筑波大学大学院博士課程

システム情報工学研究科修士論文

# 毛髪繊維の遮蔽を考慮した毛髪画像の陰影編集

佐々木 悠真

修士（工学）

（コンピュータサイエンス専攻）

指導教員　三谷純

2015 年 3 月

## 概要

人間の毛髪を写した髪型画像を他の画像に合成する技術は、髪型のシミュレーションや映像制作に用いられている。特に髪型のシミュレーションは、単純な髪型画像と顔画像の重ね合わせにより、髪型を変更した時の自分の見た目を手軽に確認することができ、個人のヘアスタイルの変更の際の手助けとなる。しかし、それぞれの画像で照明条件が異なるために、合成画像の陰影に違和感を生じてしまう。そこで本研究では、1 枚の髪型画像を入力として、毛髪上の陰影を対話的に編集できるようにすることで、異なる照明環境における毛髪の陰影を表現する手法を提案する。まず毛髪の陰影が、画像全体で滑らかに変化する陰影と、毛髪繊維同士の遮蔽より生じる細かい影からなるものと仮定する。さらにこれらの陰影について、滑らかな陰影は照明方向の違いに影響されるものの、細かい影は照明方向を変化させたときの毛髪の外観にそれほど影響を与えないと仮定する。これらの仮定に基づき、毛髪繊維における滑らかな陰影を編集することで、照明効果の変更を行う。照明効果の編集には二種類の手法を検討した。一つは、滑らかな陰影を混合ガウス分布で近似し、ガウス分布の移動により滑らかな陰影を編集する手法である。もう一つは、滑らかな陰影に Kajiya-Kay モデルを当てはめ、毛髪繊維の三次元方向ベクトル場を推定し、照明計算に用いる手法である。それぞれの手法において、数種類の髪型画像を用いた評価実験の結果を示し、それぞれの手法の有効性および課題について述べる。ユーザは単純なマウス入力によって直感的に陰影を編集し、様々な照明効果の調整結果を得られることが期待される。

# 目 次

**第 1 章　序論** **1**

**第 2 章　関連研究** **3**
2.1　毛髪の反射モデル . . . . . 3
2.2　リライティング . . . . . 4
2.3　画像中の物体における外観の編集 . . . . . 5

**第 3 章　提案手法** **7**
3.1　提案手法の概要 . . . . . 7
3.2　毛髪繊維の二次元ベクトル場の抽出 . . . . . 9
3.2.1　Watanabe らの手法 . . . . . 9
3.2.2　Kang らの手法 . . . . . 10
3.3　初期遮蔽率マップの計算 . . . . . 12
3.4　髪色マップと遮蔽率マップの改良 . . . . . 14
3.5　髪色マップの編集 . . . . . 16
3.5.1　混合ガウス分布を用いた髪色マップの編集 . . . . . 16
髪色マップへの混合ガウス分布の割り当て . . . . . 16
ガウス分布の移動による髪色マップの編集 . . . . . 17
3.5.2　Kajiya-Kay モデルを用いた髪色マップの編集 . . . . . 20
毛髪繊維の三次元ベクトル場の推定 . . . . . 20
毛髪繊維の三次元ベクトル場の平滑化 . . . . . 25
三次元ベクトル場を用いた照明効果の変更 . . . . . 25

**第 4 章　結果** **27**
4.1　混合ガウス分布を用いた髪色マップの編集結果 . . . . . 27
4.2　Kajiya-Kay モデルを用いた髪色マップの編集結果 . . . . . 34
4.3　Kajiya-Kay モデルを用いた髪色マップ編集における考察 . . . . . 39
4.3.1　毛髪繊維の二次元方向ベクトル場 . . . . . 39
4.3.2　髪色マップと遮蔽率マップ . . . . . 41
4.3.3　毛髪繊維の三次元方向ベクトル場 . . . . . 43

**第 5 章　結論** **49**

謝辞 **50**

参考文献 **51**

# 図 目 次


2.1 反射モデルごとのレンダリング結果の比較 . . . . . 4
2.2 三次元の人体の計測 . . . . . 5

3.1 提案手法の流れ . . . . . 7
3.2 髪色マップと遮蔽率マップ . . . . . 8
3.3 Watanabe らの手法により生成された毛髪繊維の二次元方向ベクトル場 . . . . 11
3.4 Kang らの手法により生成された毛髪繊維の二次元方向ベクトル場 . . . . . 12
3.5 Kang らの手法による毛髪繊維の二次元方向ベクトル場の方向の補正 . . . . . 13
3.6 初期髪色マップの計算 . . . . . 13
3.7 髪色マップと遮蔽率マップの改良 . . . . . 15
3.8 混合ガウス分布による髪色マップの表現 . . . . . 18
3.9 三次元座標系と照明・視線の模式図 . . . . . 21
3.10 環境光成分の生成 . . . . . 22
3.11 Kajiya-Kay モデルを利用した毛髪繊維の三次元方向ベクトル場の推定 . . . . 23
3.12 異なる二次元方向ベクトル場を用いた時の推定結果の違い . . . . . 24
3.13 毛髪繊維の三次元ベクトル場の推定における平滑化の有無による結果の比較 . 26

4.1 ガウス分布の移動による髪色マップの編集結果 . . . . . 30
4.2 ガウス分布の移動による髪色マップの編集結果 . . . . . 31
4.3 マウスの移動方向と移動方法の違いによる髪色マップの編集結果の比較 . . . 32
4.4 ガウス分布の移動により生成された出力画像 . . . . . 33
4.5 照明方向を変更した結果画像 . . . . . 35
4.6 照明の色や明るさを変更した結果画像 . . . . . 36
4.7 青色の髪型を入力とした場合の結果画像 . . . . . 37
4.8 複雑な形状を持つ髪型を入力とした場合の結果画像 . . . . . 38
4.9 二次元方向ベクトル場における模範解答との比較 . . . . . 40
4.10 二次元方向ベクトル場ごとの初期遮蔽率マップの比較 . . . . . 41
4.11 二次元方向ベクトル場ごとの改良遮蔽率マップの比較 . . . . . 42
4.12 二次元方向ベクトル場ごとの改良髪色マップの比較 . . . . . 42
4.13 レンダリングされた髪型画像と環境光成分の模範解答 . . . . . 45
4.14 各二次元方向ベクトル場により推定された三次元方向ベクトル場の比較 . . . 46
4.15 図 4.14 の三次元ベクトル場を用いて計算された照明変更の結果の比較 . . . . 46

4.16 二次元方向ベクトル場の模範解答を利用した三次元方向ベクトル場の比較 . . 47
4.17 二次元方向ベクトル場と照明方向の模範解答を利用した三次元方向ベクトル場の比較 . . . . . 48

# 第1章　序論

髪型は人間の見た目の印象に大きく影響する要素の一つであり、髪型の変更が自己実現の手段として用いられることも多い。しかし、毛髪は一度切断すると元に戻すことができず、また変更した髪型が自分に似合っていないと感じてしまっても、髪の長さや色、癖などといった理由から、他の髪型を試すことは難しい。そこで、予め髪型のシミュレーションを行うことができれば、実際の毛髪に手を加えることなく、自分に似合う髪型を見つけるための有用な手段となると考えられる。画像を用いて髪型シミュレーションを行うアプリケーションやサービスは、PC やスマートフォン向けのものが提供されている [30, 31, 32, 33, 34] 。これらのアプリケーションやサービスでは、ユーザは予め登録されている髪型画像と、自身で撮影した顔画像とを合成して、髪型を変更したときの自分の見た目を確かめることができる。しかしこの際、各画像のサイズの違いから生じるずれや、照明環境の違いから生じる陰影の違いによって、合成画像に違和感を生じてしまう。頭部形状の違いや生え際の形状の違和感などは、画像処理によりある程度対応できる。しかし髪型画像と顔画像の照明方向の違いに起因した、光の当たり方の違いによる違和感は、通常の画像処理では対処が難しい。この問題を解決するためには、髪型画像の陰影を調整し、顔画像に適した照明効果を持った画像になるように編集する必要がある。

しかし画像中の毛髪における照明効果の変更は、毛髪繊維の持つ複雑な光の反射特性から困難なものになっている。毛髪繊維は円筒形状をした半透明の物体であり、さらに繊維の表面には僅かに隆起した鱗片を持っている。そのため、毛髪繊維の表面やその内部で、通常の平面に対するものとは異なる複雑な光の反射や屈折、散乱を生じる。加えて、毛髪は多数の細かい繊維が複雑に折り重なっているという特徴を持っている。この特徴のために、ある繊維が他の繊維を遮蔽したために生じる細かい影が、毛髪領域全体にわたって存在している。以上の性質から、毛髪画像における陰影を光学的に解析することは困難である。

そこで本研究では、毛髪画像における陰影を編集する手法を提案する。これにより、異なる照明環境における毛髪の陰影を容易に表現できるようになる。本研究の技術的貢献は、単一の毛髪画像の陰影を編集するための手法を初めて示したことである。本手法は次のような仮定に基づく。まず毛髪の陰影が、画像全体で滑らかに変化する陰影と、毛髪繊維同士の遮蔽により生じる細かい影から構成されるとする。さらにこれらの陰影について、滑らかな陰影は照明方向の違いによって大きく変化するものの、細かい影は照明方向を変化させたときの毛髪の外観にそれほど影響を与えないと仮定する。これらの仮定に基づき、毛髪画像における滑らかな陰影を編集することで、照明効果の編集を行う。照明効果の編集について、本研究では二つの手法を検討した。一つは、滑らかな陰影を混合ガウス分布により近似し、割り当

てたガウス分布の移動によって、滑らかな陰影を編集する手法である。ユーザは単純なマウス移動により、直感的に陰影を編集できる。しかしこの手法では、毛髪繊維の流れに沿った、物理的に妥当な編集結果は得られなかった。また局所的な移動では、大域的な陰影全体を違和感なく編集することは困難であった。そこでもう一つの手法として、滑らかな陰影に毛髪の反射モデルである Kajiya-Kay モデルを利用する手法を検討した。滑らかな陰影に Kajiya-Kay モデルを当てはめることで、入力画像の照明の方向ベクトルと、毛髪繊維の各ピクセルでの方向を表す三次元方向ベクトル場を推定する。計算された毛髪繊維の三次元方向ベクトル場と、ユーザの指定した照明の方向ベクトルを用いて、入力画像とは異なる照明方向での照明効果を計算する。三次元ベクトル場の推定は二次元ベクトル場に依存し、滑らかな三次元ベクトル場が自然な照明計算の結果を得るために必要であることを示す。最終的には、ユーザが照明の方向を指定することで、様々な照明効果の調整結果を得られることが期待される。

# 第2章　関連研究

本章では本研究に関連する既存研究を紹介する。まず毛髪に光が当たった時の特徴や毛髪の照明効果を計算する研究や手法について説明する。次に、三次元の物体と画像を対象とした、照明環境を変えたときの光の当たり方をシミュレートする技術であるリライティングに関する研究や、画像を対象とした色や外観を変更する研究について紹介する。

## 2.1　毛髪の反射モデル

毛髪に光が当たった時に現れる照明の効果は、物体に当たった光が反射することにより生じている。物体表面における光の反射特性は、BRDF (Bidirectional Reflectance Distribution Function; 双方向反射率分布関数) により記述される。BRDF は、ある方向から差し込んだ光がある点に入射したときに、どれだけの光が反射されるかを示す関数である。こうした物体に光が当たった時の物理的挙動をモデル化したものを反射モデルという。

毛髪における照明効果の計算を行うためには、毛髪の反射モデルを考慮する必要がある。Kajiya と Kay [1] は毛髪が円筒形をしているという事実に基づいて、毛髪表面の光沢を表現するための反射モデルを提案した。この反射モデルは Kajiya-Kay モデルと呼ばれる。Kajiya-Kay モデルにおける反射光成分は、拡散反射成分と鏡面反射成分の 2 つから構成され、主要な物理パラメータとして、毛髪の三次元方向ベクトルと視線の三次元ベクトル、入射光の三次元方向ベクトルの 3 つのベクトルを用いて、毛髪上に現れる照明効果を計算する。

実際の毛髪では円筒形であることに加え、さらに毛髪の内部で光の屈折や散乱が発生するため、毛髪全体にぼんやりとした照明効果が現れる。Goldman [2] はこのような特徴に対して、毛髪内部での反射と屈折の効果を制御するための方向のパラメータを用いて、毛髪の半透明性を表現した。さらに、Marschner ら [3] は毛髪繊維における光の反射特性の測定結果に基づき、Marschner モデルと呼ばれる反射モデルを提案した。この Marschner モデルでは毛髪の円筒形状や半透明性だけでなく、毛髪繊維の円筒面に存在する、僅かな傾斜を持った鱗片の影響も考慮している。図 2.1 に示すように、Kajiya-Kay モデルに比べ、より実際の毛髪のものに近い照明効果を表現できている。3DCG における毛髪のレンダリングにおいて、現在一般的に用いられているのはこの Marschner モデルである。

提案手法では、毛髪上の陰影を編集するための一方法として、入力画像の毛髪領域に対して毛髪の反射モデルを仮定し、照明効果を計算する手法をとっている。計算の際に用いられるパラメータのいくつかは未知であり、それらのパラメータは何らかの方法で推定するか、適切な値を割り当てる必要がある。そうした観点から、推定すべきパラメータの比較的少ない

(a) Kajiya-Kay モデル　(b) Marschner モデル　(c) 実際の毛髪

図 2.1: 反射モデルごとのレンダリング結果の比較。画像出典: 文献 [3]

Kajiya-Kay モデルを採用した。この反射モデルにおける照明効果の計算には、ユーザの指定する照明方向や、視線方向に加え、毛髪繊維の三次元方向ベクトルが必要である。そのため画像から細かな繊維の方向を読み取る必要がある。なお、視線の方向は画像の正面から画像の中心に向いていると仮定する。

## 2.2　リライティング

ある照明環境下で撮影された物体に対して、異なる照明環境での照明効果を計算する手法はリライティングと呼ばれる。リライティングは画像合成における有用な手段であるが、物理的に妥当なリライティングを行う場合、照明の位置や方向、光の強さといった照明条件や、対象物体の形状や反射率などの情報を取得することが不可欠である。Debevec らは三次元物体のリライティングに関する研究を多く行っている [4, 5, 6, 7] 。彼らは図 2.2 のように、大掛かりな装置を用いて三次元の人間の計測を行い、必要な測定情報を用いてリライティングを行っている。この装置は、ターンテーブル上にトレッドミルを重ねたものに人間を乗せて歩かせながら、床下照明やドーム状に配置された照明によって周囲から照らし、格子状に並べられたカメラアレイで撮影するというものである。しかし彼らの研究では、こうした特別な装置が必要であることに加え、三次元の人間を対象としているため、画像中の毛髪を対象としている本研究では利用できない。

画像を入力としたリライティングでは、本来三次元の対象であれば取得可能な情報が欠けてしまっているため、三次元物体に対するリライティングとは異なるアプローチをとる必要がある。しかしながら、一枚の二次元画像から、リライティング処理を行うために必要な照明や物体の性質といった情報を正確に得るのは難しい。Okabe ら [8] は、ペンデバイスを利用した直感的な方法で、物体の形状情報や法線マップをユーザに付加させることで、物体の形状情報を補い、さらに照明を平行光源とし、物体の反射特性に Phong の反射モデルを仮定し

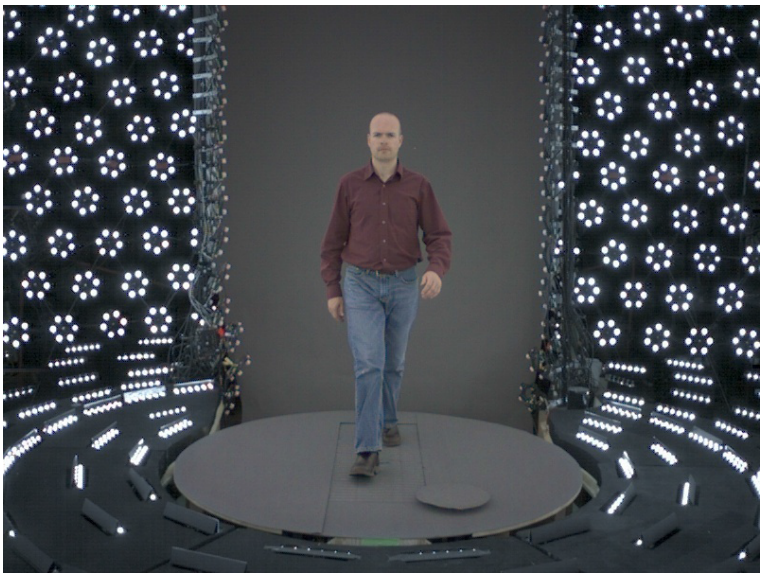


図 2.2: 三次元の人体の計測。画像出典: 文献 [7]

てリライティングを行った。また Munoz ら [11] は、物理的に測定されたデータから対象となる半透明物体の反射率を測定し、単一画像中の半透明物体のリライティングを実現した。顔画像を扱ったリライティングの研究 [9, 10] も存在している。一枚の画像だけでなく、複数の画像を入力画像として用いれば、利用できる情報が増えるため、問題が簡単になる場合もある。Teng らの研究 [12] では予め異なる照明環境でレンダリングされた球体のデータベースを利用し、入力画像のすべてのピクセルにおいて、その光の強度と勾配がデータベース内の球体とマッチするものを探索する。その後、探索された球体と、照明環境を変えた時の球体の関係を利用し、各ピクセルの照明効果を変更する。Yu ら [13] は、同じ照明環境下で複数の視点から撮影された画像を用いて照明と反射率を推定し、リライティングを行った。リライティングに必要な形状の情報を特殊なデバイスで取得しようとした研究も存在する。山田ら [14] や Ikeda [15] らの研究では Kinect のような奥行き情報を取得できる装置を利用して、撮影した画像の奥行きのパラメータを取得し、リライティングに利用している。しかし入力を一枚の髪型画像に限定したリライティングの研究はない。

## 2.3 画像中の物体における外観の編集

画像中の物体の外観を変更する手法では、対象となる物体の形状や色などの物理情報を利用して編集を行っている。Color Transfer は画像中のある物体の色を、他の画像の物体に転写する技術であり、多くの研究が存在する [16, 17, 18] 。しかしながら、入力として予め転写させたい色を持つ画像を探しださなければならないという問題がある。Levin ら [19] や Qu ら [20] は、スクリブルなどのユーザ入力によってグレイスケールの画像に色をつける手法を提案している。この手法では対象への自由な色付けが可能であるものの、元画像のハイライト

や陰影が付加する色に影響するため、編集により照明の方向が変わったように見せるのは難しい。Khan ら [21] は人間の視覚が物理的な正確さに鈍感であるという事実に基づいて、画像中の物体の材質を変更する手法を提案した。この手法では、画像からデプスマップや照明の入射方向を計算する必要があるが、そうしたパラメータの計算は、毛髪のような細い繊維の重なり合った物体では困難である。Chai ら [22, 23] は画像から毛髪の三次元モデルを生成することで、画像中の毛髪繊維の移動や切断を可能にする手法を提案した。照明効果の編集も行っているものの、ハイライトと呼ばれる物体表面に現れる光沢の強度を変更する程度に留まっており、照明効果の位置を調整できるわけではない。Shen ら [24] は物体の色を持つ反射率マップと、物体の陰影を持つ陰影マップに画像を分離する手法を提案した。これにより、物体の陰影を保持したまま、物体の色だけを編集することができる。しかし彼らの研究における陰影マップには大域的な陰影と局所的な陰影の双方が含まれているため、反射率マップを単純に編集しただけでは、照明方向が変わったようには見えない。そこで本研究では、陰影マップからさらに大域的な陰影を取り除くことで、毛髪繊維の細かな形状を維持したまま、毛髪画像の照明効果を編集することを可能にする。

# 第3章　提案手法

この章では本研究で提案する陰影編集手法について述べる。まず提案手法の流れを説明してから、毛髪画像中の滑らかな陰影と細かい影を分離する手法の手順を解説する。次に、滑らかな陰影を編集する手法に関して検討した二つの手法と、それぞれの編集手法によって結果画像を得る方法について説明する。

## 3.1　提案手法の概要

提案手法の目的は、毛髪画像中の陰影を対話的な操作によって編集することである。提案手法は入力として、一枚の毛髪画像と、その毛髪領域を指定するマスク画像を用いる。必要

図 3.1: 提案手法の流れ。髪型の映った画像とその毛髪領域を入力として利用する。まず毛髪領域を髪色マップと遮蔽率マップに分離するために、入力画像の各ピクセルの遮蔽度合いを示す初期遮蔽率マップを生成する。そして初期遮蔽率マップを用いて、髪色マップと遮蔽率マップを改良する。改良した髪色マップに編集を加え、陰影を編集する。編集は本研究で紹介する二つの手法のいずれかを用いて行われる。編集結果に遮蔽率マップを乗算することで、出力結果を生成する。

(a) 入力画像　(b) 髪色マップ　(c) 遮蔽率マップ

図 3.2: 髪色マップと遮蔽率マップ。(a) 入力画像は (b) 髪色マップと (c) 遮蔽率マップの積で表現される。

な前処理が実行された後、ユーザが毛髪画像上で単純なマウス移動を行うことで、様々な陰影の移動結果を出力として得られるようにする。

提案手法の流れを図 3.1 に示す。提案手法では初め、毛髪画像に対して次のような仮定を置く。毛髪に光が当たることにより、毛髪上には頭部全体に広がる陰影や、多数の細かい影が生じる。提案手法ではこれらの影や陰影が、頭部形状のために毛髪領域中で滑らかに変化する陰影と、毛髪繊維が他の繊維を遮ることにより生じる細かい影に大別できると仮定する。さらに、滑らかな陰影は照明方向の変化により大きく影響されるものの、細かい影は照明方向が変化してもそれほど影響を受けないと仮定する。これらの仮定に基づき、図 3.2 に示すように、毛髪画像を毛髪の色と滑らかな陰影を表す髪色マップ $\mathbf{R}$ (図 3.2(b)) と、細かい影を表す遮蔽率マップ $s$ (図 3.2(c)) に分離する。これにより、細かい影を維持したまま、滑らかな陰影を編集できるようにする。

毛髪領域中の各ピクセル $\mathbf{p}$ に対して、毛髪の色 $\mathbf{I_p}$ は髪色マップの値 $\mathbf{R_p}$ と遮蔽率マップ $s_\mathbf{p}$ の積で表現できると仮定する。すなわち、次式のように表される。

$$\mathbf{I_p} = s_\mathbf{p}\mathbf{R_p} \tag{3.1}$$

式 (3.1) に基づき、まず毛髪画像を髪色マップと遮蔽率マップに分離する。毛髪画像をうまく分離するためには、画像の持つ細かい繊維の形状を知っておく必要がある。そのため、予め各ピクセルにおける毛髪繊維の平面的な方向を表す二次元方向ベクトルを計算する。この二次元方向ベクトルは、後述する髪色マップの編集処理にも利用される。計算された二次元方向ベクトル場から粗い遮蔽率マップを計算する。しかしこの遮蔽率マップによって毛髪画像から髪色マップを分離したとき、その髪色マップには局所的な陰影やノイズが残ってしまう。そこで、この粗い遮蔽率マップを遮蔽率マップの初期値 (初期遮蔽率マップ) とし、髪色マップと遮蔽率マップの双方を改良する。これにより、毛髪繊維の方向に沿った陰影変化の影響

を強くし、髪色マップから局所的な影を取り除く。改良された各マップを、それぞれ改良髪色マップと改良遮蔽率マップと呼ぶ。分離された改良髪色マップに変更を加えることで、毛髪画像の陰影を編集する。改良髪色マップの編集手法として、本研究では混合ガウス分布を用いた手法と、Kajiya-Kay モデルを利用した手法の二つの手法を紹介し、編集を行う際は二つの手法のいずれかを利用する。混合ガウス分布を用いた手法では、毛髪領域上に複数のガウス分布を等間隔に配置し、それらのガウス分布で髪色マップが表現できるように、各ガウス分布の重みを調整する。そして配置したガウス分布をマウスの移動方向や毛髪繊維の方向に沿って移動させることで、髪色マップを編集する。Kajiya-Kay モデルを利用した手法では、照明計算を行うために、各ピクセルにおける毛髪繊維の三次元方向ベクトルを得る必要がある。しかし画像から直接三次元の方向を得るのは難しいことから、予め計算した二次元方向ベクトル場を利用し、最適化計算により三次元方向ベクトル場を推定する。その後、ユーザの与えた照明方向を用いて照明計算を行うことで、髪色マップを編集する。最後に、ユーザが毛髪画像上で行った操作に基づいて、照明効果を編集された髪色マップと、改良遮蔽率マップを乗算することで、毛髪領域中の大域的な陰影だけが変更された結果画像が得られることが期待される。

## 3.2 毛髪繊維の二次元ベクトル場の抽出

重なり合う毛髪繊維が互いに遮蔽し合うことにより生じる局所的な影に対処するためには、ある毛髪繊維が他の毛髪繊維をどの程度遮蔽しているかを検出する必要がある。毛髪繊維の形状を知ることができれば、その繊維が他の繊維に与える遮蔽の影響を計算できる。そこで、毛髪繊維の二次元方向を検出する。これにより、毛髪繊維の大まかな形状を知ることができる。毛髪繊維の方向の検出のために、毛髪画像の各ピクセルにおける、毛髪繊維の $x, y$ 方向成分を示す二次元ベクトル場を作成する。なお、座標系は画像の中心を原点とし、画像の右方向を $x$ 軸の正の方向、上方向を $y$ 軸の正の方向とする。

本節では本研究において検討した二種類の手法として、Watanabe らの手法と Kang らの手法を解説する。マスク画像で指定された毛髪領域に対してこれらの手法を適用し、毛髪繊維の二次元ベクトル場を得る。

### 3.2.1 Watanabe らの手法

Watanabe ら [25] の手法による毛髪繊維の二次元方向ベクトル場の作成手法について解説する。彼らの手法における二次元方向ベクトル場の計算は、Paris らの手法 [26] に基づいている。まず毛髪領域をグレースケールに変換する。そしてグレースケール化した画像の各ピクセルに対し、一次元フィルタを回転させながら適用する。その結果、一次元フィルタの応答値が最も大きかった方向の単位ベクトルを、毛髪の二次元方向ベクトルとして保存する。このとき、ノイズ軽減のために閾値処理を施し、一次元フィルタの応答値が閾値よりも小さければ、そのピクセルの方向ベクトルをノイズとみなし、ゼロベクトルとする。

次に、閾値処理によりゼロベクトルとしたピクセルに対して、pull-push [27] と呼ばれる手法を用いてベクトル場の補完を行う。補完にあたり、まずベクトルをテンソルを表す $2 \times 2$ 行列に変換する。これは方向ベクトル $\mathbf{v}$ と $-\mathbf{v}$ のそれぞれを区別せずに扱うためである。補完時に参照する近傍ピクセルにおける方向ベクトルの符号が互いに異なる場合、成分同士が打ち消しあうことがあるため、テンソルを用いてそれを防ぐ。

各ピクセルのベクトルをテンソルに変換した後、ゼロベクトル以外の値をもつピクセルを保護する。その後、テンソル場に対してガウシアンフィルタを適用しつつテンソル場を縮小してサンプリングし、サイズが $1/2$ に縮小されたテンソル場を作成する。この操作を繰り返し、ガウシアンピラミッドと呼ばれる画像集合を作成する。次にこの縮小されたテンソル場をバイキュービック補間により 2 倍に拡大する。そして保護されていないピクセルに対して、得られた補間値を加算していく。この縮小と拡大の処理を画像全体での値の変化が一定値以下になるまで繰り返す。最終的なテンソル場を再びベクトル場に変換することで補完が完了する。最終的なベクトル場へ変換する際には、すべてのベクトルが下方に向くようにする。

図 3.3 では本節の手法により生成される二次元ベクトル場を示す。入力画像（図 3.3(a)）から一次元フィルタによって計算された二次元ベクトル場は 図 3.3b のようになる。図 3.3(b) において、毛髪領域内で白色となっている部分は、ノイズと判断されてゼロベクトルとなった部分である。その後、これらのゼロベクトルとした部分を補完した二次元ベクトル場は図 3.3(c) のようになる。なお、本研究における毛髪繊維の二次元ベクトル場は、各ピクセルにおける二次元方向ベクトルの $x, y$ 成分の絶対値を、それぞれ R, G の色成分に割り当てて可視化した画像である。

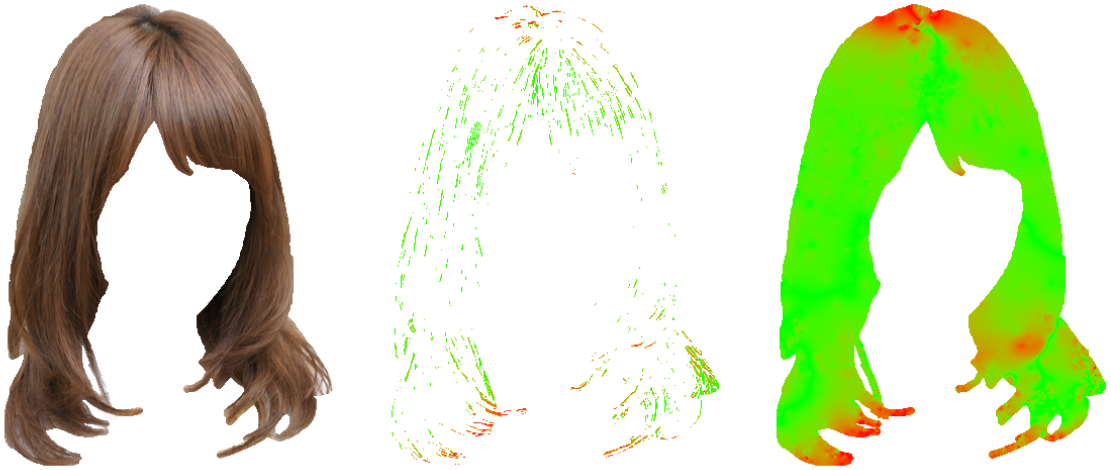
### 3.2.2 Kang らの手法

Kang ら [28] の手法は一枚の二次元画像からモノクロの線画を生成する手法であり、線画における各線を生成する基準として、edge tangent flow (ETF) と呼ばれている画像中の物体のエッジの流れを示すベクトル場を利用している。この ETF を、本研究における毛髪繊維の二次元方向ベクトルとして利用する。

ETF の生成にはバイラテラルフィルタリングに基づいた、畳み込み演算による反復的な非線形平滑化を利用する。このとき、特徴的なエッジ方向は維持され、弱いエッジは近傍の強いエッジに影響されて方向が滑らかになるように、反復的な平滑化を行う。また渦巻き状のアーティファクトを避けるために、同じ方向を持つエッジの間で平滑化を行う。

$i$ 回目の反復で得られる ETF を $\mathbf{t}^i$ とし、ピクセル $\mathbf{p}$ ベクトル値を $\mathbf{t}^i_\mathbf{p}$ とする。各ピクセルにおける ETF の初期値 $\mathbf{t}^0_\mathbf{p}$ を、入力画像の勾配方向 $\mathbf{g}^0_\mathbf{p}$ に関して、反時計回りに垂直な方向ベクトルとして得る。勾配方向 $\mathbf{g}^0_\mathbf{p}$ はソーベルフィルタにより計算される。その後、ETF を生成するためのフィルタを、 ETF の初期値に適用する。ETF を生成するフィルタは次のような式で定義される。

$$\mathbf{t}^{i+1}_\mathbf{p} = \frac{1}{k} \sum_{\mathbf{q} \in \Omega(\mathbf{p})} \phi(\mathbf{p}, \mathbf{q}) \mathbf{t}^i_\mathbf{q} \omega_s(\mathbf{p}, \mathbf{q}) \omega_m(\mathbf{p}, \mathbf{q}) \omega_d(\mathbf{p}, \mathbf{q}) \tag{3.2}$$



(a) 入力画像　(b) 補完前　(c) 補完後

図 3.3: Watanabe らの手法により生成された毛髪繊維の二次元方向ベクトル場。(a) 入力画像から計算された (b) 二次元方向ベクトル場において、ノイズとみなされる部分は一旦取り除かれる。その後、補完を行うことで、(c) 毛髪領域全体に広がる毛髪繊維の二次元方向ベクトル場が生成される。

ここで $k$ はベクトルの正規化項であり、$\Omega(\mathbf{p})$ は $\mathbf{p}$ の半径 $r$ の範囲にある近傍ピクセルの集合である。$\omega_s(\mathbf{p},\mathbf{q})$ は空間的な重みであり、次のように定義される。

$$\omega_s(\mathbf{p},\mathbf{q}) = \begin{cases} 1 & \text{if } \| \mathbf{p}-\mathbf{q} \| < r \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$\omega_m(\mathbf{p},\mathbf{q})$ は勾配の強度に関する重みであり、毛髪領域全体で最大の勾配の強度により正規化された勾配の強度 $\hat{g}(\mathbf{p})$ と減衰の割合を制御する値 $\eta$ を用いて、次の式で定義される。

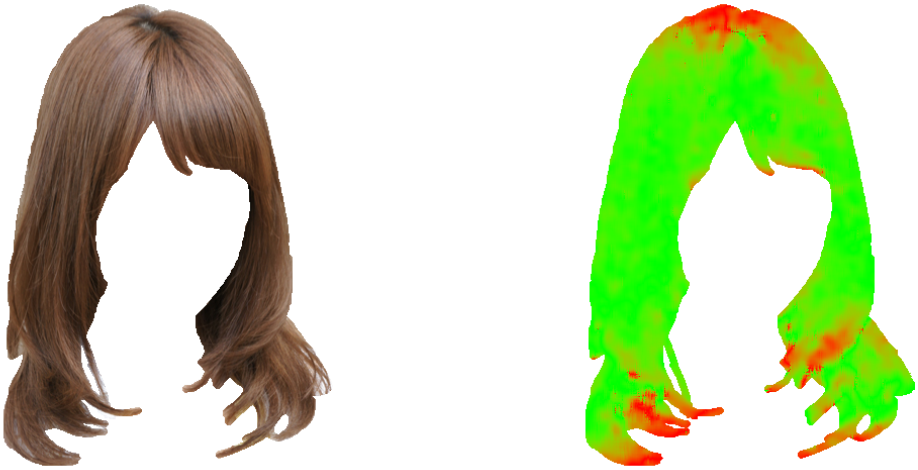
$$\omega_m(\mathbf{p},\mathbf{q}) = \frac{1}{2}(1+\tanh[\eta\cdot(\hat{g}(\mathbf{q})-\hat{g}(\mathbf{p})]) \tag{3.4}$$

$\omega_d(\mathbf{p},\mathbf{q})$ は方向に関する重みであり、対象ピクセルの方向と近傍ピクセルの方向の内積を利用して、次のように定義される。

$$\omega_d(\mathbf{p},\mathbf{q}) = \mid \mathbf{t}^i_{\mathbf{p}} \cdot \mathbf{t}^i_{\mathbf{q}} \mid \tag{3.5}$$

最後に、$\phi(\mathbf{p},\mathbf{q})$ は符号を決定する関数であり、近傍領域のベクトルが逆向きの方向を持たないようにするために、次のように定義される。

$$\phi(\mathbf{p},\mathbf{q}) = \begin{cases} 1 & \text{if } \mathbf{t}^i_{\mathbf{p}} \cdot \mathbf{t}^i_{\mathbf{q}} > 0 \\ -1 & \text{otherwise} \end{cases} \tag{3.6}$$



(a) 入力画像　　(b) 二次元方向ベクトル場

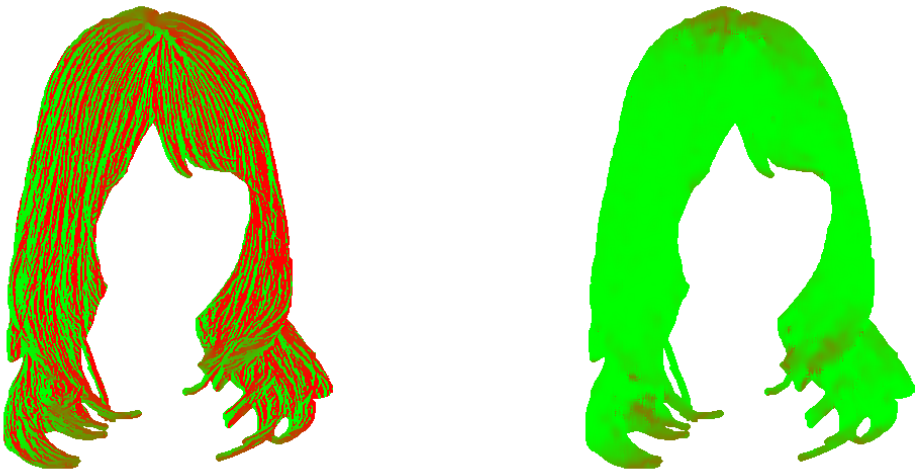
図 3.4: Kang らの手法により生成された毛髪繊維の二次元方向ベクトル場。(a) 入力画像から、Kang らの手法を利用して、(b) 毛髪繊維の二次元方向ベクトル場が生成される。

以上のように生成されるフィルタを反復的に適用し、ETF を更新する。このとき、勾配方向 $\mathbf{g}(\mathbf{p})$ は反復に応じて更新されるが、勾配の強度 $\hat{g}(\mathbf{p})$ は更新されない。たかだか 2, 3 回の反復により、滑らかな ETF が生成される。

図 3.4 では本節の手法により生成された毛髪繊維の二次元方向ベクトル場を示す。本節の手法を入力画像 （図 3.4(a)）に適用すると、図 3.4(b) のような二次元方向ベクトル場を生成できる。しかしながら、Kang らの手法により単に二次元方向ベクトル場を生成すると、同一繊維上では滑らかなベクトルを持つものの、繊維間では上下の不揃いな方向を持ってしまう。図 3.5 は $y$ 成分が正の値を持つベクトルを赤、負の値を持つベクトルを緑で表した図であるが、繊維ごとに正と負が互い違いになってしまっている。このままでは、Kajiya-Kay モデルを利用した髪色マップの編集において、三次元ベクトル場の推定がうまくいかない。そこで $y$ 成分が正の値を持っているベクトルを単に反転させることで、ベクトル場全体の繊維方向の $y$ 成分を負に補正した二次元方向ベクトル場（図 3.4(b) 及び図 3.5(b)) を作成する。

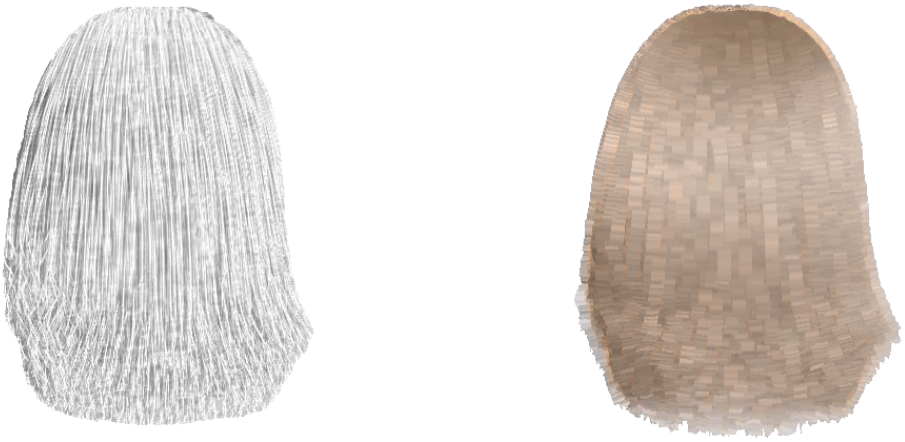
## 3.3　初期遮蔽率マップの計算

前節で計算された毛髪繊維の二次元方向ベクトル場を用いて、毛髪画像から毛髪繊維同士の遮蔽によって生じる細かい影を分離する。そのために、まず毛髪領域内の各ピクセルにおける遮蔽の度合いを示す画像を生成する。局所的に見ると、ある毛髪繊維が他の毛髪繊維によって遮蔽されていれば、その部分は周囲に比べて暗くなり、露出している部分は対照的に明るくなっている。そこで、あるピクセルの輝度と、その近傍ピクセルの中で最も明るいピクセルの輝度の比を遮蔽率として用い、初期遮蔽率マップとする。



(a) 補正前　　(b) 補正後

図 3.5: Kang らの手法による毛髪繊維の二次元方向ベクトル場の方向の補正。(a) Kang らの手法により入力画像から生成された二次元ベクトル場は毛髪繊維の方向が繊維ごとに不揃いであるため、 (b) $y$ 成分が負になるように方向を補正した二次元方向ベクトル場を作成する。



(a) 初期遮蔽率マップ　　(b) 初期髪色マップ

図 3.6: 初期遮蔽率マップの計算。(a) 計算された初期遮蔽率マップから計算された (b) 髪色マップは毛髪繊維の流れに沿った滑らかな色変化となっていない。

初期遮蔽率マップの計算には毛髪繊維の二次元方向ベクトル場を利用する。毛髪領域中の各ピクセル $\mathbf{p}$ の持つ毛髪の方向ベクトルに関して、その直交する方向に両側 5 ピクセル分の輝度分布を得る。輝度分布を得る際、バイリニア補間を用いて輝度をサンプリングする。得られた輝度分布における最大輝度 $L_{\mathbf{max}}$ と対象ピクセルの輝度 $L_{\mathbf{p}}$ を用いて、初期遮蔽率の値 $s_{\mathbf{p}}$ を次のように計算する。

$$s_{\mathbf{p}} = \frac{L_{\mathbf{p}}}{L_{\mathbf{max}}} \tag{3.7}$$

式 (3.7) から計算された遮蔽率 $s_{\mathbf{p}}$ により、式 (3.1) を用いて髪色 $\mathbf{R_p}$ を計算すると、図 3.6(b) のように、滑らかでなく、ノイズの含まれた髪色マップが生成されてしまう。理想的には、髪色マップは毛髪繊維の方向ベクトルに沿って、滑らかな色変化を持つことが求められる。そこで次節において、遮蔽率マップが局所的な影を含みつつ、髪色マップが滑らかな色変化を持つように、それぞれのマップを改良する。

## 3.4 髪色マップと遮蔽率マップの改良

前節までに計算された毛髪繊維の二次元方向ベクトル場と初期遮蔽率マップを利用して、改良髪色マップと改良遮蔽率マップを生成する。基本的には Shen らの手法 [24] と同様な方法で最適化を行う。彼らの手法では、毛髪繊維の方向を考慮していないため、毛髪に対して直接適用した場合には、大域的な陰影と局所的な影をうまく分離できない。そこで、彼らの用いたエネルギー関数に対して毛髪繊維の方向に依存した重みを導入し、毛髪繊維の遮蔽による細かい影だけを含んだ改良遮蔽率マップを分離できるようにする。具体的には、次のエネルギー関数 $E_{sep}$ の最小化により、髪色マップと遮蔽率マップの改良を行う。

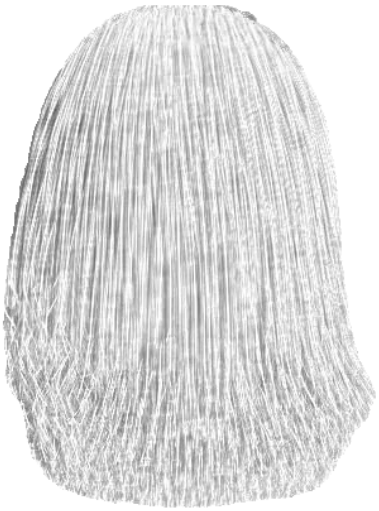
$$E_{sep} = \sum_{\mathbf{p}\in P} \parallel \mathbf{R_p} - \sum_{\mathbf{q}\in N_{\mathbf{p}}} \omega_{\mathbf{pq}} \mathbf{R_q} \parallel^2 + \sum_{\mathbf{p}\in P} \parallel \mathbf{I_p}\hat{s}_{\mathbf{p}} - \mathbf{R_p} \parallel^2 \tag{3.8}$$

ここで、$P$ はピクセル全体の集合である。$N_{\mathbf{p}}$ はピクセル $\mathbf{p}$ の近傍ピクセルの集合であり、本研究における結果では $11 \times 11$ の近傍領域を用いている。また、$\hat{s}_{\mathbf{p}}$ は $s_{\mathbf{p}}$ の逆数である。

式 (3.8) による最適化は以下の手順で行われる。まず遮蔽率マップと髪色マップの初期値を設定する。遮蔽率マップ $s$ の初期値として、前節で計算した初期遮蔽率マップを利用する。また初期遮蔽率マップと式 (3.1) から髪色マップの初期値を計算する。次に、第一項の計算を行い、髪色マップの計算を行う。このとき、重み $\omega_{\mathbf{pq}}$ を次のように決定する。

$$\omega_{\mathbf{pq}} = \frac{\mid \cos\theta_{\mathbf{pq}} \mid}{\sum_{\mathbf{r}\in N_{\mathbf{p}}} \mid \cos\theta_{\mathbf{pr}} \mid} \tag{3.9}$$

ここで、$\theta_{\mathbf{pq}}$ はピクセル $\mathbf{p}$ における毛髪繊維の方向ベクトルと、ピクセル $\mathbf{p}$ から近傍ピクセル $\mathbf{q}$ へ向かう単位方向ベクトルのなす角である。この方向依存の重みにより、毛髪繊維の方向に沿った輝度変化の影響が強くなる。その後、得られた改良髪色マップと入力画像を用いて第二項を計算し、改良遮蔽率マップを計算する。毛髪領域内のすべてのピクセルについてこれらの計算を行い、前回のエネルギー関数の値との誤差が収束するまで反復する。



(a) 初期遮蔽率マップ

(b) 改良遮蔽率マップ

(c) 初期髪色マップ

(d) 改良髪色マップ

図 3.7: 髪色マップと遮蔽率マップの改良。(b) 改良遮蔽率マップは (a) 初期遮蔽率マップに比べ繊維間の遮蔽度の差がはっきりとしており、(d) 改良髪色マップは (c) 初期髪色マップに比べ、より滑らかな色変化となっている。

図 3.7 は遮蔽率マップと髪色マップの改良前後を比較したものである。初期遮蔽率マップ（ 図 3.7(a)） に比べ、改良遮蔽率マップ（ 図 3.7(b)) は繊維間の遮蔽度合いがはっきりと現れている。また、同一繊維における遮蔽率のノイズが減少し、同一繊維上では同じような遮蔽率を持つように改良されている。また初期髪色マップ ( 図 3.7(c)) ではブロックノイズのように凸凹とした色の塊が並んだものになってしまっている。これは初期遮蔽率マップの計算時に、対象ピクセルにおける毛髪繊維方向と直交する方向だけに依存して計算を行っているためである。改良髪色マップ ( 図 3.7(d)) ではこの問題が解消され、毛髪領域全体で滑らかな色変化を持った画像が生成できている。ただし、改良髪色マップにはハイライトが白い筋として残ってしまっているという問題がある。このハイライトも毛髪繊維間の細かい影と同様に取り除かれるべきであると考えている。

次節では、髪色マップの編集手法について検討してきた二つの手法をそれぞれ解説する。

## 3.5 髪色マップの編集

本節では、髪色マップの編集を行うにあたり、本研究では検討した二つの手法について紹介する。まず混合ガウス分布を用いて髪色マップを編集する手法について解説する。次に、Kajiya-Kay モデルと呼ばれる毛髪の反射モデルを利用した髪色マップの編集手法について解説する。各手法の結果と問題点については、4 章で詳しく説明する。

### 3.5.1 混合ガウス分布を用いた髪色マップの編集

ここでは髪色マップに混合ガウス分布を割り当てることで、髪色マップの編集を行う手法について解説する。混合ガウス分布とは、複数のガウス分布によってデータの分布を近似する手法である [29] 。まず髪色マップに対する混合ガウス分布の割り当てについて説明し、次に割り当てたガウス分布を利用した髪色マップの編集方法について解説する。

#### 髪色マップへの混合ガウス分布の割り当て

混合ガウス分布を毛髪画像に割り当て、割り当てたガウス分布を移動させることにより、毛髪の陰影の直感的な編集を可能にすることを目指す。これにより、ユーザは編集したい陰影を選択し、細かく編集することができる。また、Photoshop [36] などの画像編集ソフトを利用すれば髪色マップの陰影編集は可能であるが、編集したい陰影の選択や、編集した陰影と元からあった陰影の合成などの操作には知識が必要である。こうしたことから、混合ガウス分布を利用することによって、難しい操作を必要としない、直感的な操作による編集が可能になる。

まず毛髪領域を格子に分割し、それから各セルに二次元ガウス分布を割り当てる。各セル $k$ には中心位置 $\boldsymbol{\mu}_k$ と分散 $\sigma_k^2$ をそれぞれ設定する。そしてこれらの値を用いて、ピクセル

$\mathbf{p}$ でのあるセル $k$ に配置された二次元ガウス分布の値 $G_k(\mathbf{p})$ を次の式から計算する。

$$G_k(\mathbf{p}) = \alpha \exp\Big( - \frac{\| \mathbf{p} - \boldsymbol{\mu}_k \|^2}{\sigma_k^2} \Big) \tag{3.10}$$

ここで $\alpha$ は重みであり、ユーザが手動で調節することで、ガウス分布全体の明るさを調節できる。また各ガウス分布の色 $\mathbf{C}_k$ は、次の式で決定する。

$$\mathbf{C}_k = \frac{\sum_{\mathbf{p}} \mathbf{R}_{\mathbf{p}} G_k(\mathbf{p})}{\sum_{\mathbf{p}} G_k(\mathbf{p})} \tag{3.11}$$

図 3.8 は混合ガウス分布により、髪色マップを表現した結果である。またセルサイズを変更した時の結果の比較を行っている。セルのサイズが比較的大きくても髪色マップの照明変化は表現できるものの、毛髪領域の境界付近では元の髪色マップの色が表現できず、色が明るくなりすぎている。これは本来影響を与えないはずのガウス分布の影響が強く現れてしまっているためである。セルのサイズが小さくになるにつれ、毛髪領域の縁の付近でも、より良く髪色マップを表現することができている。しかしながら、依然として元の髪色マップよりも色が明るい部分は残っている他、マスク画像で指定した毛髪領域の境界部分では、ガウス分布の影響が少なすぎるためか、極端に暗くなってしまっている。なお、ガウス分布の性質上、セルサイズを $1 \times 1$ にすることはできない。

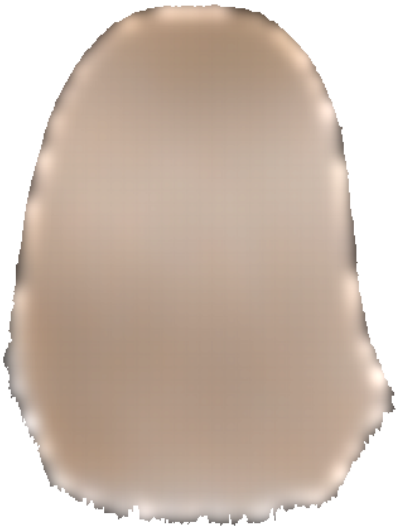
セルサイズ $2 \times 2$ の混合ガウス分布で表現された髪色マップ（図 3.8(d)) を用いれば、十分照明効果を編集できる。しかし、境界部分では極端に暗い領域が生まれてしまっているため、編集結果を顔画像と合成する際には、境界部分を補完する何らかの手法が必要だろうと考えている。

### ガウス分布の移動による髪色マップの編集

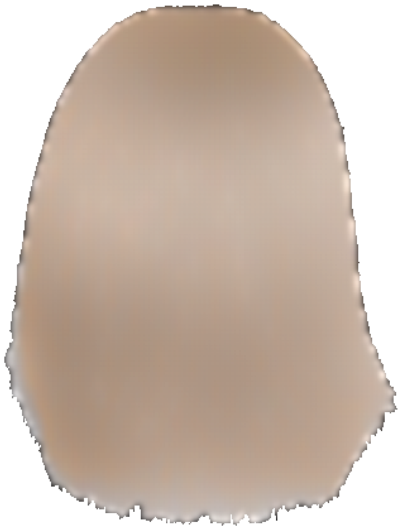
セルごとに計算されたガウス分布をマウス移動に従って移動させることで、対話的な照明効果の編集を行う。ガウス分布の移動は、マウスポインタを中心とする一定の半径内に限り実行される。マウスの移動方向に従って、一定の半径内のガウス分布を、マウスポインタからの距離と毛髪繊維の方向を利用して毛髪繊維の流れに沿って動かすことで、対話的な編集の実現を目指す。

ガウス分布の移動を行う際、単純にガウス分布をマウスの移動方向に従って動かす方法と、それに加えて各ピクセルでの毛髪繊維の方向を考慮した方法を利用して移動を行う。まずマウスの移動方向に従ってガウス分布を動かす方法では、各ガウス分布はマウスの移動方向と移動距離に依存して移動する。このとき、マウスポインタからの距離が遠いほど移動量を抑えることで、マウスポインタ周囲の照明効果を直感的に動しつつ、意図しないガウス分布は移動されないようにする。次に、毛髪繊維を利用した方法では、まず各ガウス分布 $k$ の中心点 $\mathbf{p}_k = (x_k, y_k)$ における毛髪繊維の二次元方向ベクトル $\mathbf{t}_{\mathbf{p}_k}$ を取得する。その方向ベクトルに、マウス移動の方向ベクトル $\mathbf{m}$ を射影した移動ベクトルを計算し、そのベクトルに従ってガウス分布を移動させる。これは毛髪繊維上ではガウス分布を移動しやすくし、別の毛髪繊維

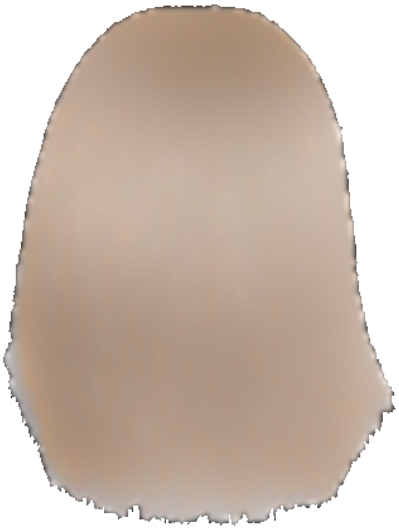
(a) 髪色マップ



(b) セルサイズ 10 × 10



(c) セルサイズ 4 × 4



(d) セルサイズ 2 × 2

図 3.8: 混合ガウス分布による髪色マップの表現。(b) や (c) のように、ガウス分布を割り当てるセルのサイズが大きすぎると、本来影響を与えないはずのガウス分布の影響が強く表れてしまい、毛髪領域の境界付近では (a) 元の髪色マップの色が表現できずに色が明るくなりすぎている。(d) に示すようにセルのサイズが小さいと、毛髪領域の縁の付近でもよりよく髪色マップの色を表現できているものの、ガウス分布の影響が少なすぎるためか、境界部分はセルのサイズに関わらず極端に暗くなってしまっている。

に対しては移動しにくくするためである。これらの考えから、ガウス分布の移動関数 $M(\mathbf{p}_k)$ を次のように定義する。

$$M(\mathbf{p}_k) = \omega_g v \mathbf{t}_{\mathbf{p}_k} \Big( \frac{\mathbf{t}_{\mathbf{p}_k} \cdot \mathbf{m}}{\parallel \mathbf{t}_{\mathbf{p}_k} \parallel \parallel \mathbf{m} \parallel} \Big) \tag{3.12}$$

ここで $v$ は移動の強さを決める変数であり、ユーザが自由に操作できる。本研究では $v = 0.2$ としている。また $\omega_g$ は距離に関する重みであり、次のように定義される。

$$\omega_g = \exp(-(\mid x_k - x_{current} \mid^2 + \mid y_k - y_{current} \mid^2)) \tag{3.13}$$

ここで、$\mathbf{p}_{current}$ はマウスポインタの現在位置であり、 $\mathbf{p}_{current} = (x_{current}, y_{current})$ とする。

### 3.5.2 Kajiya-Kay モデルを用いた髪色マップの編集

本節では Kajiya-Kay モデルを用いた髪色マップの編集手法について解説する。まず Kajiya-Kay モデルの計算に必要な毛髪繊維の三次元方向ベクトル場と、三次元ベクトル場の推定に必要な照明の方向ベクトルの推定手法を解説した後、毛髪繊維の三次元方向ベクトル場の問題点を解決するための平滑化手法について説明する。最後に Kajiya-Kay モデルを用いた照明効果の編集方法を説明する。

#### 毛髪繊維の三次元ベクトル場の推定

髪色マップから入力画像の照明方向と共に、毛髪繊維の三次元ベクトル場を推定する手法を解説する。改良髪色マップに Kajiya-Kay モデルを当てはめることで、照明方向を指定することにより照明効果を編集できるようにする。これにより、毛髪領域全体で、毛髪として物理的に妥当な照明効果を計算により求めることができる。

3.2 節で計算された毛髪繊維の二次元方向ベクトル場を用いて、各ピクセルにおける毛髪繊維の方向ベクトルの $z$ 方向成分と、入力画像における照明の方向ベクトルを、誤差関数の最小化により推定する。ここで座標系は 3.2 節の $x, y$ 軸に加え、画像の手前方向を $z$ 軸の正の方向とする右手座標系とする。また照明は平行光源とし、視線のベクトルは $z$ 軸の負の方向に平行な単位方向ベクトルとする。三次元座標系と照明、視線の模式図は図 3.9 のようになっている。

Kajiya-Kay モデルでは、入力画像の各ピクセルにおける照明効果 $\mathbf{R_p}$ は、拡散反射光 $\Psi_d$, 鏡面反射光 $\Psi_s$, 環境光 $\Psi_a$ の和で計算される。

$$\mathbf{R_p} = \Psi_d + \Psi_s + \Psi_a \tag{3.14}$$

また、拡散反射光と鏡面反射光はそれぞれ以下の式で計算される。

$$\Psi_d = \mathbf{k}_d(1 - (\mathbf{t_p} \cdot \mathbf{l})^2) \tag{3.15}$$

$$\Psi_s = \mathbf{k}_s((1 - (\mathbf{t_p} \cdot \mathbf{l})^2)(1 - (\mathbf{t_p} \cdot \mathbf{e})^2) - (\mathbf{t_p} \cdot \mathbf{l})(\mathbf{t_p} \cdot \mathbf{e}))^n \tag{3.16}$$

ここで $\mathbf{k}_d, \mathbf{k}_s$ はそれぞれ拡散反射係数と鏡面反射係数である。$\mathbf{t_p}$ はピクセル $\mathbf{p}$ における毛髪繊維の単位方向ベクトルであり、$\mathbf{l}$ は照明の単位方向ベクトル、$\mathbf{e}$ は視線の単位方向ベクトルである。また、$n$ は光沢度である。本研究における計算では、環境光成分 ( 図 3.15(b)) は入力画像 ( 図 3.15(a)) の毛髪領域中で最も輝度の低い入力画像の色が毛髪領域全体に一様に広がっているものであると仮定する。本節では拡散反射光成分と鏡面反射光成分の二つの成分の推定について議論する。

式 (3.14) を用いて、毛髪繊維の方向ベクトルの $z$ 方向成分と、入力画像の光の入射方向を推定する手法について説明する。まず照明の方向ベクトルを離散的に少しずつ変化させながら、それぞれの照明方向において照明計算を行う。照明方向は極座標において、方位角 $\phi$ を $0° \leq \phi \leq 360°$, 仰角 $\theta$ を $0° \leq \theta \leq 90°$ の範囲で設定される。本研究のすべての結果では、方

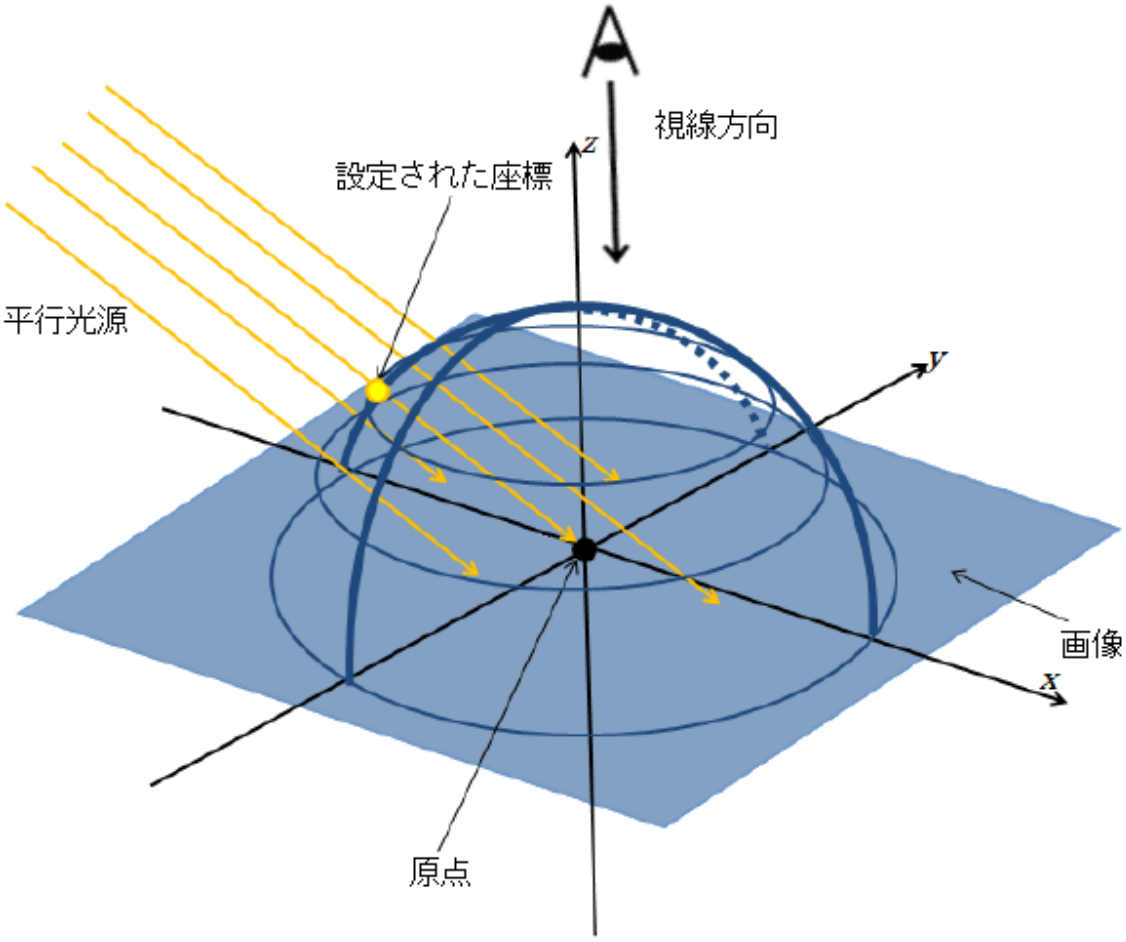



図 3.9: 三次元座標系と照明・視線の模式図。照明は極座標で設定された座標から画像の中心に向く平行光源とする。視線は $z$ 軸の負の方向に平行な単位方向ベクトルとする。

位角を 8 分割、仰角を 6 分割して計算している。設定された各照明方向について、毛髪繊維における三次元方向ベクトルを各ピクセルで推定する。各ピクセルにおける毛髪繊維の三次元方向ベクトルの $x, y$ 方向成分については、毛髪繊維の二次元方向ベクトル場の値を利用する。三次元方向ベクトルの $z$ 方向成分については、誤差関数の最小化により推定する。すなわち、ピクセル $\mathbf{p}$ における髪色の値 $\mathbf{R_p}$ と、毛髪繊維の三次元方向ベクトル $\mathbf{t_p}$ 及び照明の方向ベクトル $\mathbf{l}$ から、式 (3.14) により計算された髪色 $\Psi_\mathbf{p}$ を用いて、誤差 $E_\mathbf{p}$ を次のように定式化する。

$$E_\mathbf{p} = \| \mathbf{R_p} - \Psi_\mathbf{p} \|^2 \tag{3.17}$$

式 (3.17) を最小化するような $z$ 成分を持つ毛髪繊維の三次元方向ベクトル $\mathbf{t_p}$ を、レーベンバーグ・マーカート法を用いた最適化を利用して、毛髪領域中のすべてのピクセルで計算する。

次に、照明方向と毛髪領域内で最適な毛髪繊維の三次元方向ベクトル場を決定する。各ピ

(a) 入力画像　　(b) 環境光成分

図 3.10: 環境光成分の生成。(a) 入力画像の毛髪領域において輝度の最も低い色で、毛髪領域全体を塗りつぶすことで、(b) 環境光成分を得る。

クセルで計算された $E_{\mathbf{p}}$ の最小値の総和を最小にする毛髪繊維の三次元方向ベクトル場 $\mathbf{T}$ と照明の方向ベクトル $\mathbf{l}$ をそれぞれの最適値とする。すなわち、次の関数 $E_{est}$ を最小にする各値を求める。

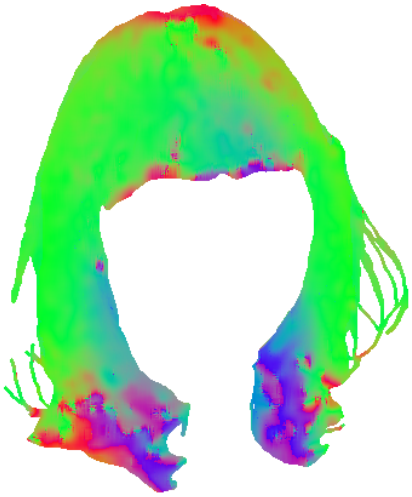
$$E_{est} = \sum_{\mathbf{p}} \min E_{\mathbf{p}} \tag{3.18}$$

図 3.11 は本節の手法により計算された毛髪繊維の二次元方向ベクトル場及び三次元方向ベクトル場を可視化した図である。なお、本研究において示す毛髪繊維の三次元方向ベクトル場は、各ピクセルにおける方向ベクトルの $x, y, z$ 成分の絶対値を、それぞれ R, G, B の色成分に割り当てて可視化した画像である。推定された毛髪繊維の三次元方向ベクトル場は図 3.11 (d) のような結果となる。理想的には、毛髪繊維の三次元方向ベクトル場は、実際の毛髪繊維の流れに従って、滑らかに接続していることが望ましい。しかし図 3.11(d) の結果では、所々で波打つようなベクトル場の変化が見られ、また方向が急激に変化している領域も存在している。図 3.11(b) に示した二次元方向ベクトル場においても同じようなベクトル場の変化が見られる。また図 3.12 でも示すように、予め生成された二次元方向ベクトル場に、三次元ベクトル場は大きく影響される。これらのことから、推定される三次元ベクトル場が二次元ベクトル場におけるノイズの影響を強く受けていることがわかる。そこで次に、三次元ベクトル場を平滑化することにより、このノイズの影響を減少させることを試みた。

(a) 入力の髪型画像

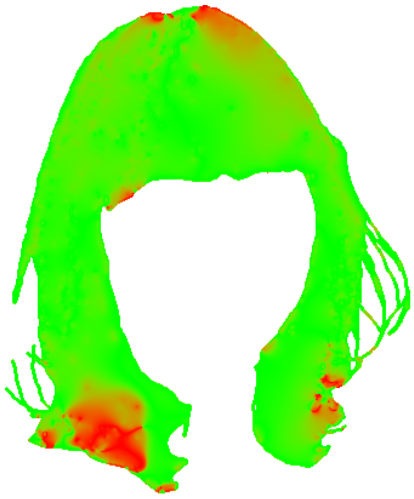
(b) 毛髪繊維の二次元方向ベクトル場

(c) 髪色マップ

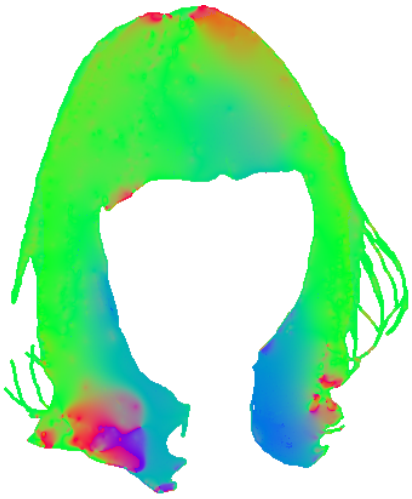


(d) 毛髪繊維の三次元方向ベクトル場

図 3.11: Kajiya-Kay モデルを利用した毛髪繊維の三次元方向ベクトル場の推定。

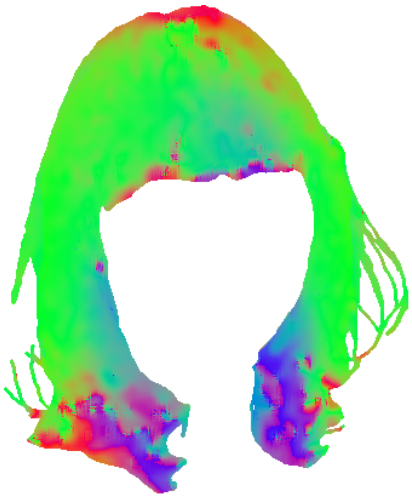


(a) Watanabe らの手法による毛髪繊維の
二次元方向ベクトル場



(b) (a) を用いて生成した毛髪繊維の
三次元方向ベクトル場

(c) Kang らの手法による毛髪繊維の
二次元方向ベクトル場



(d) (c) を用いて生成した毛髪繊維の
三次元方向ベクトル場

図 3.12: 異なる二次元方向ベクトル場を用いた時の推定結果の違い。

### 毛髪繊維の三次元ベクトル場の平滑化

毛髪繊維の三次元方向ベクトル場に含まれたノイズを減少させるために、髪色マップと Kajiya-Kay モデルによる計算結果との誤差を出来るだけ小さくしつつ、ベクトル場が滑らかになるように平滑化を行う。まず 3.5.2 節の手法により照明の方向ベクトルと毛髪繊維の三次元方向ベクトル場を推定する。このとき得られた照明の方向ベクトルは正しい方向が得られたものとして、照明の方向を固定する。推定された毛髪繊維の三次元方向ベクトル場を初期値として、式 (3.18) に平滑化項を加えたエネルギー関数 $E'_{est}$ を最小化する毛髪繊維の三次元ベクトル場を求める。

$$E'_{est} = \sum_{\mathbf{p}} (E_{\mathbf{p}} + \lambda \frac{1}{\omega_{sum}} \sum_{\mathbf{q} \in N_{\mathbf{p}}} \omega_{\mathbf{pq}} (z_{\mathbf{p}} - z_{\mathbf{q}})^2) \tag{3.19}$$

ここで $\lambda$ は平滑化項の重みであり、本研究では $\lambda = 10000$ としている。$z_{\mathbf{p}}, z_{\mathbf{q}}$ は各ピクセルの $z$ 方向成分であり、$N_{\mathbf{p}}$ はピクセル $\mathbf{p}$ の近傍ピクセル集合である。また、$\omega_{sum} = \sum_{\mathbf{q} \in N_{\mathbf{p}}} \omega_{\mathbf{pq}}$ とする。ここであるピクセルとその近傍ピクセルの遮蔽率の差が小さければ、それらのピクセルは同じ毛髪繊維上に存在する可能性が高いと考えられる。また、あるピクセルへの方向の影響は、ピクセル間の距離が近いほど強くなると考えられる。これらのことから、二点間の重み $\omega_{\mathbf{pq}}$ を次のように計算する。

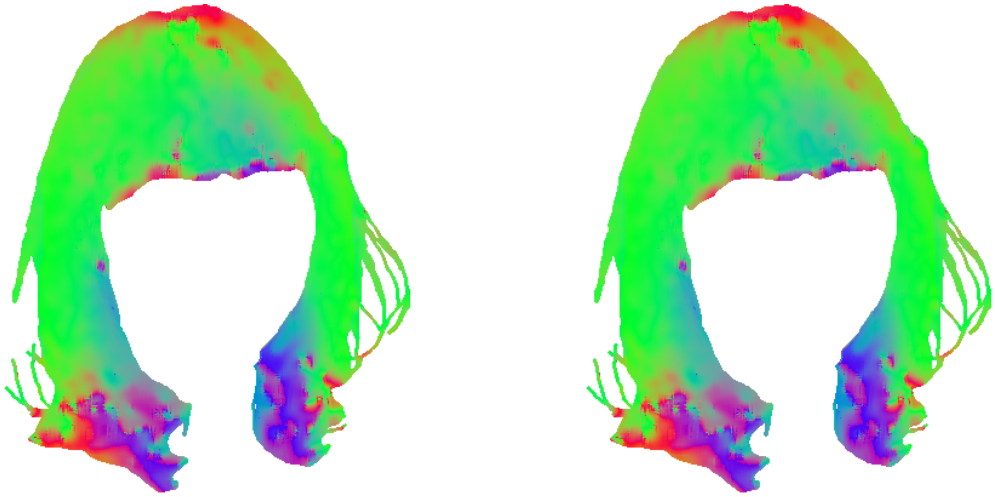
$$\omega_{\mathbf{pq}} \exp - \frac{\| \mathbf{p} - \mathbf{q} \|^2}{\sigma_d^2} \exp - \frac{| s_{\mathbf{p}} - s_{\mathbf{q}} \|^2}{\sigma_o^2} \tag{3.20}$$

ここで、$\sigma_d$ は距離の項における重みであり、$\sigma_o$ は遮蔽率の差の項における重みである。本研究ではそれぞれ $\sigma_d = 0.1, \sigma_o = 3$ としている。

図 3.13 は本節で解説した平滑化手法による三次元ベクトル場の結果を示している。平滑化処理を施したベクトル場と施していないベクトル場を比較すると、ほとんど違いは見られない。従って平滑化処理が適切に行われていないため、平滑化手法についての別の方法を検討することが今後の課題である・。

### 三次元ベクトル場を用いた照明効果の変更

本節の手法で推定された毛髪繊維の三次元方向ベクトル場を利用して、式 (3.14) から髪色マップを新たに計算し、その結果画像に 3.4 節で得られた改良遮蔽率マップを乗算したものを出力画像とする。ユーザは照明の方向ベクトルを手動で設定することにより、出力結果に対して対話的に変更を加える事ができる。



(a) 平滑化あり　　(b) 平滑化なし

図 3.13: 毛髪繊維の三次元ベクトル場の推定における平滑化の有無による結果の比較。

# 第4章　結果

本章では、本研究における実験環境を示してから、3.5.1 節と 3.5.2 節において述べた手法により編集された髪色マップ、および編集された髪色マップに改良遮蔽率マップを乗算した結果画像を示す。そして提案手法による成果に加え、いくつかの比較画像を示すとともに、提案手法の問題点と考察を述べる。

本研究における実験環境は、Intel Core i7-4771 3.50GHz, 16GByte のメモリを搭載した PC である。実装においては、実装言語として C++、ライブラリとして OpenGL, GLUT, GLUI, DevIL, GSL, Dlib を利用した。本研究で解説した各手法の処理時間は表 4.1 に示す通りである。なお、混合ガウス分布を利用した編集手法と、Kajiya-Kay モデルを利用した編集手法の双方で、髪色マップの編集はリアルタイムに行うことができる。

## 4.1　混合ガウス分布を用いた髪色マップの編集結果

3.5.1 節において解説した混合ガウス分布を用いた髪色マップの編集結果を示す。

図 4.1 及び 4.2 ではガウス分布を利用した編集の過程を示している。入力された髪型画像から生成された髪色マップに混合ガウス分布を割り当て、各分布をマウス移動に従って移動することにより編集した画像である。図 4.1 では毛髪繊維の二次元方向ベクトルを利用せずに髪色マップを編集している。合成結果としては、頭頂部付近にハイライトが集まったような画像が生成できている。混合ガウス分布を利用した編集では、ユーザが編集したい陰影を細かく編集できるため、違和感のある部分を自由に修正できる。その一方で、ガウス分布の移動されて暗くなった部分がうまく修正できずに、波立ったような陰影変化が生成されてしまっている。これはガウス分布の移動が局所的なためである。広い範囲のガウス分布を動かそうとすると、各所にあるガウス分布を細かく移動する必要がある。そのため、ガウス分布が無くなった部分を明るくするために他の部分からガウス分布を移動させる、という操作を繰り返し行うことになる。この操作により、移動の仕方によっては陰影変化が滑らかでなくなってしまうことがある。図 4.2 は黒色の毛髪に対して、毛髪繊維の二次元方向ベクトルを利用して編集を行った結果である。黒色の毛髪では不自然な色変化が目立ちにくく、違和感が少ない編集が可能である。

図 4.3 は毛髪繊維の方向ベクトルの利用したか否かによる移動結果の違いを比較したものである。毛髪繊維の二次元方向ベクトルを利用しない場合、ガウス分布は縦横のどちらの方向に関しても、マウスの移動に従って移動する。一方で、毛髪繊維の二次元方向ベクトルを利用した場合、縦方向の移動に関しては毛髪繊維を利用しない場合と同じような移動をする

表 4.1: 各処理における処理時間。各合計時間は、手作業によるパラメータの指定やリライティングの処理時間を除く前処理の合計時間である。各処理時間の単位は秒である。画像サイズは（横 × 縦）の形式で記述している。総ピクセル数はマスク画像で指定された毛髪領域に含まれるピクセルの数を表す。Watanabe らの手法、Kang らの手法は、それぞれの手法における二次元方向ベクトル場の計算時間を示す。初期遮蔽率マップ、各マップの改良は、初期遮蔽率マップの計算、遮蔽率マップと髪色マップの改良のそれぞれの処理時間を示す。混合ガウス分布の生成は、セルサイズが $2 \times 2$ の場合の処理時間を示す。推定と平滑化は三次元ベクトル場の生成にかかった時間を表す。また各手法の合計時間は、二次元方向ベクトル場の計算を Watanabe らの手法で行った場合のものを示している。

| | 図 4.1 | 図 4.2 | 図 4.5 | 図 4.7 | 図 4.8 |
|---|---|---|---|---|---|
| 画像サイズ | 375 × 500 | 375 × 500 | 360 × 450 | 480 × 640 | 550 × 640 |
| 総ピクセル数 | 73279 | 31154 | 93978 | 82401 | 175212 |
| Watanabe らの手法 | 4.9<br>(23.3%) | 2.1<br>(23.9%) | 4.2<br>(10.0%) | 8.0<br>(17.4%) | 6.3<br>(7.7%) |
| Kang らの手法 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 |
| 初期遮蔽率マップ | 2.3<br>(11.0%) | 0.9<br>(10.0%) | 2.7<br>(6.4%) | 2.7<br>(5.9%) | 5.7<br>(7.0%) |
| 各マップの改良 | 14.2<br>(66.7%) | 5.7<br>(64.8%) | 18.2<br>(42.9%) | 15.1<br>(32.6%) | 32.5<br>(39.0%) |
| 混合ガウス分布の生成 | $4.1 \times 10^{-3}$<br>(0.02%) | $1.7 \times 10^{-3}$<br>(0.02%) | | | |
| 混合ガウス分布による編集の合計時間 | 21.5 | 8.8 | | | |
| 推定と平滑化 | | | 16.9<br>(42.9%) | 19.6<br>(43.5%) | 37.6<br>(46.3%) |
| Kajiya-Kay モデルによる編集の合計時間 | | | 42.2 | 45.6 | 82.4 |

が、横方向の移動は比較的抑えられている。しかしながら、この移動手法では移動された軌跡上に、クレータのような穴が生成されてしまう。図 4.4 は提案手法によりガウス分布を移動させた結果である。ある場所にあるガウス分布を単純に移動させると、図 4.4(c) に示すように、その部分のガウス分布が取り除かれ、不自然に暗い領域が生成されてしまい、編集時はこの暗い部分を埋めながら編集を行う必要があるため、編集が難しくなってしまう。またマウスを素早く移動させると、図 4.4(d) に示すように、マウスの移動に従って、移動した軌跡上のガウス分布がマウスポインタ上に集中し、集中したガウス分布がまとめて移動してしまうため、亀裂のような穴が開いてしまう。

これらの問題に対処するには、ガウス分布の移動を毛髪領域全体で大域的に行う手法が考えられる。例えば、移動対象をすべてのガウス分布とし、マウスの移動方向と毛髪繊維の流れに沿って移動させるという方法が考えられる。ただし、混合ガウス分布を用いた編集は、毛髪上の陰影を局所的に編集する場合に有効な手法であるため、大域的な陰影を変化させることに対しては、Kajiya-Kay モデルを利用とした手法の方が有効である。

(a) 髪色マップ　(b) ガウス分布により表現された髪色マップ

(c) 髪色マップの編集結果　(d) 合成結果

図 4.1: ガウス分布の移動による髪色マップの編集結果。(a) 入力として与えられた髪色マップを (b) 混合ガウス分布で表現し、その後 (c) ガウス分布を移動した。髪色マップの編集後、遮蔽率マップを乗算し、(d) 出力画像とする。

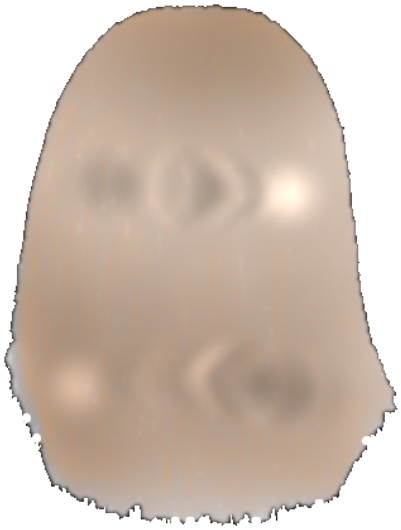
(a) 入力の髪型画像

(b) ガウス分布により表現された髪色マップ

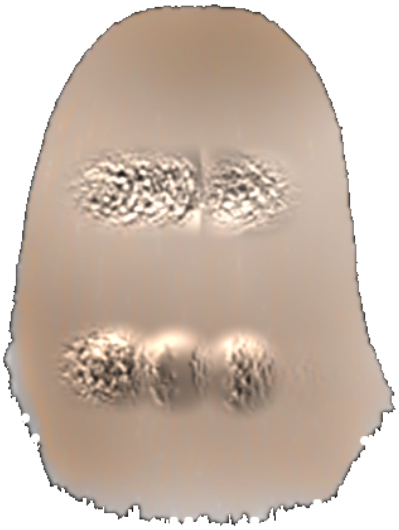
(c) 髪色マップの編集結果

(d) 合成結果

図 4.2: ガウス分布の移動による髪色マップの編集結果。(a) 入力画像から髪色マップを生成し、それを (b) 混合ガウス分布で表現する。その後 (c) ガウス分布を移動した。髪色マップの編集後、遮蔽率マップを乗算し、(d) 出力画像とする。

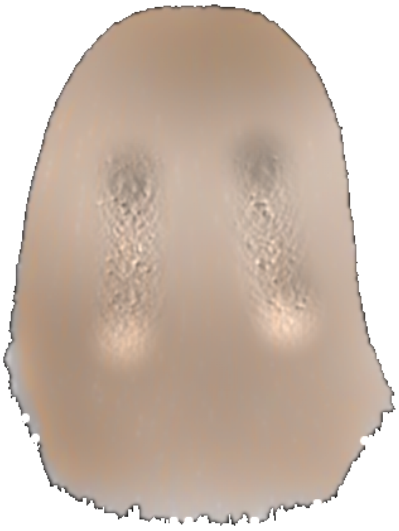


(a) 横移動: 毛髪繊維方向無し



(b) 横移動: 毛髪繊維方向有り

(c) 縦移動: 毛髪繊維方向無し



(d) 縦移動: 毛髪繊維方向有り

図 4.3: マウスの移動方向と移動方法の違いによる髪色マップの編集結果の比較。毛髪繊維の二次元方向ベクトルを利用しない場合、ガウス分布は (a) 横と (c) 縦のどちらの方向に関しても、マウスの移動に従って移動する。一方で、毛髪繊維の二次元方向ベクトルを利用した場合、(d) 縦方向の移動に関しては毛髪繊維を利用しない場合と同じような移動をするが、(c) 横方向の移動は比較的抑えられている。ただし、毛髪繊維の方向ベクトルを利用した場合、移動した軌跡に沿ってクレータのような穴が生成されてしまう。

(a) 入力の髪型画像

(b) 混合ガウス分布で表現された髪型画像

(c) 編集結果 1

(d) 編集結果 2

図 4.4: ガウス分布の移動により生成された出力画像。(c) 編集結果 1 のように、ある一部のガウス分布だけを移動させると、元々あったガウス分布がその場所から取り除かれるため、不自然に暗くなってしまう。(d) 編集結果 2 のように、マウスを素早く移動させると、マウスポインタにガウス分布が集中してしまい、移動の軌道に沿って亀裂のようになった真っ黒な部分が生成されてしまう。

## 4.2　Kajiya-Kay モデルを用いた髪色マップの編集結果

3.5.2 節において解説した Kajiya-Kay モデルを利用した髪色マップの編集結果を示す。図 4.5 は提案手法により生成された三次元方向ベクトル場を用いて、様々な照明方向で照明計算を行った結果である。照明方向の違いにより異なる照明効果を表現できているものの、図 4.5 のように、照明方向によっては明らかに暗すぎる結果となってしまっている。また図 4.5(b) は入力画像 ( 図 4.5(a)) と同じ照明方向から照明計算した結果であるが、入力画像と同じような画像は得られていない。これは毛髪繊維の三次元方向ベクトル場の不正確さや、Kajiya-Kay モデルが多重散乱を考慮していないことによるものである。また三次元方向ベクトル場において滑らかにベクトルが変化していない領域 ( 図 4.5 では特に右上部分 ) では不自然な照明効果が生成される。また図 4.6 は照明に明るさや色を変更した例であり、それらのパラメータの変更は問題なく行うことができる。

現在の提案手法では、毛髪の色と影の区別のつきにくい青い色の髪型や、複雑な方向変化を持つ髪型では、違和感の少ない結果の生成は難しい ( 図 4.7、図 4.8 ) 。図 4.7 の青色の髪型画像を入力とした例では、髪色マップにハイライトや局所的な影がはっきりと残ってしまっており、遮蔽率マップに含まれるべき細かい影を取り除けていないことが分かる。そのため、照明計算の結果画像には頭頂部などに不自然な照明効果が現れるだけでなく、ハイライトが全く現れていない。また図 4.8 の複雑な方向変化を持つ髪型では、三次元方向ベクトル場がうまく推定されないため、与えられた照明方向によっては、毛髪領域の至るところに、穴が開いたように暗い領域が生成されてしまう。こうしたことから、三次元方向ベクトル場をより正確に推定する手法について検討する必要がある。

(a) 入力画像

(b) $\phi = 45^\circ$

(c) $\phi = 135^\circ$ における髪型画像

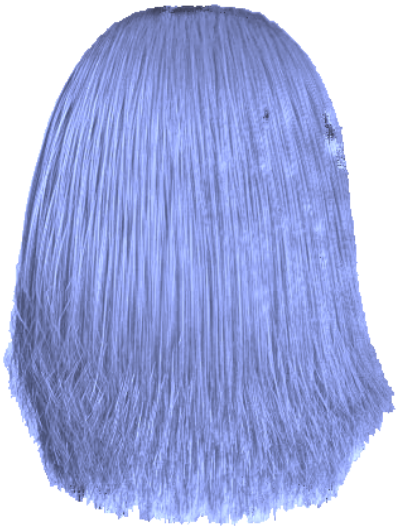
(d) $\phi = 135^\circ$

図 4.5: 照明方向を変更した結果画像。全ての結果について、仰角 $\theta = 45^\circ$ であり、方位角 $\phi$ を変更させて結果画像を生成している。入力画像においては $\phi = 45^\circ$ である。(a), (b) は各照明方向における正しい照明効果を示し、(b), (d) は提案手法において生成された結果である。

(a) 光を弱くした場合

(b) 光を強くした場合

(c) 光を橙色にした場合

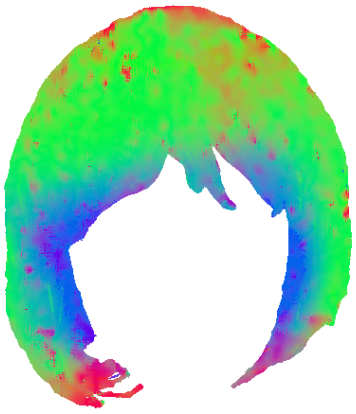


(d) 光を青色にした場合

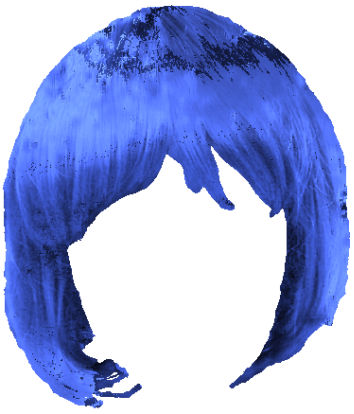
図 4.6: 照明の色や明るさを変更した結果画像。すべての結果について、仰角 $\theta = 45°, \phi = 135°$ として、照明の色や明るさを調整している。

(a) 入力画像

(b) 髪色マップ



(c) 毛髪繊維の三次元ベクトル場

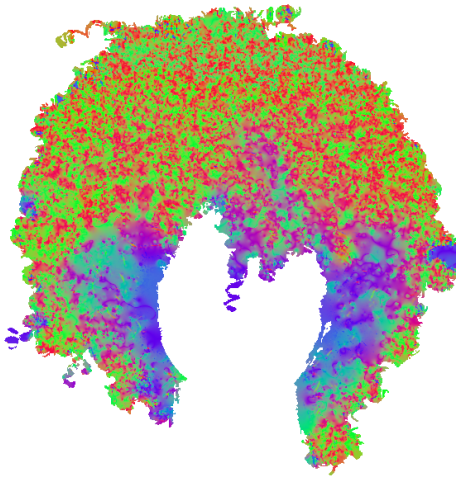


(d) 照明方向を変更した結果

図 4.7: 青色の髪型を入力とした場合の結果画像。(a) 入力画像から生成された (b) 髪色マップにはハイライトや局所的な影がはっきりと残っており、遮蔽率マップに含まれるべき細かい影を取り除けていない。そのため、(d) 照明計算の結果画像には頭頂部などに不自然な照明効果が現れるだけでなく、ハイライトが全く現れていない。なお、照明方向は仰角 $\theta = 45°, \phi = 135°$ に変更している。

(a) 入力画像

(b) 髪色マップ



(c) 毛髪繊維の三次元ベクトル場

(d) 照明方向を変更した結果

図 4.8: 複雑な形状を持つ髪型を入力とした場合の結果画像。(a) 複雑な構造を持つ髪型の入力画像では、(c) 三次元方向ベクトル場がうまく推定されないため、(d) 結果画像において穴のような黒い領域が至る所に生じてしまう場合がある。

## 4.3 Kajiya-Kay モデルを用いた髪色マップ編集における考察

本節では、Kajiya-Kay モデルを利用した毛髪画像の陰影編集手法における現在の問題点を、模範画像を用いた実験に基づいて考察する。実験には、POV-Ray [35] というレンダリングソフトを利用してレンダリングされた、既知のパラメータを持った毛髪画像を使用した。入力として用いている図 4.13(a) の毛髪画像は、照明の方向として方位角 $\phi = 45°$, 仰角 $\theta = 45°$ を与えて生成されたものである。またレンダリング時の各毛髪繊維のデータから取得した各毛髪繊維の三次元方向から毛髪繊維の二次元方向ベクトル場及び三次元方向ベクトル場を生成し、それぞれの模範解答としている。

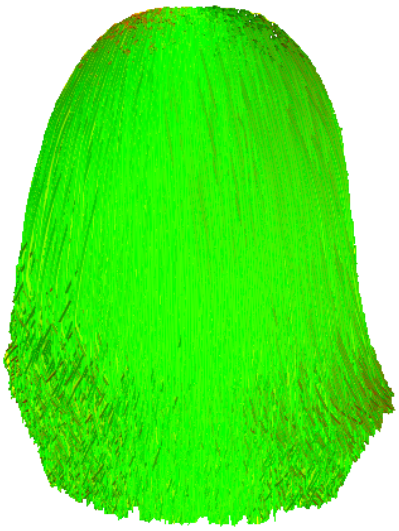
二次元方向ベクトル場、三次元方向ベクトル場、照明の方向の模範解答の与え方を変えたときの結果を示し、各手法における成果と問題点を示す。

### 4.3.1 毛髪繊維の二次元方向ベクトル場

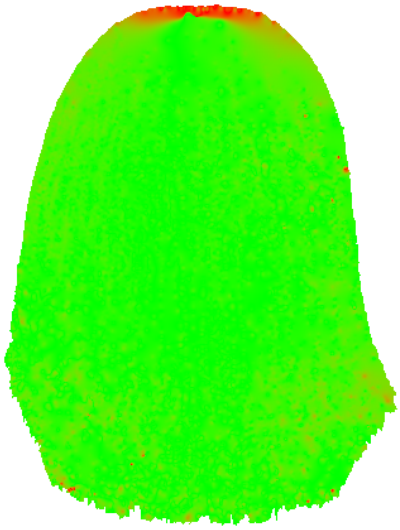
図 4.13 はレンダリングされた入力の毛髪画像 ( 図 4.13(a) ) から得られた毛髪繊維の二次元方向ベクトル場の模範解答 ( 図 4.13(b) ) と、本研究で毛髪繊維の二次元方向ベクトル場を計算する手法として紹介した Watanabe らの手法 ( 図 4.13(c) ) 及び Kang らの手法 ( 図 4.13(d) ) により生成された毛髪繊維の二次元方向ベクトル場を比較した図である。各二次元方向ベクトル場について、各ピクセルにおける色は毛髪繊維の方向ベクトルの各成分の絶対値を表し、赤いほど $x$ 軸方向、緑であるほど $y$ 軸方向の絶対値の大きい成分を持っていることを示している。模範解答においては、各ピクセルの方向ベクトルはほとんど一様に $y$ 軸方向の絶対値の大きい成分を持っており、また入力画像における毛髪繊維の形状に沿って、滑らかなベクトル場を形成している。一方で、本研究で解説した二つの手法による二次元方向ベクトル場では、模範解答のような滑らかなベクトル場を生成できていない。Watanabe らの手法による二次元方向ベクトル場では、頭頂部においては本来滑らかに変化するはずのベクトル場が急激に変化してしまっている。また毛髪領域全体に表れている小さな穴のような部分では、ベクトル場が小刻みに波打っていると考えられる。Kang らの手法では、Watanabe らの手法では小さな穴のようになっていた部分が、それよりも少し大きなノイズとして現れてしまっており、やはり滑らかなベクトル場を得ることはできていない。また頭頂部や毛髪領域の右上部分にベクトル場が滑らかになっていない部分が存在する。これは二次元方向ベクトル場を生成した際に発生したノイズであると思われる。本研究では Kang らの手法により生成された二次元方向ベクトル場に対して、$y$ 成分が正のベクトルを単純に反転することで、ベクトル場全体で $y$ 成分を負に補正するという処理をしている。この処理によりノイズが発生している可能性もある。

図 3.12 において示したように、二次元方向ベクトル場の計算結果は三次元方向ベクトル場の推定結果に大きく影響する。そのため、より毛髪繊維の流れに沿った滑らかなベクトル場の生成できれば、三次元方向ベクトル場の推定や、髪色マップの編集における結果の質を高めることができると考えている。

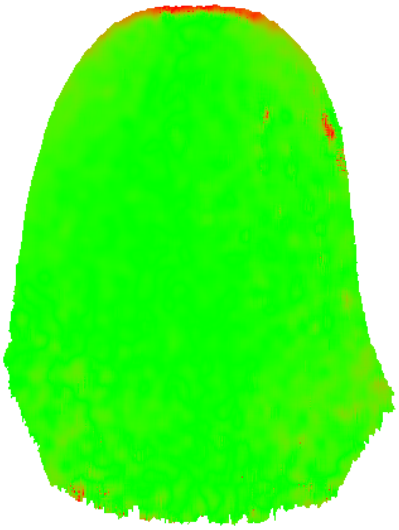
(a) 入力画像



(b) 模範解答



(c) Watanabe らの手法



(d) Kang らの手法

図 4.9: 二次元方向ベクトル場における模範解答との比較。(b) 模範解答の二次元方向ベクトル場では、頭頂部から毛先にかけて滑らかにベクトルが変化している。しかし (c) Watanabe らの手法や (d) Kang らの手法によるものでは、波打ったようなベクトル変化になっており、所々にノイズも発生している。

### 4.3.2　髪色マップと遮蔽率マップ

ここでは、髪色マップと遮蔽率マップが二次元方向ベクトル場の違いによりどの程度の変化があるかを示す。図 4.10 は Watanabe らの手法と Kang らの手法のそれぞれで生成された初期遮蔽率マップを示す。Kang らの手法については、毛髪繊維の二次元方向ベクトル場の $y$ 方向の補正の影響を示すために、方向を補正した場合としていない場合の双方の二次元方向ベクトル場での結果を示している。また図 4.11 では改良遮蔽率マップ、図 4.12 では改良髪色マップを同様の方式により生成したものを比較している。それぞれ図を見ると分かるように、初期遮蔽率マップ、改良遮蔽率マップ、改良髪色マップのいずれについても、二次元方向ベクトル場の違いにより、大きく異なる結果は生成されなかった。従って、本研究で解説した手法によって初期遮蔽率マップを計算し、改良遮蔽率マップ及び改良髪色マップを生成する際の初期値として利用すれば、二次元方向ベクトル場の質にかかわらず同等の結果が生成できることが分かった。

(a) 図 4.13(c) による初期遮蔽率マップ　(b) 補正なしの図 4.13(d) による初期遮蔽率マップ　(c) 図 4.13(d) による初期遮蔽率マップ

図 4.10: 二次元方向ベクトル場ごとの初期遮蔽率マップの比較。二次元ベクトル場はそれぞれ、(a) Watanabe らの手法、(b), (c) Kang らの手法のものをそれぞれ利用している。(b) 補正なしは、二次元ベクトル場の $y$ 成分を負に補正していないことを示す。二次元方向ベクトル場の違いによって各初期遮蔽率マップに大きな違いはない。

(a) 図 4.10(a) による改良遮蔽率マップ (b) 図 4.10(b) による改良遮蔽率マップ (c) 図 4.10(c) による改良遮蔽率マップ

図 4.11: 二次元方向ベクトル場ごとの改良遮蔽率マップの比較。二次元ベクトル場はそれぞれ、(a) Watanabe らの手法、(b), (c) Kang らの手法のものをそれぞれ利用している。初期遮蔽率マップは、それぞれ図 4.10 の対応する画像を用いている。二次元ベクトル場の違いにより、遮蔽率マップの改良の質に大きな差はない。

(a) 図 4.10(a) による改良髪色マップ (b) 図 4.10(b) による改良髪色マップ (c) 図 4.10(c) による改良髪色マップ

図 4.12: 二次元方向ベクトル場ごとの改良髪色マップの比較。二次元ベクトル場はそれぞれ、(a) Watanabe らの手法、(b), (c) Kang らの手法のものをそれぞれ利用している。初期遮蔽率マップは、それぞれ図 4.10 の対応する画像を用いている。二次元ベクトル場の違いにより、髪色マップの改良の質に大きな差はない。

### 4.3.3 毛髪繊維の三次元方向ベクトル場

ここでは、三次元ベクトル場の模範解答を示すと共に、二次元方向ベクトル場や照明方向が既知であるか推定されたかによって、三次元方向ベクトル場の推定結果がどのように変化するかを示す。本節で示す各結果画像における計算において、与えた二次元方向ベクトル場と照明方向のデータが既知であるか推定されたかどうかを表 4.2 にまとめる。また、図 4.15 で示した照明方向の変更結果について、照明の方向ベクトルとして画像の中心から極座標で $\theta = 45°, \phi = 135°$ に向かう平行光源を与えている。図 4.13(a) の髪型画像は、POV-Ray というレンダリングソフトを用いてレンダリングされた画像である。模範解答に関しては、図 4.13 の髪型画像をレンダリングする際のデータから、毛髪繊維の三次元方向ベクトル場と二次元方向ベクトル場、照明方向、環境光成分を生成した。毛髪繊維の三次元方向ベクトル場と二次元方向ベクトル場は、各毛髪繊維の座標データからベクトルを生成し、各ピクセルの方向ベクトルとして生成した。照明方向はレンダリングした際に自分で指定した方向となっている。環境光成分については、POV-Ray におけるレンダリングの際に、拡散反射成分、鏡面反射成分をゼロとして、環境光成分だけを持つ髪型画像をレンダリングすることで生成した。なお、環境光成分の模範解答は本節のすべての結果に適用されている。

3.2.2 節において解説した Kang らの手法において、毛髪繊維の方向ベクトルの $y$ 成分を負に補正することが三次元ベクトル場の推定結果や照明方向の変更結果にどのような影響を与えるかを示す。 図 4.14 では、本研究で解説した Kang らの手法により生成された二次元方向ベクトルの $y$ 成分を負に補正した場合としていない場合の二つの二次元ベクトル場を利用して推定された三次元方向ベクトル場を示す。図 4.14(a) において、頭頂部は極端に $z$ 成分が大きく、それ以外の部分では $z$ 成分がほとんどなく、全体としては平らな方向ベクトル場となってしまっている。一方で、図 4.14(b) では頭頂部や髪の下部では $z$ 成分が強く、中央の部分では $z$ 成分が小さいというように、ベクトルの変化が滑らかになっている。この $y$ 成分の補正処理に対する影響は、図 4.15 において示す照明方向の変更結果ではより分かりやすく現れる。$y$ 成分を負に補正していない場合の三次元方向ベクトル場を利用した結果 ( 図 4.15 ) では、陰影が波打ったようになっているが、$y$ 成分を負に統一した場合の三次元方向ベクトル場を利用した結果 ( 図 4.15 ) においては、それが比較的滑らかになっている。

図 4.16 では三次元方向ベクトル場の模範解答と、各二次元方向ベクトル場 ( 模範解答、Watanabe らの手法、Kang らの手法 ) を用いて推定された三次元方向ベクトル場の比較を示す。図 4.16(b) は、三次元ベクトル場の模範解答 ( 図 4.16(a) ) と比較して、模範解答と同じような三次元方向ベクトル場を推定できている。従って、二次元方向ベクトル場が正確であれば、提案した推定手法によって三次元方向ベクトル場の推定が可能であると考えられる。ただし、頭頂部に関してはノイズが発生していることから、正確な推定手法とは言えない。また図 4.16 及び 4.17 では、二次元方向ベクトル場のノイズによって $x$ 軸方向の成分が強くなってしまっている部分が見られる。また、三次元方向ベクトル場の模範解答と同様に、頭頂部や左下部分に $z$ 軸方向に向く成分を持っているものの、繊維ごとではなくその領域に満遍なく分布してしまっているため、二次元方向ベクトル場には繊維ごとの細かい方向の違いを区別できるようなベクトル場の生成手法が必要であると考えられる。

図 4.17 は二次元方向ベクトル場の模範解答に加え、入力画像をレンダリングした際の照明方向を与えて、三次元方向ベクトル場を推定した結果である。しかしながら、どの三次元方向ベクトル場においても、毛髪領域全体が極端に大きい $z$ 成分を持つ結果となってしまった。本研究で解説した三次元方向ベクトル場の推定手法においては、ある照明方向を与えたとき、髪色マップと Kajiya-Kay モデルによる計算結果の誤差の総和が最小となるような三次元ベクトル場を最適値としている。この時、扱うデータが正確であったとしても Kajiya-Kay モデルが実際の毛髪の色を正確に計算できるモデルではない ( 図 2.1 参照 ) ため、最小となる誤差の総和を導く照明方向は、実際の照明方向とは異なる場合がある。図 4.16(b) においても、推定された照明の方向は、$\theta = 45°, \phi = 270°$ であり、実際の照明方向 ($\theta = 45°, \phi = 45°$) とは異なっている。こうしたことから、Kajiya-Kay モデルでは本研究における毛髪の方向ベクトル推定には不十分である。しかしながら、Marschner モデルでは推定すべきパラメータの数の多さから、パラメータ推定がより困難となるため、どのように三次元方向ベクトル場の推定を行うかが今後の課題の一つである。

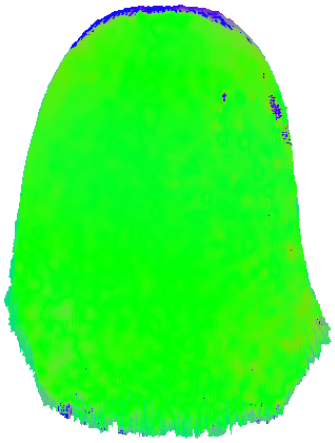
表 4.2: 各結果画像における条件。各結果画像において、二次元方向ベクトル場と照明方向についてどのようなパラメータを利用して、毛髪繊維の三次元方向ベクトル場の推定を行ったかを示す。

| | 二次元方向ベクトル場 | 照明方向 |
|---|---|---|
| 図 4.14 (a) | Kang らの手法 | 未知 |
| 図 4.14 (b) | Kang らの手法 | 未知 |
| 図 4.16 (b) | 模範解答 | 未知 |
| 図 4.16 (c) | Watanabe らの手法 | 未知 |
| 図 4.16 (d) | Kang らの手法 | 未知 |
| 図 4.17 (b) | 模範解答 | 模範解答 |
| 図 4.17 (c) | Watanabe らの手法 | 模範解答 |
| 図 4.17 (d) | Kang らの手法 | 模範解答 |

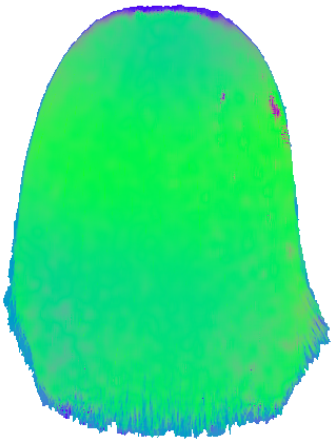
(a) 髪型画像

(b) 環境光成分の模範解答

図 4.13: レンダリングされた髪型画像と環境光成分の模範解答。(a) 髪型画像をレンダリングした際に用いた照明の方向と、各繊維のデータから生成した毛髪繊維の三次元方向ベクトル場と二次元方向ベクトル場を、それぞれの模範解答とする。(b) 環境光成分は、(a) 髪型画像を生成する際に得られる。



(a) $y$ 成分を負に統一していない二次元方向ベクトル場による三次元方向ベクトル場



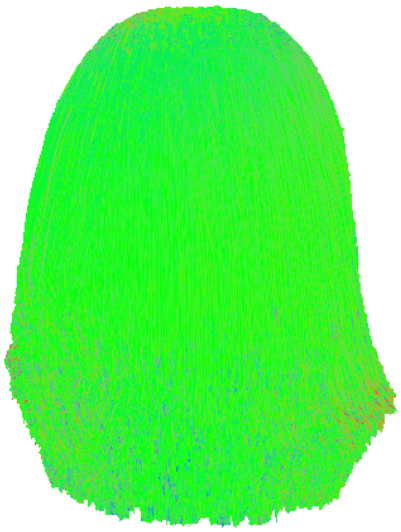
(b) $y$ 成分を負に統一した二次元方向ベクトル場による三次元方向ベクトル場

図 4.14: 各二次元方向ベクトル場により推定された三次元方向ベクトル場の比較。(a) Kang らの手法で計算された二次元方向ベクトル場の $y$ 成分を統一せずに利用して推定された三次元方向ベクトル場。(b) Kang らの手法で計算された結果におけるベクトルの $y$ 成分を統一した二次元方向ベクトル場を利用して推定された三次元方向ベクトル場。

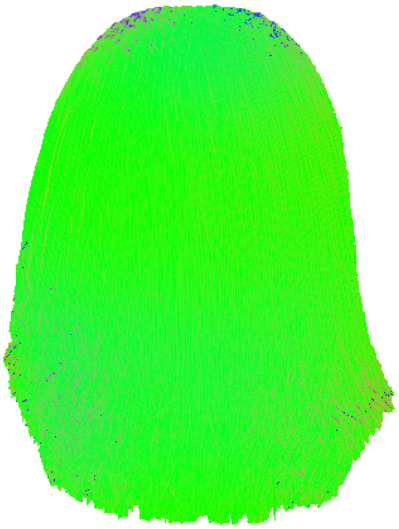
(a) 図 4.14(a) を用いた照明変更の結果

(b) 図 4.14(b) を用いた照明変更の結果

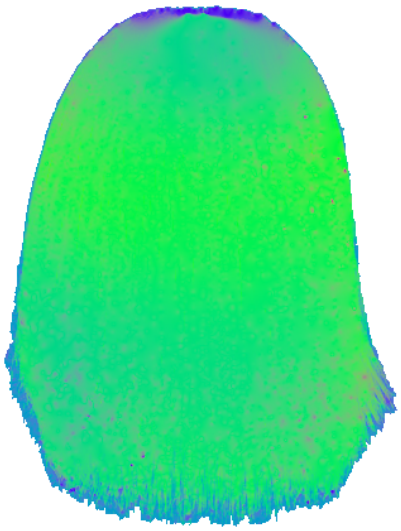
図 4.15: 図 4.14 の三次元ベクトル場を用いて計算された照明変更の結果の比較。(a) 図 4.14(a) を用いた結果では、特に右中央部分がノイズの影響から波打った陰影変化になっているが、(b) 図 4.14(b) を用いた結果ではそれが滑らかになっている。



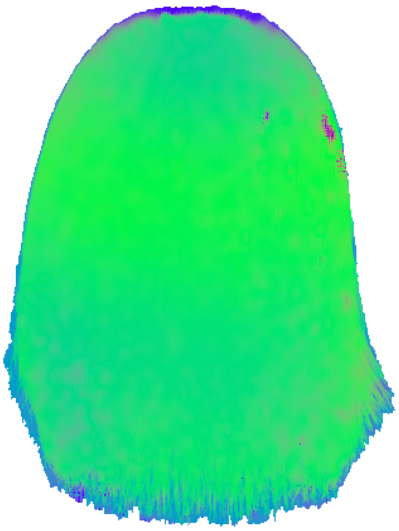
(a) 三次元方向ベクトル場の模範解答



(b) 二次元方向ベクトル場の模範解答による三次元方向ベクトル場

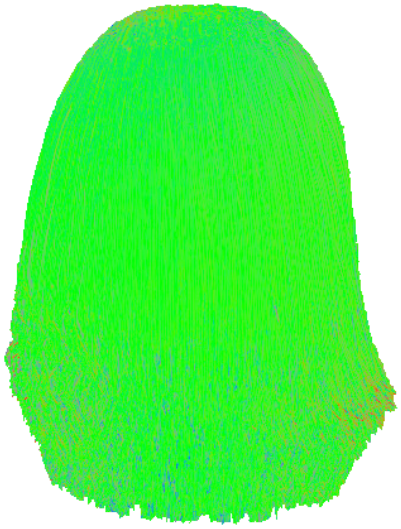


(c) Watanabe らの二次元方向ベクトル場による三次元方向ベクトル場

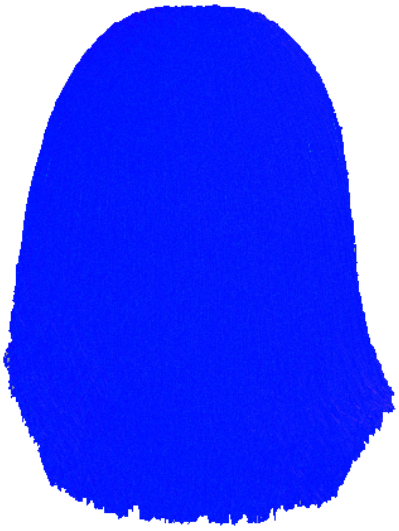


(d) Kang らの二次元方向ベクトル場による三次元方向ベクトル場

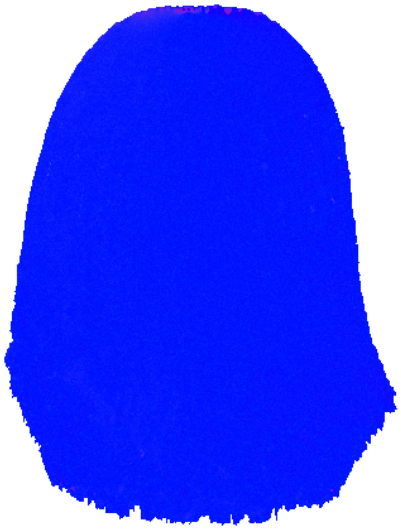
図 4.16: 二次元方向ベクトル場の模範解答を利用した三次元方向ベクトル場の比較。



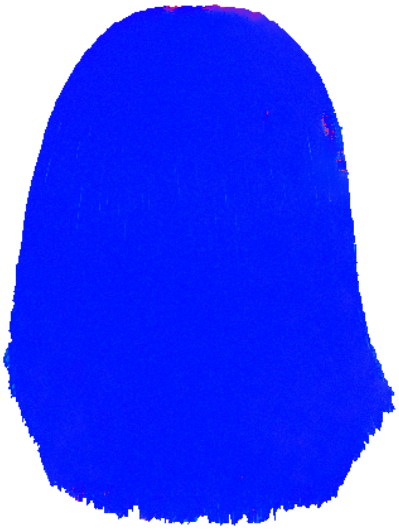
(a) 三次元方向ベクトル場の模範解答



(b) 二次元方向ベクトル場の模範解答による三次元方向ベクトル場



(c) Watanabe らの二次元方向ベクトル場による三次元方向ベクトル場



(d) Kang らの二次元方向ベクトル場による三次元方向ベクトル場

図 4.17: 二次元方向ベクトル場と照明方向の模範解答を利用した三次元方向ベクトル場の比較。

# 第5章　結論

本研究では、毛髪画像における陰影の編集手法を提案した。提案手法では、毛髪が大域的な陰影を含む髪色マップと、局所的な影を表す遮蔽率マップとの積で表現できると仮定し、毛髪繊維の遮蔽による細かい影を維持するために、画像から遮蔽部分を粗く検出した初期遮蔽率マップを計算した。次に、初期遮蔽率マップを用いて、髪色マップと遮蔽率マップの改良を行った。改良された髪色マップに対して編集を行うことで、毛髪の陰影の編集を行った。髪色マップの編集手法として、混合ガウスマップを用いた手法と、Kajiya-Kay モデルを利用した二つの手法を紹介した。混合ガウス分布を用いた手法では、髪色マップを複数のガウス分布により表現し、ガウス分布を移動させることにより髪色マップを編集した。Kajiya-Kay モデルを利用した手法では、改良された髪色マップに Kajiya-Kay モデルを当てはめ、髪色マップから毛髪繊維の三次元方向ベクトル場と入力画像の照明の方向ベクトルを推定した。そしてユーザの与えた照明方向に基づいて照明計算を行い、結果を出力する。双方の手法において、ユーザは対話的な操作により髪色マップを編集し、陰影を変更した髪型画像を出力結果として得ることができる。

提案手法では細かい毛髪繊維と、それによって生じる細かい影を扱うため、各処理の結果の質は、入力された毛髪画像の画質に大きく依存する。画質の低さから毛髪繊維間の細かい影が不鮮明になってしまうと、髪色マップと遮蔽率マップの分離と改良がうまくいかない場合がある。

今後の課題として、毛髪繊維の三次元方向ベクトル場の推定手法、及び推定された三次元方向ベクトル場の平滑化手法の改良を検討する必要がある。現在の推定手法では照明方向の推定ができていない。また提案した平滑化項を加えたエネルギー関数による平滑化は、元の三次元方向ベクトル場とそれほど変わらないだけでなく、ノイズを広げた結果を生成してしまう。毛髪繊維の流れを反映した滑らかな三次元方向ベクトル場と、元画像の照明方向を推定できる手法が求められる。また毛髪画像の髪色マップと遮蔽率マップの分離と改良について、提案手法は髪色マップにおけるハイライトの影響を完全に除去できていない。そのため、陰影だけを編集するためには、残ったハイライトを除去する手法についても検討していきたい。

# 謝辞

本研究を進めるにあたり、ご指導を頂きました本学システム情報工学科の金森由博助教、三谷純准教授に深謝いたします。また、本論文をご精読頂き有用なコメントを頂きました本学システム情報工学研究科の酒井宏教授、福井和広教授に心より感謝いたします。多くの有益なご助言を頂きました本学システム情報工学科の福井幸男名誉教授、西原清一名誉教授に感謝いたします。また本論文の執筆に際しまして、校正にご協力頂いた飯塚里志先輩を始め、非数値処理アルゴリズム研究室の皆様に深くお礼を申し上げます。

# 参考文献


[1] J. T. Kajiya and T. L. Kay. 1989. Rendering fur with three dimentsional textures. In Proc. of 1989 ACM SIGGRAPH, vol. 23, no. 3, pp. 271-280.

[2] D. B. Goldman. 1997. Fake fur rendering. In Proc. of SIGGRAPH, ACM Press, pp. 127-134.

[3] S. R. Marschner, H. W. Jensen, M. Cammarano, S. Worley and P. Hanrahan. 2003. Light scattering from human hair fibers. ACM Trans. on Graphics, vol. 22, no. 3, pp. 780-791.

[4] A. Wenger, T. Hawkins and P. E. Debevec. 2003. Optimization color matching in a lighting reproduction system for complex subject and illminant spectra. In Proc. of the 14th Eurographics workshop on Rendering, pp. 249-259, Eurographics Association.

[5] T. Hawkins, A. Wenger, C. Tchou, A. Gardner, F. Goransson and P. Debevec. 2004. Animatable facial reflectance fields. In Rendering Techniques'04, pp. 309-321.

[6] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins and P. Debevec. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination. In ACM SIGGRAPH 2005 Papers, pp. 756-764.

[7] C-F. Charbert, P. Einarsson, A. Jones, B. Lamond, W-C. Ma, S. Sylwan, T. Hawkins and P. Debevec. 2006. Relighting human locomotion with flowed reflectance fields. In ACM SIGGRAPH 2006 Sketches.

[8] M. Okabe, G. Zeng, Y. Matsushita. T. Igarashi, L. Quan and H. Y. Shum. 2006. Single-view relighting with normal map painting. In Proc. of Pacific Graphics, pp. 27-34.

[9] J. Chen, G. Su, J. He and S. Ben. 2010. Face image relighting using locally constrained global optimization. In Proc. of ECCV, Part IV, pp. 44-57.

[10] Y. Wang, Z. Liu, G. Hua, Z. Wen, Z. Zhang and D. Samaras. 2007. Face Re-Lighting from a single image under harsh lighting conditions. In Proc. of IEEE International Conference on Computer Vision and Pattern Recognition 2007, pp. 1-8.

[11] A. Munoz, J. I. Echevarria, F. Seron, J. Lopez-Moreno, M. Glencross and D. Gutierrez. 2011. BSSRDF estimation from single image. Computer Graphics Forum, vol. 30, no. 2, pp. 455-464.

[12] X. Teng and T-J. Cham. 2011. Image Relighting by Analogy. Proc. of the 7th International Conference on Advances in Visual Computing - Volume Part I, pp. 78-89.

[13] T. Yu, H. Wang, N. Ahuja and W-C. Chen. 2006. Sparse lumigraph relighting by illumination and reflectance estimation from multi-view images. Proc. of the 17th Eurographics Conference on Rendering Techniques, pp. 41-50.

[14] 山田 裕貴, 金森 由博, 福井 幸男, 三谷 純. 2013. “Kinect を用いたリライティング”. 映像情報メディア学会誌 研究速報, vol. 67, no. 5, pp. J152-J154.

[15] T. Ikeda, F. de Sorbier, H. Saito. 2013. Arbitrarily Shaped Objects Relighting Using an RGB-D Camera. Proc. of the 2013 2nd IAPR Asian Conference on Pattern Recognition, pp. 631-636.

[16] E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley. 2001. Color transfer between images. IEEE Comput. Graph. Appl, vol. 21, no. 5, pp. 34-41.

[17] Y-W. Tai, J. Jia and C-K. Tang. 2005. Local color transfer via probabilistic by expectation-maximization. Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01, pp. 747-754.

[18] R. M. H. Nguyen, S. J. Kim and M. S. Brown. 2014. Illuminant aware gamut-based color transfer. Computer Graphics Forum, vol. 33, no. 7, pp. 319-328.

[19] A. Levin, D. Lischinski and Y. Weiss. 2004. Colorization using optimization. ACM Trans. Graph., vol. 23, no. 3, pp. 689-694.

[20] Y. Qu, T-T. Wong and P-A. Heng. 2006. Manga colorization. ACM Trans. Graph., vol. 25, no. 3, pp. 1214-1220.

[21] E. A. Khan, E. Reinhard, R. W. Fleming and H. H. Bulthoff. 2006. Image-based material editing. ACM Trans. on Graphics, vol. 25, no. 3, pp. 654-663.

[22] M. Chai, L. Wang, Y. Weng, Y. Yu, B. Guo and K. Zhou. 2012. Single-view hair modeling for portrait manipulation. ACM Trans. on Graphics, vol. 31, no. 4, pp. 116:1-116:8.

[23] M. Chai, L. Wang, Y. Yu, B. Guo and K. Zhou 2013. Dynamic hair manipulation in images and videos. ACM Trans. on Graphics, vol. 32, no. 4, 75:1-75:8.

[24] J. Shen, X. Yang, Y. Jia and X. Li. 2011. Intrinsic image using optimization. In Proc. of IEEE Conf. Computer Vision and Pattern Recognition, pp. 3481-3487.

[25] H. Watanabe, Y. Kanamori, J. Mitani and Y. Fukui. 2011. An image-based hair styling system layered texture pieces. In Proc. of Pacific Graphics 2011, pp. 7-12.

[26] S. Paris, H. Briceno and F. Sillion. 2004. Capture of hair geometry from multiple images. ACM Trans. on Graphics (Proc. of the SIGGRAPH conference), vol. 23, no. 3, pp. 712-719.

[27] S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. F. Cohen. 1996. The lumigraph. In Proc. of the 23rd annual conference on Computer graphics and interactive techniques, SIGGRAPH '96, pp. 43-54.

[28] H. Kang, S. Lee and C. K. Chui. 2007. Coherent line drawing. In Proc. of the $5^{th}$ International Symposium on Non-photometric Animation and Rendering, pp. 43-50.

[29] S. Richard. 玉木 徹, 福嶋 慶繁, 飯山 将晃, 鳥居 秋彦, 栗田, 波部 斉, 林 昌希, 野田 雅文 ( 訳 ). 2013. コンピュータビジョン -アルゴリズムと応用-. 共立出版.

[30] 株式会社イン フィニシス. Change My Image 3. http://www.infinisys.co.jp/product/cmimage2/index.shtml

[31] バイロメディア有限会社. Sturk. http://sturk.jp/

[32] らしさドットコム株式会社. Rasysa. http://www.rasysa.com/

[33] AITIA Corporation. ヘアスタイルナビ. https://itunes.apple.com/jp/app/heasutairunabi/id516243021

[34] mandom corp. 髪コレ. https://itunes.apple.com/jp/app/fakore/id408828994?mt=8

[35] POV-Ray. 2004. version 3.6, Persistence of Vision Pty. Ltd., Persistence of Vision Raytracer. http://www.povray.org/

[36] $\mathrm{Photoshop}^{©}$. Adobe. https://www.adobe.com/jp#