修 士 論 文

果実における細毛表現手法の提案

北陸先端科学技術大学院大学
知識科学研究科知識科学専攻

高見澤　大輔

2012 年 3 月

修 士 論 文

果実における細毛表現手法の提案

指導教員 宮田一乗　教授

北陸先端科学技術大学院大学
知識科学研究科知識科学専攻

1050031 高見澤　大輔

審査委員：　宮田一乗　教授 (主査)
西本一志　教授
梅本勝博　教授
DAM　HIEU　CHI　准教授

2012 年 2 月

目 次

図 目 次

第1章　はじめに

この章ではまず研究の背景について述べ，現在存在している問題について触れる．続いて設定した問題とその解決方法の概要を述べる．関連研究ではどういった解法がなされていて，どのような問題があるのかを述べる．

1.1　背景

コンピュータグラフィックス（以下 CG と表記）では，コンピュータを使用することによってユーザの望む風景や物体，自然現象だけではなく現実には存在しないようなさまざまなイメージまで表現することが可能である．そのため現在では，CG は映画やドラマのような映像作品だけではなく，ゲームや広告など多岐に渡って利用されている．また一方で，そのような商業利用だけではなく個人でも趣味として積極的に利用されている．

この背景としてパーソナルコンピュータ（以下 PC と表記）や CG を作成するためのソフトウェアが普及し，個人でも簡単に手に入れられるようになったことがある．このことで CG 作成のための環境が用意しやすくなっただけではなく，個人が製作した作品を発表する場も増えている．これはインターネットの発達が大きく関係している．インターネットの発達と共に Web 上では大容量のデータをやり取りすることが可能になり，テキストデータのみではなく音声や画像，映像などさまざまな情報がやり取りされるようになった．ソーシャル機能をもった音楽共有サービスである SoundCloud[23] ではフリー音源のやり取りが行われ，youtube[27] や Ustream[24]，ニコニコ動画 [18] ではプロアマ問わず音楽・映像コンテンツの配信が行われている．これらのサービスに共通する特徴は作品や配信に対して，ユーザの反応が直接帰ってくることである．従来ではプロが商用に販売や公開していた音楽や映像などのコンテンツが，個人でも気軽に時間や場所を選ばず数多くの人間へ提供されるようになった．

このことによってそれまで個人では難しかった作品を多くの人間に対して発表し，感想を得ることが気軽に出来るようになり，コンテンツ制作に対するモチベーションへと繋がっている．同様に公開されたコンテンツを別のユーザが手に入れやすくなり，作品に触れる機会が増えたことでコンテンツ制作に取り組むユーザが増えている．このようにして近年では個人でも CG を作成する環境を整えやすく，作成した CG を発表する場が増えている．このことで商用に限らず個人による CG 制作が頻繁に行われていて，Google SketchUp[8] ではユーザが作成した 3DCG モデルが 3D ギャラリーにおいて多く公開されているほかに，ディスカッショングループによるユーザ同士の交流も行われている．加え

て，2012 年 2 月現在 MikuMikuDance[15] に関連した動画がニコニコ動画に 6 万件以上投稿されている．

CG を利用する機会が増えている一方で，CG 表現の難しさがある．CG ではユーザの表現したい事物を一から手動で作成することも可能ではあるが，それには多くの作業や表現を実現するための知識が必要で，大規模なシーンやアニメーションを作成する場合には膨大な時間が求められる．また作成した CG が必ずしもユーザの満足のいくものとは限らず，人間が観て違和感のないようなリアリティのある表現を実現することは容易ではない．特に自然物は人間が普段から目にする機会も多く，また成長の過程などで周囲にあるさまざまな要素から強い影響を受けるため違和感なく再現することは難しい．そこでユーザの手間を削減し，満足いく表現を実現するためにさまざまな研究 [1][11][12][13][14][19][20][30] がされている．文献 [1] では毛筆の再現を行なっており，筆先のしなりを計測することで高速でリアルな毛筆表現を実現している．文献 [11] はユーザの作成したウロコをユーザの描くストロークによって 3D モデル上に自動で配置するウロコ表現のための手法である．また文献 [12] はフリーハンドのペインティングを支援する研究で，ユーザの描くストロークから描画対象を判定しキャンバスの背景に描画対象と類似する画像のシルエット表示することで支援する．文献 [13] は入力された 3D 形状からポップアップカードを自動で作成するポップアップカード製作支援手法である．文献 [14] は現実に存在する建物の構造をもとに，ユーザの設定した間取りから建物の内装も考慮した 3DCG を生成する手法である．文献 [19] は複数の始点から描いたシルエットをもとに 3DCG を再現する．文献 [20] はキャンバスに絵を描くのと同様の感覚で 3 次元空間上でのペインティングを実現した．文献 [30] でも 3 次元空間上でペインティングを行い，実際に 3D モデルを 3D プリンタで出力して現実に再現できるツールである．

このように現在 CG は身近なものになりつつあり，手軽で高品質な表現を実現する研究がされている．

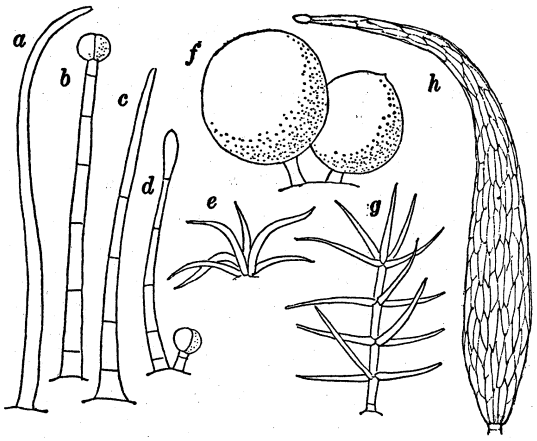
1.2 目的

本研究では自然物の表現の中でも植物表面に生えている細毛に着目し，その表現手法を提案する．植物の表面には細毛と呼ばれる毛状の器官が生えていて，紫外線や外敵から身を守る目的の他にも保湿などさまざまな目的を持っている．細毛は植物ごとに目的や形状が異なり解剖学的にも重要な特徴であるが，それだけではなく植物の見た目の特徴としても大きな役割を持っている．植物は根や茎，葉などの部位から成り立ち，それぞれで細毛は異なる特徴を示す．またひとが手にすることの多い果実についても細毛は重要な特徴のひとつである．南アジア原産のランブータンという果実はマレー語で「毛の生えたもの」を意味していることからも，果実表面の細毛が大きな特徴となっていることが分かる．

しかしながらこの細毛の表現に着目した研究は非常に少なく，代表的な研究は文献 [5] のみである．この研究では L-system（1.3.7 節で詳述）という形式文法を用いて表現を行なっているが，この手法ではユーザ自身がルールを定義して表現を行うために，表現する

対象の正しい知識が必要不可欠である．このため難度が高い．また細毛の分布については確率によって操作しているため，ユーザにとっては直感的ではない．

そこで本研究ではこれらの問題の解決を目指す．先行研究では知識のルール化という難易度の高い作業と分布を確率によって操作していることが問題となっている．前者の問題は，まず細毛の形状を作成し，それを分布させる工程に分解することで解決する．後者の問題は，細毛の分布を構成する際にその粗密を操作可能にすることで解決する．



a:ビワ, b キリ, c ヒメムカシヨモギ, d ヒヨドリジョウゴ,
e アオギリ, f ツルナ, g スズカケノキ, h イヌワラビ

図 1.1: 細毛の例
文献 [16] より図 57 を引用.

1.3 関連研究

1.3.1 細毛

文献 [16] によれば植物の表面には細毛（もしくは毛状突起）と呼ばれる毛状の器官が生えていて，主な役割は害虫から身を守ることや保湿など，植物自身を護ることである．一方で種子を運ぶ目的の散布毛（ワタやヤナギ属）や根から水分を吸収する根毛，粘液を分泌する腺毛（サクラソウ属），消化液を分泌する消化毛（ムシトリスミレ属），他にも先端に大きな細胞を持ち水分を蓄える嚢状毛，毒を持つ刺毛など多種多様な用途を持っている．これらの細毛は植物の表皮細胞が外側へと突き出た器官であることは共通しているが，単細胞のものや多細胞のもの，先に述べたようなさまざまな機能が存在しているため多様な形態を示している．他にも毛の形状を有していて表皮だけではなく維管束などを持つものも存在するが，これと細毛を区別するためにこれを毛状体と呼んでいる．また形状

や機能が細毛と近しいので消化毛（モウセンゴケ属）や刺毛（イラクサ属）と呼ぶものもある．

このように細毛は機能や用途によって形状に違いがあり植物解剖学上重要な要素となっている．ただしそのすべてが解明されているわけではない．ただし，文献 [6] によると細毛に共通する特徴として，表皮細胞は細胞同士が連なることで防護の役割を果たすのとは異なり，気孔や根毛と同様に細毛同士が適度な距離を保つことで機能している．このことから細毛は植物表面を一定の距離を保って分布していると言えるので，領域内を一様にサンプリングすることで細毛の分布を表現することが出来る．つまり本研究で利用する Poisson-disk Sampling（詳細は 2.3.1 節で述べる）で細毛の分布を代用することが可能と言える．

1.3.2 細毛と毛髪の違い

細毛と似た形状で人間の持つ毛髪がある．細毛は前節で述べたように表皮細胞が突起することによって構成される器官であるのに対して，毛髪は皮膚と同じくタンパク質を主成分としていて，皮膚内部の毛根が細胞分裂することによって発生している．

毛髪は毛根にある毛球という部位が，血管から栄養を受け取る事で細胞分裂し，成長している．毛根は皮膚内部に存在しているため，普段目することはなく見た目の特徴にはならない．毛髪の見た目において最も重要なのが図 1.2 で皮膚から外へ伸びている毛幹と呼ばれる部位である．この部位は 3 層構造（図 1.3）になっていて外側からキューティクル，コルテックス，メデュラと呼ばれている．キューティクルは半透明なうろこ状の組織で，硬いタンパク質で構成されていて毛髪の保護を担っている．コルテックスはキューティクルよりも柔らかいタンパク質で髪の水分をコントロールすることで毛髪の柔らかさを生み出している．メデュラは毛髪の中心にあって，蜂の巣状の細胞組織である．必ずしも存在しているわけではなく毛髪の太さに依存していているが，厳密な働きはわかっていない．

毛髪はこれらによって柔らかさやツヤなどが決まり，毛髪の質感に大きな影響を与えている．特にキューティクルはうろこ状の形をしていて，少しずつずれながら重なっているので，このキューティクルが光を反射してツヤが生まれている．キューティクルが傷ついていれば髪はくすみ，なめらかな状態であれば光を反射してツヤが生まれる．また毛髪は，植物全体でたかだか数万本しか細毛が生えないのに対して頭部だけでおよそ 10〜15 万本が生えている．太さも約 0.08mm と非常に細い（文献 [10]）．このため毛髪の表現は細毛に比べ複雑で，毛髪の不透明度 [28] や物理的な動き [21] などの問題がありもっともらしく表現することは難しい．

これに対して植物の細毛は，特別な形状のものも存在するが表皮細胞が変形したもので構造が単純であり，周囲とのインタラクションも少ないので表現が容易と言える．

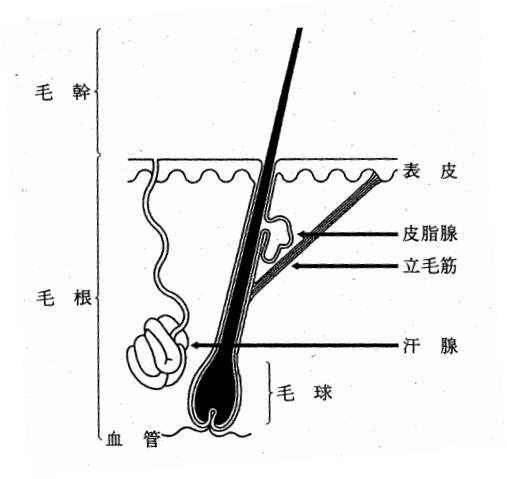


図 1.2: 毛髪の構造
文献 [10] より図 2.1 を引用.

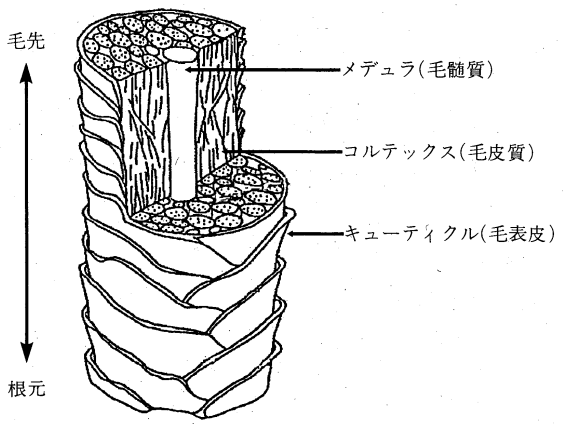


図 1.3: 毛幹の構造
文献 [10] より図 2.5 を引用.

1.3.3 毛髪の表現

文献 [29] によれば，毛髪は見た目の特徴を表す重要な要素ではあるが，計算負荷が高いため毛髪の表現は嫌厭されてきた．しかしハードウェアの発展と共に適切な表現がなされるようになるだろうと述べている．毛髪表現の大きな問題点として複雑な幾何形状やリアルなレンダリングの困難さ，毛髪の細さなどが上げられ，これらの問題を解決するためにさまざまな研究がなされている．

Selle[21] らの研究では 10 万本の毛髪を幾何的にシミュレートすることを目標としている．毛髪の動きはその数と複雑な運動から再現の難しい現象のひとつではあるが，アニメーションや視覚効果などで重要な要素となる場面が多々存在する．文献 [21] 以前の研究ではこういった複雑なシミュレーションには毛髪の大まかな動きに着目して再現していたが，Selle らは細かく入り組んだ毛髪に着目する手法を追求した．先行研究では毛髪を，髪の集まった塊として捉えていたので，自由度が低く複雑な髪型には対応できない．他にも毛髪同士や体への衝突がうまく取り扱えないなどの欠点があり，これらの問題を解決するために Selle らは Mass-Spring Model を導入した．これによって計算時間が改善され，体への毛髪の衝突も正確に再現することが可能になった．Selle らの研究で議論されるような問題は，毛髪が人間の頭部に 10 万本以上生えているために毛髪同士が干渉しあったり，人間が動くことによって衝突することが原因である．細毛についてこの問題を考えると，前者は細毛の密度が毛髪に比べて粗であるため干渉し合う可能性は低い．後者については，細毛は植物のもつ器官なので人間や動物のように激しく運動することはないので問題にはならない．

続いて Yuksel[28] らは毛髪のような半透明な物体の影付けを高速に計算する手法を提案している．Yuksel らが提案する deep opacity maps method は，ピクセル上に分布する半透明レイヤから深度マップを計算して opacity shadow map を得る手法を拡張している．

この手法では少ないレイヤで高い品質の shadow が計算出来る上，ノイズが少ない．加えて，高速でメモリが少なく標準的な PC で利用可能な手法である．細毛でも光の散乱が起き，果実の見た目にも影響を与えている場合が存在する．しかし本研究では細毛の形状や分布に着目しているため，細毛の材質については考慮していない．今後結果のリアリティを高めるためには考えなければならない問題である．

Wei[25] らの研究では，複数視点から撮影した毛髪の画像をもとに幾何形状を再現する手法を提案している．これまでの毛髪をモデリングする手法はユーザが長時間かけて行うものや特別な環境下での計測によるものであったが，この手法では小型のカメラで柔軟に計測が可能である．複数の画像を使うことで毛髪の向きやボリュームを計算し，精度よく幾何形状を再構築することが出来る手法である．この手法はカメラで毛髪の幾何形状を再現するもので，細毛についても同様のアプローチを取ることは出来るが，この手法ではピクセル以上の詳細な情報は取ることが出来ない．毛髪の場合には複数の毛髪が集まって束となり流れを作っている．このことから毛髪の一本一本が識別できなくても，束の流れから毛髪の向きが分かるので再現することが出来る．細毛の場合には細くても毛髪のような束が出来ないので Wei らの手法をそのまま流用することは難しい．

このように毛髪をより正確に表現するために多様な研究がなされている．毛髪に関する問題は，構造の複雑さや本数の多さに起因するものの，各研究ではその特徴を利用して解決手段を導き出しているために，細毛に流用することは難しい．しかしながら，細毛は毛髪に比べて単純な構造をしているために問題として捉えやすく，1.3.1 節で細毛の分布を Poisson-disk Sampling で近似出来るように単純な方法で解決することも可能である．

1.3.4 毛皮の表現

Kajiya らは毛皮が生えているような複雑なサーフェースを持つ物体をレンダリングするための手法 [9] を提案している．毛皮の生えたようなサーフェースではイメージを生成するときにエイリアシングの問題が発生してしまう．Kajiya らはこの問題は，ジオメトリが不適切なスケールで使われているためで，改善するためにはジオメトリで表現するよりもテクスチャを使用する方が良いと考えた．そこで問題解決のために，3 次元空間のテクスチャマップであるテクセルを提案している．これによってライティングモデルは自由に配置可能で，いかなるジオメトリにも制限されずに複雑なサーフェースを表現出来ると述べている．テクセルは現在テクスチャを構成する単位として，画像を構成する単位であるピクセルと対応する概念として非常に重要である．

1.3.5 その他の表現手法

文献 [4] は物体表面にウロコや羽，トゲを表現するための手法である．この文献 [4] ではトゲやウロコを生やすモデルは，陰関数曲面（詳しくは 2.2.2 節で述べる）によって表現されており，モデル自体を変形させることで表現している．文献 [4] の手法では事前に変形させるための操作を用意しておき，陰関数を数学的に変化させて形状を操作している．そのため文献 [4] の手法による変形は曲面がなめらかに接合していて，トゲを生やしたときにもトゲとモデルの間に境界が存在しない．細毛は本来植物の表皮が隆起して出来た器官であるため，植物の表皮となめらかに接合している．本研究の手法では細毛の形状定義と配置を個別に行なっているため，モデルと細毛は接合していない．したがって，より正確さを求めるならばこの手法と同じように，モデルと細毛がなめらかに接合する手続きを取り入れる必要がある．一方でトゲの制御には数学の高度な知識が必要なため，操作が難しいという問題がある．

文献 [11] は文献 [4] と同様に物体の表面にウロコ状の構造を付加する手法である．ウロコは身体を守る目的で身体を覆っていてその数が多いため手作業で表現することは難しい．そこでこの手法ではユーザがストロークを描画するだけでモデル表面にウロコが生成される．ウロコには大きさや分布，向きが存在しているのでこのストロークによってそれらを制御している．またウロコは自己相似形なので，ひとつのウロコをユーザがデザインし，それをもとにモデル上に分布させる．この手法の方針は本手法の方針と近く，対象の 3D モデル上に方向を決めてオブジェクトを配置している．ただし詳細な実装については異なっている．大きく異なる点は，この手法ではウロコを自然に物体上に配置するためにモデルのメッシュ構造を変形させているのに対して，我々の手法では細毛を付加する 3D モデルのメッシュの構造は分布に使うのみで，変形は考慮していない．今後の本研究の発展において参考にすべき研究のひとつと言える．

文献 [20] ではキャンバスに筆で絵を描くように 3DCG を作成することを目的とした研究である．事前に作成した 3D モデルをキャンバスの代わりにし，ペンタブレッドで絵を描くように色を塗ったり，凹凸を編集することが出来る．他にも毛髪や毛皮を描画するためのツールも用意されてはいるが，厳密には毛髪や毛皮を描画するわけではなくストロークを描画空間内にどのように配置するかを設定するのみである．このためこのツールを使ったとしても毛髪や毛皮がすぐに描けるわけではなく，相当の練習が必要だと考えられる．

1.3.6 Modeling Hairy Plants

Fuhrer らは L-system を用いた植物のモデリングにおいて細毛を考慮した表現手法を提案 [5] しており，細毛はさまざまな用途を担っているだけではなく，植物自身の見た目についても重要な要素である（図 1.4 を参照）と述べている．

細毛は生えている部位によってその形状が異なるため，細毛を表現するためには柔軟にパラメータを制御する必要がある．

後方から差す光が細毛によって散乱し葉の周囲に白い輪郭が見える

図 1.4: ユスラウメ
文献 [5] より図 1 を引用

そこで Fuhrer らは，既存の L-system による植物モデリングのフレームワークにパラメタライズした細毛を描画する機能を追加した．細毛は太さと曲がり，ねじれのパラメータを制御することでその形状が表現される．L-system はルールベースの表現手法なので細毛もユーザがルールを定義することで任意の表現を行うことが出来る．図 1.5 に定義されたルールとその結果を示す．この例では植物の茎に相当する部分を描画しながら描画位置を考慮して細毛をモデリングしている．図 1.5（a）の 5 から 8 行目で細毛のパラメータである長さ len と曲がり具合 inc を設定し，同時に細毛の密度を決定するパラメータ den も設定している．5 行目で茎全体の長さに対する描画位置の相対位置を求め，その値から定義されたルール $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ により各パラメータを設定して，10 行目で細毛の描画を行っている．これによって描画された結果が図 1.5（b）の画像である．図 1.5（b）を見るとわかるように画像の左から右にかけて細毛の密度は低くなり，細毛の巻いていた形状はだんだんと直線に近い形状になって長さも短くなっている．このルールでは茎の長さに対する相対位置で細毛のパラメータが変化するため，異なった形状の茎に対してこのルールを設定しても茎の根元から先にかけてだんだんとまっすぐで短い細毛が生えるような結果を得ることが出来る．このようにして事前に細毛のパラメータを決定するルールを定義しておくことで柔軟に細毛を表現することが可能になる．

しかしながらルールが決定的ならばルールから描画結果を一意に定めることは出来るが，描画結果からルールを求めることは困難で，図 1.5 の例でも図 1.5b から図 1.5a のルールを求めることは難しい．つまり任意の描画結果を望む際にその結果を得るためのルールを定めることは困難である．

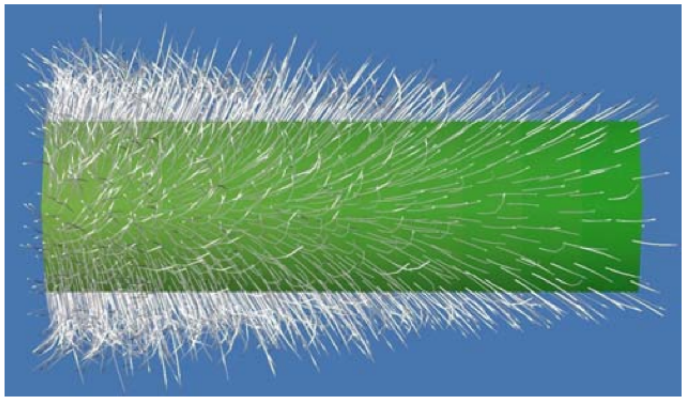
1. #define l 80 /* length of axis */
2. #define $\Delta s1$ /* turtle step */

3. Axiom: @Gs @Hf(1) @Hp(0,1) B(0)

4. B(s): $s \leq l$
5. { $relativeDistance = s/l$;
6. $len = \mathscr{F}_1(relativeDistance)$
7. $inc = \mathscr{F}_2(relativeDistance)$;
8. $den = \mathscr{F}_3(relativeDistance)$;
9. } $\longrightarrow$
10. @Hl(len) @Hi(inc) @Hd(den)
11. @Hp($relativeDistance, 1 - relativeDistance$)
12. f(Δs) @Gc B($s + \Delta s$)

13. B(s): $s \geq l \longrightarrow$ @Ge

(a)



(b)

図 1.5: 細毛のルールとその結果
文献 [5] より引用

1.3.7 L-system

L-system は 1968 年に Aristid Lindenmayer によって考案された形式文法のひとつで植物の成長プロセスや自然物の構造，フラクタル図形などを記述することが出来る．L-system は一般的に文字集合 V と初期状態 ω，および文字の置換規則 P によって定義されるタプル $G = \{V, \omega, P\}$ によって表される．例えば，$V = A, B$，$\omega = A$，$P = (A \rightarrow AB), (B \rightarrow A)$ と定義すると図 1.6 のように変化していく．このとき，文字それぞれをひとつの細胞と考えるとステップがひとつ進むごとに A 細胞は A 細胞と B 細胞に分裂し，B 細胞は A 細胞に変化するというように見なすことが出来，細胞の成長過程を記述していると考えられる．

L-system によって図形を描画する際には文字ひとつひとつが描画に関する命令に相当し，これによってステップを進めるごとに文字が置換されて，割り当てられた命令にしたがって図形が描画される．以下に直角によるコッホ曲線を描画する例を示す．コッホ曲線

$n = 0:$ A
$n = 1:$ AB
$n = 2:$ ABA
$n = 3:$ ABAAB
$n = 4:$ ABAABABA
$n = 5:$ ABAABABAABAAB

図 1.6: L-system の例

はフラクタル図形のひとつで直線を 3 等分し，分割した 2 点を頂点とした正三角形を無限に繰り返す図形である．この例では正三角形ではなく分割した 2 点を頂点とする正方形が繰り返し作図される．コッホ曲線は $V = F$,$\omega = F$, $P = (F \to F + F - F - F + F)$ で定義され，文字 $+$,$-$ は定数で変化しない文字である．この時文字 F は直線を描画する命令で，定数 $+$,$-$ はそれぞれ左と右へ $90°$ 直線の方向を回転させる命令である．図 1.7 に各ス

$n = 0:$ F
$n = 1:$ F+F-F-F+F
$n = 2:$ F+F-F-F+F+F+F-F-F+F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F
$n = 3:$ F+F-F-F+F+F+F-F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F+
F+F-F-F+F+F+F-F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F-
F+F-F-F+F+F+F-F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F-
F+F-F-F+F+F+F-F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F+
F+F-F-F+F+F+F-F-F+F-F-F+F-F+F-F-F+F+F+F-F-F+F

図 1.7: コッホ曲線の例

テップでの文字列の変化を，図 1.8 に $n = 4$ の場合のコッホ曲線を図示した結果を示す．

このように L-system では細胞の成長過程やコッホ曲線のような幾何図形を表現できる他にも植物の成長過程をルール化することで植物を描画することも出来る．図 1.9 に L-system を用いて植物を描画した例を示す．コッホ曲線の例と比べて複雑になるものの茎の伸び方や葉の付き方，形状などをルール化することで同様に L-system を用いて描画することが出来る．

以上のように L-system では自然物の成長過程や幾何形状を描画することが出来るが，既に存在している任意の構造からルールを生成することは難しい．これは表現しようとしている構造について正確に理解している必要があることと，構造からルールを定義しなければならないことに起因する．加えて L-system 上では結果の一部分のみを任意に変形させることが出来ないので，途中経過を確認しながら調整するようなことは出来ない．本研究で着目している細毛に関しては，一度ルール化が行えれば異なるモデル間でもルールを併用出来るもののやはり扱いが難しい．そこで L-system に代わる容易に扱える表現手法が求められる．

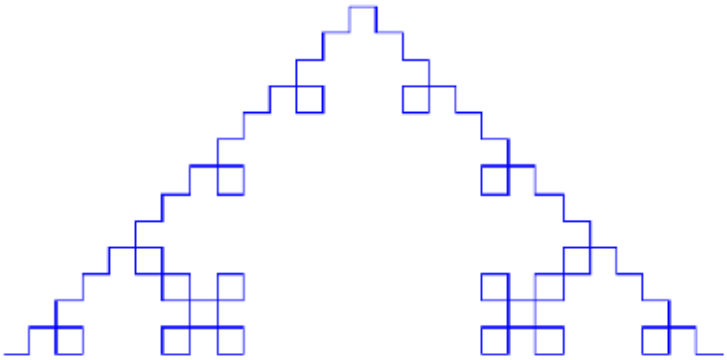


図 1.8: $n = 4$ の時のコッホ曲線

図 1.9: L-system で植物を描画した例
文献 [5] より図 8 を引用

第2章　実現方法

この章では細毛表現を実現するための方法ついて詳述する．2.1 節では細毛を観察して得られた知見について述べ，2.2 節では細毛の形状について，その定義と CG 表現方法の順に述べる．続いて 2.3，2.4 節で細毛の分布と配置方法を説明する．

2.1　細毛の観察

細毛の生え方を理解するために細毛の観察を行った．実際に多種多様な果実を手に入れて観察することは困難であったので，書籍やインターネットによる画像検索でさまざまな果実を検索し観察を行った（図 2.1）．植物の細毛はその用途によってさまざまな形状

ランブータン（Ahoerstemeier による撮影[1]）

オナモミ（Franco Folini による撮影[2]）

図 2.1: 観察した果実の例

をもち，分類上重要な要素である．オナモミのトゲ状の細毛のように特徴的な細毛を持っている果実では，その植物特有の特徴であるため他の植物に応用することは困難である．他にもキウイフルーツのように短く太い毛という以外に特徴がなく，乱雑に生えているものなどが多く果実全体に対して言える細毛の特徴は発見出来なかった．しかしながら観察

[1] この画像はクリエイティブコモンズライセンス CC BY-SA 3.0 および CC BY-SA 1.0 のもと掲載している

[2] この画像はクリエイティブコモンズライセンス CC BY-SA 3.0 および CC BY 2.5 のもと掲載している

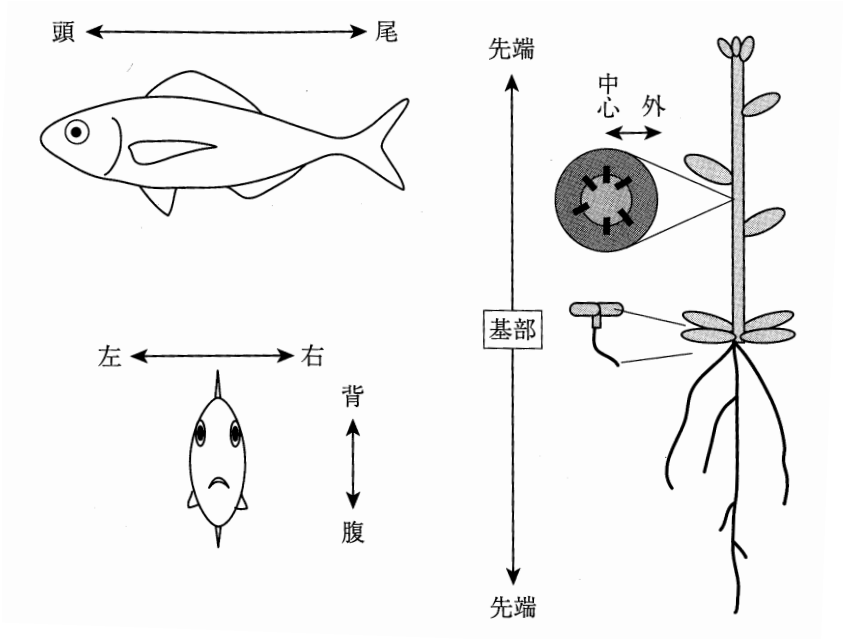


図 2.2: 生物の体の軸
文献 [22] より図 1-3 を引用

した果実は形状はさまざまではあったが，ヘタから果実の先端の方向に軸を設定すると対象な構造を持っていることを発見した．このことは文献 [22] の第 1 章 pp.19-20 を参照すると体の軸（図 2.2）と呼ばれるもので，生物の発生と深い関係がある．動物では前後左右上下に軸を持っていて，これによって各器官がどこに存在するべきか認識することが出来る．植物については上下と内外の軸しか持っていないため，植物は前後や左右を認識することが出来ない．このため前後左右どちらに葉や花が付いているのか分からない．これに従えば果実についてもヘタから先端へ向かう軸の左右を認識出来ない．つまり果実ではこの軸上の位置情報にのみ影響を受けて成長していると言える．このことから果実の細毛もヘタから先端へ向かう軸上の位置が重要で，ヘタから先端へ向かう軸と直交する軸の重みは少ないと考えられる．そこでこの軸に対する位置情報を利用して細毛を操作する手続きを行えば，細毛の表現が可能になる．また細毛の形状は植物によって形状の差が大きいので，細毛は自由に形状を作り果実上に配置することで果実の細毛表現を実現出来る．

2.2 細毛形状

2.2.1 形状の定義

細毛を表現するために形状を定義する．細毛は毛状の器官なので細毛の曲がり方と太さの変化によってその形状を表現することが出来る．図 2.3 に示す細毛の例では，図中の s と t における細毛の太さを比較すると，t における太さが s のものよりも細くなっている．

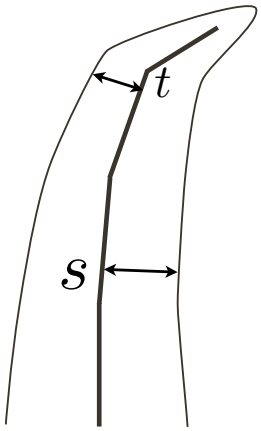


図 2.3: 細毛の例

また細毛は先端に向かうに従って右に曲がっている．この曲がり方と s から t への太さの変化を定義することで細毛が表現できる．

そこで細毛の中心軸を表す曲線と太さの変化を表す太さ関数によって細毛の形状を定義する．

中心軸を表す曲線

中心軸を表す曲線は自然スプライン曲線を用いて表現する．スプライン曲線は与えられた制御点すべてを通りながらなめらかにつながった曲線で，隣り合ったふたつの制御点間の区間ごとに曲線を表す多項式を別々に求めることで曲線を表現する．多項式の次数が 3 次で端点の 2 次導関数を 0 とするスプライン曲線を自然スプライン曲線と呼ぶ．

以下にスプライン曲線を求める解法を述べる．

$$S_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i \quad (0 \leq i \leq n-1) \tag{2.1}$$

スプライン曲線では n 個の制御点 $p_i = (t_i, s_i) \quad 0 \leq i \leq n$ に対して $n-1$ 区間の 3 次式 (2.1) が定められる．変数 t は媒介変数で x，y，z 座標に対してそれぞれ異なった多項式を求めることで 3 次元空間中の曲線が定義される．係数 a_i，b_i，c_i，d_i は未知数であるため，拘束条件を設定しなければならない．以下に自然スプライン曲線の定義を満たすための 5 つの拘束条件を記述する．

1. $S_i(t_i) = s_i$
2. $S_i(t_{i+1}) = S_{i+1}(t_{i+1}) = s_{i+1}$

3. $S_i'(t_{i+1}) = S_{i+1}'(t_{i+1})$

4. $S_i''(t_{i+1}) = S_{i+1}''(t_{i+1})$

5. $S_0''(t_0) = S_{n-1}''(t_n)$

条件 1 は多項式が必ず制御点を通るための条件で区間の開始点を曲線が必ず通ることを保証し，条件 2 は連続した 2 つの区間で共有する制御点を曲線が通る条件である．この条件 1 と 2 によってすべての多項式が表す曲線はその区間を表す制御点 2 つを通ることが保証されるのですべての連続した区間は制御点で接続され，ひとつの連続した曲線を得ることが出来る．条件 3 と 4 は制御点で曲線がなめらかに接続することを保証するための条件で，条件 3 によって接続した 2 つの多項式は制御点でその接線の傾きが等しくなり条件 4 によって曲率が等しくなるので制御点においてふたつの曲線はなめらかに接続される．条件 5 は多項式を一意に定めるための条件である．この 5 つの条件から連立方程式を解くことによって係数が定められ任意の曲線を得ることが出来る．図 2.4 は $p_0 = (-0.75, 0.5), p_1 = (0, 0.25), p_2 = (0.5, 0.5), p_3 = (0.75, 0)$ の制御点からなる 2 次元の自然スプライン曲線である．各制御点で区間がなめらかにつながっていることが見て取れる．

同様に制御点を 3 次元空間で定義することによって 3 次元の自然スプライン曲線を描くことが出来る．現在の実装では，制御点は 3 次元空間上の座標値を直接システムに渡すことで指定している．なので座標を指定することが出来れば外部のモデリングソフトや図形描画ソフトで自然スプライン曲線を描画した後に，その制御点を本システムに渡すことで細毛の中心軸を表現可能である．このように定義された曲線を利用することで細毛の中心軸とする．

太さ関数

次に細毛の太さを表現する方法について述べる．現実の細毛は自然物なのでその形状には凹凸が存在しているが，これを厳密に再現することは難しい．しかしながら，実際には細毛の凹凸を人間の目で捉えることは出来ないので，本研究では簡単化のために細毛の中心軸に対して垂直な断面は常に円となるように細毛の太さを定義する．

細毛の太さは細毛の中心軸からの距離として定める．中心軸が直線の場合に距離が一定ならば円柱，始点から終点に向けて距離を一定の値で減少させれば円錐を表現することが出来き（図 2.5），中心軸からの距離をより複雑な関数によって定義すればより複雑な形状を表現することが出来る．太さを表す関数 F を根元からの軸上の位置 t を入力とし細毛の中心軸からの距離を返す関数として定義する．t は自然スプライン曲線を表現するための媒介変数で n を制御点の個数とすると $0 \leq t \leq n$，太さは負になることはないので関数 F は $0 \leq d$ の範囲の値を返す．

$t = t'$ のとき媒介変数 t' がスプライン曲線の i 番目の区間に含まれるとすると，スプライン曲線の多項式より $S_i(t') = (x_s, y_s, z_s)$ が求まる．この点と，式 2.2 で表せる距離 d' だ

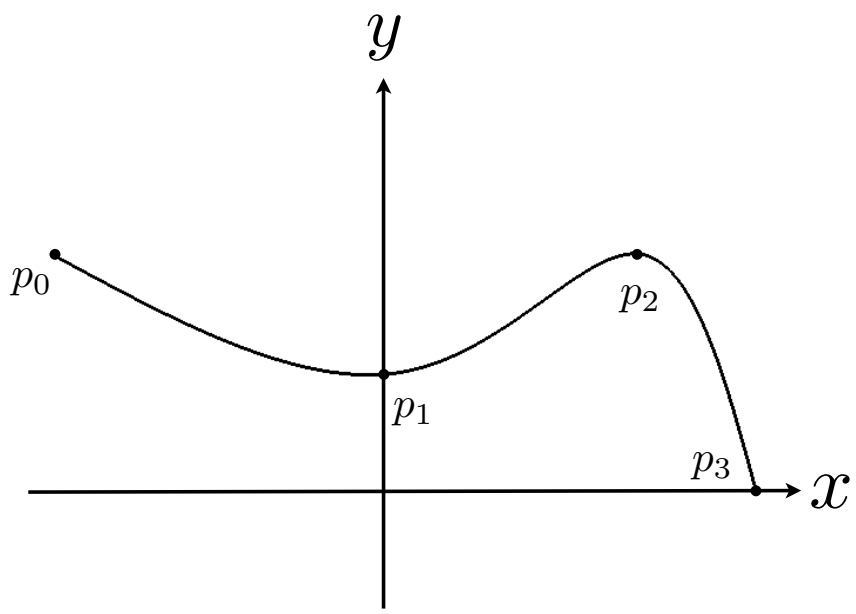


図 2.4: スプライン曲線

け離れた点が細毛の表面上の点となる．

$$(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 = d'^2 \tag{2.2}$$

2.2.2 細毛の CG 表現

つづいて形状定義に従って細毛を CG として表現する方法を考える．スプライン曲線上の任意の点 t を選んだとき式 2.2 と，細毛の太さを返す関数 $F(t)$ から

$$(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 - F(t)^2 = 0 \tag{2.3}$$

を解くことで面を張る座標を求めることが出来る．このような複数の変数の関係を表した式を陰関数と呼び，陰関数によって張られる曲面を陰関数曲面と呼ぶ．

陰関数曲面では 3 次元空間上で，与えられた陰関数を満たす座標群に面を張ることで形状を表現する．式 2.3 は曲面を表すことから，式 2.3 の左辺に 3 次元空間上の座標を与えて計算した結果が負ならば曲面よりも内側，正なら曲面よりも外側と分かる．また異なる陰関数の領域を比較して図 2.6 のような操作をすることが出来る．図 2.6a では負領域の論理和によってふたつの領域が合成されていて，図 2.6b では領域内の値の差を計算することによって円形の穴が開けられている．

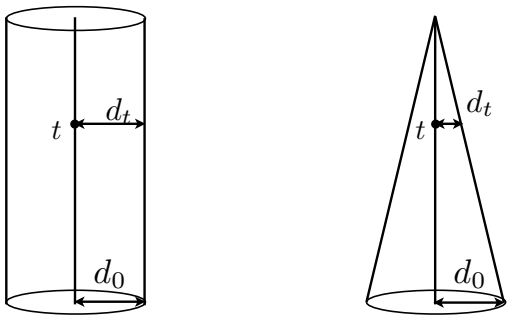


図 2.5: 太さ関数

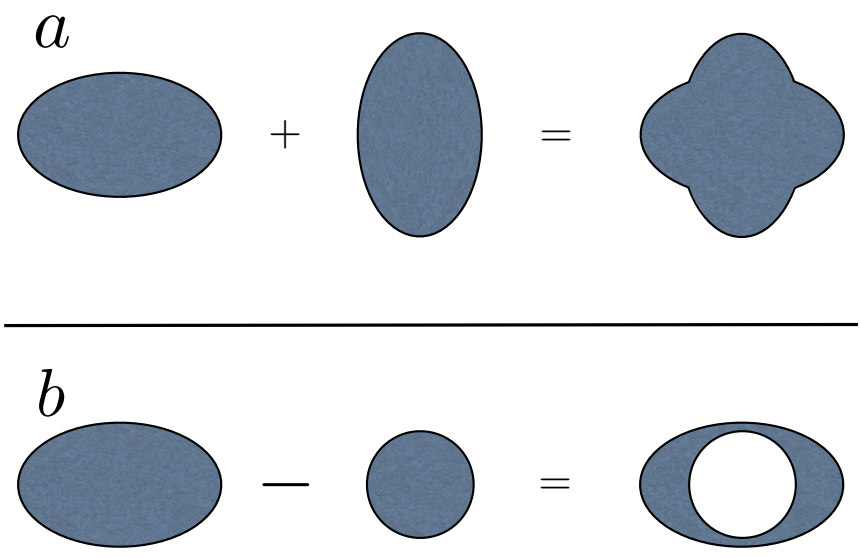


図 2.6: 陰関数曲面の操作

陰関数モデルでは陰関数を設定する必要があるが，太さ関数は式 2.3 のように利用することで陰関数として捉えられ，陰関数によるモデリング手法に流用することが出来る．

陰関数モデリングには *Jules Bloomenthal* の *Polygonizer*[3] を利用する．*Polygonizer* では定義した陰関数に空間上の座標を与えていき，計算結果を評価することによって曲面の内外を判定し曲面を近似する．式 2.3 を陰関数として利用すれば，細毛の内部の座標は太さ関数の返す値よりも中心軸に近い距離にあるので式 2.3 の左辺に代入すると負になり，細毛表面上の座標では 0，細毛外部では正になるので，*Polygonizer* によって細毛の形状を CG 表現することが出来る．

そのために細毛の中心軸と空間上の任意の座標との距離を求める必要がある．自然スプライン曲線と空間上の任意の座標 $\mathbf{q}$ との距離を求めるには，まず任意の座標 $\mathbf{q}$ と最も近い自然スプライン曲線上の座標 $\mathbf{p}$ を求める必要がある．これには参考文献 [2] より，

$$\frac{\partial|\mathbf{p}-\mathbf{q}|}{\partial t}=0 \tag{2.4}$$

を解くことで求めることが出来る．

$$3a\cdot at^5+5a\cdot bt^4+2(2a\cdot c+b\cdot b)t^3+3(a\cdot e+b\cdot c)t^2+(2b\cdot e+c\cdot c)t+ce=0 \quad (2.5)$$
$$where \quad e=d-q$$

式 2.4 の左辺を展開すると式 2.5 になるので，この 5 次式を解くことで媒介変数 t を求めることが出来，自然スプライン曲線の多項式から点 $\mathbf{p}$ が導かれる．係数 $a\sim d$ は各区間の多項式**??**の係数と等しい．すべての区間に対して式 2.5 を解くことで最もふさわしい t を求めることが出来るが，明らかに点 $\mathbf{p}$ が含まれない区間（図 2.7 の頂点 2 と 3 の区間には含まれないことが見て取れる）に対しても式 2.5 を解かなければならない．そこで計算量を削減するために区間の絞り込みを行う．図 2.7 のように任意の点 $\mathbf{q}$ を定めたとき，スプライン曲線上で点 $\mathbf{q}$ と最も近い点を $\mathbf{p}$ とする．このとき，スプライン曲線上で既知である制御点と点 $\mathbf{q}$ の位置関係を調べると，制御点 1 と最も近いことが見て取れる．制御点は最大で 2 つの区間に含まれるので，頂点 $\mathbf{q}$ と最も近い制御点を含む区間に対して式 2.5 を解くことで，すべての区間から点 $\mathbf{p}$ を求めるよりも計算量を減らすことが出来る．

以上により導いた点 $\mathbf{p}$ と点 $\mathbf{q}$ の距離を事前に定義した太さ関数による中心軸からの距離と比較することで *Polygonizer* に必要な内外の判定を行うことが出来る．図 2.8 は領域の内外を判定した後にメッシュを作成する概略図である．曲面が求まった後に曲面が交差している立方体を四面体に分解し，四面体と交差した 3 点からメッシュを作成する．

図 2.9 にモデリングした結果を示す，中心軸は図 2.4 を用いて太さは 2 次関数（2.6）で細毛の根元から先端へ減少するように設定している．

$$F(t)=0.1-0.1\frac{t^2}{(n-1)^2} \tag{2.6}$$

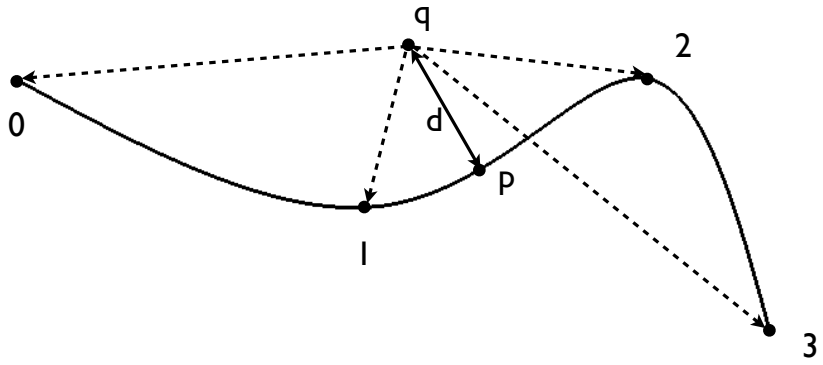


図 2.7: 自然スプライン曲線との距離

2.3 細毛分布

細毛の形状を作成した後には，実際にモデル表面に細毛を配置する必要がある．細毛の分布は先行研究 [5] より Poisson-disk Sampling で近似出来ることから，入力モデル表面に Poisson-disk Sampling に基づいて生成した頂点群を利用することで細毛の位置を決定する．

2.3.1 Poisson-disk Sampling

Poisson-disk Sampling は任意の領域内をサンプリングする手法のひとつで，領域内を一様にサンプリングすることが出来る．

Poisson-disk Sampling ではサンプリングを行う際に座標の周囲に排他領域（図 2.10 の円で囲まれた領域）を設定し，新たに選ばれた任意の座標 p の排他領域内に他の頂点がなければ p を採用し排他領域内にすでに採用された他の頂点があれば p を棄却する（図 2.10b）．これによって任意の領域内を一様にサンプリングすることが出来る．

しかしながら候補点を無作為に選び出すようなナイーブな方法によってこのサンプリングを行うと領域内がある程度サンプリングされた後には，排他領域内の座標が選び出されることが多くなるため候補点が棄却されることが増えてしまい，計算時間が増加してしまう．そこで本研究では *Xiang*[26] らの手法を用いることで計算時間の削減を行った．この手法ではまずモデル上に密な初期座標群を作成し，その座標群からマルチスレッドで候補

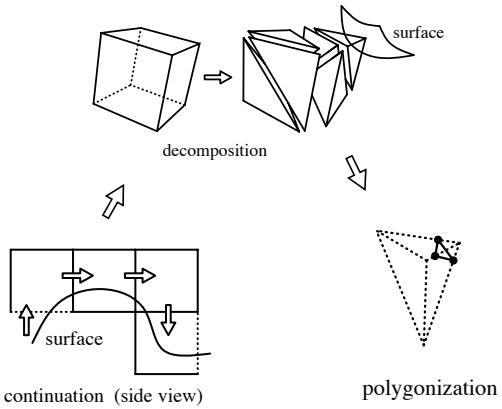


図 2.8: Polygonization
文献 [3] 図 1 を引用

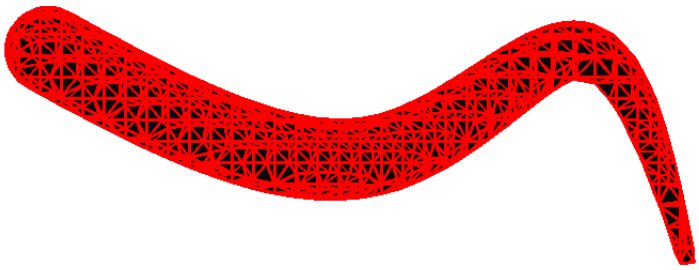


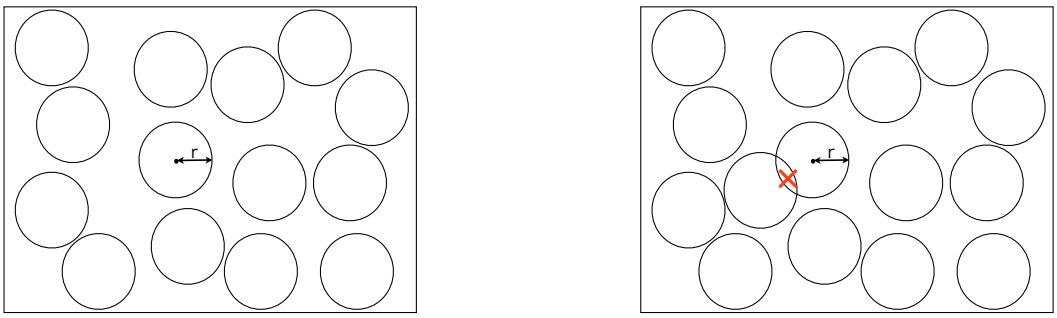
図 2.9: 細毛のモデリング結果

点を選出し排他領域の判定を行いサンプリングする手法である．以下に続く節で，初期座標群の生成法とサンプリング法について述べる．

2.3.2 初期座標群の生成

サンプリングを行う前処理としてモデル表面に初期座標群の生成を行う，サンプリングは生成した初期座標群から無作為に座標を選び出す．3D モデルはメッシュで表現されるので各メッシュが表す平面上に十分な数の頂点を無作為に生成することで初期座標群を生成することが出来る．

初期座標群の生成は文献 [17] の手法を参考に行った．文献 [17] ではすべてのメッシュが三角化されている．m_i は i 番目のメッシュで a_i はメッシュの面積，S_i はそれまで訪問したメッシュの面積の和とすると，図 2.11 のような配列を作成することが出来る．次に 0 か



(a) (b)

図 2.10: Poisson-disk Sampling

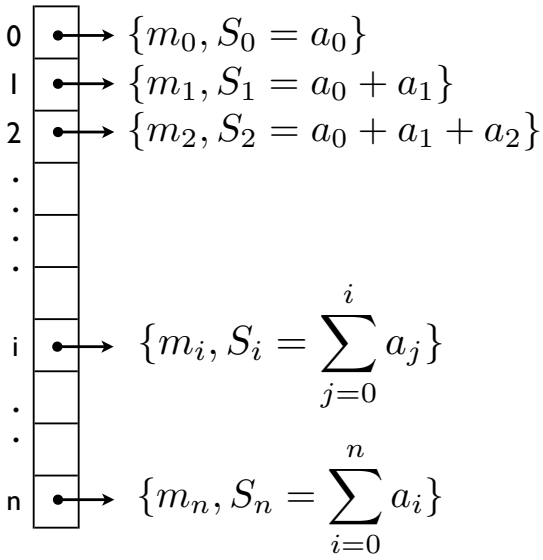


図 2.11: Osada らの手法

ら S_n の範囲で無作為に数をひとつ選び，配列を二分探索することでメッシュをひとつ選び出す．これによって選ばれたメッシュから式 2.7 に従って座標を生成する．r_1，r_2 は 0 から 1 の乱数，A，B，C は三角形を構成する頂点である．

$$P = (1 - \sqrt{r_1})A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C \tag{2.7}$$

この作業を繰り返せば確率的に面積の大きなメッシュには多くの座標が生成され，面積の小さいメッシュには比較的少ない数の座標が生成される．Osada らの行った実験によれば，モデルの持つ頂点数が 64 個だった場合に 1024^2 個の座標を生成すれば形状を再構成するのに十分な密度だと述べている．形状の再構成に十分な密度ということは，メッシュ上で座標の生成されていない領域が小さいので密と言える．

この手法を用いてサンプリングに用いる密で偏りのない座標群を生成する．

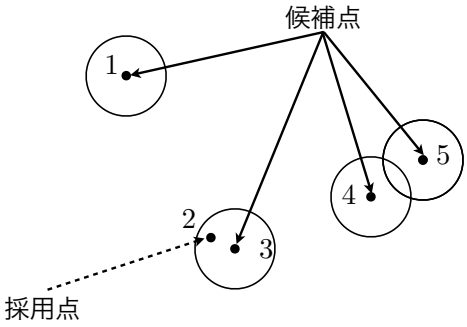


図 2.12: サンプリングの例

2.3.3 サンプリング

Xiang[26] らの手法を参考にして，2.3.2 節で生成した初期座標群を使用してサンプリングを行う．

以下の処理を CPU のマルチスレッドによって，並列処理する．

まず初期座標群から無作為に候補点 p_i を選び，取り除く．この時に一意に定まるような優先度を設定する．

$$pripority(p_i) = \frac{rand() \times T + i}{RAND\_MAX \times T} \tag{2.8}$$

式 2.8 は優先度を計算する式で，$rand()$ は乱数を返す関数，$RAND\_MAX$ は乱数の最大値，T はスレッドの総数，i はスレッドの番号である．次に選んだ頂点に対して Poisson-disk Sampling と同様に排他領域の計算を行う．マルチスレッドで処理を行うので排他領域内に他の候補点が存在する場合があり，その場合にはどちらの頂点を棄却するか式 2.8 によって設定した優先度に従って決定する．候補点が棄却されなかった場合には初期座標群から候補点の排他領域内に存在するすべての座標を削除する．このことでサンプリングされた頂点の排他領域内に頂点が存在しなくなるため，排他領域内の座標が候補にならない．よって排他領域内の明らかに棄却される座標が候補にならなくなるため，ナイーブな方法よりも計算量を削減することが出来る．この一連の動作を初期座標群からすべての座標が取り除かれるまで繰り返す．図 2.12 の例では候補点 1 は排他領域内に他の頂点がないので採用されるが，候補点 3 は排他領域内にすでに採用された頂点 2 が存在しているため棄却される．右のふたつの候補点 4，5 はお互いに排他領域が衝突しているので，お互いの優先度を比較して優先度の低い候補点が棄却される．このようにして領域内を一様にサンプリングすることが可能になっている．

以上の方法によって高速な Poisson-disk Sampling を実現し，モデル表面上に頂点群を生成することが出来る．図 2.13 はおよそ 2 万個の頂点をサンプリングした結果である．同

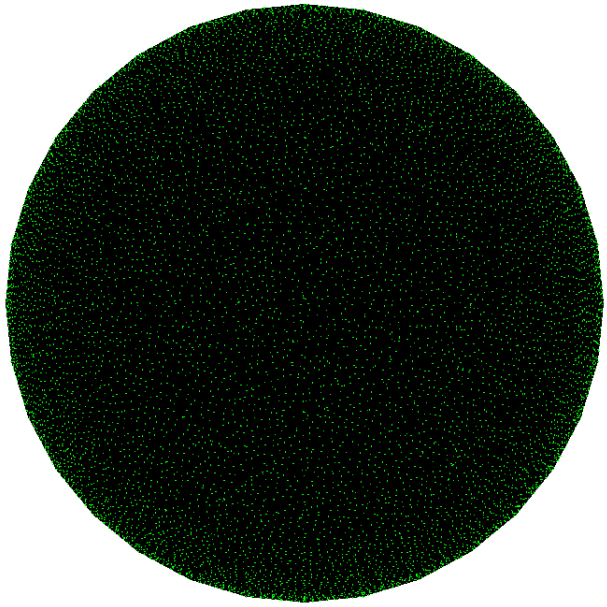


図 2.13: サンプリング結果

じ実行環境でナイーブな方法でおよそ 2 万頂点のサンプリングを行ったときには 10 分ほどの時間がかかったのに対して，この手法では 3 分ほどで結果が得られた．

サンプリング密度の操作

本来 Poisson-disk Sampling は任意の領域に対して一様なサンプリング結果を返す手法である．しかし本研究では果実表面に生えている細毛の分布に対して利用するため必ずしも一様な分布が最適とは限らず，密度の差が生じる場合も存在する．

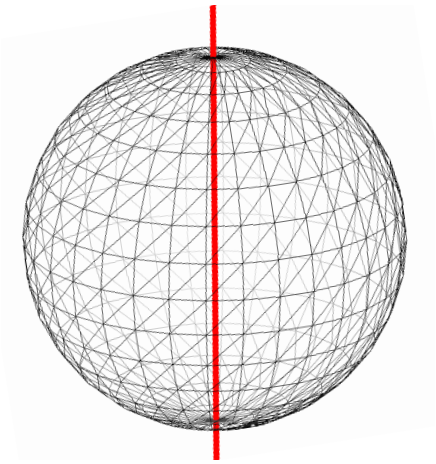
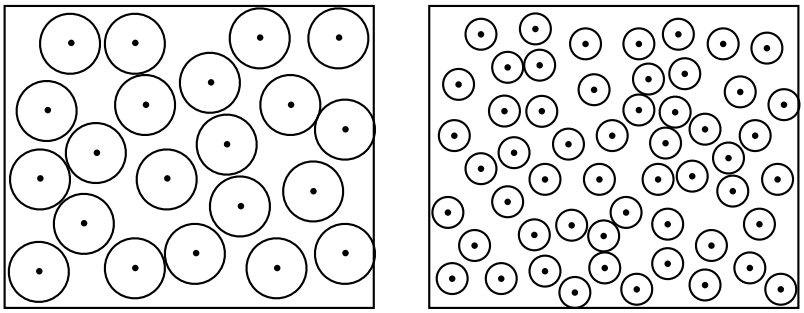


図 2.14: 果実の軸



$n = 21$ $n = 56$

図 2.15: 大きさの異なる排他領域の結果

そこで Poisson-disk Sampling を改良し，密度を操作可能にすることで分布の偏りを表現する．Poisson-disk Sampling の詳細は 2.3.1 節ですでに述べてある通り，座標を中心とする排他領域を設定し，排他領域の衝突を検査することで一様な分布を実現している．Poisson-disk Sampling の密度はこの排他領域の大きさに依存していて，排他領域が小さければ密度は高くなり排他領域が大きければ密度は低くなる（図 2.15）．このことから排他領域を変化させながらサンプリングすることで，密度を操作することが出来る．

図 2.16 は排他領域の大きさを変化させたときの様子を示している．この例では y 軸の正の方向へ移動するにつれて排他領域が大きくなっていて，だんだんと頂点間の間隔が広くなり密度が下がっていることが分かる．

本研究では図 2.14 のような果実のヘタから先端にかけて直線を軸として捉えて，この軸上での位置から排他領域の操作を行う．Xiang らのサンプリング手法で候補点を選んだ際に候補点が軸上のどの位置にあるか計算出来るので，この軸上で排他領域の大きさを変化させるように値を設定すれば（例えば排他領域の半径），候補点の位置から排他領域の大きさを補間して決定することが出来る．図 2.17 の例では，ヘタ付近で排他領域の大きさを 1，先端付近では 2 と設定している．選ばれた候補点の位置が軸上のちょうど中間地点にあった場合，内分の比から排他領域の大きさが 1.5 と定まる．図 2.17 の例ではヘタと先端の 2ヶ所のみに値を設定しているので，その間は線形に増加するのみであるが，より細かく値を設定すればより細かい粗密の変化を表現することが出来る．以上の方法で排他領域の大きさを操作することが出来るので，細毛の粗密を操作することが可能になる．図 2.18 は排他領域の大きさを操作してサンプリング結果に密度の差を生じさせた結果である．この場合ではヘタ付近では排他領域の半径を 0.1，先端付近では 3 に設定している．

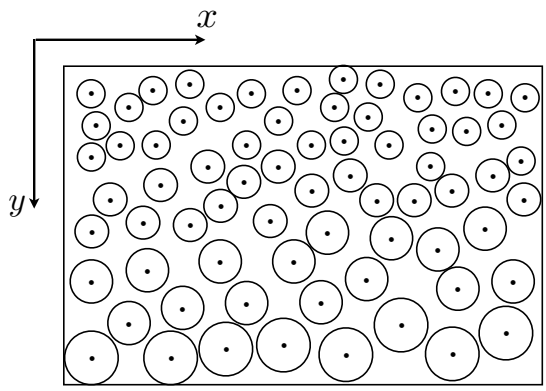


図 2.16: 排他領域を変化させた様子

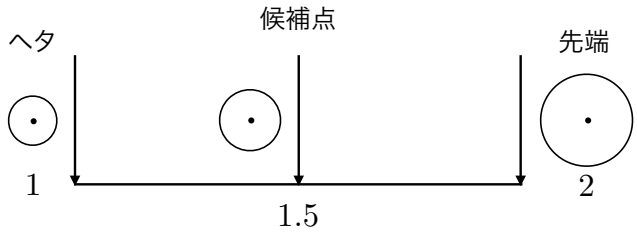


図 2.17: 排他領域の補間

このことによってヘタ付近では排他領域が小さくなるのでサンプリング結果が密に，逆に先端付近では排他領域が大きくなるのでサンプリング結果が粗になっていることが見て取れる．

このようにしてモデルの表面上に生成した頂点群に対して細毛を生やすことによって細毛表現を実現する．

2.4 細毛の配置

2.3 節で述べた方法によって細毛を生やすためのモデル上の座標群を得ることが出来，この座標群に対して 2.2 節で定義した細毛を実際に配置することで細毛を表現する．このとき細毛を配置する座標以外に細毛の生える向きが重要である．細毛は植物の表皮から外に向かって生えているので細毛を配置する際にも当然モデルの外へ向かって配置する必要がある．そこで細毛に相応しい向きを決めるためにモデル表面の法線ベクトルを求める．法線ベクトルは式 2.9 で定義されるベクトルで，$\vec{a}$ と $\vec{b}$ が張る面に対して垂直なベクトルである（図 2.19）．

$$a \times b = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x) \tag{2.9}$$

その向きは外積の計算する順序によって決定され式 2.9 の場合には $\vec{a}$ を始点にして $\vec{b}$ へ回転して進む右ねじの方向である．計算順序が逆のときには $\vec{b}$ から $\vec{a}$ へ回転しながら進む右ねじの方向になるので先の場合とは逆方向になる．このように法線ベクトルは任意の面に対して垂直なベクトルであるので，モデル表面の法線ベクトルを求めることでモデル表面から外側に向かうベクトルを求めることが出来るが，計算順序を誤るとモデルの外側ではなく内側を向いてしまう．しかしながら，モデルのメッシュ構造を考慮すればモデルの外側へ向かう法線ベクトルを求めることが出来る．図 2.20 のようにメッシュは 3 つの

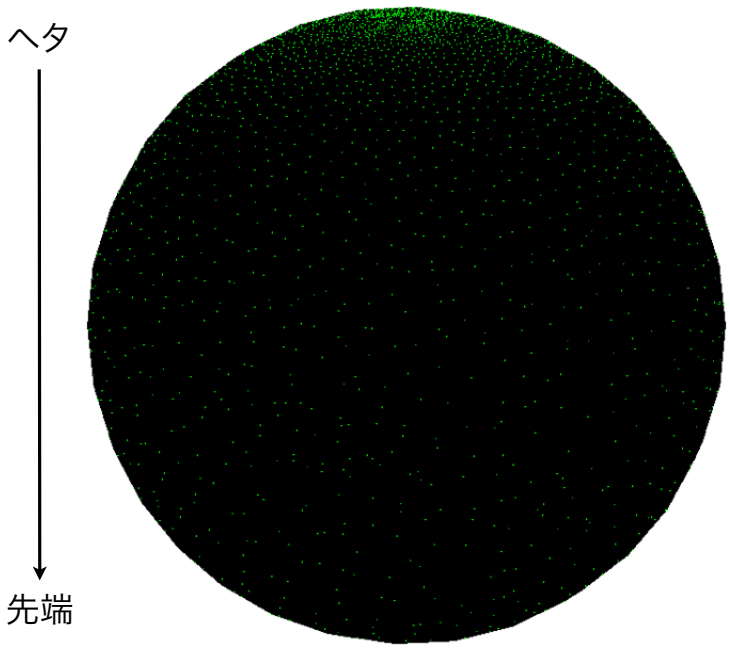


図 2.18: 粗密のある分布

頂点とそれらがなす三角形によって表現されている．一般的に頂点は v_0 を始点として反時計回り，もしくは時計回りに頂点が格納されている．頂点の順番が分かるので外積の計算順序を定めることが出来る．例えば図 2.20 のように頂点が反時計回りに格納されている場合には $\overrightarrow{v_0v_1} \times \overrightarrow{v_0v_2}$ のように，任意の頂点と次の頂点が成すベクトルが先に来るように外積を計算することで，モデルの外側へ向かう法線ベクトルを得ることが出来る．

続いてこの法線ベクトルをもとに細毛を配置する方法について述べる．法線ベクトルはすでに述べてあるようにモデル表面に対して垂直な外へ向かうベクトルである．一方で細毛もまた図 2.21 のように細毛の根元から毛先へと向かう方向ベクトルを定義することが出来る．この細毛の方向ベクトルと法線ベクトルを一致させることで細毛の方向を定め細毛をモデル表面に配置することが出来る．これを実現するために細毛の方向ベクトル $\overrightarrow{d}$ を回転させて法線ベクトル $\overrightarrow{n}$ と一致させることを考える．これにはふたつのベクトルのなす角度と細毛の方向ベクトルを回転させるための回転軸が必要になる．まず前者についてはベクトル $\overrightarrow{d}$ とベクトル $\overrightarrow{n}$ はそれぞれのベクトルは既知であるのでベクトルの内積からふたつのベクトルのなす角度を求めることが出来る．ベクトル $\overrightarrow{a}$，$\overrightarrow{b}$ の内積は式 2.10 で表され，これを θ について解けば逆三角関数からふたつのベクトルのなす角度が得られる．

$$\overrightarrow{a} \cdot \overrightarrow{b} = |\overrightarrow{a}||\overrightarrow{b}| \cos\theta \tag{2.10}$$

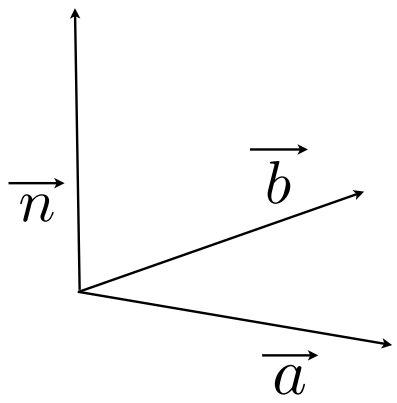


図 2.19: 法線ベクトル

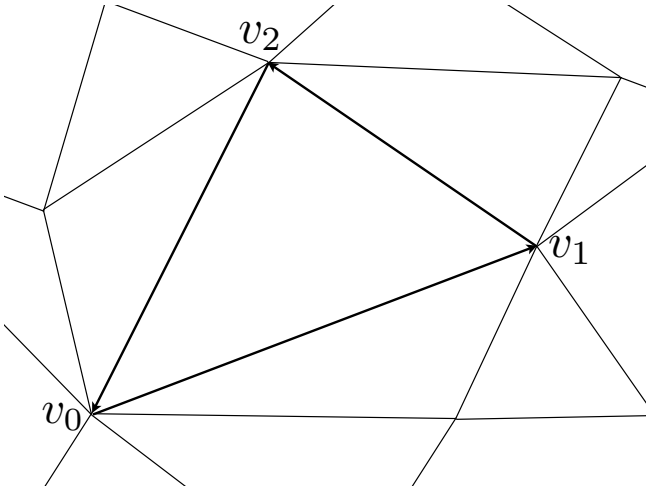


図 2.20: メッシュの構造

$$\cos\theta = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{|\overrightarrow{a}||\overrightarrow{b}|}$$

$$\theta = \arccos(\frac{\overrightarrow{a} \cdot \overrightarrow{b}}{|\overrightarrow{a}||\overrightarrow{b}|})(0 \leq \theta \leq \pi) \tag{2.11}$$

(2.12)

この角度だけ細毛の方向ベクトル $\overrightarrow{d}$ を回転されば法線ベクトル $\overrightarrow{n}$ と一致する．次にベクトルを回転させるための回転軸について述べる．3 次元空間では x，y，z 軸によって空間を表現していてそれぞれの軸を順番に任意の角度だけ回転させることで等しい回転を再現することが可能ではあるが，それぞれの軸について順番に回転させる必要がありまた回転させることによって複数の軸が同軸となってしまい自由度が失われるジンバルロックと呼ばれる減少が存在する．そこで本研究ではそれらの問題を避けるために空間中に任意

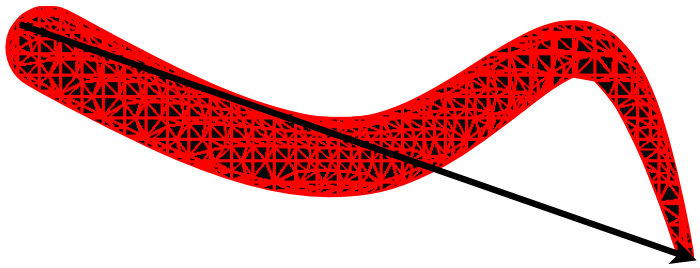


図 2.21: 細毛の方向ベクトル

の回転軸を設定してその軸について回転させることによって，空間中の回転を表現する．行列 2.13 は任意軸の回転を表す行列として知られている．$\vec{v} = (x_v, y_v, z_v)^t$ は回転軸を表すベクトルで $\vec{v}$ を回転軸として θ 回転させることを表している．回転方向は回転軸の指す方向に対して右ねじの方向である．

$$\begin{pmatrix} {x_v}^2(1-\cos\theta)+\cos\theta & x_v y_v(1-\cos\theta)+z_v\sin\theta & x_v z_v(1-\cos\theta)-y_v\sin\theta & 0 \\ x_v y_v(1-\cos\theta)-z_v\sin\theta & {y_v}^2(1-\cos\theta)+\cos\theta & y_v z_v(1-\cos\theta)+x_v\sin\theta & 0 \\ x_v z_v(1-\cos\theta)+y_v\sin\theta & y_v z_v(1-\cos\theta)-x_v\sin\theta & {z_v}^2(1-\cos\theta)+\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.13)$$

回転角 θ はふたつのベクトルの内積によって得られるので，回転軸 $\vec{v}$ が定まれば $\vec{d}$ を θ だけ回転させて $\vec{n}$ と一致させることが出来る．$\vec{d}$ を回転させて $\vec{n}$ に一致させることからベクトルが回転することによる軌跡は $\vec{d}$ と $\vec{n}$ が張る平面の一部であることが分かる．つまり $\vec{d}$ は $\vec{d}$ と $\vec{n}$ の張る平面内を移動するので，この平面に対して垂直なベクトルを回転軸とすれば良い．平面に対して垂直なベクトルは平面内の異なるふたつのベクトルから外積を計算することで求められる．ただし外積は計算順序によってベクトルの向きが異なるため正しい計算順で無ければ誤った回転軸となってしまうが，回転は回転軸の進行方向に対して反時計回りに行われるので $\vec{d} \times \vec{n}$ の順序で計算すれば，正しい向きのベクトルが得られる．このとき正しい向きとは，$\vec{d}$ を $\vec{n}$ に重ねるようように回転させて進むネジと同じ方向へ向かう望むベクトル（図 2.22）である．このベクトルと回転角 θ によって回転行列（2.13）が定まり，$\vec{d}$ を $\vec{n}$ に一致させることが出来る．

以上の方法によりモデルの表面上に正しい方向を向いた細毛を配置することが出来る．図 2.23 はモデルの各メッシュ上に法線方向を向いた細毛を配置した結果である．各細毛の座標はメッシュの重心を計算することで決定している．

行列 2.13 を利用した回転の操作は回転角度と回転軸を定めることが出来れば容易に計算出来る．これを利用することで細毛を配置する方法を上述したが同様の方法によって細毛の曲がる方向や

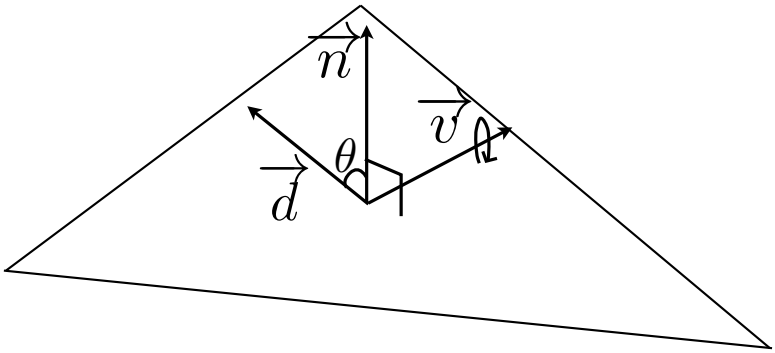


図 2.22: 細毛の方向ベクトルと法線ベクトル

根元の角度も操作することが可能である．回転操作をする際に回転角や回転軸をユーザによって任意の値を与えることによって制御することは可能であるが，すべての細毛について細かく値を設定することは困難である．しかしながら細毛の曲がる方向や向きは，果実のヘタから果実の先を通るような軸（図 2.14）に従いながら生えているので，細毛の位置情報を利用して機械的に制御することが可能である．例えば，細毛がヘタ付近では法線方向に生えていて先端に近づくにつれて細毛が果実に沿うような場合には，ヘタと先端で傾ける角度を設定してその間を補完することで表現出来る．傾ける角度は図 2.24 にあるように法線ベクトルとの成す角度で表し，図では画像の下方向を正の向きとしている．上方向に傾けた場合には負の角度を設定することで表現可能である．また角度が大きすぎる場合には細毛がモデルの中を貫通してしまう．これを避けるために，細毛に最大限角度をつけたとしても面に沿うだけなので角度を $-90^\circ \leq \theta \leq 90^\circ$ の範囲として，それを超える場合には $\pm 90^\circ$ を割り当てる．

図 2.25 の例ではヘタ付近では傾ける角度を 0°（法線方向），先端付近では 45° に設定していてその間の角度はヘタからの距離に応じた比率で角度を計算している．これによってヘタから先端に近づくにつれてだんだんと細毛が果実に沿うように生えていることが分かる．

真っ直ぐな細毛の場合には対称な形状になるが細毛が曲がっている場合には，その曲がる方向が法線ベクトルを回転軸として自由度を持つ．図 3.14 のように，それぞれの細毛が無作為な方向へ曲がっている場合には制御する必要はないが，細毛の曲がる方向が統制されている場合には制御する必要がある．しかしながら現在の実装では実現が出来ていないため今後の課題といえる．

このように細毛が果実に対して生える性質を利用することで細毛の形状と合わせてさまざまな細毛を表現することが可能である．

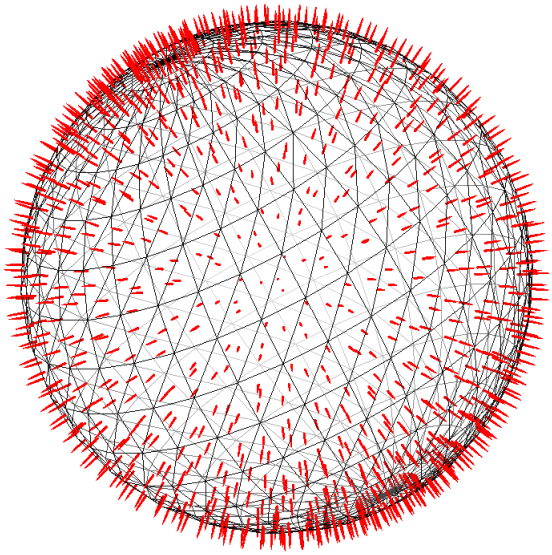


図 2.23: 面法線を細毛によって可視化した様子

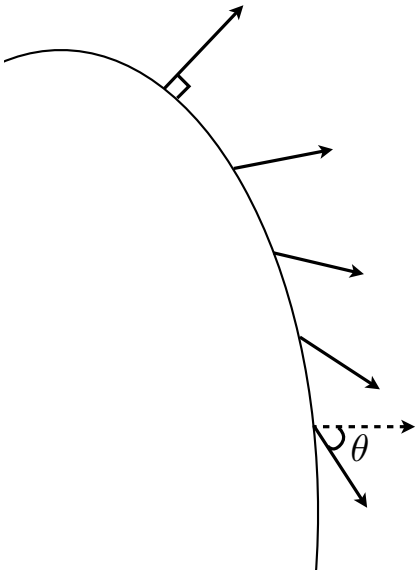
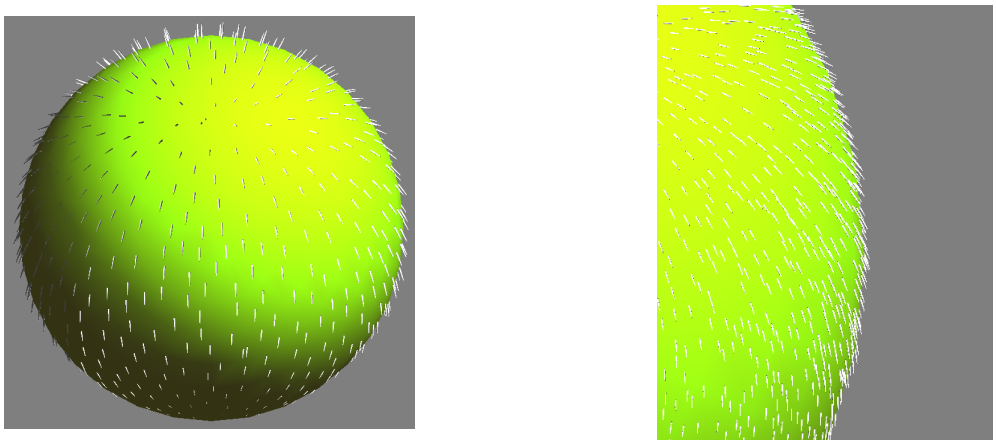


図 2.24: 角度の付け方



(a) (b)

図 2.25: 細毛に角度をつけた結果

第3章　結果

本研究で考案した手法によって作成した結果についての述べ，考察する．

本手法では 1. 細毛の定義，2. 分布の作成，3. 細毛の配置という工程で果実における細毛を表現した．

まず細毛の定義では細毛の中心軸を表す曲線と細毛の太さを決める太さ関数を定めることで細毛を定義することが出来る．中心軸は自然スプライン曲線を用い，中心軸上の位置を入力とする太さ関数によって図 2.9 のような形状だけではなく，図 3.1 から図 3.6 のような形状を作成することが出来る．自然スプライン曲線は 2 つ以上の制御点を与えることで制御点間をなめらかにつなぐ曲線を作成出来るので任意の中心軸を作成可能である．図 3.3 や図 3.4 のような曲がった細毛も表現できる．また太さ関数は中心軸上の位置に対して細毛の半径を返す関数で，中心軸の根元から先端方向に対して減少するような関数を設定すればだんだんと細くなる細毛（図 2.9）を作れるだけではなく，定数を返せば円筒に近い形状（図 3.5）を作ることが出来る．更に中心軸上の位置によって関数を切り替えるように設定すると，図 3.5 と図 3.6 は同じ中心軸を使用しているが，まったく異なる形状を作ることも可能である．中心軸と太さ関数のふたつを自由に設定することで，任意の形状の細毛を定義することが出来る．

次に分布の作成ではモデル表面上に細毛を配置する頂点群を生成する．細毛の分布は Poisson-disk Sampling で近似できることが知られている．したがってモデル表面上で Poisson-disk Sampling を行うことで細毛を分布させる頂点群（図 2.13）を生成することが出来る．しかし密度の差を表現したい場合には不十分である．そこでサンプリングを行う際に用いる排他領域（図 2.10）の大きさを制御することによって分布の粗密を表現する．ヘタから先端へ向かう軸上で排他領域の大きさを設定する．排他領域の判定を行う際に軸上の位置から排他領域の大きさを求めることによって分布の粗密を実現している．このようにしてモデル表面に一様なだけではなく，粗密に差があるような分布を作成することが出来る．

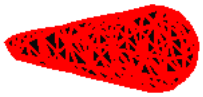
最後に細毛の配置では，1. で作成した細毛を 2. で生成した座標群に配置することでモデル表面上に細毛を生やしている．座標だけでは細毛を生やす方向が定まらないので，図 2.21 にあるような細毛の方向ベクトルと配置させる座標での法線の外積を計算することで回転軸を求めて，細毛の方向ベクトルと法線ベクトルが成す角度だけ細毛を回転させることで細毛が外側へ向くように配置している．同様の考え方で法線ベクトルと図 2.14 の外積による回転軸で細毛を回転させることで，細毛を任意の角度傾けることが出来る．傾ける角度を手作業ですべての細毛に割り当てることは困難なので，分布の粗密を制御する方法と同様に軸上で角度を割り当てることで実現している（図 2.25）．

図 3.11 では表面に細くて短く，まっすぐな細毛を生やした結果で桃に近い見た目になっている．図 3.12 は円錐形の細毛を定義して分布させることで，表面にトゲが生えた状態を表現している．図 3.13 では手前に来るほど密度が高くなっていて，モデル表面に沿った細毛が生えている．手前側は細毛の密度が非常に高く，密集した様子がよく分かる．図 3.14 では先が曲がった細毛をモデル表面上に生やしている．図 3.16，3.17 では球面以外のモデルに細毛を生やしている．図 3.17 でも，中央付近の細毛の密度が高く両端では低くなっているように粗密の制御が実現出来ている．

以上の工程によってさまざまな細毛表現を実現することが可能となっている．

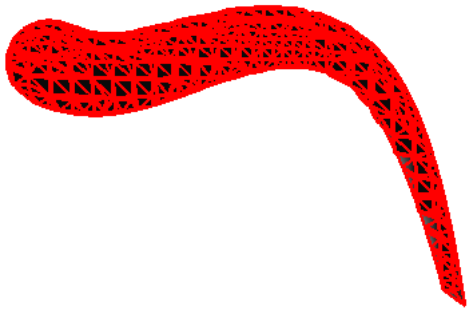
太さ関数：d(t)=$0.1 - 0.1\frac{t^2}{(num-1)^2}$

図 3.1: 細毛の結果 1



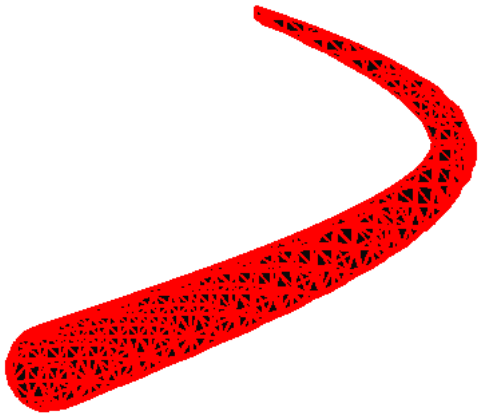
太さ関数：$d(t) = 0.1 * t$

図 3.2: 細毛の結果 2



太さ関数：$d(t) = 0.1 - 0.1\frac{t^2}{(num-1)^2}$

図 3.3: 細毛の結果 3

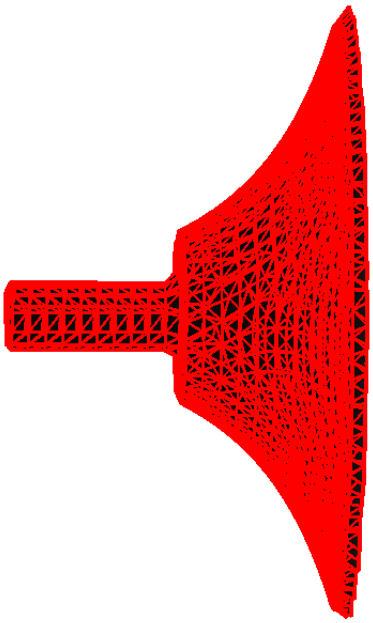


太さ関数：$d(t) = 0.1 - 0.1\frac{t^2}{(num-1)^2}$

図 3.4: 細毛の結果 4

太さ関数：$d(t) = 0.1$

図 3.5: 細毛の結果 5



太さ関数：
$if(t < 0.5)d(t) = 0.1\ else\ d(t) = 0.1 + t^2$

図 3.6: 細毛の結果 6

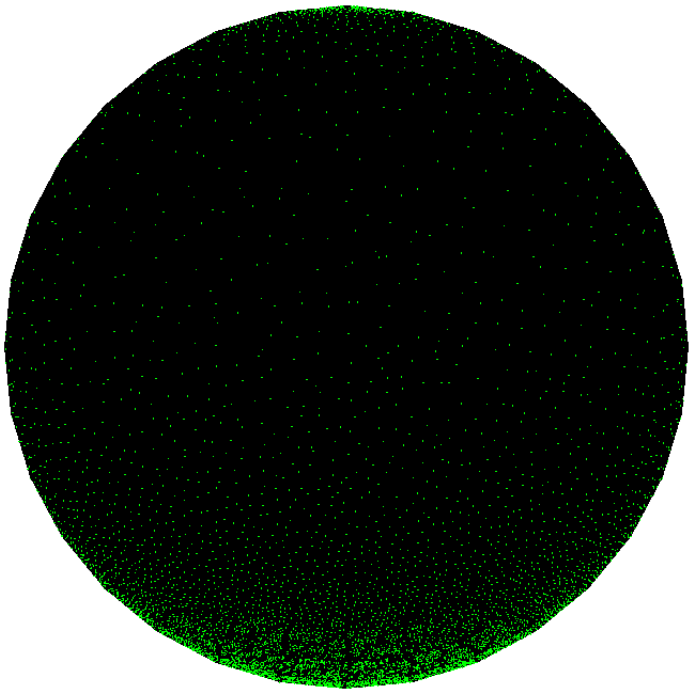


図 3.7: 粗密を操作した結果 2

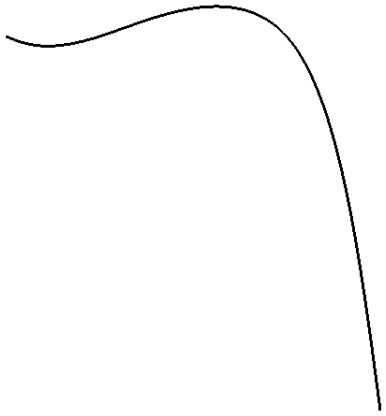
植物の細毛を表現する手法として Fuhrer らの手法 [5] がある．本手法と Fuhrer らの手法を比較すると，Fuhrer らの手法は L-system を用いた表現手法で，ルールさえ設定することが出来ればどのような表現も可能であることに加えて，一度ルールを設定すればひとつのルールを繰り返し利用することが出来るので汎用性が高い．しかしながらこのルールの設定は表現しようとしている対象の知識が重要で，正しい知識がなければ表現が難しい．さらに細毛の分布については，先に Poisson-disk Sampling を行なって生成した座標群から確率関数に従って分布を決定しているため，どのような結果になるのか想像しづらい．

これに対して本手法では，1. 細毛の形状定義，2. 分布の作成，3. 細毛の配置と工程を分けていることで Fuhrer らの手法のようにルールを設定する必要がない．またルールの作成という行為は一般的に行われることではなく難度が高いが，本手法のように部品を作成してから部品を配置する工程は CG 制作において日常的に行われている工程である．そのため Fuhrer らの手法と比べて使用の難度が低いといえる．また細毛の分布については，本手法では Poisson-disk Sampling の排他領域を直接操作して分布の粗密を制御するため，Fuhrer らの手法よりも直感的であるといえる．

以上のことから本手法では Fuhrer らの手法で問題であったルールを定義する困難さと直感的ではない確率関数による分布を，前者については工程を分解して作業をわかりやすくすることで解決し，直感的ではない分布の作成を直接排他領域の大きさを操作することで粗密の制御を実現している．

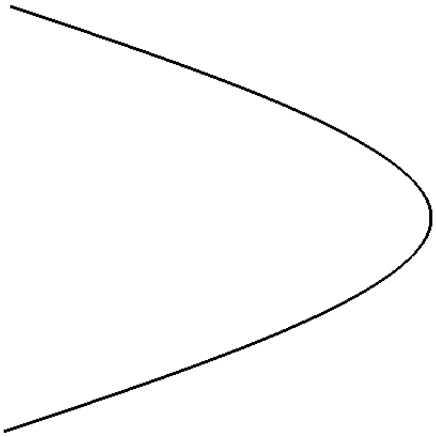
(-0.5,0,0)，(0,0,0)，(0.5,0,0)

図 3.8: 中心軸 1



(-0.25,0.5,0)，(0,0.5,0)，(0.5,0.5,0)，
(0.75,-0.5,0)

図 3.9: 中心軸 2



(-0.5,-0.5,0)，(0,0,0)，(-0.5,0.5,0)，

図 3.10: 中心軸 3

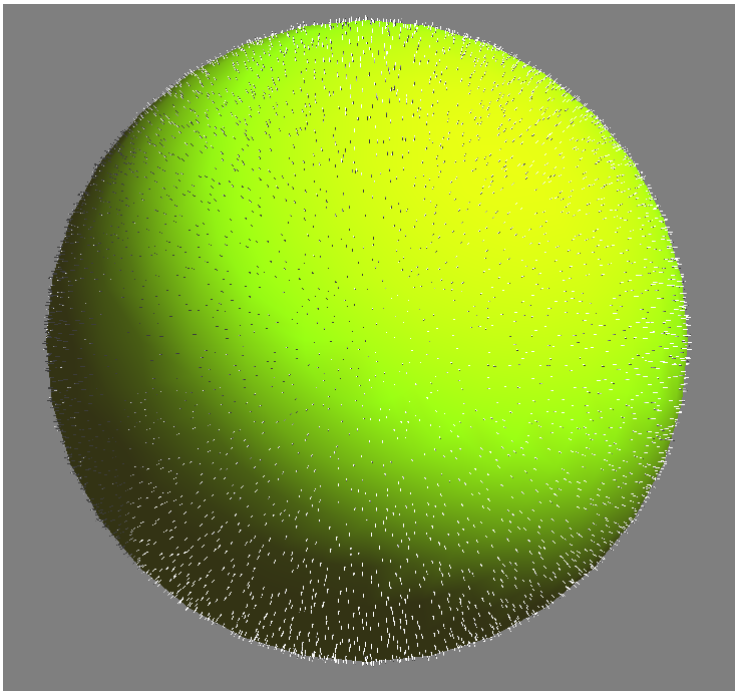


図 3.11: 球体の表面に細毛を生やした結果 1

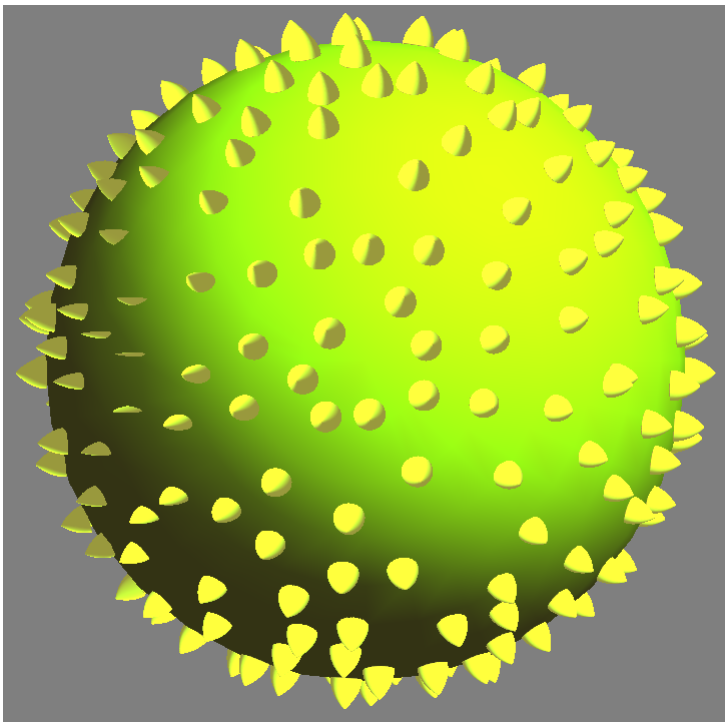


図 3.12: 球体の表面に細毛を生やした結果 2

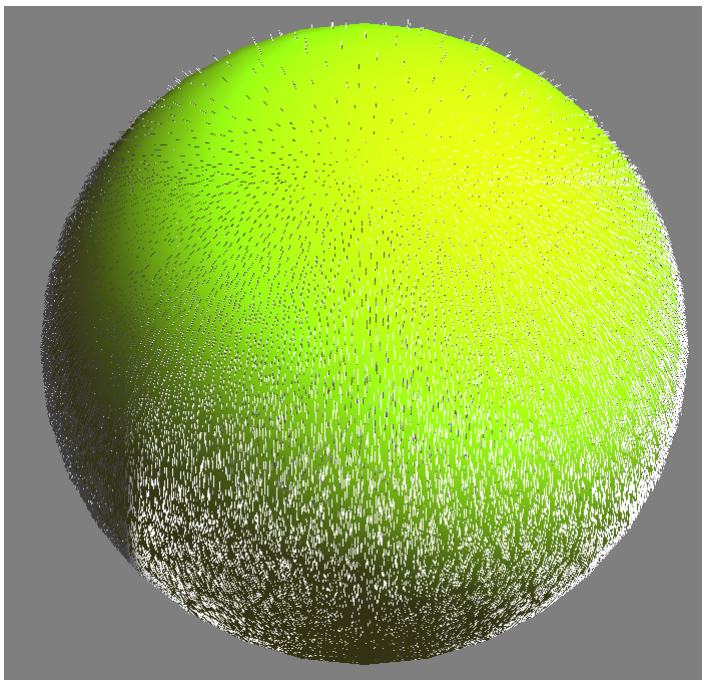


図 3.13: 球体の表面に細毛を生やした結果 3

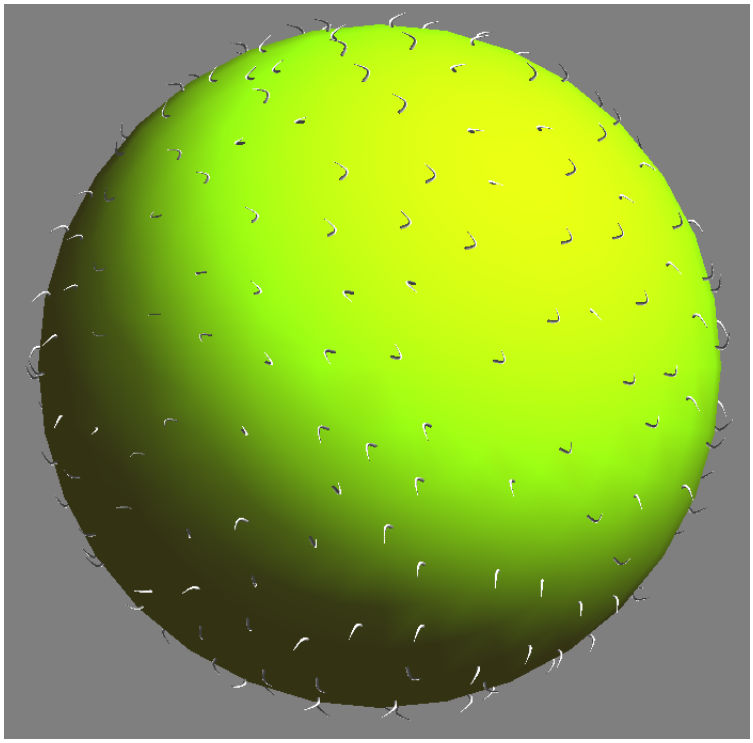


図 3.14: 球体の表面に細毛を生やした結果 4

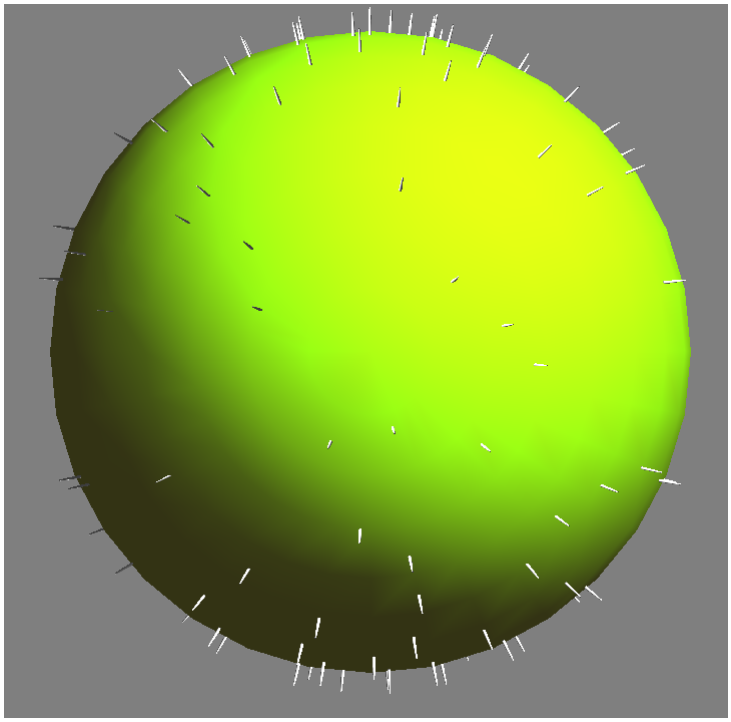


図 3.15: 球面の表面に細毛を生やした結果 5

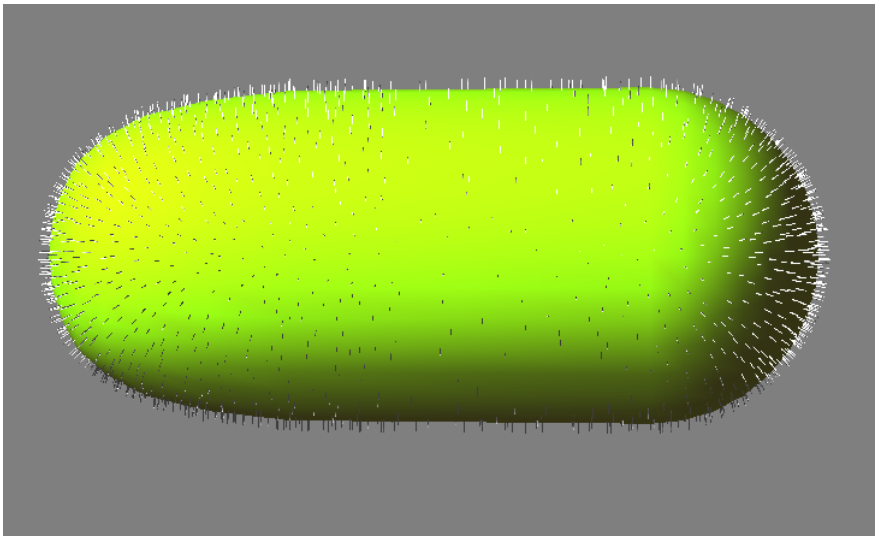


図 3.16: 細毛を生やした結果 1

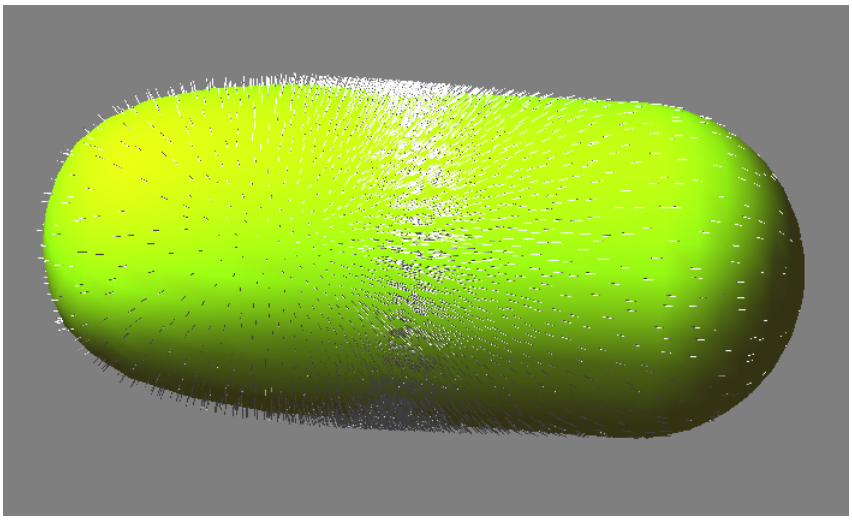


図 3.17: 細毛を生やした結果 2

第4章　まとめ

本研究では果実における細毛表現手法の提案を行った．

既存の手法 [5] では難易度の高いルールの定義と直感的ではない確率による分布の操作が問題になっていた．前者の問題に対して細毛作成の工程を分解することで解決し，後者の問題に対しては分布を構成する際に粗密を操作することによって解決した．

第 2 章では実現するための方法について詳細を述べている．第 2.2 節で細毛形状の作成方法について述べた．細毛はその中心軸と太さ関数によって定義し，ユーザがこれらを自由に設定することで任意の細毛を作成することが出来た．第 2.3 節で細毛の分布を構成する方法について述べた．細毛の分布は Poisson-disk Sampling で近似できることが知られていて，サンプリングの際に排他領域の大きさを変化させることで分布の粗密を操作することが可能になった．

以上のように既存手法での問題を解決した．以下では現在ある課題と今後の展望について述べる．

4.1　今後の課題

本手法では，果実のヘタから先端へ向かう軸を中心に細毛を制御していて，モデル形状を考慮していないため表現が出来ないものが存在する．例えばイチゴは果実の表面に種子があって，その種子から細毛が伸びている．そのため細毛を生やすためには種子の位置を理解しなければならないず，無作為に分布を作成している本手法では表現することが出来ない．

また現在の実装では細毛同士の衝突を扱っていないため，細毛同士が衝突していても貫通してしまう．非常に細かい細毛の場合には，目で判断しても貫通を判別しづらいので問題にはならないが，ランブータンのような太い細毛が生えている場合にははっきりと確認出来るので衝突を考慮しなければならない．

4.2　今後の展望

本研究では果実に対象をしぼった表現手法であるため，植物全体の細毛を完全に表現することは難しい．今後は果実だけではなく植物全体に対象を広げ，さまざまな細毛の表現を実現することを目標とする．

謝辞

研究にあたり指導教員である宮田先生を始め，研究室の先輩方には多大なご心配をおかけしたことと思います．論文を書く時期になっても実装がまるで終わっておらず，執筆に臨む段階ではありませんでした．そういったことから宮田先生や先輩方から実に熱のこもった激励の言葉を頂いて，私に出来るのかと悩みもしましたが奮起することが出来，より一層熱心に研究することが出来ました．修士論文を書き始めた頃はもしかしたら書き終わらないんじゃないかと不安にも苛まれましたが，無事書き上げることが出来てこのような形に著せたことは宮田先生や先輩方のお陰であります．簡単ではありますがここにその謝辞を記して終わろうと思います．繰り返しになりますが多大なご心配をおかけしながら，決して見捨てずに助言頂けたこと非常に感謝しております．

参考文献

[1] William Baxter, Naga Govindaraju. Simple Data-Driven Modeling of Brushes. Symposium on Interactive 3D Graphics 2010.

[2] Jules Bloomenthal. Techniques for Implicit Modeling. Xerox PARC Technical Report 1989, P89-00106.

[3] Jules Bloomenthal. An Implicit Surface Polygonizer. Graphics Gems IV, 1994

[4] Kurt W. Fleischer , David H. Laidlaw , Bena L. Currin , Alan H. Barr. Cellular Texture Generation. SIGGRAPH 1995.

[5] Martin Fuhrer, Henrik Wann Jensen, Przemyslaw Prusinkiewicz. Modeling Hairy Plants. In Proceedings of Pacific Graphics 2004, pp. 217-226.

[6] 江上信雄. 生物学　上, 飯野徹雄編. 東京大学出版会, 1983 年.

[7] Eric Landreneau, Scott Schaefer. Scales and Scale-like Structures. Eurographics Symposium on Geometry Processing 2010.

[8] Google Inc. Google SketchUp. http://sketchup.google.com/

[9] James T, Kajiya, Timothy L. Kay. Rendering Fur with Three Dimensional Textures. SIGGRAPH 1989.

[10] 花王生活科学研究所. ヘアケアの科学. 裳華房 1993 年.

[11] Eric Landreneau, Scott Schaefer. Scales and Scale-like Structures. Comput. Graph. Forum 2010 1653-1660

[12] Yong Jae Lee, C. Lawrence Zitnick, Micheal F. Cohen. ShadowDraw: Real-time User Guidance for Freehand Drawing. SIGGRAPH 2011 papers.

[13] Xian-Ying Li, Chao-Hui Shen, Shi-Sheng Huang, Tao Ju, Shi-Min Hu. Popup: Automatic Paper Architectures from 3D Models. SIGGRAPH 2010 papers.

[14] Paul Merrell, Eric Shkufza, Vladlen Koltun. Computer-Generated Resident Buiding Layout. SIGGRAPH Asia 2010 papers.

[15] 樋口優. MikuMikuDance. http://www.geocities.jp/higuchuu4/

[16] 小倉譲. 植物解剖および形態学. 養賢堂, 1962 年.

[17] OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. Shape Distributions. ACM Trans. Graph. 21(2002), pp. 807—832.

[18] ニワンゴ. ニコニコ動画. http://www.nicovideo.jp/

[19] Alec Rivers, Fredo Durand, Takeo Igarashi. 3D Modeling with Silhouettes. SIGGRAPH 2010 papers.

[20] Johannes Schmid, Martin Sebastian Senn, Markus Gross. OverCoat: An Implicit Canvas for 3D Painting. SIGGRAPH 2011

[21] Andrew Selle, Michael Lentine, Ronald Fedkiw. A Mass Spring Model for Hair Simulation. ACM Transactions on Graphics SIGGRAPH 2008, ACM TOG 27, 64.1-64.11 (2008)

[22] 「植物の軸と情報」特定領域研究班. 植物の生存戦略「じっとしているという知恵」に学ぶ. 朝日新聞社.

[23] SoundCloud Ltd. SoundCloud. http://soundcloud.com/

[24] Ustream. Ustream. http://www.ustrea.tv

[25] Yichen Wei, Eyal Ofek, Long Quan, Heung-Yeung Shum. Modeling Hair from Multiple Views. SIGGRAPH 2005 papers.

[26] Ying Xiang, Shi-Qing Xin, Qian Sun, Ying He. Parallel and Accurate Poisson Disk Sampling on Arbitrary Surfaces. SIGGRAPH Asia 2011 Sketches

[27] YouTube, LLC. YouTube. http://www.youtube.com

[28] Cem Yuksel, John Keyser. Deep Qpacity Maps. Computer Graphics Forum(Proceedings of EUROGRAPHICS 2008), 2008

[29] Cem Yuksel, Sarah Tariq. Advanced Techniques in Real-time Hair Rendering and Simulation. SIGGRAPH 2010 Course Notes.

[30] Karl D.D. Willis, Juncong Lin, Jun Mitani, Takeo Iharashi. Spatial sketch: bridging between movement & fabrication. TEI 2010.