

# 3D ヘアモデリングにおける髪束両端の描画によるヘアカード生成

高野正統<sup>1)</sup> 森本有紀<sup>1)</sup> (正会員)

1) 九州大学芸術工学府

## Hair Card Generation by Drawing Both Ends of a Hair Bundle in 3D Hair Modeling

Masamune Takano<sup>1)</sup> Yuki Morimoto<sup>2)</sup>(Member)

1) Graduate School of Design, Kyushu University

2) Faculty of Design, Kyushu University

morimoto@design.kyushu-u.ac.jp

### アブストラクト

我々は、直感的かつ素早くヘアモデリングのため、頭部を囲う形や平面や任意の形状の 3D キャンバスを用い、髪束両端の輪郭線の手描き入力によって髪束を表すヘアカードを生成するシステムを提案する。1本の線を軸として髪束を生成する既存手法ではユーザの意図を詳細に反映することが難しい。本手法では手描きのイラスト作成のように髪束を二本の輪郭線からモデリングことでより自由に詳細な形状を指定しより直感的に詳細を指定しやすい。またその際に用いる 3D キャンバスは、頭部を覆う等位面や、任意の平面・3D モデル、入力線のスイープと補間によるキャンバスの作成機能により柔軟な指定が可能である。また、作成した髪束を複製することにより、髪全体を素早く生成する。本研究では、本システムで作成した結果画像や被験者実験によって、紙に絵を描くような直感的な操作で髪型造形を行う UI の有用性を示した。

キーワード：ヘアモデリング, スケッチ, ユーザインターフェース

### Abstract

We propose an intuitive and quick hair modeling system that uses a 3D canvas of any shape, including a head enclosure, flat surface, and hand-drawn outlines of hair bunches. The existing method of generating a bundle of hair using a single line as an axis is intuitive. Still, it isn't easy to reflect the user's intention in detail because it only adjusts the shape and automatically determines the thickness of the line as an axis. This method proposes a system that allows users to specify detailed figures more freely and intuitively by modeling a hair bundle from two contour lines, as in hand-drawn illustrations. The 3D canvas used in this process can be flexibly specified by isosurfaces covering the head, arbitrary planes and 3D models, and a function for creating a canvas by sweeping and interpolating input lines. In addition, the entire hair is quickly generated by duplicating the created hair bundle. The resulting images created by this system and subject experiments demonstrate the usefulness of the UI for hair modeling with intuitive operations similar to drawing on paper.

**Keywords:** Hair modeling, sketch, user interface

## 1. はじめに

本稿ではお絵描きの感覚で 3D のヘアモデリングを行うシステムを提案する(図 1)。3DCGにおいて、人物の髪型のモデリングは非常に煩雑なことが知られている。例えば一般的な 3D ヘア

モデル作成では、数百本の 3D 曲線を手作業で入力し変形する方法[1]や、髪束を模した筒状のオブジェクトをドラッグによる頂点毎の座標操作やスカルプトなどの手法で変形し配置する作業を繰り返す方法[2]などが挙げられる。しかしいずれも、クリックやドラッグを十数回行う動作を、数十本におよぶ曲線や筒

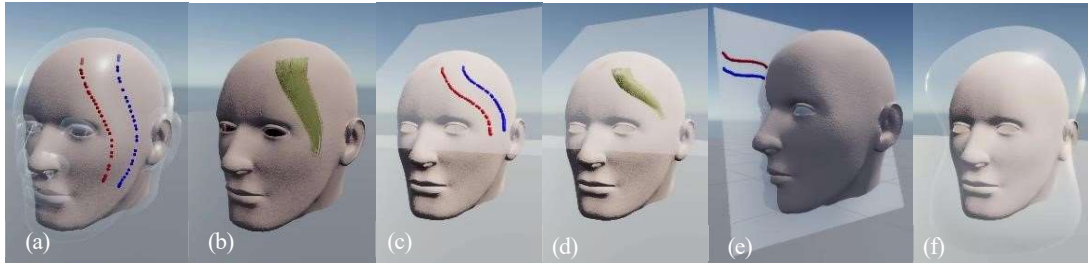


図 1 手描き入力で2本の線を入力する(a)と、その線を輪郭としたヘアカードが生成される(b)。頭皮に接した平面キャンバス(cd)やカメラの視線に垂直な平面キャンバス(e)、任意の形状のキャンバス(f)の使用で多彩な髪型を生成することができる。

状オブジェクトに対して行う必要がある。

こうした既存の造形プロセスの一つに、ヘアカードと呼ばれるオブジェクトを使用した手法がある。ヘアカードとは、短冊状の3D オブジェクトに髪の毛束を模したテクスチャ画像をマッピングする(図2)ことで、髪の毛束を表現したものである[3]。ヘアカードを多数配置することで、実際に一本一本をカーブやポリゴンで表現するのに比べ、より少ないポリゴン数で髪型を表現することができる。この手法は髪を一本一本表現する方法に比べてデータ量も少なく済むため、主にゲームの3DCGにおいてよく用いられている。本手法ではこのヘアカードを用いたヘアモデリングを支援するシステムを提案する。

本手法の主な新規性は、ヘアカードを用いた髪型のモデリングにおいて、ヘアカードの左右の両端をスケッチで入力することにより、大量の入力の手間を回避しつつ、より柔軟な形状指定を実現する点である。

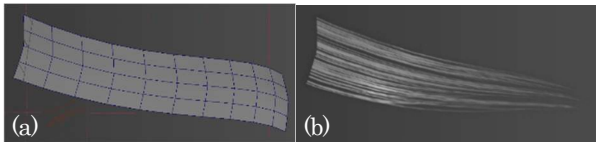


図2 ヘアカード

## 2. 先行事例

ヘアモデリングはその複雑さから基本的に難しい作業であり、これまでに多くの研究がおこなわれている[4]。また複雑かつ大量の髪の毛を自由に編集するため、このような研究では直感的かつ簡便なUIも同時に模索されている。3D 曲線を一つ一つ手作業で入力し、これをもとに髪型を造形する方法[1]は一般的な手法の一つである。また、頭皮上に複数の3D 曲線を入力し、この曲線の一つ一つをガイドとしたヘアカードを頭皮上に配置する手法はより簡便かつ一般的な方法である[5]。現実の人間の

頭髮に近い、精密な表現が可能な一方、ガイド入力の本数が数十本から数百本に及び、ユーザがこれを手作業で描いていくのは煩雑である。上記のような3D 曲線の入力をより便利に行う手法として、一つの入力線に無数の髪を付随させ、束を生成するもの[6]や、頭部のUVテクスチャ上で毛の密度や長さを指定するもの[7]がある。しかし、このような自動生成手法では髪の毛の一つ一つの束の外形を直感的に指定することはできない。

主にトゥーン調や、3D プリントを想定した人物造形などを対象として、3D スカルプトと筒状のオブジェクトの配置によって髪をモデリングするZbrush [2]などの手法がある。この手法では、髪束の一本一本でそれぞれ細かい形状の調整や、テクスチャマッピングなどの操作が煩雑である。

VRoid Studio [8]ではスケッチ入力に沿って頭部を覆うメッシュ上にヘアカードを配置する。この手法は頭部を覆うキャンバスやヘアカードを使用する点で本提案手法に類似している。主な違いとして、VRoidでは髪束の配置後にデフォルトの形状をパラメータや別画面上の曲線によって調整する必要があるが、外形を直接描画するものではない。

HairBrush [9]は、基本的にVR空間上で入力した曲線をガイドとし、3D 髪型データベースを考慮して髪の毛束を生成することで直感的にヘアモデリングを行う手法である。また、生成した一つの髪束の周辺の髪型を逐次提示する。造形にかかる時間は既存手法に比べて早い。予め用意された数百種類の髪型データベースの髪パーツを組み合わせるため、データにないパーツを使った造形はできない。

DeepSketchHair [10]は畳み込みニューラルネットワークを用いた髪型生成であり、髪型全体の輪郭線と髪の毛の流れを示した線の2D入力を、ニューラルネットワークによって3Dでの髪の毛の流れ情報に変換することで髪型造形を行う。髪の毛の一本一本を正確に表現したモデルである一方、リアルタイムでの表現では大きな処理負荷がかかり、任意の領域内の髪束として動かすこ

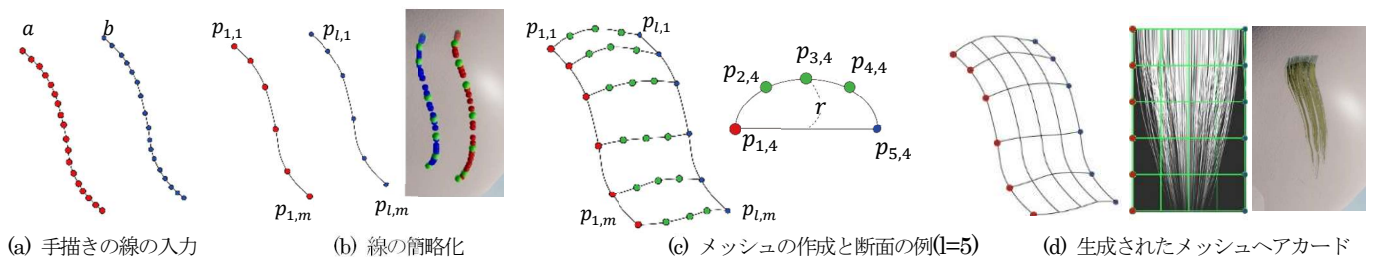


図3 ヘアカード作成の概要

とができない他、一度の入力で全体を作るため細かい髪型の調整ができない。

OverCoat [11]は 3D モデルに手描きで描画や着色を行う手法である。3D モデルの表面から任意の距離に面を作り、それをキャンバスとする UI により、層状の 3D ペインティングを実現する。本手法では、ヘアカードの浮きや髪のシルエットを柔軟に指定できるように、この UI を取り入れる。

### 3. 本手法

本手法では、頭部モデルと3Dキャンバスに対して、以下の手順の繰り返しにより髪をモデリングする。

1. 3Dキャンバスの形状を決める
  2. 3Dキャンバスに対し、手描き入力で2本の線を与えることで、ヘアカードのメッシュモデルを作成する
  3. オプションとして生成したヘアカードを複製する。
- 以下の節では、上記の三つの機能について述べる。

#### 3.1 手描き入力によるヘアカードの生成

初期状態の 3D キャンバスは、頭部モデルと同一である。また提案する他の 3D キャンバスも含め、いずれも同様にヘアカード生成に用いることができる。本手法ではヘアカードの生成は、以下の手順で行う(図3)。

1. 3D キャンバス上に髪束を表す2本の線を描画する。
2. 入力線を簡略化する。
3. 定義した位置に頂点を配置、そしてヘアカードのメッシュモデルを生成する。

##### 3.1.1 3D キャンバスへの描画

ユーザはモデルに対する描画により、3D 空間にヘアカードの左右の輪郭線を配置する。まず、マウスドラッグによりスクリーン上に点を配置する。点の配置の間隔は、本システム上では  $1.0 \times 10^{(-5)}$ メートルとした。次に、この空間のカメラの位置より配置された点にレイを飛ばし、このレイが頭部、または 3D キャンバスと交差した位置に点を配置する (図4a)。点は描画順に配置され(図4b)、描画した最初の一点は頭部モデルの表面、残りは 3D キャンバスの表面に配置する。これによって、実際の人間の髪のように毛根が頭皮に接し、他が離れているようにする。

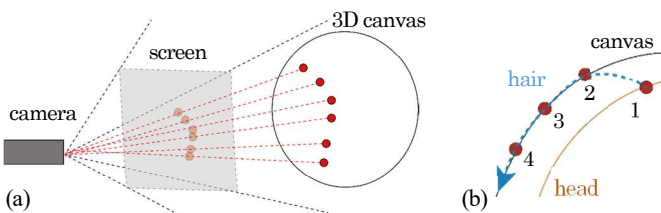


図 4 3D キャンバスへの描画 (左) と点の配置 (右)

##### 3.1.2 入力線の簡略化

2本の線a、bに沿って、髪の毛を形成する頂点を配置する(図

3b)。これは、入力された2本の線の頂点数を同じにする他、任意の頂点数に増減することで、滑らかな見た目や処理負荷の軽減を実現するためである。頂点簡略化の方法は、Douglas-Peucker polyline simplification [12] を採用した。この方法ではポリラインの形状的特徴を保ちつつ頂点数を減らし、簡略化することができる。まず、ポリラインを構成する全ての点のうち、始点と終点を結ぶ線分と他の全ての点との距離を計算し、最も距離の長かった点を検出する。次に、元の始点と終点の間に検出された点を加えたポリラインを定義し、この各線分に対し、検出されなかった点の中から最も距離が遠いものを検出し、これをポリラインの新たな頂点として間に追加する。同様に残りの頂点を処理し、指定した点数になったら処理を終了する。図3bの右図は簡略化の実際の例で、青と赤の点が元の描画線、緑の点が簡略化後の点を示す。

##### 3.1.3 ヘアカードのメッシュモデルの生成

一般的なヘアカードは、カードの断面を曲線形にする、あるいは、法線マッピングにより実際は平面でありながら曲面であるかのような質感を表現することで髪の厚みを表現する。本手法では前者を採用したヘアカードを作成する。

2本の描画線a、bを簡略化して作成した同じ頂点数の2つのポリライン $p_{1,j}$ 、 $p_{l,j}$ を左右のエッジとし、水平方向を $l$ 点で分割した短冊状メッシュ $p_{i,j}$ を生成する(図3c)。このとき  $i=1,2,\dots,l$ ,  $j=1,2,\dots,m$ ,  $m$  はポリラインの頂点数で短冊状メッシュの鉛直方向の頂点数、分割点数  $l = 2s + 1$ ,  $s$  は分割数を定めるパラメータである。各 $p_{1,j}$ と $p_{l,j}$ の中間の頂点 $p_{i,j}$ を以下のようにすることで、面の中間部分を法線方向に膨らませる。

$$p_{i,j} = \frac{\{(2s-1)-s\}p_{1,j} - sp_{l,j}}{2s+1} + i(2s-i)rd_j,$$

$$d_j = (p_{l,j} - p_{1,j}) \times (p_{1,j+1} - p_{1,j})$$

ここで、 $d_j$ は $p_{1,j}$ 、 $p_{1,j+1}$ 、 $p_{l,j}$ の3点からなる面の法線、 $r$ は短冊の膨らみ具合を指し、 $r = |p_{1,1} - p_{l,1}|/200$ とする。これにより、短冊状オブジェクトの $p_{1,j} - p_{l,j}$ 方向の断面(図3c右)の形状を一般的なヘアカード同様に曲線形にし、髪束の厚みを表現する。本研究では経験則的に $s = 2$ とする。

##### 3.1.4 ヘアテクスチャのマッピング

前項3.1.3で配置した頂点 $p_{i,j}$ を元に、ポリゴンメッシュを生成し、テクスチャマッピングを行うことでヘアカードを生成する。テクスチャは予め用意された2D画像(図5左)を使用する。一つのテクスチャ内に、密度の異なる複数の髪束を含み、UV座標の指定によって、使用する領域を変更し、密度の指定を行うことができる(図5右)。テクスチャマッピングに必要なメッシュを構成する頂点 $p_{i,j}$ に対応するテクスチャのUV座標 $(u_{i,j}, v_{i,j})$ は以下のように計算する。

$$u_{i,j} = u_{min} + \frac{i}{l}(u_{max} - u_{min}),$$

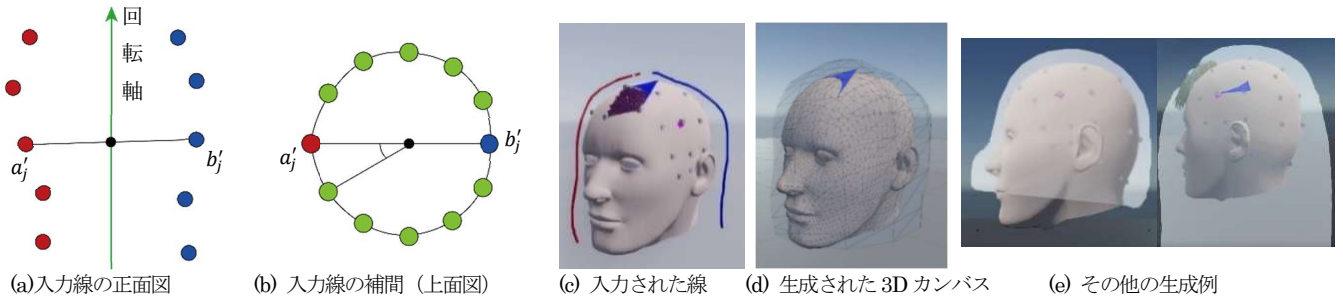


図6 描画による3Dキャンパスの生成

$$v_{i,j} = v_{min} + \frac{j}{m}(v_{max} - v_{min}).$$

このとき $u_{min}$ ,  $u_{max}$ ,  $v_{min}$ ,  $v_{max}$ はそれぞれ UV 座標の水平方向および鉛直方向の最小値、最大値で、ユーザが指定することで一つのテクスチャに入った複数の髪束を指定できる(図5左)。この座標を用い、短冊状メッシュにテクスチャ画像をマッピングすることでヘアカードを生成する(図3(d))。

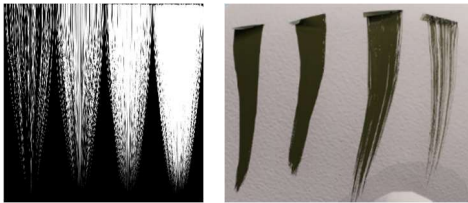


図5 密度の異なるヘアテクスチャとマッピングの結果

### 3.2 任意形状の3Dキャンパスの作成と読み込み

本手法における3Dキャンパスは、頭皮に沿った形状や、平面型、ユーザのスケッチによる半自動生成など、様々な形状を切り替えて使用することで、様々な髪型を作成できる。

#### 3.2.1 平面型の3Dキャンパス

逆立った形状の髪束、例えば、スパイキーヘアやオールバックなどの表現を目的とし、平面型の3Dキャンパスを提案する。平面は二種類定義できる。一つは、頭部モデル上でクリックした点の接面を用いる接面キャンパス(図1c)、一つは頭部モデルの中心にスクリーンと並行な平面を配置した視点キャンパス(図1d)である。また、この視点キャンパスは、後述する3Dキャンパスの半自動生成でも用いる。

#### 3.2.2 スイープ補間による3Dキャンパスの半自動生成

本機能によって、ユーザは2本の入力線によって、手軽に3Dキャンパスを作成する。ユーザはスクリーンと平行な平面キャンパス上で、頭皮を囲うように2本の線を描画し、3.1.2項と同様に簡略化を行い頂点数を揃えたポリラインを $a'$ ,  $b'$ とする。このポリラインを頭部の中心を通る鉛直軸回転によるスイープを行うと同時に、対応する $a'$ ,  $b'$ の頂点に対し回転軸からの距離を角度に応じて補間することで、2本の線の形状を徐々に補間した形状の3Dキャンパスを生成する(図6)。本稿で示す結果では、回転によるスイープで補間する角度は $30^\circ$ とした。

### 3.3 ヘアカードの複製

本手法では、ユーザが手描き入力で生成したヘアカードを周囲の領域に複製する。また、複製の際、曲面に応じて法線方向に沿うよう回転する。

#### 3.3.1 複製する領域

ユーザがヘアカードを生成した時、そのヘアカードの含まれる領域全体にヘアカードを複製する。ヘアカードを複製する領域は予め分けられている(図7)。この領域分けは、実在の人間の髪の毛の生える範囲に基づいた描画における領域ごとの髪の毛の生え方の描き分けのしやすさなどの経験則に基づく[13]。例えば人物の髪を絵画で描く時、前髪、もみあげ、襟足など、領域ごとに分けると描きやすいとされている。

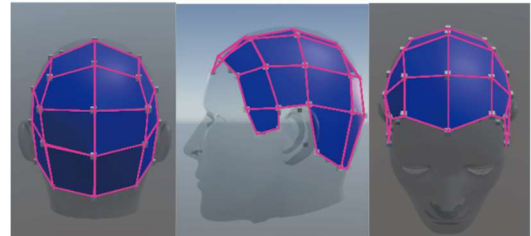


図7 矩形による領域分け

実装当初は、予め手描き入力で生成した3枚のヘアカードで囲まれた領域内にヘアカードを補完する実装をしていたが、逐一3箇所ヘアカードを描かないといけない煩雑さや、ヘアカードの生えていない箇所が見落とされる可能性があることから、一枚のヘアカードを近傍領域に複製する方式に変更した。今後、生成したヘアカードの補間などが考えられる。

#### 3.3.2 ヘアカードの複製

ユーザは、領域内でのヘアカード複製数、および浮き具合を指定できる。領域内のヘアカード複製数を増やせば、領域内に多くの髪束を作成し、密な髪表現となる。本手法では、一領域の水平方向および鉛直方向におけるヘアカードの配置数 $m_L$ ,  $m_H$ はそれぞれ以下のように定義する(図8左)。

$$m_L = \frac{Ld_L}{W}, \quad m_H = \frac{Ld_H}{r}$$

このとき  $L$  は矩形領域の水平方向の幅、 $H$  は鉛直方向の幅、 $W$  はヘアカードの幅、 $d_L$  および  $d_H$  は正の調整パラメータで、本研究ではヘアカード同士の不自然な重なりや処理負荷の軽減の観点から経験的に  $d_L = 4.8$ 、 $d_H = 0.03$  とした。一般的なヘアカードによるモデリングでは多少のヘアカードの重なりは見た目により影響がなければ許容される。本研究でも同様に、後述するインタラクティブな GUI により、ユーザがパラメータ調整によって編集する。

浮き具合は任意の正の数であり、これを高めると、ヘアカードは毛先になるほど頭部法線方向に引っ張られ、逆立つ(図8右)。毛先を浮き上がらせることで、ヘアカード同士の重なりをある程度軽減することができる。頂点  $p_{i,j}$  の浮き具合を示す 3 次元ベクトル  $D_i$  は以下のように求める。

$$D_i = \left\{ \frac{d}{r} * \frac{i}{n} * \left( 1 - \frac{T - p_{i,j,y}}{T - B} \right) \right\} n,$$

このとき  $n$  は  $p_{i,1}$  のヘアカードが接する頭部の法線の単位ベクトル、 $T$  および  $B$  は頭部の最高点と最低点の  $y$  座標、 $p_{i,j,y}$  は  $p_{i,j}$  の  $y$  座標、 $r$  はヘアカードの浮き具合(3.1.3)、 $d$  はユーザが指定可能な浮き具合の調整係数で、初期値は  $2.5 \times 10^{-5}$  としている。

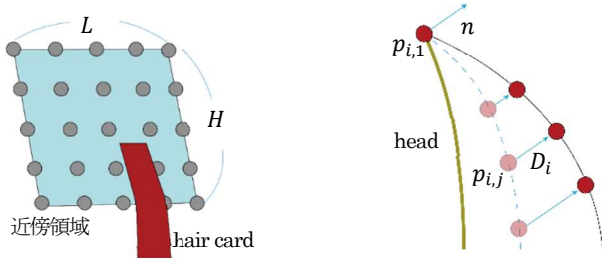


図8 ヘアカードの配置(左)と浮き具合(右)

### 3.3.3 水平方向のヘアカードの回転

前項で変形したヘアカードを、さらに生成場所の頭の法線に合わせて  $y$  軸回転を行う。中心軸はヘアカードの根元を通る鉛直な軸とし、複製元のヘアカードの根元位置の頭の法線と複製先の頭の法線のなす角度だけ回転する。

## 3.4 本システムの UI

本システムの UI は、Unity Engine によって実装されている。ユーザは、上記に示すような操作に加え、生成したヘアカードの削除やカメラアングルの調整などの操作を、キーボードやマウスを用いて行う。本システムの画面(図9)の左上の制御パネルでは「髪の密度」「髪の浮き具合」「髪束の数」の三つのパラメータを指定できる。「髪の密度」は、3.1.4 項の  $u_{min}$ 、 $u_{max}$  の値を変えることでヘアカードに適用するテクスチャの UV 座標を変更し、パネルの下のプレビューにて確認できる。「髪の浮き具合」は、3.3.2 項の  $d$ 、「髪束の数」は、3.3.2 項の  $d_L$ 、 $d_H$  を指定できる。左下のパネルで、現在ヘアカードを作る状態か、キャンバスを作る状態か、適用している 3D キャンバスの種類が平面キャンバスか、任意の立体キャンバスなのかを表示、画面右下には、

ストローク数や生成したヘアカードの総数をテキスト表示する。

ヘアカードの複製を行った結果が想定通りの見た目にならなかった場合、複製されたヘアカード群を削除し、新たに制御パネルのパラメータを指定しなおし、再度複製の操作を行う。複製したヘアカード群に対してパラメータを与え直すことで、インタラクティブにヘアカード群の見た目を変更する。



図9 本システムを用いた作業画面

## 4. 結果

本手法を用いて複数の髪型を作成した(図10)。また、髪型をモデリングする被験者実験を行った。システム実行時の環境は、CPU は AMD Ryzen 9 3900X、GPU は GeForce RTX 2080 Ti、メモリは 64GB、OS は Windows 10、開発環境に Unity 2021.3.2f1 を採用した。本稿の結果画像は、Arnold Renderer によってレンダリングし、髪型を構成するポリゴン数の合計は、およそ 20 万個であった。

### 4.1 生成結果と考察

本システムを用いた目標の髪型の生成では、概ね近いものを 20 分程度で作成することができた。実際に使用した目標画像は著作権のため論文への掲載を控え、同様の画像を描画したものを図 10 に示す。本システムでは一度作ったヘアカード形状の調整機能は実装しなかったため、一度の描画で形状を決める必要があった。微調整の機能は追加で実装することは可能である。また、微調整しなくてもある程度思い通りのモデルが得られた。このほか、毛の密度に関しても調整することができれば、より目標に近い画像生成を行うことができると考えられる。

自由な髪型のモデリングでは、平面キャンバスを用いることで、より立体的に跳ねた髪型を表現することができ、柔軟で独創的な髪型の生成が可能であった。しかしより独創的な髪型を実現しようとする場合にはヘアカードは平面的であるため、極端な突起形状などには適さない。また、頭部キャンバスの描画部分がユーザに対して垂直な場合は比較的直感的なモデリングとなるが、そうでない場合に直感的な結果が得られない。例えば、お絵描きで髪を描く場合、頭部を顔正面から見て頭部の側面の髪束をスケッチした際にキャンバス上に描画がされない、キャンバス上へ投影された描画が歪んでしまうなどである。

表1 実

被験者
被験者
被験者

## 4.2 被験者実験

本システムの検証のため、本システムを用いたモデリングを2名の被験者に行ってもらい、感想などのフィードバックを得た。本実験は科学的な検証としては十分な被験者数に満たないため、参考程度のもので記載する(4.2.1)。

また、本システムを使って作成した髪型モデルに関し、主にその完成度などについてアンケートを行った(4.2.2)。

### 4.2.1 実験方法

被験者に、本システムの使い方や、主な機能を15分程度で説明し、各種操作法のマニュアルを提供した上で各操作を行ってもらった後、提示した2つの画像を目標とした髪型の生成(図11)、および目標画像なしで自由な髪型の生成をそれぞれ25分で行う実験を実施した。被験者数は2名で、20代、男女一名ずつ、3Dモデリングの経験は半年および3か月、絵を描く経験年数は5年と1年であった。実際に提示した目標画像は著作権への配慮のため論文に掲載するのを控え、同様の画像を作成しなおしたものを図11に示す。

また、これらの作成に関する作業時にかかった時間、手描き入力回数(入力を取り消した分を含む)、生成されたヘアカードの総数、生成された髪型モデルを記録した結果を表1に示す。表1の結果は被験者実験(4.2)のものであるが全体の参考としてここに記載する。

実験のあとに7項目のアンケートを行った。数値による評価は5段階で、またその理由を記述する形式とした。その結果を図11に示す。質問項目は著者間で議論の上、選定した。

質問1「髪をモデリングする際、絵を描くような直感性がありましたか? (1:無かった~5:あった)」での評価値は両被験者ともに4であり、理由については、「実際にマウスで二本の線を描くことで髪束を作り上げることができた」、「ある程度正確な形状の髪束が生成されていた」などの回答が得られた。

質問2「全体として、造形作業は大変でしたか?(1:楽だった~5:大変だった)」では評価値は1,2であり、理由として「ある程

度の時間はかかるが、他のツールと比べ楽」という共通の回答が得られた。

質問3「提示された髪型の作成結果に対する満足度(1:不満~5:満足)」では評価値は3,2であり、理由として「ある程度の造形はできたが、イラスト特有のふんわりとした髪表現があまりできなかった」「逆立った毛束を作るのがあまり上手いかなかった。髪を毛を立てさせるようなパレットを作成しようと試みたが、パレットの作成が上手いかなかった。」などとあった。

質問4「自由に作った髪型の作成結果に対する満足度(1:不満~5:満足)」では評価値は3,5とやや別れた。評価3の被験者のコメントとして「ハーフツインテールの髪型にしようと考えていたが、時間が足らず、セミロングの女性になってしまった」「最初よりも操作に慣れ、綺麗な髪束を作ることができた」、評価5の被験者のコメントとしては「マウスで入力する線がカーブしても、その曲線に沿った形状の髪束が生成され、イメージ通りの髪型を作ることが出来た。」などとあった。以下は記述回答のみの質問5-7である。

質問5「動作が行き詰ったところや、難しいと感じた部分はありますか?」に対して「慣れれば難しさは感じなかった」「毛束を複製した時に、頭の中の方向に生える・頭の中に髪束が埋もれてしまうことがあった」などとあった。

質問6「現在のシステムに加えてどのような機能があればよいと思いますか?」に対しては、undoや、作ったキャンバスの拡大縮小、作成した毛束の編集などの回答が得られた。

質問7「その他、感想や気になったことなど」に対して、「普通にモデリングするよりも楽しく髪束をモデリングできた」「直感的に描いた曲線のそのままの形で毛束を作成できて、CGソフトで制作するよりもイメージ通りの髪型を作るのが簡単」などの回答が得られた。

被験者実験とアンケート結果に対する考察として、プロの3DCGアーティストが人物の髪型を造形すると、Xgenなどのヘアカードによる写実的なモデリングでは数十時間を要する[5]。また、Zbrush [2]のような筒状オブジェクトによるデフォ



図10 生成結果。特定の髪型を参考にして作成された髪型(上段)、参考を見ずに自由に作成された髪型(下段)

ルメされたトゥーン調のヘアモデリングでも、数時間を要する。これらに対し、本手法での写実的なモデリングではかかった時間は 20 分前後と非常に短くなっている。さらに、既存手法では数百本の 3D 曲線の一つ一つマウスドラッグなどで変形・配置するが、本手法ではおよそ 70 回の手描き入力で済む。このように既存手法の手間の多さや制作時間の長さという課題を本手法では大幅に改善することができた。

一方、本手法は頂点が 3D カンバスの形状に沿って配置されるため、やや柔軟性に欠け、実際のお絵描きの場合と異なる形状が作成されるとも感じられた。対策として、生成されたヘアカードの形状の編集機能の追加が考えられる。また、配置の自動化は便利な反面、手作業のモデリングほどの詳細な指定ができない。この他、本システムのバグとして描画した入力線の 8 ~9 割程度にヘアが短くなってしまいが、本実験では被験者からその点に関する指摘はなかった。

また、参考と比較のため VRoid Studio を用いて同様の目標の髪型を作成する実験を行い、インタビューを行った。VRoid Studio は商用ソフトである一方、本システムは研究用途に絞った最低限の実装であるため総合的ユーザビリティの点で VRoid に劣る。しかし、髪束の両端を直接描画できる機能が VRoid にあれば現在の VRoid の UI より便利そう、目標画像 1 のような頭部から離れた髪束は提案する平面キャンバスが向いている、といった意見が得られた。



目標画像 1/2 (上/下)	被験者 1 の造形	被験者 2 の造形
		
		

自由に作成した髪型 被験者アンケート結果

図 11 被験者実験

#### 4.2.2 様々な生成結果に関するアンケート

次に、15 人を対象に、生成された髪型をレンダリングした 12 の画像について、アンケートを行った。アンケートは、資料を

見て作られたものはその再現度を、自由に作られたものについてはその完成度を、1(低い)~5(高い)の 5 段階で評価し(表 2)、その評価値の理由を記述する形式とした。アンケートの結果、髪型の完成度に関して 3 を上回る評価が多く、完成度の高いモデリングを行えることを示している。評価値が 3 未満のものに対しては、髪の長さや密度が足りないという指摘が多かった。これに関して、描画した線と髪のテクスチャの長さが合っていなかったということと、デフォルトの毛先の密度の設定が低すぎたことが課題として挙げられる。

表 2 様々な結果画像に関する評価(15 名)

図	中央値	平均値	最頻値	図	中央値	平均値	最頻値
8a	4	4.40	4,5	8g	3	3.00	3
8b	4	3.80	4	11a	2	2.40	2,3
8c	3	3.33	4	11b	2	1.80	1
8d	5	4.87	5	11c	2	2.47	2
8f	4	3.80	4	11d	1	1.67	1

#### 4.3 将来課題

本研究では任意の形状の 3D カンバスへの手描き入力を行う UI を提案し、様々なシルエットの髪型に対応することができた。しかし、現状ではいくつか制限がある。

まず、ヘアカードの生成の際、ヘアカード同士の突き抜けが起こる(図 12)。既存の 3DCG でも髪同士の突き抜けは見た目には大きな影響がない限りは許容されているが、本手法では見た目が不自然になる場合がある。これを解消するために他のヘアカードを移動させると、既に配置されたヘアカード部分の見栄えを損なう可能性がある。一方、周囲のヘアカードに合わせて手描き入力の線を変形させた場合、ユーザの狙った通りの入力ができなくなる。この解消のためには、ヘアカードの配置をユーザの意図をできるだけ反映する制約を課した上で最適化するような方法が考えられる。ちなみに本手法によってヘアカードと頭部モデルの突き抜けも起こりえるが大きくは発生せず、またこれは実際の制作でも許容範囲とすることが多い。

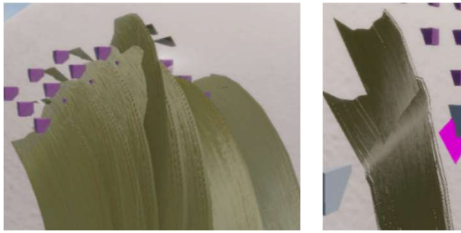
本手法ではヘアカードのテクスチャは一つの髪型モデリングにおいて同じものを使用しているが、ヘアカードのサイズによって伸び縮みが起こるため、サイズの異なる場合の対応が必要である。

また、本手法は巻き髪や結んだ髪、編んだ髪の造形には対応していない。巻き髪の場合、らせん状の輪郭線の入力、結んだ髪の場合、円筒状にヘアカードの成形、三つ編みの場合、3 つのヘアカードに編み構造を与える必要がある。このような特殊なヘアカードの成形が場合には個別に対応する必要がある。

髪型の完成度を高めるためには、近隣のヘアカードの形状に大きな差異が出て不自然にならないようにする必要がある。そのため、ヘアカードを複製する際、周囲にある既存のヘアカードの形状とブレンドする、生成したヘアカードの形状の編集を可能にする、などの対応が考えられる。

他に、ヘアモデリングと同時にリギングを行うことができれば、造形作業の合間に物理シミュレーションを行って見栄えを

確かめることができる。



(a) 左右の重なり (b) 上下の重なり  
図 12 ヘアカード同士が重なっている例

## 5. 総括

本手法は、ヘアカードの両端の 2 本の輪郭線を描画によって入力することにより、髪型を生成するシステムを提案した。また、近傍領域へのヘアカードの複製機能やキャンパスの生成機能を備えたヘアモデリングシステムを構築し、ユーザスタディによって入力の手間の軽減効果や一定のユーザビリティを備えていることを示した。今後の研究では、既存手法との比較実験、ヘアカード同士の突き抜けの解消や、自動的にリギングを行い、アニメーションをシミュレーションするなどによって、さらなるユーザビリティの向上へとつなげていきたい。

## 参考文献

- [1] Maya Tutorial : creating curves for hair using maya curve modelling techniques, <https://vimeo.com/33487648>, last viewed: 2022.1.19
- [2] Sculpt & Model Your Own Custom HairBrush & Hairstyle in Zbrush, <https://flippednormals.com/product/sculpt-model-your-own-custom-hair-brush-hairstyle-in-zbrush-2485>
- [3] Creating Hair Cards for Realtime Characters, last viewed: 2022.1.19. <https://cgcookie.com/courses/creating-hair-cards-for-realtime-characters>
- [4] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, Marie-Paule Cani, Ming C. Lin: A Survey on Hair Modeling: Styling, Simulation, and Rendering. IEEE Trans. Vis. Comput. Graph. 13(2): 213-234 (2007)
- [5] XGenの学習パス, last viewed: 2022.1.19. <https://knowledge.autodesk.com/ja/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2020/JPN/Maya-CharEffEnvBuild/files/GUID-C0470142-600B-4615-8110-EC779934DF5F-htm.html>
- [6] Shahzad Malik: A Sketching Interface for Modeling and Editing Hairstyles. SBM 2005: 185-194
- [7] Benjamín Hernández, Isaac Rudomín: Hair Paint. Computer Graphics International 2004: 578-581
- [8] VRoid, last viewed: 2023.11.16. <https://vroid.com/en>
- [9] Jun Xing, Koki Nagano, Weikai Chen, Haotian Xu, Li-yi Wei, Yajie Zhao, Jingwan Lu, Byungmoon Kim, and Hao Li, 2019. HairBrush for

Immersive Data-Driven Hair Modeling. UIST. ACM, 263–279.

<https://doi.org/10.1145/3332165.3347876>

[10] Yuefan Shen, Changgeng Zhang, Hongbo Fu, Kun Zhou, and Youyi Zheng, 2021. DeepSketchHair: Deep Sketch-Based 3D Hair Modeling. IEEE Transactions on Visualization and Computer Graphics 27, 7, 3250–3263. <https://doi.org/10.1109/TVCG.2020.2968433>

[11] Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner. 2011. OverCoat: an implicit canvas for 3D painting. ACM Trans. Graph. 30, 4, Article 28.

<https://doi.org/10.1145/2010324.1964923>

[12] Wu, Shin-Ting, Silva, Adler C. G. da, & Márquez, Mercedes R. G. (2004). The Douglas-peucker algorithm: sufficiency conditions for non-self-intersections. Journal of the Brazilian Computer Society, 9(3), 67-84.

[13] 分け目・つむじの位置で悩まない！髪のための描き方, last viewed: 2022.1.19.

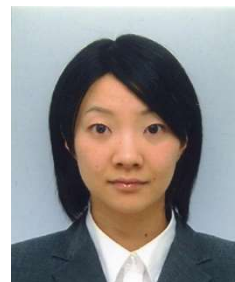
<https://www.clipstudio.net/oe kaki/archives/152318>

### 高野 正統



2021年 九州大学芸術工学部画像設計学科卒業。2022年 九州大学芸術工学府芸術工学専攻修士課程修了。

### 森本 有紀



2008年九州大学芸術工学府修了, 同年東京大学にて日本学術振興会特別研究員, 2009年より独立行政法人理化学研究所研究員, 2012年より東京電機大学未来科学科講師, 芝浦工業大学助教などを経て, 2016年より九州大学助教。コンピュータグラフィクスに関する研究に従事。情報処理学会, ACM SIGGRAPH 各会員。博士(芸術工学)。