

インタラクティブなヘアスタイリングシステムの開発

システム情報工学研究科 コンピュータサイエンス専攻 博士前期課程 2 年
200820683 林本法也

指導教員 福井幸男

2009 年 06 月 04 日

1 はじめに

髪は、人間の頭に約 10 万本存在し、その太さも直径 0.05mm ~ 0.15mm と、非常に細いが、髪型によって各個人の個性を表現したり、髪型によって他人に与える印象が変化するなど、大変重要な存在である。また、ある髪の運動によって、体全体の動きを推測することもできる。現状では、個人が好みの髪型に変えるには、雑誌などに記載されている写真などを参考にするか、あるいは個人が口頭で、美容師に髪型の全体像を伝達するのが一般的である。

しかし、雑誌の写真を参考にする場合は写真の顔が自分の顔ではないので、選んだ髪型が自分に本当に合っているかどうかの判断が難しい。また、雑誌の写真は 2 次元であり、任意のアンクルから撮影された写真のみしか掲載されておらず、全体像を把握するには情報量が乏しい。美容師に口頭で伝える場合には、自分の説明能力や、美容師の読解力や想像力に依存してしまい自分が望んでいた髪型とズレが生じてしまう可能性も十分あると考えられる。

上述した問題を解決するには、ユーザが髪型を簡単に生成することができ、それを他者に容易に提示できることのできるヘアスタイリングシステムの開発が必要になってくると考えられる。

そこで、本稿では 3 次元空間上における新たなヘアスタイリングシステムを提案する。提案システムは最終的に目指す目標は以下の 3 つを予定している。

1. 直観的な操作により、容易に髪型を編集できる機能
2. 個人の顔を 3 次元空間上に再構築し、任意の髪型との合成
3. 顔の形状を測定し、その顔型に最適な髪型を提案する機能

本研究では、上記の 3 つの内、1. のインタラクティブな操作による髪型の編集を可能にする機能を中心に、提案・開発を行う。

2 関連研究・技術

本節では、既存のヘアスタイリングシステムにおける関連研究や、サービス、製品について述べる。

3 次元空間上におけるヘアスタイリングに関する研究としては、Tae-Yong と Ulrich Neumann による Multiresolution Hair Modeling(MHM) システム [1] などが挙げられる。

MHM の特徴としては、「多重解像度」という概念を利用して、汎用シリンダを用いた髪モデルを任意の頭部ポリゴンモデルに構築して髪型を生成している。また、ある髪の形状やパラメータを別の場所へとコピー & ペースト機能を有している他、シェーディングやアンチエイリアシング、セルフシャドウイングについても言及している。

また、Web サービスの分野では、個人の顔写真を 3 次元形状に再構築して、仕上がりのヘアスタイルを提案するコンサルテーションソフトとして、シュワルツコフ & ヘンケル株式会社が提供している「Virtual Preview」 [2] といったアプリケーションがある。

Virtual Preview の主な特徴としては、「モーションポートレート技術」という技術を用いて、写真から 3 次元形状を構築している。また、表情エンジンによって多彩な表情の再現も可能となっている。

3 髪モデル

本節では、本研究で提案する髪モデルと髪の挙動を再現するアルゴリズムについて述べる。

3.1 Hair Sheet モデル

本研究では、髪モデルを構築する際に Hair Sheet モデルという構造モデルを提案する。これは、複数のキーとなる髪の毛を定義し、このキーとなる髪の毛間を曲線で補間することでシート形状に髪の毛をグルーピングし、制御するモデル手法である。Hair Sheet モデルの構成要素を図 1 に示す。

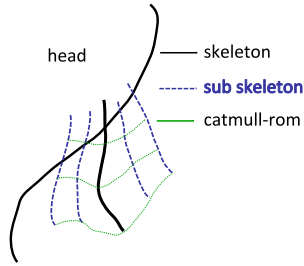


図 1: Hair Sheet モデル

Hair Sheet モデルは主に 3 つの要素から構成される。

Skeleton は、Hair Sheet の中心に位置する髪の毛 (以下 Hair Strand とする) を示している。Skeleton は、ポリライン (点列) として定義される。各点は、質量などの情報を持つパーティクルである。

Sub Skeleton は、Skeleton 以外の Hair Strand として定義される。構成は、Skeleton と同等である。

Catmull-Rom は、点列を補間するスプライン曲線の一つで、Skeleton と Sub Skeleton の同じインデックス番号に属するパーティクルを通るように曲線で補間する。

3.2 Shape Matching

本研究では、髪モデルの挙動を計算コストを極力抑えて効率よく再現するために、Shape Matching [3] という形状変形アルゴリズムの概念を導入した。

Shape Matching とは、オブジェクト (ポリゴンモデルと仮定する) の材質パラメータなどに依存せずに、各頂点の初期状態の位置座標と現在の位置座標から目標座標を算出し、その目標座標に向かうことによって、オブジェクトを変形することができる手法である。

3.3 Shape Matching に基づく髪モデルの伸縮・曲げ処理

Hair Strand の伸縮と曲げを制御するために 3.2 節で述べた Shape Matching の概念を応用した。その処理の流れを図 2 に示し、手順を以下に示す。

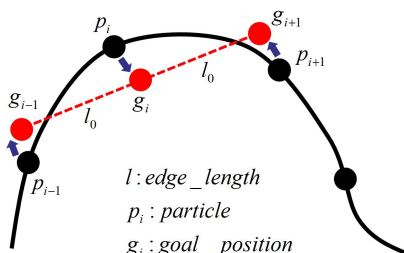


図 2: 髪の毛の伸縮・曲げ処理

以下に、Hair Strand の伸縮・曲げ処理にかかる力の計算の手順を示す。

1. ある Hair Strand における 3 つのパーティクル p_{i-1}, p_i, p_{i+1} に注目する
2. p_{i-1} と p_{i+1} とを結ぶベクトルを計算する。また 3 点の重心座標も計算する
3. 2 で求めたベクトルが、重心座標を通るように平行移動する
4. ベクトルを正規化し、事前に設定したエッジ長 l_0 間隔に目標座標 g_i を配置する

1 ~ 4 の処理を (Hair Strand に含まれるパーティクルの数 2) 番目のパーティクルまで行う。この処理を行うと、目標座標値が複数回計算されるパーティクルが発生するが、その場合は各処理で得られた座標値を計算回数で割り、その平均値をそのパーティクルの目標座標位置とする。

上記の処理によって各パーティクルにかかる力 $force_i$ は次式で表わされる。

$$force_i(p_i) = gravity + \alpha \times ((\sum_{j=1}^m g_j) / m) - pos(p_i)$$

但し、 g は目標位置座標、 $pos(p_i)$ は p_i の位置を、 m は p_i において g_i が計算された回数をそれぞれ表す。

3.4 衝突検知処理

次に Hair Strand が頭部モデルに衝突した場合の処理について述べる。3.3 節で述べた処理のみを考慮すると、Hair Strand が頭部モデル内に埋没してしまうので、その衝突を防ぐために、衝突を検知しそれに対処する手法が必要になってくる。処理手順を Algorithm1 に示す。また、図 3 にその概観を示す。

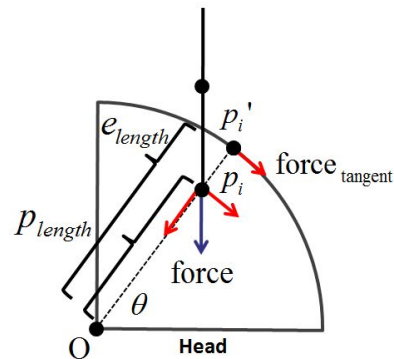


図 3: 衝突検知

Algorithm 1 衝突検知アルゴリズム

p_{length} : p から楕円体の中心 O までの距離
 e_{length} : θ にある楕円面上の点から O までの距離
if $p_{length} < e_{length}$ **then**
 $p_i = p'_i$
 $force_{tangent} = force_i \times \cos \theta$
 $force_i = force_{tangent}$
end if

4 提案システム

4.1 システム概要

本研究で提案するシステムの概要を図4に示す。

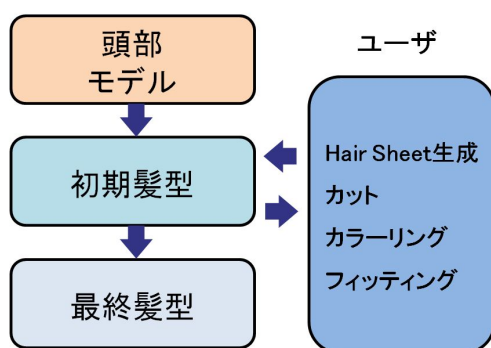


図4: 提案システム

まず初めに、3次元空間上に頭部モデルを構築する。そして、初期設定としていくつかの Hair Sheet を頭部モデル上に追加する。そして、ユーザによる髪を切る操作（カット処理機能）やフィット機能による操作によって、髪型を編集する。また、ユーザが行った操作による髪型の更新は、リアルタイムで実行し再描画され、ユーザにフィードバックされる。

4.2 カット処理

カット処理アルゴリズムを Algorithm2 に示す。

ユーザは、マウス操作によって、3次元空間上にマウスの軌跡（マウスストローク）を描くことができる。このマウスストロークがカット処理時のカットラインとなる。システムは、全ての Hair Strand がマウスストロークと交差しているかどうかを判定し、交差しているならば、交点を算出し、その交点で Hair Strand を分割する処理を行う。最後に交点を Hair Strand の先端のパーティクルとして、新たに格納する。

Algorithm 2 カット処理アルゴリズム

マウスストロークを 3D 空間に射影
for all edge in all Hair Strands **do**
 for all edge in mouse stroke **do**
 交差判定
 if 交差している **then**
 交点算出
 Hair Strand に格納
 end if
 end for
end for

4.3 フィット処理

フィット処理アルゴリズムを Algorithm3 に示す。

ユーザは、まずマウスのドラッグなどで、変形させる領域を選択する。選択領域内にある Hair Strand からキーとなる曲線を算出する。キー曲線と各 Hair Strand との位置情報などを保持しつつ、ユーザが描くマウスストロークによって、このマウスストロークとキー曲線をマッチングさせる。キー曲線の変形によって各 Hair Strand も追従して変形される。

この処理によって、髪の前毛やパーマなどの特徴が表現できることが可能になると考えられる。

Algorithm 3 フィット処理アルゴリズム

マウスによる選択エリアの設定
マウスストロークを 3D ワールド座標系に変換
for all Hair Strand in selected area **do**
 選択エリア内の Hair Strand のキー曲線を算出
 キー曲線と各 Hair Strand との距離情報を算出・保持
 キー曲線とマウスストロークのマッチング
end for

5 実装・評価

本節では、現段階での実装状況を述べる。

5.1 実行環境

本研究における実行環境を表1に示す。

5.2 実装状況・実行時間計測

現時点での実装は、3次元空間上の平面に頭部モデルを生成し、その上部に Hair Sheet を 435 枚配置した。実行時の図を図5に示す。

OS	Windows Vista Business Service Pack1
CPU	Intel(R) Core2 DUO E6750 2.66GHz
メモリ	2.00GHz
グラフィック	NVIDIA GeForce 8600 GT 1022MB
開発・言語	Java (JRE1.6.0) Eclipse

表 1: 実装環境

1 枚の Hair Sheet に含まれる Hair Strand の数は 40 本と設定しているため、Hair Strand の総数は約 17000 本である。また、実行時間を計測した結果、Hair Strand の各パーティクルにかかる力の計算や頭部との衝突検知などの数値計算処理にかかる時間は、Hair Strand の数が増えるほど $O(n)$ の比でほぼ増加する傾向にあった。フレームレートは、おおよそ 30 ~ 70fps の範囲で計測された。

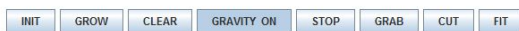
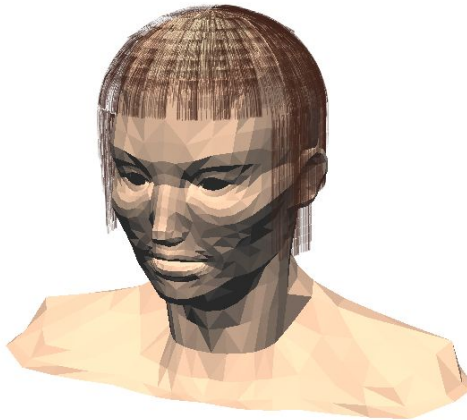


図 5: Hair Sheet 生成時の結果

図を見ると、頭部上に薄い筋が発生しているのが確認できる。これは、Hair Sheet の座標値 (y 座標値) が等間隔で割り当てられているためと考えられる。

また、Hair Strand が頭部に埋没している箇所も何か所か挙げられるが、これは、頭部を楕円体で擬似的に表現しているためにこのような埋没箇所が発生したと考えられる。このような問題を解決するためには、Hair Strand がの初期配置座標をランダムに割り当てたり頭部モデルの形状を高精度で近似できるモデルの構築が必要であると考えられる。

5.3 評価方法の提案

本節では、提案するシステム、及び手法の評価について述べる。本研究では主に以下の 2 つの実験の

方向で実施することを予定している。

髪モデルの検証

実世界での髪と仮想空間上に構築した髪モデルの挙動の差異を比較する。その差異からより実世界の挙動に近いパラメータを設定することで、提案する髪モデルの有用性を検証する。

システムの検証

システムのユーザビリティや満足度などを複数ユーザを対象にアンケート形式で回答していただく。可能ならば、実際に本システムを用いて、美容師と共に髪型を決定していただき、そのフィードバックも得たいと考えている。

6 まとめと今後の課題

本稿では、髪のをモデリングする手法として、Hair Sheet モデルというモデリング手法を提案し、実装を行った。

また、重力などによる髪の挙動を計算コストが低く安定した解を得ることのできる形状変形手法である Shape Matching の概念を用いてシミュレートした。

今後の課題としては、まずは、カット機能など各種機能が未実装なので、早急に行う。また、髪のボリュームを再現するために髪と髪に働く力も考慮する必要がある。そして、頭部モデルを 3 次元形状として疑似的に表現する際の精度の上昇も挙げられる。

参考文献

- [1] TY Kim and U Neumann. Interactive multiresolution hair modeling and editing. *International Conference on Computer Graphics and Interactive Techniques Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, Vol. 21, pp. 620–629, 2002.
- [2] Virtual preview. <https://virtual-preview.com/LE/>.
- [3] M.M. Müller, B. Heidelberger, M. Teschner, M. Gross. Meshless deformations based on shape matching. *Proceedings of ACM SIGGRAPH 2005*, Vol. 24, No. 3, pp. 471–478, 2005.