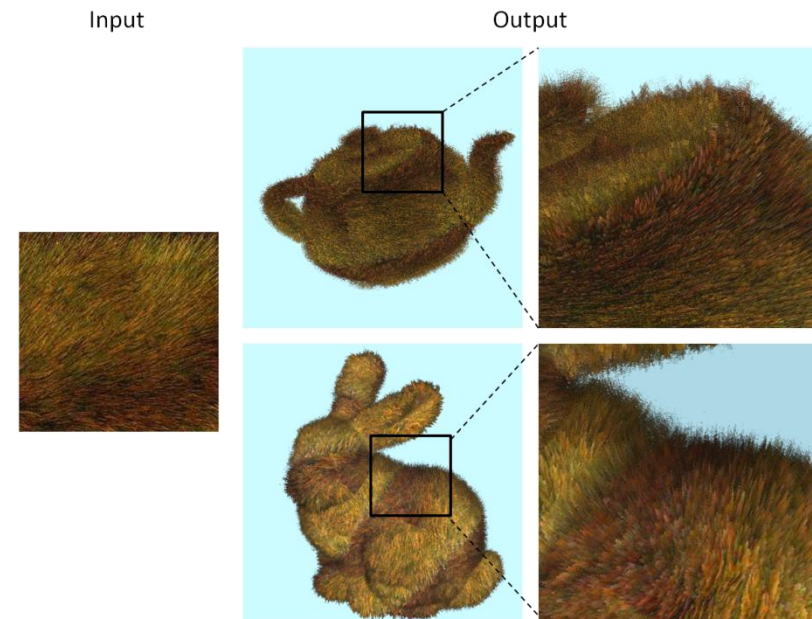


図4 Teapot モデル, Bunny モデルへの適用

4.3 複数の入力画像に対する描画結果

図6に複数の入力画像に対する描画結果を示す。描画結果において、各入力画像に対応した毛並りが3Dオブジェクト上に再現されていることがわかる。結果については、4.2節と同様に34832頂点のBunnyモデルを用い、15層のShell Textureを積層させることにより描画を行なっている。

5. 考察

4.1節の結果より、本手法は実写画像に見られる毛並りをShell法により立体的に再現できていると言える。実際4.2節、及び4.3節の結果から、3Dオブジェクトに対し適用した場合においても、入力画像に対応した毛並りが反映された描画結果が出力されている。

各描画結果に示されるように、本手法により入力画像中の毛並り情報をShell法によって任意のオブジェクト上に再現させることが可能である。実際、本稿の結果においては3種類の入力画像に対し、各々適切にShell Textureの生成および3Dオブジェクト上への毛並りの再現が行われている。

本手法では毛並りの表現を行うため、積層するテクスチャは各層異なるものを使用している。しかし各層のテクスチャは入力画像一枚から生成されているため、描画の際に保持するテクスチャは実質入力画像一枚のみに留めることが可能となっている。

6. 今後の課題

本手法の課題としては、以下のような事柄が挙げられる。

- 毛皮の厚みについて実写との対応がとれていない
- テクスチャ間の繋ぎ目が目立つ
- 一部毛並りの違和感が見られる
- ライティングの考慮がなされていない

まず実際の毛皮の厚みとの対応関係について、現在の描画結果では、毛皮の厚みはShell Textureの積層幅と積層枚数の積で与えられる。つまり頂点の設定、及び積層枚数の指定により決定されるため、入力の実写画像には依存しない。毛皮の実写画像一枚から元の毛皮の厚みを推定することは困難なため、入力に応じて積層幅を決定することは難しいが、積層幅をユーザー側で制御できるようにし、より実物に毛皮の質感を近づけるよう調整を行う余地を持たせるなどといった改善は可能であると考えられる。

次にテクスチャ間の繋ぎ目について、4.2節の茶色の入力画像によるBunnyモデル

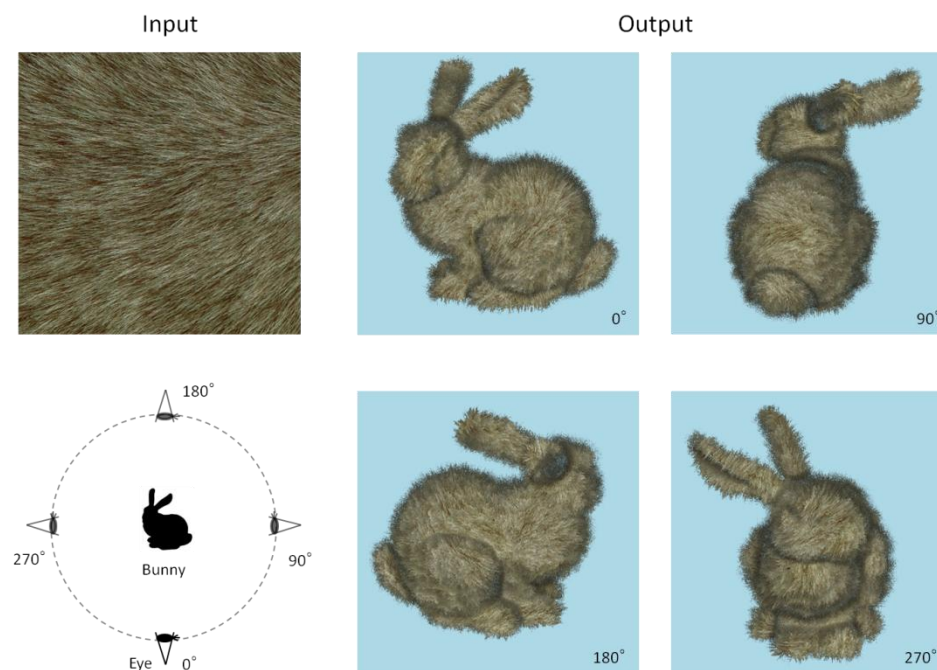


図5 複数視点からの描画結果

の描画結果に顕著に見られるように、入力画像に色のばらつきがある場合には、UV座標の張られ方により3Dオブジェクト上でテクスチャ間の繋ぎ目が目立つという問題が挙げられる。本手法は輝度値を用いた高低推定を行なっている以上、斑模様のように極端な色相の違いが見られる入力画像を対象とはしないが、単色の毛皮の入力画像においても、撮影時の環境によっては、色のばらつきが発生するため、別途入力画像に対し色味の補正等が必要だと考えられる。

一部に見られる毛並りの違和感については、4.3節の黄土色の入力画像に対し、Bunnyモデルの耳部分において特にこの現象が顕著である。この結果のように、実写画像中の毛が長く、毛並方向が明瞭な入力を用いた場合、3DオブジェクトにおけるUV座標の張られ方によっては、実際の動物が持つ毛並方向にそぐわない結果が得られることがある。特にBunnyモデルにおける耳など、曲率の大きな地点においては、外側の

テクスチャが曲率の変化が大きい方向に引き伸ばして描画されるため、これが入力
の毛並方向と一致すると、より違和感の強い結果が得られることになる。この問題の解
決のためには現在のところ部分的な UV 座標自体の張り替えなど、コストのかかる手
段のみが挙げられている状態のため、妥当な解決手法の提示は今後の課題としていき
たい。

また、写実的な毛皮の描画には、光源依存のライティング効果の付加も必要である。
本稿でのライティングは各層におけるテクスチャマッピングの際に Lambert シェー
ディングのみを施しているが、本来ならば毛の一本ごとに対し各々影やハイライトな
ども生成されるはずであるため、これらのライティング効果の追加は今後の課題と
して挙げられる。

7. まとめ

本稿では、一枚の実写画像から毛並の再現が可能な Shell Texture を生成することを
目的とし、実写画像中の輝度値と毛並における高低の対応を仮定することにより、
Shell Texture の生成ならびに毛並の実時間描画を行った。今後の課題としては、任
意の曲面に対して違和感を生じない妥当な毛並を与える方法の検討などが挙げられ
る。

参考文献

- 1) Shinji Ogaki, Yusuke Tokuyoshi, and Sebastian Schoellhammer. 2010. An empirical fur shader. In ACM SIGGRAPH ASIA 2010 Sketches (SA '10). ACM, New York, NY, USA, Article 16, 2 pages. DOI=10.1145/1899950.1899966 <http://doi.acm.org/10.1145/1899950.1899966>
- 2) Jerome Lengyel, Emil Praun, Adam Finkelstein, and Hugues Hoppe. 2001. Real-time fur over arbitrary surfaces. In Proceedings of the 2001 symposium on Interactive 3D graphics (I3D '01). ACM, New York, NY, USA, pp.227-232. DOI=10.1145/364338.364407 <http://doi.acm.org/10.1145/364338.364407>
- 3) J. T. Kajiya and T. L. Kay. 1989. Rendering fur with three dimensional textures. SIGGRAPH Comput. Graph. 23, 3 (July 1989), pp.271-280. DOI=10.1145/74334.74361 <http://doi.acm.org/10.1145/74334.74361>

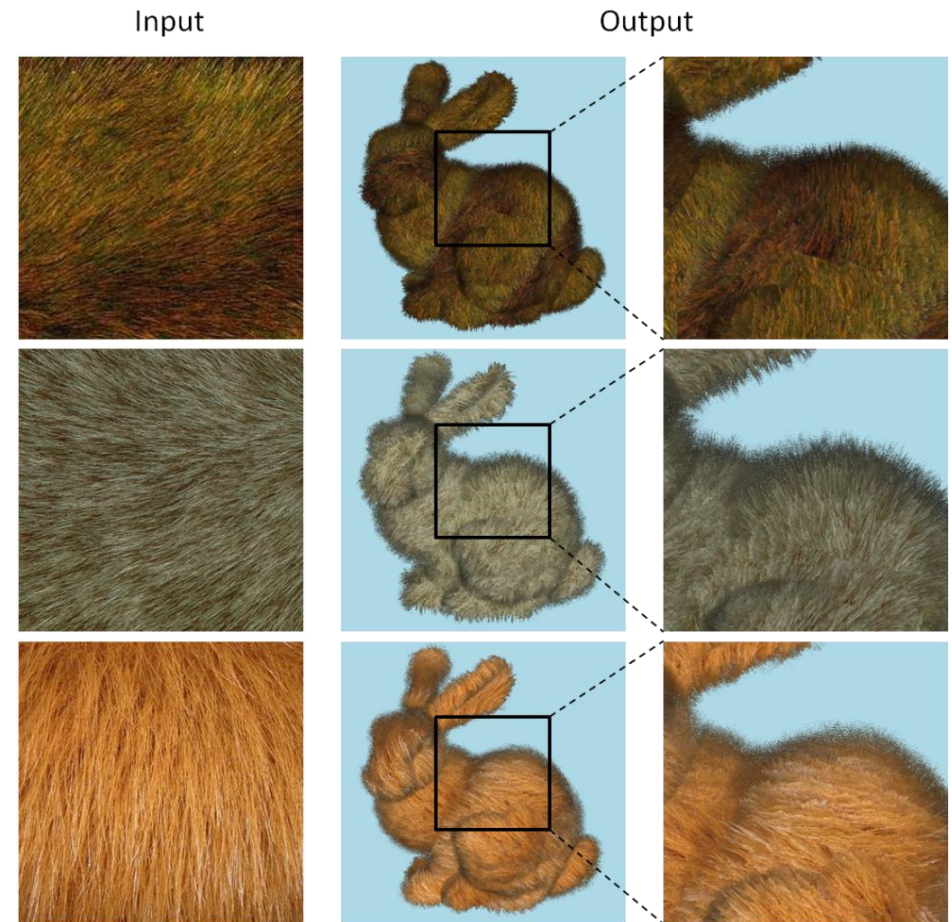


図 6 複数種の入力画像に対する描画結果