

Cartoon hair : blobby model based approach

酒井, 健行 / SAKAI, Takeyuki

(発行年 / Year)

2014-03-24

(学位授与年月日 / Date of Granted)

2014-03-24

(学位名 / Degree Name)

修士(理学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

2013 年度 修士論文

Cartoon Hair: Blobby Model Based Approach

酒井 健行

Takeyuki Sakai

法政大学大学院情報科学研究科情報科学専攻

E-mail: 12t0006@cis.k.hosei.ac.jp

2014 年 2 月 17 日

指導教員

Vladimir Savchenko 教授

概要

本論文では、3DCG キャラクターの髪の毛の表現手法の提案を行う。本手法では、キャラクターの髪の毛をスケルトンベースの陰関数曲面を用いて表現する。陰関数曲面の一種である Blobby モデルを利用することで、従来のポリゴンモデルでは表現することが困難だった髪の毛の分離と結合を容易に表現することができる。本手法では Blobby モデルのパラメータを持った髪の毛のスケルトンを入力データとして扱う。入力されたスケルトンは、ノード間で各パラメータをスプライン関数を用いて補間され、そのスケルトンから髪の毛の形状となる陰関数曲面が生成される。その陰関数曲面をレイマーチング法を用いてレンダリングし、ノンフォトリアリスティックシェーディングを適用して画像として出力する。本研究では、提案手法のモデルに使われるスケルトンを作成するための GUI の実装も行った。本システムを使うことでデザイナーは通常の GUI 操作だけでなく、デプスカメラを用いた指のトラッキングにより、直感的な操作でスケルトンを作成することができる。また本研究では、髪の毛の物理ベースアニメーションについても提案をする。質点バネモデルを用いて、本手法の立体的な髪の毛の動きをシミュレーションする。

Abstract

This paper describes a new hair representation technique for drawing 3D CG characters. In our approach hair models are represented by skeleton-based implicit surfaces. Implicit modeling allows representing union and division of wisps of hair by using a blobby model. Our method gets hair skeletons as input data. The skeleton is interpolated using a spline function and then implicit surface of hair shape is generated by the skeleton. And ray marching method is used to compute intersection of ray and surface of hair implicate function. Finally, non-photorealistic rendering is applied to generate resulting image as output. In our approach hair skeletons are created by operating GUI skeleton editor or tracking hand motion using depth camera. This system allows the artist to create skeleton with the help of intuitive operations. Physically-based animation of the hair model is also proposed. In our approach, spring-mass model is used to animate a solid hair shape.

目次

1	序論	1
2	関連研究	3
3	セルルック 3DCG におけるキャラクターの表現	4
3.1	セルルック 3DCG の現状	4
3.2	セルシェーディングにおける課題	5
3.3	形状表現における課題	6
3.4	動きにおける課題	6
3.5	その他の課題	7
4	提案手法	8
5	スケルトンベースの Blobby モデルを用いた髪の毛の形状表現	9
5.1	陰関数曲面	9
5.2	Blobby モデル	10
5.3	髪の毛のスケルトン	10
5.4	パラメータの補間	11
5.5	髪の毛の形状関数	12
6	レイマーチング法による髪の毛のレンダリング	14
6.1	レイマーチング法	14
6.2	陰関数の評価	14
6.3	バウンディングボックスによる高速化	15
7	セルルックな髪の毛のシェーディング	17
7.1	法線の計算	17
7.2	トゥーンシェーディング	17
8	物理ベースの髪の毛アニメーション	20
8.1	バネを用いた髪の毛の物理モデル	20
8.2	バネの自然長の最適化	21
9	デプスカメラを用いたユーザインタフェース	23
9.1	デプスカメラを用いたインタフェース	23
10	結果	25
10.1	髪の毛のアニメーション	25

1 序論

現在まで日本のテレビアニメにおいて、絵はアニメーターの手によって描かれることが一般的である。しかしながら、近年ではその手描きの作画と併用して三次元コンピュータグラフィックス (以下 3DCG) による作画が増えてきている。またその 3DCG の使用も、車やロボットなど物に限らず、最近ではキャラクターに用いるケースも増えている。3DCG のキャラクターは、ダンスシーンや戦闘シーンなど、キャラクターの動きの多い場面で、カットによって手描きのキャラクターと切り替えながら使われている。この場合 3DCG のキャラクターは所謂セルルック CG と呼ばれる手描きに似せた 3DCG が用いられる。

3DCG における人間の髪の毛の表現は多くの論文で議論されてきた。3DCG による毛の表現に関する初期の研究には、3 次元テクスチャを利用して毛の描画を行った Kajiva らの研究 [5] が挙げられる。Ward らの調査研究 [17] では髪の毛に関する様々な手法についての比較がなされ、また、Volino らの研究 [15] においても髪の毛の表現についての各種研究についてまとめられている。

セルルックな 3DCG における髪の毛は、日本の手描きアニメーションに見られる表現をベースにしている。手描きアニメーションでの髪の毛の表現はフォトリアルな髪の毛の表現とは大きく異なる。フォトリアルな表現では細い線の集まりで描画されるが、セルルックな表現では髪の毛のまとまりを表現した形状を持ったモデルで表現される。またセルルックな表現では物理的に正しい動きをしない、カメラの方向によって形が変わり 3D 空間上での整合性が保たれていないなど、の特徴もある。

本論文では、日本のアニメにおける 3DCG キャラクターの髪の毛のモデリングに焦点を当てる。これまで、アニメにおける 3DCG キャラクターの髪の毛はポリゴンモデル、もしくはパラメトリック曲面により作成されてきた。しかしながら、ポリゴンモデルは大きな変形を表現することが難しく、動きに制限がかかる。これらのモデルでは、図 1 に示すような手描きのアニメーションで見られる髪の毛の房がわかる、または合わさるといった表現が困難である。そのため、手描きアニメの髪の毛に近づけるためにはより柔軟な髪の毛の表現モデルが必要となる。



図 1. 手描きアニメーションに見られる髪の毛の変形の例.

2 関連研究

ノンフォトリアリスティックなキャラクターの髪の毛に関する研究は、フォトリアルな髪の毛の研究に比べると数が少ない。また、ノンフォトリアリスティックな髪の毛を対象にした研究においても、形状表現について扱ったものは少なく、シェーディングや形状を作成する負担を軽減する手法について扱ったものが多い。

Noble らの研究 [9] では髪の毛の房を表現する NURBS 曲面を、キーヘアークーブから生成している。この手法では、まずデザイナーにより作成された髪の毛全体を表す一つの NURBS 曲面から、キーヘアークーブを生成する。その後キーヘアークーブに対し物理ベースのアニメーションをつけた後、1 本のキーヘアークーブにつき 1 本立体的な髪の毛の房を表す NURBS 曲面を生成している。

Mao らの研究 [8, 7] では、簡単なスケッチから髪の毛のポリゴンを生成している。この手法では、髪の毛を生やす領域と数本のスケッチによる簡単な髪の毛の概形を入力するだけで、頭全体に髪の毛のスケルトンを生成し、その後スケルトンから髪の毛のポリゴンに変換している。

Shin らの研究 [13] では、パーティクルとビルボードを用いた髪の毛のレンダリングを提案している。この手法では、髪の毛の形状を定義するパーティクルをデザイナーが作成し、その入力されたパーティクルにそってビルボードを生成する。生成されたビルボードは髪の毛の形状に合うようにそれぞれリサイズする。生成された髪の毛に対し、シルエットライン、拡散反射成分、ハイライトと各種シェーディングを行い最終的な絵が出来上がる。またこの研究では、髪の毛のレンダリングに関する計算のほとんどを GPU 上で行っている。

Cassol らの研究 [2] では、いくつかのパラメータを指定することで様々な髪型を自動的に生成する手法が提案されている。

Sugisaki らの研究 [14] では、手描きの髪の毛に対してラフスケッチを用いてアニメーションを付ける手法が提案されている。

Côté らの研究 [3] では、手描きの髪の毛に対するノンフォトリアリスティックなライティング手法が提案されている。手描きのキャラクターに対して髪の毛の領域を指定し、擬似的なライトを使い明るい箇所と暗い箇所を塗り分けている。

Koyama らの研究 [6] では、物理演算におけるデフォルメ手法が提案されている。カメラの位置に応じて物理演算を変化させることによって、3 次元空間上で正確に定義することが難しいセルルックな髪の毛の形状を表現することを可能としている。

3 セルルック 3DCG におけるキャラクターの表現

ここ数年のテレビアニメに使われているセルルックな 3DCG の進歩はめざましく、手描きアニメとの差がなくなりつつある。セルルックレンダリング専用のレンダラ、シェーダや、コンポジットソフト向けのプラグインが市販されるなど、ソフトウェア環境も充実してきている。しかしながら、未だ多くの点において、ソフトウェアで解決できずにデザイナーのスキルに頼っているのが現状である。

3.1 セルルック 3DCG の現状

以前からロボットや車、宇宙船など所謂メカものには広く 3DCG による作画が使われてきた。これらはメカものについては、未だ多くの手描きファンがいるものの、比較的 3DCG による作画が受け入れられているように見える。メカの 3DCG では、シェーディングこそフォトリアルなものとは異なるが、厳密に手描きを模倣した表現を目指しているわけではない。むしろ手描きでは表現できなかった映像を作るために 3DCG が用いられる。手描きアニメーションにおけるメカは作画のコストの都合上細部が簡略化して描かれる。また、同様の理由からカメラの回り込みなど、絵を新しく描き直さなければいけない動きも避けられる傾向にある。メカの 3DCG はそういった動きや演出の制約を取り除き、緻密で迫力のある映像を作るために使われる。そのため、3DCG では手描きでは表現されない細部まで描き込みがされ、また、1 秒間に使われるフレーム数も多い。これはメカ 3DCG が単なる手描きの模倣ではなく 3DCG ならではの表現を求められていることを意味する。

しかし、キャラクターの 3DCG では少し事情が異なる。最近のテレビアニメでは 3DCG によるキャラクターの作画が使われることが増えてきているが、メカ CG とは異なり 3DCG が使われている作品でもあくまで手描きによる作画がメインとなっている。つまり、3DCG のキャラクターは同じ作品中で手描きのキャラクターと混ぜて使われることが多い。近年多く見られる 3DCG キャラクターの使用例は多数のキャラクター踊るライブシーンである。この場合、3DCG はライブシーンの踊っているキャラクターのみに使われ、その他の日常のシーンでは手描きによる作画が使われる。場合によっては同じカットの中でも近景のキャラクターは手描きによる作画、遠景のキャラクターは 3DCG による作画と使い分けを行うこともある。そのように使われる 3DCG キャラクターは手描きと切り替わった場合の違和感をなくすために、限りなく手描きに近い絵である方が好ましい。このことから 3DCG によるキャラクターは手描きを模倣することを明確に求められていることがわかる。

キャラクターの 3DCG では特に、単にセルルックな処理をするだけでは手描きと同じ映像を得ることはできない。そのため、プロダクションでも使用される専用のソフトウェアが増えてきたにもかかわらず、デザイナーによる試行錯誤と手作業による修正を必要としている。また、そういった手間をかけて作られた映像であっても手描きと全く区別が付か

ないという段階には至っていない。

3.2 セルシェーディングにおける課題

セルルック CG での大きな課題の一つに陰影の付け方が挙げられる。シェーディングはセルルック CG のもっとも重要な要素と言っても過言ではない。一般的にアニメの様に光の当たる明るい部分と陰影の部分とをはっきりと 2 色で塗り分ける手法はトゥーンシェーディングと呼ばれている。最も基本的なトゥーンシェーディングでは、通常のフォトリアルなライティングと同様にピクセルの輝度値を計算した後、閾値を用いて塗り分けを行う。この方法では確かに手描きのようにはっきりとした塗り分けがされるものの、その陰影の付き方自体は手描きのそれを再現できるわけではない。現在のセルシェーディングでもこの考え方が基本になってはいるものの、ただ単純に作られたモデルにライトを当て二値化するのではなく、手描きに近づけるための様々な工夫が加えられている。

そもその問題として、手描きアニメーションで付けられる陰影は現実のライティングと一致するものではない。手描きアニメーションのライティングは、これまでの手描きアニメーション文化で培われた経験とセオリーに基づいていて、現実世界のライティングを再現していない。また、キャラクターの顔にかかる陰影に至っては、心理表現など演出の一つとして特別な付けられ方をすることも多い。それを 3DCG で考えると、光の当たる角度がパーツによってバラバラである、本来陰影になる部分が明るい、といったことが起こりうる。そのため、単純にトゥーンシェーディングを適用するだけでは手描きの陰影を再現することにはならない。

この問題を解決するために一般的に取られる手法として大きく 3 つの手法が挙げられる。まずひとつはテクスチャに陰影を描き込む方法である。これは通常のライティングをせずに直接テクスチャに色として陰影を描き込むことで、実際のライトに関係なく思った通りの陰影を出すことができる。髪の毛など形状が変わりやすく陰影のコントロールが難しいオブジェクトに対してはよく使われる。しかしこの方法では、当然テクスチャに陰影は描き込まれた形状のまま変化せず、キャラクターが動いても、光源が変わっても、陰影は変化することがない。キャラクターが動かない、もしくは逆に動きが大きく変化がないことが目立たないシーンでは有効であるが、使い方によっては 3DCG の硬さが目立ってしまうことになる。

次にオブジェクトの法線を制御する方法である。オブジェクトの輝度は主に光源の情報とオブジェクト表面の法線によって計算される。そこで、シェーディングに使う法線を実際の形状の法線ではなく、法線マップによって理想の陰影が出る法線を指定しようという考え方である。ライティングの計算を行うため、キャラクターの動きや光源の動きに合わせて陰影も動き、テクスチャに描き込む方法と比べ比較的自然的な陰影が描画できる。

最後はレタッチである。レンダリングした後の画像に対して手描きによる修正を行う。一番確実に理想的な陰影を表現することができるが、フレーム単位で手作業で描き込むため修正の手間はかかる。しかしながら、必要なカット数によっては、むしろテクスチャや

法線マップを試行錯誤で作成するよりもアニメーターのノウハウが活かせる分手軽に行える。また、前述の手法により、ある程度陰影を理想の形状に近づけた後、どうしても気に入らないフレーム、パーツをレタッチにより修正する、ということは多い。

これらの解決手段はいずれもデザイナーの手によるもので、手描きに近づけるために相応のコストを払っていることになる。また、どれか一つを使えばすべてが解決するというものでもなく、カットによって、パーツによって、これらの手法を組み合わせることで映像が作られている。

3.3 形状表現における課題

もう一つの課題としてモデルの形状が挙げられる。手描きアニメーションにおけるキャラクターは、3次元空間上で再現できないことが多い。これは特に髪型や顔において顕著で、シェーディングと同様に手描きアニメーションでのセオリーに基づき、カメラの方向によってキャラクターの輪郭やパーツの配置が変わる。一例を上げると、手描きアニメーションにおけるキャラクターの口は、キャラクターを横から見た場合、頬の輪郭の内側に収まり、口を開ければ口の中が見えるように描かれることがおおい。これは明らかに3次元のモデルを作成した際に破綻する点である。その他にも、輪郭や目、鼻の位置など、通常の3次元モデルではある角度から見た場合には手描きのキャラクターと同じ形状になっていても、角度を変えると大きく違和感があるといったことが起こる。

この問題を解決するために、形状モーフィングが用いられることが多い。カメラのアンクルに合わせて形状を変化させることで、どの角度から見ても手描きアニメーションと同様の形状を再現しようというものである。モデルを複数作るわけではなく、形状を修正するだけなので比較的成本も低い。また、シェーディングについて様々な修正を加えているため、フォトリアルなCGキャラクターとは異なり、形状の変化によるシェーディングへの影響もきにならない。

しかし、モデルのほとんどはポリゴンで作られているため、形状の変化について制約を受ける。髪の毛が最も顕著に現れる例で、髪の毛がなびく場合、房の先端が別れるなどの形状変化が起こるが、これをポリゴンで再現することは難しい。こういった形状変化に対する試みは、実際の制作においてあまり行われていないのが現状である。

3.4 動きにおける課題

キャラクターの動きについては多くの課題が残っている。現在のキャラクターCGは、前述の努力により1フレームだけ切り抜いて絵として見た場合には、手描きと区別がつかないまでになっている。しかし、アニメーションとして動いているキャラクターを見た場合には、違和感が生まれる。動きについても多くの試みがされている。まず、通常の3DCGアニメーションと異なり1秒間のフレーム数を手描きアニメーションと同じ8フレームにすることが多い。また、ダンスなどのキャラクターのモーションはモーション

キャプチャーによって動きをとるが、モーションキャプチャーのデータをそのまま適用するのではなく、カットに合わせ動きを手作業で調整している。髪の毛や服など、所謂揺れものについても物理シミュレーションを行わず手付けによるキーフレームアニメーションを用いることも多い。他にもキーフレームの補完にデフォルメを加えたり、モーション自体にもアニメ的な動きを取り入れている。しかし、これだけの試みがされているにもかかわらず、キャラクターの動きが大きいシーンでは違和感が出てしまっているのが現状である。

3.5 その他の課題

他にもセルルック CG の課題は多く、手描きアニメーションに見られるデフォルメ表現を再現するために様々な試みがされている。もちろんデザイナーの手作業だけではなく、ソフトウェアの面での改良も行われている。アクションシーンで見られるデフォルメされたパースを再現するためのカメラシステムもセルルック CG 用のレンダラに搭載されている。ポストエフェクトに付いてもモーションブラーなどセルルック CG に合わせた特別なアルゴリズムが開発されている。

このように、セルルック CG では手描きと同様の映像を再現するために、様々な工夫がなされている。しかし、多くは未だにソフトウェアの改良によるものではなく、デザイナーのノウハウや試行錯誤に頼った手作業による修正である。言い換えればソフトウェアによる支援について研究の余地が大きく残されていることを意味する。もちろん、すべてをソフトウェアによる自動的な処理で代替できるわけではなく、デザイナーの感性や演出によって修正が必要な部分が残る。しかしながら、デザイナーの行う作業を部分的にでもアルゴリズム化できれば、デザイナーの負担が減り、またアニメーターが持つノウハウがない人間でも、質の高いセルルック CG が作れるようになる。そのため、これからもセルルック CG 用のソフトウェアの発展が期待される。

4 提案手法

本研究ではセルルックな髪の毛を表現するために以下の項目について実装と実験を行った。

- モデリング
- レンダリング
- シェーディング
- アニメーション
- グラフィカルユーザインタフェース

本研究では、スケルトンベースの陰関数曲面を用いて表現した髪の毛の形状に対して、ノンフォトリアリスティックシェーディングを適用している [10, 11]。描画に関する処理の流れは次のとおりになっている。

1. 髪の毛のスケルトンを入力する
2. スケルトンの各パラメータをスプライン関数を用いて補間する
3. 陰関数曲面を用いて髪の毛の形状を生成する
4. ノンフォトリアリスティックシェーディングを適用しレンダリングする

また、本手法に用いる髪の毛のスケルトンを編集するためのグラフィカルユーザインタフェース（以下 GUI）も合わせて提案する。本研究の GUI を使用することで、デザイナーは指の動きで髪の毛のスケルトンを作成することができる。

さらに、物理ベースの髪の毛のアニメーションについても実装を行った。弾性体シミュレーションに用いられる質点バネモデルを使い、髪の毛の動きをシミュレーションしている。

5 スケルトンベースの Blobby モデルを用いた髪の毛の形状表現

本研究の目標の一つとして、セルルックな髪の毛を表現する上で手描き表現に近づけるために従来の方法では困難だった、髪の毛の房の分離結合を容易に表現することが挙げられる。そのために本研究では、キャラクターの髪の毛をスケルトンベースの陰関数曲面により表現する。

従来からセルルックな 3DCG キャラクターの髪の毛は一般的なポリゴンモデル、もしくはパラメトリック曲面を用いてモデリングされてきた。しかしながらポリゴンモデル、パラメトリック曲面共に、オブジェクトが裂けるといった大きな変形を苦手とする。手描きアニメーションにおけるキャラクターの髪の毛は図 1 に示したとおり、風によってなびいた時などに、まとまった大きな房の先端が複数に分かれ、また 1 つに戻るといった変形が多く見られる。このアニメーションをポリゴンモデルにより表現するためには、アニメーション中に動的にモデルを切り替えるなどの手間が必要となる。これには 1 人のキャラクターに対して髪の毛のモデルを複数作成し、それをアニメーションを作る上で適切に切り替える必要があるため、デザイナーにとって大きな負担となる。

本研究ではスケルトンベースのアプローチにより髪の毛の変形を可能にしている。スケルトンベースのモデリングでは、実際の形状はスケルトンから自動的に生成されるため、デザイナーは形状の骨組み (スケルトン) を作るだけで良い。また本手法で用いる陰関数曲面の一種である Blobby モデルはプリミティブ同士をその距離に応じて滑らかに結合させ、アニメーション中の分離、結合を表現しやすい。この Blobby モデルとスケルトンベースのモデリングを用いることで、従来のポリゴンモデルでは表現が難しかった、大きな変形が容易に行える柔軟なモデルを作る。

5.1 陰関数曲面

陰関数曲面は 3DCG における形状の表現方法の一つで、数学的な関数で表現されたモデルである [1]。一般化された陰関数曲面は式 (1) で表される。式が示す通り、陰関数曲面はある方程式の解の集合で定義される曲面である。またベジェ曲面、NURBS 曲面に代表されるパラメトリック曲面や、ポリゴンモデルと異なり、物体の内側外側が定義されるソリッドなオブジェクトを表現できるモデルである。

$$f(x) - C = 0 \quad (C \neq 0) \quad (1)$$

代表的な陰関数曲面として球を例に挙げる。 (c_x, c_y, c_z) を中心とする半径 r の球の陰関数表現は式 (2) で表すことができる。

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2 = 0 \quad (2)$$

陰関数曲面を用いたモデリング手法には、球や楕円体、直方体など単純な式で表すことができるプリミティブをブーリアン演算により組み合わせる手法や、点群データを元に基底関数の組み合わせや部分部分で適切な関数をフィッティングさせることで曲面を再構成する手法、Blobby モデルのようにプリミティブを滑らかに結合させ有機的な形状を作る手法など様々あり、これらを組み合わせることで複雑な形状を表現している。

5.2 Blobby モデル

Blobby モデルは、濃度関数の等値面をオブジェクトの表面とする陰関数曲面の一種で、一般にはメタボールとも呼ばれる。全体の濃度関数をプリミティブによる局所的な濃度関数の重ね合わせで表現される。Blobby モデルでは式 (1) における関数 $f(x)$ をプリミティブ p の濃度関数 $f_p(x)$ を用いて、式 (3) のように定義できる。局所的な濃度関数は距離に応じて減衰する関数で、一般にはガウス関数、距離の二乗逆数、ガウス関数を近似した区分的二次多項式などが用いられる。

$$f(x) = \sum_p f_p(x) \quad (3)$$

Blobby モデルはプリミティブ同士を滑らかに結合させるため、有機的な形状を表現するために用いられる。プリミティブには球が多く使われる他、直方体や円柱なども存在する。オブジェクトの分離や結合を表現しやすく、水などの流体のレンダリングに用いられることが多い。本研究では、この分離結合を表現しやすいという特性を用いてセルルックな髪の毛における房の分離結合を表現する。

5.3 髪の毛のスケルトン

本手法では髪の毛の形状は多数のスケルトンから作られる。このスケルトンはデザイナーによって作成され、そのスケルトンの形状によりキャラクターの髪型をデザインする。一本のスケルトンは一本の細い髪の毛の房を表し、複数のスケルトンが集まることによって、手描きアニメに見られる太い房を表現する。複数のスケルトンを用いて1つの太い房を表現することによって、髪の毛がなびいた時に房の先が複数にわかれる表現を可能にする。髪の毛のスケルトンの例を図2に示す。

スケルトンの各ノードは、ノードの位置情報の他に Blobby モデルを表現するためのパラメータを持つ。Blobby モデルでは、等値面を取るための閾値をオブジェクト全体で1つ、各プリミティブにローカルな濃度関数の分布半径と重みをそれぞれパラメータとして持つ。本手法のモデルでも同様に、髪の毛のモデル全体で閾値を1つと、各ノードに濃度関数の分布半径と重みを持っている。このパラメータは髪の毛のデザインに影響するため、デザイナーが設定を行う。

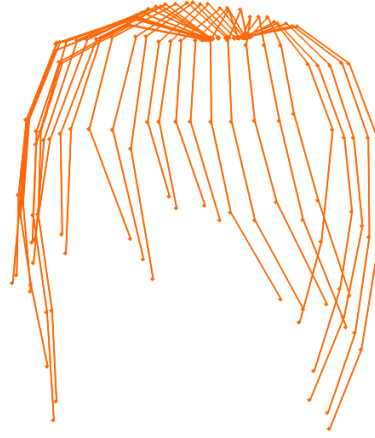


図 2. 本手法で用いる髪の毛のスケルトンの例.

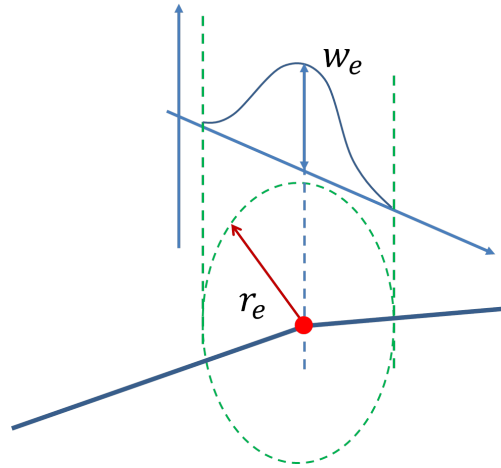


図 3. 髪の毛のスケルトンのノードが持つパラメータ. r_e は濃度関数の分布半径, w_e は濃度関数の重みをそれぞれ表す.

5.4 パラメータの補間

各ノードが持つ位置を含む全てのパラメータをスプライン関数で補間する. スケルトンの各ノード間ではノードが持つパラメータを補間する必要があるが, ただ線形に補間したのでは最終的な形状が角ばってしまう. そこで, なめらかな曲線の髪の毛を表現するために, スケルトンのノード間を 3 次の自然スプライン関数を用いて補間する.

スプライン曲線は, 与えられたコントロールポイントを通るなめらかな曲線で, コントロールポイントで区切られた区間ごとに多項式が与えられる. 3 次のスプライン関数の各区間 $S_j(t_j)$ は式 (4) で与えられる. ここで, a_j, b_j, c_j, d_j はそれぞれスプライン関数の係数である. N 個のコントロールポイントが与えられた場合, 区間は $j = 0 \dots N-1$ となり, t_j は式 (5) で定義される. 3 次のスプライン関数はすべての点において 0 次から 2 次

までの導関数がすべて連続となる．また自然スプラインでは両端の 2 次導関数を 0 と定義する．これらの条件 (式 (6, 7, 8)) から，多項式のすべての係数を求めることができる．

$$\mathbf{S}_j(t_j) = \mathbf{a}_j t_j^3 + \mathbf{b}_j t_j^2 + \mathbf{c}_j t_j + \mathbf{d}_j \quad (0 \leq t_j \leq 1) \quad (4)$$

$$t_j = t - j \quad (j = 0, 1, \dots, N-2) \quad (5)$$

$$\mathbf{S}'_j(1) = \mathbf{S}'_{j+1}(0) \quad (6)$$

$$\mathbf{S}''_j(1) = \mathbf{S}''_{j+1}(0) \quad (7)$$

$$\mathbf{S}''_0(0) = \mathbf{S}''_{N-2}(1) = \mathbf{0} \quad (8)$$

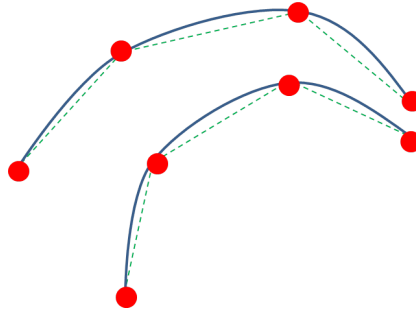


図 4. スケルトンのスプライン補間.

5.5 髪の毛の形状関数

本手法の髪の毛のモデルは Blobby モデルをベースにしている．スケルトンベースの Blobby モデルを用いた本手法では，スケルトン同士が離れた状態ではそれぞれ細い円柱となり，近づくことで複数のスケルトンの滑らかにつながる．これにより，複数のスケルトンがまとまっている時には一つの大きな房として描画され，アニメーション中に分かれることで複数の細い毛先を見ることができる．

各スケルトンを中心に分布する濃度関数を重ねあわせた関数の等値面を取ることで髪の毛の形状を表現している．本手法の髪の毛の形状を表す関数は式 (9) で表され， $f_{\text{surface}}(\mathbf{v}) = 0$ となる点 $\mathbf{v}$ が髪の毛の表面となる．ローカルな濃度関数を重み付けして足し合わせ，閾値で表面を作っていることがわかる．本研究では，ローカルな濃度関数 $f_{\text{density}}(x)$ に，式 (10) に示すガウス関数を近似した区分的二次多項式を用いた．

$$f_{\text{surface}}(\mathbf{v}) = \sum_{e \in S} f_{\text{density}}\left(\frac{D_e(\mathbf{v})}{r_e}\right) w_e - r_{\text{threshold}} \quad (9)$$

$$f_{\text{density}}(x) = \begin{cases} 0 & (1 < x) \\ \frac{3}{2}(1-x)^2 & (\frac{1}{3} < x \leq 1) \\ 1-3x^2 & (x \leq \frac{1}{3}) \end{cases}, \quad (10)$$

ここで、 $\mathbf{v}$ は関数を評価する点、 S はスケルトンが持つ辺の集合、 D_e は点と辺との距離、 r_e は濃度関数の分布半径、 w_e は濃度関数の重み、 $r_{\text{threshold}}$ は等値面を取る閾値となっている。また、濃度関数の分布半径と重みは、スプラインで補間されている値で、辺の中で点との距離が最小になる位置での値となる。

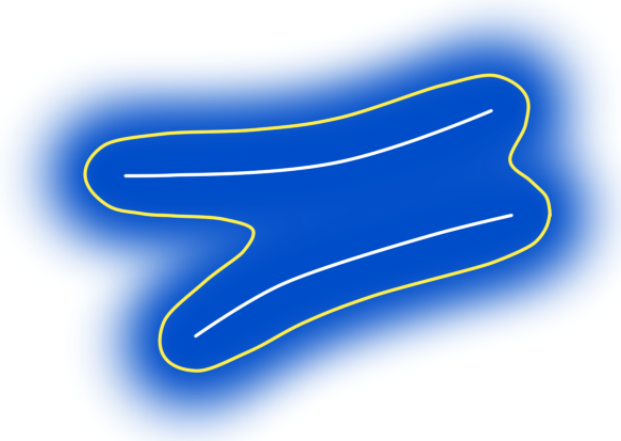


図 5. 本手法のスケルトンベースの Blobby モデルの考え方. 2 本の細い線をスケルトンとすると、その周りに濃度関数が広がり、その等値面をとったものが髪の毛の形状となる。

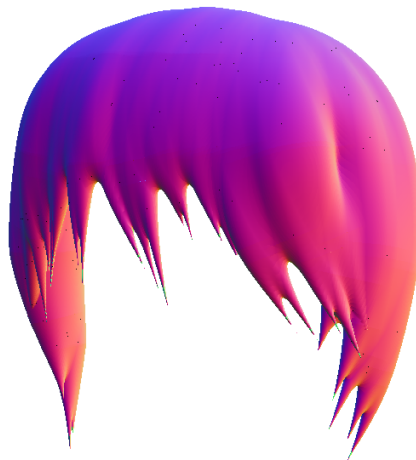


図 6. 本手法を用いた髪の毛のモデリングの例.

6 レイマーチング法による髪の毛のレンダリング

本手法では陰関数曲面を表現された髪の毛をレイマーチング法を用いてレンダリングを行っている。陰関数曲面の描画手法として、レイマーチング法を用いて直接レンダリングする他に、マーチングキューブ法を用いて一度ポリゴンモデルに変換してから、通常のレイトレーシング法やラスターライズにより描画する方法がある。しかし、一般的にマーチングキューブ法はシャープな形状を再現することが難しく、多くのポリゴンを必要とする。そこで本研究ではポリゴン化を行わずにレイマーチング法により直接描画を行っている。

6.1 レイマーチング法

レイマーチング法はレイトレーシング法の一種で、視線（レイ）とオブジェクトとの交点を数値的に求める手法である。通常のレイトレーシング法ではレイとオブジェクトの交点を解析的に求める。しかし、本手法の髪の毛のモデルはポリゴンと異なり複雑な関数であるため、交点を解析に求めることは不可能である。そこで、数値的な手法であるレイマーチング法を用いて交点を求める。

レイマーチング法では、まずレイトレーシングと同様にカメラの位置から対象となるピクセルの方向ベクトルを作る。そしてそのベクトルの方向に向かってカメラ側から点（レイ）を少しずつ進めながら交点を探索する。陰関数曲面はその関数の値によってオブジェクトの内側と外側が定義できる。そこで、カメラからレイを進めながら、その点での関数を評価する。その点がオブジェクトの外側であればそのままレイを進め、内側に入っていれば前のステップとの間にオブジェクトの表面があることになる。レイが物体の内側に入った場合、その点を近似的に表面とする場合もあるが、より正確に求める場合には二分法などを用いてより表面に近い点を探す。

レイを少しずつ進め、そのたびに関数の評価を行うため、1つのオブジェクトにつき1回の計算で済む通常のレイトレーシング法とくらべて計算コストが非常に高い。そのため、推定されるオブジェクトの距離に応じてレイのステップを調整する、探索する範囲を限定するなど、効率化を図ることが重要になる。

6.2 陰関数の評価

レイマーチング法による交点の探索では、レイを進めながらその点における関数を評価する。本手法では髪の毛を式 (9) の関数で表している。この関数ではスケルトンの各エッジを一つのプリミティブとして扱っている。濃度関数はエッジを中心に分布し、エッジとの距離で減衰しているが、補間されたエッジはスプライン曲線であるため距離を数値的に求める必要がある。また、本手法で用いたローカルな濃度関数は、分布半径より外側の値が0となるため、レイから離れたエッジの計算を省くことで高速化を図る。

6.2.1 スプライン関数とレイの距離

陰関数を評価する際に式 (9) に示したとおりスケルトンとレイとの距離を求める必要がある。スケルトンは 3 次のスプライン関数により補間されているため、3 次スプライン関数と点との距離を求める事になる。この場合最小化すべき関数 (11) は 6 次関数となり、解析的に求めることができない。

$$f(t) = (S_x(t) - v_x)^2 + (S_y(t) - v_y)^2 + (S_z(t) - v_z)^2, \quad (11)$$

ここで、 S_x, S_y, S_z は 3 次元空間上のスプライン曲線の各成分を、 v_x, v_y, v_z はレイの座標の各成分を表す。

そこで本研究では [16] で提案されている、二次最小化とニュートン法を組み合わせた手法を用いて距離を求めている。この手法では、まず二次最小化を使い関数を最小化する t のおおよその位置を求めた後にニュートン法を使い t を求める。

二次最小化は関数を放物線に近似し関数の極値を求める方法である。関数から 3 点選びその 3 点を通る 2 次関数を作る。その 2 次関数の極値点をもととの関数の極値点の候補とする。得られた極値点と極値点を中心に 2 点を取り、再び 3 点で 2 次関数を作る。これを繰り返すことで、近似された 2 次関数の極値点がもともとの関数の真の極値点に近づく。もともとの関数の形が 2 次関数に近い場合は近似がうまく働くが、2 次関数とずれると真の極値点に収束しづらい。そこで、二次最小化により極値点の周囲の点を得た後ニュートン法を用いることで真の極値点に収束させる。

ニュートン法 (ニュートン・ラフソン法) は高次方程式の解を数値的に求める方法である。極値点は関数の 1 次導関数が 0 の値を取る点であるため、式 (11) の 1 次導関数のゼロ点を求めることになる。ニュートン法では、まずある点における関数の接線を求める。その接線と軸との交点を求め、再び接線を求める。それを繰り返すことで軸と接線との交点が解に近づいていく。しかしながらニュートン法は、初期値によっては値が発散し解が求まらない事がある。そこで、二次最小化により求まった点を初期値として使うことで、収束しやすくなる。

本研究では収束したかどうかに関わらずそれぞれのイテレーション回数を 4 回ずつとして実装している。

6.3 バウンディングボックスによる高速化

レイマーチング法による交点の探索にかかる時間は、探索する区間の大きさに大きく影響される。そこでバウンディングボックスを用いて探索区間を限定し高速化する。バウンディングボックスを用いることで、交点を持つ可能性がある区間のみを探索し、また、計算する必要のない関数の評価を省くことができる。

バウンディングボックスは、ベジェ曲線の凸包性を利用することで構築する。ベジェ曲線の特性により、曲線はコントロールポイントを結んだ多角形の内部にある。そこで、ス

ケルトンを補間している 3 次の自然スプライン曲線の各区間を一度 3 次のベジェ曲線に変換し、そのコントロールポイントを囲うようにバウンディングボックスを作る。3 次ベジェ曲線 $B_j(t_j)$ は 4 つのコントロールポイント $v_{j0}, v_{j1}, v_{j2}, v_{j3}$ を用いて式 (12) と定義される。そして、式 (12) を t_j について解くことで式 (13) を得る。

$$B_j(t_j) = t_j^3 v_{j3} + 3t_j^2(1-t_j)v_{j2} + 3t_j(1-t_j)^2 v_{j1} + (1-t_j)^3 v_{j0} \quad (12)$$

$$\begin{aligned} B_j(t_j) = & (v_{j3} - 3v_{j2} + 3v_{j1} - v_{j0})t_j^3 \\ & + (v_{j2} + 2v_{j1} + v_{j0})t_j^2 + (v_{j1} - v_{j0})t_j + v_{j0} \end{aligned} \quad (13)$$

式 (13) と式 (4) の各項から連立方程式 (14) を得る。この連立方程式 (14) を解くことでコントロールポイント $v_{j0}, v_{j1}, v_{j2}, v_{j3}$ を求めることができる。

$$\begin{cases} v_{j3} - 3v_{j2} + 3v_{j1} - v_{j0} = a_j \\ v_{j2} + 2v_{j1} + v_{j0} = b_j \\ v_{j1} - v_{j0} = c_j \\ v_{j0} = d_j \end{cases} \quad (14)$$

ここで求められたコントロールポイント $v_{j0}, v_{j1}, v_{j2}, v_{j3}$ の最大値最小値を用いてバウンディングボックスを作る。また、各スケルトンの濃度関数の分布範囲はスケルトンから各ノードが持つ分布半径であるため、バウンディングボックスを作る際にコントロールポイントの座標から分布半径の最大値分広げる必要がある。

各スケルトンの濃度関数が 0 以外の値を持つのはそれぞれのバウンディングボックスの中だけである。そこではじめにレイとバウンディングボックスとの交差判定を行い、交差したバウンディングボックスの内部のみをレイマーチングにより探索する。またレイを進める上で各点の陰関数を評価する際にもバウンディングボックスを用いて、点が自身のバウンディングボックスに入っていないスケルトンについては評価を省くことで、高速化をしている。

また、レイと交差するバウンディングボックスを探索する方法として、通常のレイトレーシング法の高速化にも用いられる、バウンディングボリューム階層という手法を用いている。バウンディングボリューム階層は、バウンディングボリューム（本手法ではバウンディングボックス）で木構造を作り効率的に交差するオブジェクトを探索する手法である。

7 セルルックな髪の毛のシェーディング

本研究では髪の毛のモデルにセルルックなシェーディングを適用している．アニメに多く見られる表現では，髪の毛は光の当たる明るい部分と陰影の 2 色，もしくはハイライトを追加した 3 色で塗り分けられている．陰影は光が手前からあたっているとすると，髪の毛の房の分け目や，髪の毛の端に多く見られる．また，ハイライトはいわゆる天使の輪と呼ばれる水平な線が髪に横切るものが見られる．本研究のシェーディングはこの特徴を再現することを目標にしている．

7.1 法線の計算

式 (15) で定義される陰関数曲面の点 (x, y, z) での法線 $\mathbf{N}$ は式 (16) で得られる．

$$f(x, y, z) = 0 \quad (15)$$

$$\mathbf{N} = \left(\frac{\partial f(x, y, z)}{\partial x}, \frac{\partial f(x, y, z)}{\partial y}, \frac{\partial f(x, y, z)}{\partial z} \right)^T \quad (16)$$

本手法の髪の毛の関数の偏微分は解析的に求めることができない．そこで式 (16) と式を変形し数値的に求める．

$$\mathbf{N} = \begin{pmatrix} \frac{f(x + \Delta x, y, z) - f(x, y, z)}{\Delta x} \\ \frac{f(x, y + \Delta y, z) - f(x, y, z)}{\Delta y} \\ \frac{f(x, y, z + \Delta z) - f(x, y, z)}{\Delta z} \end{pmatrix} \quad (17)$$

7.2 トゥーンシェーディング

本手法では一般的なトゥーンシェーディングをベースとしたシェーディングを実装した．まず各ピクセルごとにサーフェスの法線と光源の情報から輝度値を計算した後，閾値を使い 2 色に塗り分ける．セルルック特有の表現を再現するために，拡散反射による色とハイライトによる色を個別に計算した後合成している．本シェーディングはキャラクターの頭を基準として，上下方向を Z 軸としたローカル座標系上で計算している．

ハイライトは法線ベクトルと光源への方向ベクトルの垂直方向の角度差を使って計算される．これは髪の毛に見られる異方性のハイライトを表現するためである．ハイライトの計算手順を以下に示す．

1. 法線ベクトルと光源への方向ベクトルを頭を基準としたローカル座標系にマッピングする．

2. 法線，光源方向それぞれの 3 次元ベクトルを Z 軸に沿った垂直方向の成分 (式 (19) で表される) と，XY 平面状に投影したベクトルに沿った平行方向の成分 (式 (20) で表される) からなる 2 次元ベクトルに変換する。
3. ステップ 2 で 2 次元ベクトルに変換された，法線と光源方向の 2 つのベクトルの内積をとり，それをハイライトの輝度とする (式 (18) で表される)。

拡散反射による輝度は法線ベクトルと光源への方向ベクトルの水平方向の角度差と，法線ベクトルの垂直方向の成分を利用して計算される。2 つの要素はそれぞれ法線と光源方向の水平方向の角度差は式 (23) で，法線の垂直方向の成分は式 (22) によって定義される。この内水平方向の角度差は，法線ベクトルと光源への方向ベクトルを頭を基準としたローカル座標系に投影したものの内積をとった形になっている。拡散反射の輝度はこの 2 つの要素を掛けあわせることで計算される (式 (21) で表される)

$$I_{\text{specular}} = N_{\text{latitude}} L_{\text{latitude}} + N_{\text{longitude}} L_{\text{longitude}} \quad (18)$$

$$V_{\text{latitude}} = V \cdot E_z \quad (19)$$

$$V_{\text{longitude}} = \sqrt{(V \cdot E_x)^2 + (V \cdot E_y)^2} \quad (20)$$

$$I_{\text{diffuse}} = I_{\text{latitude}} I_{\text{longitude}} \quad (21)$$

$$I_{\text{latitude}} = 0.5 (N_{\text{latitude}} + 1) \quad (22)$$

$$I_{\text{longitude}} = 0.5 \left(\frac{L \cdot E_x}{L_h} \frac{N \cdot E_x}{N_h} + \frac{L \cdot E_y}{L_h} \frac{N \cdot E_y}{N_h} + 1 \right), \quad (23)$$

ここで， N は法線ベクトル， L は光源方向への単位ベクトル， E_x ， E_y ， E_z はキャラクターの頭を基準としたローカル座標系の基底ベクトルを表す。また， V は N か L のいずれかを表し，例えば N_{latitude} は (19) の V を N に置き換えた式で計算される。ここまでの計算に使用した各輝度の成分を可視化したものが図 7 となる。

拡散反射成分とハイライトの輝度値を計算した後に，それぞれ閾値から領域を分け色を決定する。ユーザが指定するパラメータは，ハイライトの領域を決定する閾値 t_{specular} ，明るい領域と陰の領域とを分ける閾値 t_{diffuse} ，そしてハイライト，明るい領域，陰領域のそれぞれを塗りつぶす色 c_{specular} ， c_{diffuse} ， c_{shade} の 5 つである。これらのパラメータを使い，式 (25,26) に示す通りにそれぞれ色を決定し，最終的に 2 つの色を足し合わせる (式 (24))。

本手法のシェーディングを適用した例を図 8 に示す。明るい部分と陰の部分がはっきりと塗り分けられ，一般に天使の輪と呼ばれる平行なハイライトが入っていることがわかる。

$$C = C_{\text{specular}} + C_{\text{diffuse}} \quad (24)$$

$$C_{\text{specular}} = \begin{cases} c_{\text{specular}} & (t_{\text{specular}} < I_{\text{specular}}) \\ 0 & (I_{\text{specular}} \leq t_{\text{specular}}) \end{cases} \quad (25)$$



図 7. シェーディングに用いる輝度情報をそれぞれ可視化したもの. I_{specular} (左上), I_{diffuse} (右上), $I_{\text{longitude}}$ (左下), I_{latitude} (右下).

$$C_{\text{diffuse}} = \begin{cases} c_{\text{diffuse}} & (t_{\text{diffuse}} < I_{\text{diffuse}}) \\ c_{\text{shade}} & (I_{\text{diffuse}} \leq t_{\text{diffuse}}) \end{cases} \quad (26)$$

ここで t は輝度の閾値, c は色を表す.



図 8. 本手法によるシェーディングの例.

8 物理ベースの髪の毛アニメーション

本研究では物理ベースの髪の毛のアニメーションを実装した。髪の毛のアニメーションにはキーフレームアニメーションが用いられることもあるが、本手法ではスケルトンの数が多く手作業でのアニメーションはコストが高い。そのため、物理ベースのアニメーションを適用し、頭の動きを入力することでその動きに合わせた髪の毛の動きを計算する。

8.1 バネを用いた髪の毛の物理モデル

本研究では質点バネモデルを用いた髪の毛のアニメーションを実装した。質点バネモデルは弾性体シミュレーションに用いられるモデルで、弾性体を多数の質点とそれをつなぐバネとで表現したものである [4]。髪の毛のシミュレーションは弾性体シミュレーションの一種として扱われ質点バネモデルが使われることが多い [12]。

バネ室点モデルでは弾性体を質点とそれらをつなぐバネとで表現し、質点の運動方程式を解くことでオブジェクトの動きを計算する。図 9 に本手法で構築した質点とバネのコネクションを示す。このモデルでは、各ノードごとに 1 つの主質点と 8 つの補助質点が置かれる。主質点はノードの座標と同じ位置に置かれ、この質点の動きが実際の髪の毛のスケルトンの動きとなる。補助質点はノードを含むスケルトンの接ベクトルに垂直な面上に置かれ、ノードから Blobby モデルのパラメータである分布半径だけ離れた位置に円形に置かれている。補助質点とそれをつなぐバネは、スケルトンの変形やねじれを復元し、髪の毛の形状を保つ役割を持つ。バネはそれぞれ、同じノードの主質点と補助質点、同じノードの隣合う補助質点、隣のノードの主質点同士、隣のノードの主質点と補助質点、隣のノードの対応する補助質点同士、2 つ隣のノードの補助質点同士を結んでいる。

質点 p にかかる力を示したものが式 (27) となる。

$$\mathbf{f}_p = \sum_{s \in S} k_s (l_s - l_{s,\text{natural}}) \mathbf{d}_{s,p} + \mathbf{g}m_p - c_{\text{damp}}\mathbf{v}_p, \quad (27)$$

ここで、 S はバネの集合、 k_s はバネ s のバネ定数、 l_s はバネ s の長さ、 $\mathbf{d}_{s,p}$ は質点 p から質点 p につながるバネ s の逆側の質点への方向ベクトル、 $\mathbf{g}$ は重力加速度、 m_p は質点 p の質量、 c_{damp} は粘性減衰係数、 $\mathbf{v}_p$ は質点 p の速度をそれぞれ表す。また、本研究では重力以外の外力を考慮せず、髪の毛に対する空気抵抗などの力はまとめて粘性減衰力として扱う。

本研究で実装したシミュレーションのフローチャートを図 10 に示す。また、本研究では時間積分に 4 次のルンゲクッタ法を使用している。4 次のルンゲクッタ法での推定誤差は時間ステップの 5 乗のオーダーとなり、1 ステップの計算量は増えるものの時間ステップをオイラー法と比べ大きく取ることができるため効率が良い。

通常アニメーションは 1 秒間に 24~60 フレームが使われるが、本研究のシミュレーションでは 1 ステップを 10^{-4} 秒としている。キャラクターのアニメーションは通常アニメーションと同様のフレームレートで入力されるため、フレーム間のステップは線形補

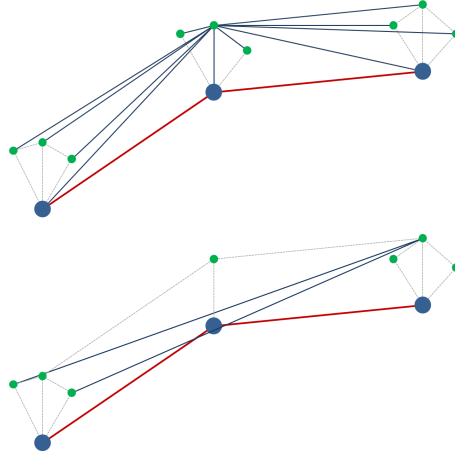


図 9. 本手法で用いる質点とバネのコネクション．上の図は，一つの補助質点からのコネクションで，隣り合う質点や隣のノードの質点とそれぞれつながっている．下の図はねじれや変形に対して働くバネで隣りのノードのさらに一つ先のノードの質点と繋がれている．

間している．また，単位系を合わせるために，入力されたデータを MKS 単位系に変換している．

8.2 バネの自然長の最適化

髪の毛の挙動は重力に大きく影響を受ける．そのため，たとえキャラクターが動いていない状態であっても，髪の毛は重力によって垂れ下がり入力されたスケルトンの形状を保つことはできない．質点をつなぐバネの定数を大きくすればある程度変形を小さくすることができるが，それではキャラクターが動いた際にも変形をしなくなってしまう．入力されたスケルトンの形状がデザイナーがデザインした本来のキャラクターであり，何もしていない状態でその形状が崩れてしまうのは好ましくない．またデザイナーが重力の影響を考慮してスケルトンを作成することは非常に困難である．そこで本研究では，シミュレーションの開始前に初期状態で質点へかかる力が釣り合うように，質点をつなぐ各バネの自然長を調整することで形状が崩れることを防いでいる．

重力とバネからの力が釣り合い形状が保たれている状態では式 (28) が満たされる．

$$\mathbf{f}_p = \sum_{s \in S} k_s (l_{s, \text{initial}} - l_{s, \text{natural}}) \mathbf{d}_{s,p} + \mathbf{g} m_p = \mathbf{0}, \quad (28)$$

ここで $l_{s, \text{initial}}$ は初期形状でのバネ s の長さ, $l_{s, \text{natural}}$ は重力と釣り合うように調整されたバネ s の自然長を表す．式 (28) をすべての質点について連立し， $l_{s, \text{natural}}$ について解くことで，重力と釣り合うバネの自然長を求めることができる．

また，立てられる方程式の数と求める変数の数はそれぞれ，質点の数とバネの本数に対応している．バネの本数は質点の数と比べ多いため，連立方程式の解は一意に定まらない．そこで本研究では，特異値分解を用いた一般化逆行列を利用し解を求めた．特異値分

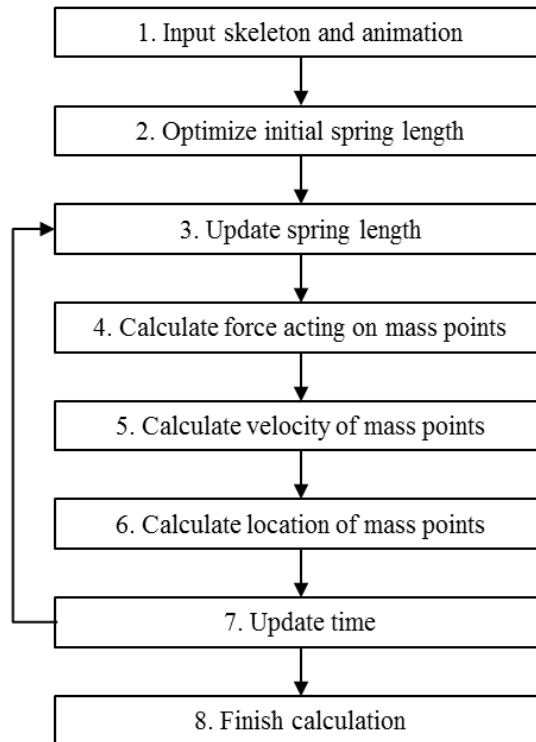


図 10. 本手法で実装した物理ベースアニメーションのフローチャート.

解を用いることで得られる方程式の解は，取りうる解の中でベクトルのノルムを最小化するように求まる．それによりバネの長さの偏りが小さくなることが期待できるため，本手法に適していると考ええる．

前述の処理により，シミュレーション開始時に髪型が崩れてしまうことを防いでいる．しかし，実際のコンピュータ上での計算では誤差が乗るため式 (28) の右辺を厳密に $\mathbf{0}$ にすることができない．それにより時間ステップが進むに連れ誤差が蓄積し，キャラクターが動いていない状態で一部のスケルトンが振動を始めてしまう．そこで，本研究では大きさが 10^{-6} という非常に小さい力は $\mathbf{0}$ として扱っている．

9 デプスカメラを用いたユーザインタフェース

本手法で入力データとして用いるスケルトンは、既存の 3DCG ソフトではプラグイン、アドオンなしに作成することができない。そこで、本研究ではスケルトンを作成するためのグラフィカルユーザインタフェースを実装した。本研究で実装した GUI の画面キャプチャを図 11 に示す。

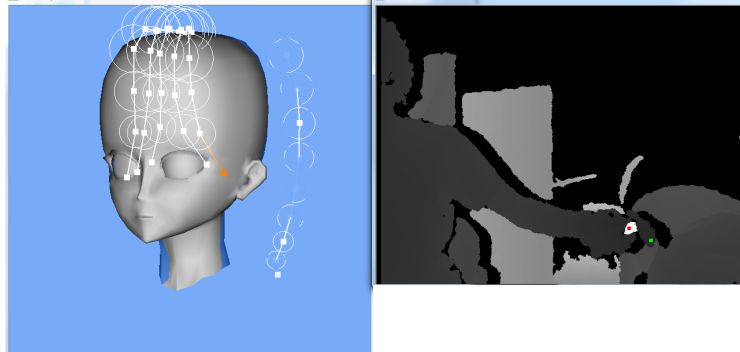


図 11. 本研究で実装した GUI の画面キャプチャ。左側のウィンドウがスケルトンを編集するためのインタフェース，右側がデプスカメラから取得した深度情報とそこから検出した指先の情報。

本研究ではグラフィカルユーザインタフェースの実装に GUI 開発フレームワークである Qt のオープンソース版を利用している。

9.1 デプスカメラを用いたインタフェース

本手法では大きな髪は複数のスケルトンで表現することを想定しているため、デザイナーは数百のスケルトンノードを作り、パラメータの編集をする必要がある。これはデザイナーにとって大きな負担となる。そこで、直感的な操作でスケルトンを作成することができるユーザインターフェースが必要になる。

本研究では、本手法に用いる髪の毛のスケルトンを指の動きによって作成する事ができるインタフェースを実装した。指の動きはデプスカメラから得られる深度情報を元にトラッキングする。デプスカメラを用いて 2 本の指先をトラッキングし、その 2 本の指の位置と間隔からノードの位置とノードが持つ濃度関数の半径を設定する。

2 本の指先はカメラ画像内で最もカメラに近い 2 つの出っ張った領域 (図 11 内に見える白い領域) として定義している。この指先の領域を探索するプロセスは以下ようになる。

1. 深度画像の中で最もカメラに近いピクセルを探す。
2. ステップ 1 のピクセルから周囲に 1 ピクセルずつ領域を拡張していく。
3. 拡張するピクセルの深度とステップ 1 のピクセルの深度との差が設定された閾値を

超えるか，外側のピクセルが内側のピクセルより手前だった場合領域の拡張を打ち切る．

4. ステップ 3 で確定した領域を 1 つめの指先領域とする．
5. 深度画像から 1 つめの指先領域を除き最も近いピクセルを探す．
6. ステップ 2-3 と同様に領域を拡張する．
7. ステップ 6 で確定した領域を 2 つめの指先領域とする．
8. 2 つの指先領域の中心をそれぞれの指先の位置とする．

2 本の指先の中点をノードの位置，距離の半分をノードの濃度関数の半径とする．

1 つめの指領域と 2 つめの指領域はフレーム間で独立に検出される．そこで，前のフレームの 2 点と現在のフレームの 2 点でそれぞれ距離を比較し，近い組を同じ指としてラベル付けしている．また，デプスカメラから得られる深度情報は完全ではなく，指先の深度情報が取れないフレームもある．その場合，突然あるフレームで検出される指の位置が飛んでしまう事がある．本手法では複数フレームで指の位置を平滑化しているが，指の検出がうまくいかなかった場合には，平滑化で得られる位置も大きくハズレてしまう．そこで，検出できなかった場合の影響をなくすために，複数フレームでの平均を求めた後，平均から大きく外れた値を除外した後再び平均を求めている．

本研究ではデプスカメラに Microsoft 社製の Kinect for Windows を利用した．

10 結果

本手法を用いて、手描きアニメーションに見られる髪の毛の変形を再現したものを図 12 に示す。また、使われたスケルトンを図 13 に示す。左の形状はすべてのスケルトンが先端で集まるように整えられたもので、右の形状は左で使われたスケルトンの一部を変形させて作られたものである。本手法により表現される髪の毛はスケルトンを変形させるだけで図 1 で見られる手描きと同様の変形を 3DCG で再現することができる。

10.1 髪の毛のアニメーション

本手法を用いた髪の毛のアニメーションの例を図 14 に示す。このアニメーションは図 15 に示すサイドテールと呼ばれる髪型のスケルトンを用いて、頭を左右に揺らした場合の髪の毛の動きをシミュレーションしたものである。髪の毛が頭の動きに合わせて揺れていることがわかる。また、揺れた髪の毛の先は複数の細かい房に分かれ、アニメーションの中で再び結合している。シェーディングについても、明るい部分と陰影とが塗り分けられ、いわゆる天使の輪と呼ばれるハイライトが描画されていることがわかる。

10.1.1 コネクションの違いによるモーシヨンの違い

バネのコネクションによるモーシヨンの違いを示す。バネのコネクションは髪の毛の動きとアニメーション中の髪の毛の形状に大きく影響する。本手法ではバネを隣り合うノードの質点同士だけでなく、そのさらに 1 つ先のノードの質点同士もつないでいる (9 の下の図を参照)。このバネにより、大きな動きの中でも髪の毛の形状をもととのキャラクターの髪型を保つことができる。このバネの有無によるアニメーション中の形状の違いを図 16 に示す。1 つ先のノードとのコネクションがない場合、頭が動き出した直後からサイドテールがバラバラに分かれ、動きが止まった後も元の形から大きく崩れてしまっている。一方 1 つ先のノードとのコネクションがある場合、サイドテールが比較的まとまったまま動いており、動きが止まった後も元の形には戻っていないものの、コネクションがない場合と比べると崩れ方が小さい。

10.1.2 バネの自然長の調整の有無による違い

バネによる力と重力と釣り合うように初期状態のバネの自然長の調整をしない場合、動きがない状態であっても重力により髪型が崩れてしまう。それを示したものが図 17 である。バネの自然長を調整していない場合、頭が動く前からサイドテールが重力の影響で垂れ下がっていることがわかる。バネの自然長を調整した場合は、初期状態で各質点にかかる力が釣り合っているため頭が動くまでは髪型が崩れていない。

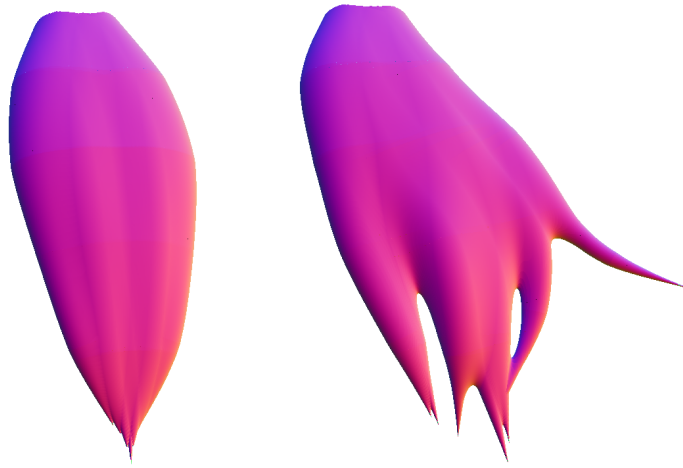


図 12. 手描きアニメーションと同様の変形を再現した例.

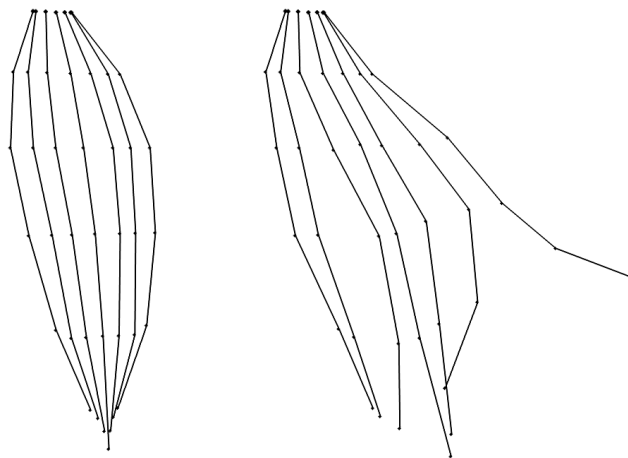


図 13. 図 12 で用いたスケルトン. スケルトンの構造は変わらず形状が変化している.



図 14. 本手法を用いたアニメーションとレンダリングの例.



図 15. 実験に用いたスケルトン.



図 16. コネクションの違いによる髪型の変化. 左は隣のノードとのコネクションのみ, 右はさらに 1 つ先のノードとのコネクションを持つ場合. 1 つ先のノードとコネクションを持っているほうが形状を保っていることがわかる.



図 17. バネの自然長の調整の有無による髪型の変化. 左は何も処理を行わなかった場合の髪, 右は初期形状で重量と釣り合うようにバネの自然長を調整した場合の髪. 共にシミュレーション開始から 1 秒後の状態. 何も処理を行わなかった場合は髪が重力の影響で垂れ下がり元の形を保っていないことがわかる.

11 まとめと今後の課題

本研究では、セルルックな髪の毛を表現するために、以下の項目について研究実装を行った。

- Blobby モデルを用いた髪の毛のモデリング
- レイマーチング法を用いたレンダリング
- セルルックな髪の毛のシェーディング
- 物理ベースな髪の毛アニメーション
- デブスカメラを用いたスケルトン編集 GUI

本手法を用いることで、従来のポリゴン、パラメトリック曲面を使ったモデルでは難しかった、より手描きに近い変形を表現することができた。また、手描きアニメーションに見られる特徴的な陰影を描画した。髪の毛の動きを物理ベースのモデルを使うことで、頭の動きに合わせた動きをシミュレーションした。デザイナーがスケルトンを作成する負担を減らすために、デブスカメラを用いて指先をトラッキングすることでスケルトンを生成する。直感的なインタフェースの実装も行った。

しかしながら、本研究の手法はまだ実用段階にはない。今後本手法を実際のアニメ制作に利用できる段階に持っていくために解決すべき課題を以下に挙げる。

まず形状について挙げる。本研究では髪の毛の形状を表現するための濃度関数としてベーシックな Blobby モデルと同様にガウス関数の近似関数を使っている。本来、手描きの髪の毛は図 1 の右に示したように、各房の分かれ目がシャープになっている。しかし、Blobby モデルを使った本研究の髪の毛の各房はすべてなめらかな曲面で融合するため、そのシャープな分かれ目が表現できていない。これを解決するために、スケルトンベースの表現として単純な Blobby モデルを用いるのではなく、髪の毛の表現に特殊化した関数を考える必要がある。

次にアニメーションについて挙げる。本研究では、髪の毛の各スケルトンはそれぞれが完全に独立して動いている。現状では髪の毛の房同士、または髪の毛とキャラクターの体の間で衝突の判定は行っていない。そのため、本手法のモデルに適した髪の毛の衝突判定と衝突を考慮した動きについて考える必要がある。本手法の髪の毛のモデルでは、各髪の毛の房は距離に応じて結合するため、それぞれのスケルトンは別のスケルトンの形状の中に入る可能性がある。しかし、キャラクターの体と髪の毛の形状とが交差することは好ましくない。

また髪の毛の房が独立して動いていることによる問題がもう一点ある。図 14 から分かる通り、揺れた髪の毛が散らばり、それぞれ 1 つのスケルトンで作られた細い髪の毛ばかりになっている。これはデザイナーにより作られた元の髪の毛の形状とは大きく異なっており、また、セルルックな髪の毛の特徴である大きな房が表現できていない。より手描きに近い形状を表現するためには、毛先に近い部分ではそれぞれのスケルトンごとに独立して

動くものの、大まかにはある大きな房で動くことが好ましい。これを解決するために、パネについて一つのスケルトンの中の質点同士のコネクションだけでなく、スケルトン間でのコネクションについても考える必要がある。

次にシェーディングについて挙げる。本手法ではトゥーンシェーディングとして代表的な手法をベースに実装を行った。しかしながら、まだまだ手描きと同様の絵になっているとは言えず、また表現に幅がない。そのため、さらに手描きの表現に近づけるためには、アニメーターによる塗りを模倣するシェーディング手法を考える必要がある。

また、現在セルルックな 3DCG はアニメーションだけではなく、ゲームなどインタラクティブなアプリケーションにおいても必要がある。しかしながら本手法では現状レンダリングにはレイマーチング法を用いる必要があるため、リアルタイムでの描画が難しい。そこで、GPU による高速化や、バウンディングボックス以上に計算を効率化できる手法を用いて高速化を図る必要がある。

このようにまだまだ課題は多いものの、本研究により、セルルックな髪の毛の表現にスケルトンベースの陰関数曲面を用いることが有効な選択肢の一つであることを示すことができた。

謝辞

本研究を進めるに当たりご指導を賜りました法政大学情報科学部の Vladimir Savchenko 教授に深く感謝申し上げます。同学部の小池崇文教授には研究に関する助言，学会発表の支援等を頂き大変お世話になりました。ライトトランスポートエンタテインメント株式会社の藤田将洋氏，Pixar Animation Studios の Ryusuke Villemin 氏には髪の毛のレンダリング手法，高速化手法についてご助言頂きました。法政大学大学院情報科学研究科の杉本宗一郎氏，小泉悠馬氏には，研究において物理数学に関するアドバイスを多く頂きました。同研究科の同輩の皆様には，日頃より研究についての議論，アドバイスを頂きました。法政大学情報科学部の後輩の皆様は，大学院での研究生生活を送る上で心の支えとなりました。皆様に心より感謝申し上げます。

参考文献

- [1] Jules Bloomenthal and Brian Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
- [2] V.J. Cassol, F.P. Marson, and S. Raupp Musse. Procedural hair generation. In *Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on*, pages 185–190, 2009.
- [3] Martin Côté, Pierre-Marc Jodoin, Charles Donohue, and Victor Ostromoukhov. Non-Photorealistic Rendering of Hair for Animated Cartoons. In *Proceedings of GRAPHICON'04*, 2004.
- [4] Xavier Provot Institut and Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *In Graphics Interface*, pages 147–154, 1996.
- [5] J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '89, pages 271–280, New York, NY, USA, 1989. ACM.
- [6] Yuki Koyama and Takeo Igarashi. View-dependent control of elastic rod simulation for 3d character animation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 73–78, New York, NY, USA, 2013. ACM.
- [7] X. Mao, S. Isobe, K. Anjyo, and A. Imamiya. Sketchy hairstyles. In *Proceedings of the Computer Graphics International 2005*, CGI '05, pages 142–147, Washington, DC, USA, 2005. IEEE Computer Society.
- [8] Xiaoyang Mao, H. Kato, A. Imamiya, and K. Anjyo. Sketch interface based expressive hairstyle modelling and rendering. In *Computer Graphics International, 2004. Proceedings*, pages 608–611, 2004.
- [9] P. Noble and Wen Tang. Modelling and animating cartoon hair with nurbs surfaces. In *Computer Graphics International, 2004. Proceedings*, pages 60–67, 2004.
- [10] Takeyuki Sakai and Vladimir Savchenko. Skeleton-based anime hair modeling and visualization. In *Cyberworlds (CW), 2013 International Conference on*, pages 318–321, 2013.
- [11] Takeyuki Sakai and Vladimir Savchenko. Skeleton-based cartoon hair modeling using blobby model. In *SIGGRAPH Asia 2013 Posters*, SA '13, pages 17:1–17:1, New York, NY, USA, 2013. ACM.
- [12] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 64:1–64:11, New York, NY, USA, 2008. ACM.

- [13] Jung Shin, Michael Haller, and R. Mukundan. A stylized cartoon hair renderer. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '06, New York, NY, USA, 2006. ACM.
- [14] Eiji Sugisaki and Yizhou Yu. Simulation-based cartoon hair animation. In *In 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG' 05*, pages 117–122, 2005.
- [15] Pascal Volino and Nadia Magnenat-Thalmann. Real-time animation of complex hairstyles. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):131–142, 2006.
- [16] Hongling Wang, Joseph Kearney, and Kendall Atkinson. Robust and efficient computation of the closest point on a spline curve. In *In Proceedings of the 5th International Conference on Curves and Surfaces*, pages 397–406, 2002.
- [17] Kelly Ward, Florence Bertails, Tae yong Kim, Stephen R. Marschner, Marie paule Cani, and Ming C. Lin. A survey on hair modeling: styling, simulation, and rendering. In *IEEE TRANSACTION ON VISUALIZATION AND COMPUTER GRAPHICS*, pages 213–234, 2006.