

修士論文

平成 20 年度 (2008)

バブルメッシュ法を用いた
3D スキャナデータの再メッシュ化

東京工科大学大学院
バイオ・情報メディア研究科メディアサイエンス専攻

佐藤 和弥

修士論文

平成 20 年度 (2008)

バブルメッシュ法を用いた
3D スキャナデータの再メッシュ化

指導教員 渡辺 大地 講師

東京工科大学大学院
バイオ・情報メディア研究科メディアサイエンス専攻

佐藤 和弥

論文の要旨

論文題目	バブルメッシュ法を用いた 3D スキャナデータの再メッシュ化
執筆者氏名	佐藤 和弥
指導教員	渡辺 大地 講師
キーワード	3D スキャナ、特徴稜線、バブルメッシュ法 メッシュ、3次元再構成
[要旨]	<p>近年、3DCG コンテンツにおける新しいモデリング手法として、3D スキャナから形状を測定し取得した点群データから3次元モデルを生成する方法がある。数分で3次元モデルを生成でき、3D スキャナの操作も容易なことからモデリング作業の効率化が望める。3D スキャナで生成した形状は、メッシュの大きさが均一で稜線方向が規則的である。これを3DCG コンテンツで利用するためにはアニメーションを考慮し、特徴部位はメッシュを細かくそれ以外の部位はメッシュを粗く、かつ筋肉や骨の流れに沿った稜線方向を持つメッシュへと修正する必要がある。しかし、3D スキャナで測定する頂点数が膨大なため、メッシュ修正に大幅な時間がかかってしまい非効率である。また、3D スキャナデータを修正が容易なメッシュを持つ3次元モデルへと再構成する手法は数多くあるが、メッシュの細かさや稜線方向をユーザーが任意に制御することで修正が容易なメッシュを再構築する手法は提案されていない。</p> <p>本研究では、3D スキャナデータからユーザーが求めるメッシュの細かさや稜線方向を反映した修正が容易なメッシュの再構築手法を提案した。メッシュの細かさや稜線方向をユーザーが任意に制御可能なメッシュ生成手法として平方充填バブルメッシュ法がある。本手法では、形状の特徴を表すキャラクターラインを特徴稜線に近似して指定し、キャラクターラインを反映した平方充填バブルメッシュ法によって再メッシュ化を行った。その際に、ユーザーがメッシュの細かさや稜線方向を任意に指定することで、ユーザーが求めるメッシュの細かさや稜線方向を反映した修正が容易なメッシュを再構築可能である。</p>

A b s t r a c t

Title	Re-meshing of 3D scanner data using the bubble mesh method
Author	Kazuya Satou
Advisor	Lecturer Taichi Watanabe
Key Words	3D scanner,Feature edge,Bubble mesh method Mesh,3D reconstruction
[summary] <p>In recent years, there is the method of generating a three-dimensional model from the point-cloud data which measured and measured form from 3D scanner as the new modeling technique in 3DCG contents. The three-dimensional model of the object measured in several minutes can be generated, and increase in efficiency of modeling work can be desired from operation of 3D scanner being easy. The form generated with 3D scanner has a uniform mesh, and the regular direction of a ridgeline. In order to use this from 3DCG contents, in consideration of animation, it is necessary to correct the feature part to a mesh with the direction of a ridgeline which was coarse in the mesh as for the other fine part, and met the flow of muscles or a bone in the mesh. However, since the number of the peaks measured with 3D scanner is huge, mesh correction takes large time and it is inefficient. Moreover, although there are many techniques of reconstructing 3D scanner data to a three-dimensional model in which correction has an easy mesh, The technique of reconstructing a mesh with an easy correction because a user controls the fineness of a mesh and the direction of a ridgeline is not proposed.</p> <p>In this research, the reconstruction technique which is a mesh with an easy correction reflecting the fineness of a mesh and the direction of a ridgeline for which a user asks from 3D scanner data was proposed. It is considered as the mesh generation technique which can control the fineness of a mesh and the direction of a ridgeline, and there is the squarely-packing bubble mesh method. In this technique, the character line showing the feature of form was approximated as the feature edge, Re-meshization was performed by the squarely-packing bubble mesh method reflecting a character line. In that case, correction reflecting the fineness of a mesh and the direction of a ridgeline for which a user asks can reconstruct an easy mesh, because a user specifies the fineness of a mesh and the direction of a ridgeline.</p>	

目次

第1章	はじめに	1
1.1	研究背景と目的	2
1.2	論文構成	8
1.3	論文内の表記	8
第2章	修正が容易なメッシュを持つ3次元モデル	9
2.1	修正が容易なメッシュ	10
2.2	キャラクターラインと特徴稜線	13
2.2.1	キャラクターライン	13
2.2.2	特徴稜線	15
第3章	平方充填バブルメッシュ法を用いた再メッシュ化	16
3.1	提案手法の流れ	17
3.2	3D スキャナデータからの特徴稜線抽出	18
3.3	3D スキャナデータへのバブル配置	20
3.3.1	特徴稜線上の頂点への要素サイズ算出	21
3.3.2	要素サイズ補間形状の生成	23
3.4	バブルの平方充填	28
3.4.1	ベクトル場と隣接点	28
3.4.2	バブル間力の算出	32
3.4.3	バブルの移動と消滅	33
3.5	バブルからの4角形メッシュ生成	34
第4章	再構成結果	36
4.1	適用実験	37
4.2	考察	41
第5章	おわりに	43
	謝辞	46
	参考文献	48

目次

1.1	3次元モデルの比較	3
2.1	自動車のキャラクターライン	14
2.2	自動車の3次元モデル	14
3.1	既存手法にて抽出した特徴稜線	20
3.2	本手法によって抽出した特徴稜線	20
3.3	頂点に集まる面の隅の角度の総和	22
3.4	入力形状と特徴稜線	23
3.5	要素サイズ	23
3.6	Delaunay 条件を満たす3角形分割	25
3.7	Delaunay 条件を満たさない3角形分割	25
3.8	入力形状と要素サイズ	26
3.9	生成した補間形状	26
3.10	補間形状からの要素サイズ指定	27
3.11	バブル間力のポテンシャル場	29
3.12	ベクトル場と隣接点	30
3.13	顔に適したベクトル場	31
3.14	バブル間力の方向	32
3.15	3角形メッシュの4角形化	35
4.1	3D スキャナデータに対する本手法の適用	38
4.2	本手法を適用したモデル	40
4.3	メッシュ修正を施したモデル	40

第 1 章

はじめに

1.1 研究背景と目的

近年、コンピュータの高性能化により3次元コンピュータグラフィクス(以下3DCG)の需要が高まっている。それに伴い、3DCGコンテンツ制作の機会が増え作業の効率化が重要な課題となっている。しかし、3DCGコンテンツ制作は3次元形状の生成や複雑なアニメーション作成など手間取る作業が多く、また専門的なスキルや経験を求められるため制作者が限られている。そのため、3DCGコンテンツ制作における作業の効率化向上や簡易的な操作によって制作が行えることを目的とした研究が数多くなされてきた。本研究では、3DCGコンテンツ制作工程の1つであるモデリングに着目している。

一般的にモデリングはMaya[1]、3dsMAX[2]に代表されるモデリングソフトを用いて行う。立方体や球などの基本形状を基に頂点や面を分割するか、形状のアウトラインや特徴部を描いた稜線を基に面を張っていき分割していくことで3次元モデルを生成する。これには頂点や面に様々な操作や設定をする必要があり、形状によっては操作や設定に膨大な時間を費やすことになる。また、モデリングにはモデルが3次元空間上でどのような配置となるかを把握することや、モデリングソフトの操作や設定によってモデルがどのような挙動をするかを理解することなどが必要になってくる。これらの能力やモデリングの習熟度には個人差があるため、出来上がる3次元モデルのクオリティは制作者のモデリング能力に依存することになってしまう。

近年では、新しいモデリング手法として3Dスキャナにより取得した物体の表面座標値を点群データとして取り込み、3次元モデルを再構成してデータ化する手法が注目されつつある。この手法は以前から提案され主に自動車業界などのCADで利用されていたが、3Dスキャナが高価で他の業界では容易に利用することができなかったため、広く普及することは難しかった。しかし、近年の3Dスキャナの小型化、低価格化により、ビデオゲームや3DCGアニメーションといった3DCGコンテンツ制作においても容易に利用することが可能になり、新しいモデリング手

法として注目されてきている。

3D スキャナを用いると、登場キャラクターの例となる実際の人物やモックアップモデルの測定に数秒、測定で取得した表面座標値を点群データにし3次元モデルを再構成してデータ化するまでが数分で済む。これは、既存のモデリングソフトを利用した場合に比べ制作時間の削減ができることが明らかであり、膨大な制作時間がかかる3DCGコンテンツにおいては有効である。また、モデリングに必要な空間把握能力や習熟度、物の形状を的確に捉える観察力などがなくとも、3D スキャナの操作が非常に容易なため簡単に3次元モデルを生成することができる。モデルに対する形状修正もモデリングソフトを用いて修正する必要はなく、対象物に対して施した修正を再度スキャニングすることによって簡単に3次元モデルを修正することが可能である。しかし、3D スキャナによる形状生成や再スキャニングによる修正だけでは3D スキャナデータを3DCGコンテンツで利用することは難しい。その要因に3D スキャナデータの持つメッシュ構成がある。

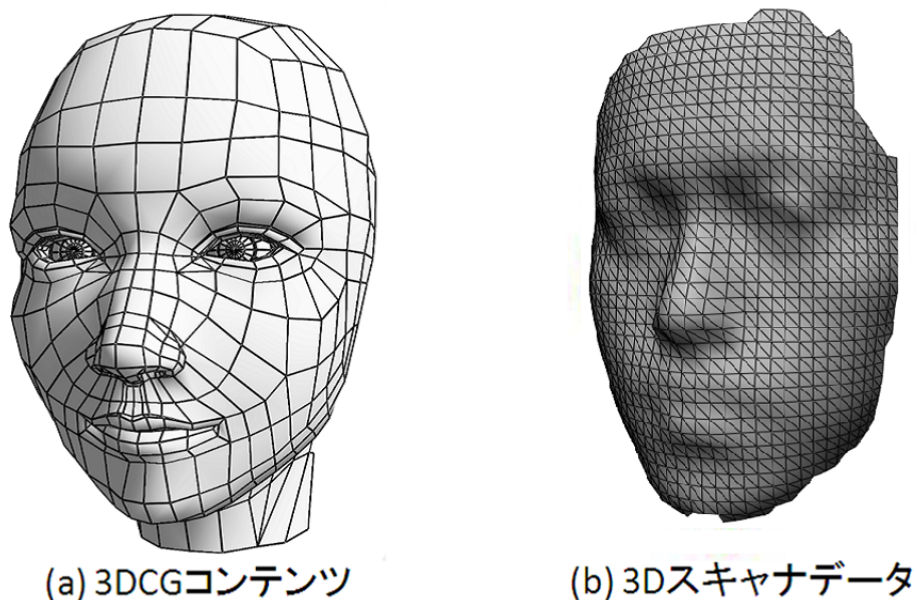


図 1.1: 3次元モデルの比較

図 1.1(a) は3DCG コンテンツ向けに作成した顔の3次元モデル、図 1.1(b) は3D

スキャナで生成した顔の3次元モデルである。図1.1(a)の特徴として、目、鼻、口といった顔の特徴的な部分はメッシュが細かくなっており、稜線の方向が顔の筋肉や骨の流れに沿っている。これらはアニメーション設定の簡易化や動作時のメッシュ同士の干渉回避など、アニメーションを考慮したメッシュ構成となっている[3]。また、4角形メッシュで構成することでメッシュの細かさや稜線方向が把握しやすく、3角形メッシュに比べ形状編集が容易である。

一方、図1.1(b)は3Dスキャナで生成した顔の3次元モデルである。3Dスキャナは形状を正確に表現することを重要視しており、測定する頂点数を多くすることで精度の高い形状を表現している。そのため、メッシュの構成までは考慮されおらず密で均一な3角形メッシュによってモデルを構成している。これでは、メッシュの細かさが均一で稜線方向も規則的になっているため、メッシュに対する修正を施す必要がある。この場合にメッシュの修正点として必要となるのは、頂点数の削減、メッシュの細かさの調整、稜線方向の調整である。頂点数の削減は一部モデリングソフトにて自動的に行えるものの、それ以外の修正に関してはユーザーが全て手作業で行っている[4]。メッシュの修正は頂点や面を移動、結合などすることで行い、これには測定する頂点数が膨大なためメッシュ修正に大幅な時間がかかり効率が悪くなっているのが現状である。

3Dスキャナデータをメッシュの修正が容易になるように再構成する手法は幾つか提案されている。主な手法として、不規則点群からの3次元モデル再構成、メッシュの再構築を行う再メッシュ化がある。不規則点群とは規則性を持たないランダムな点群座標値のデータ集合であり測定する機器に依存することはない。3Dスキャナで測定した点群は不規則点群に属する。

既存の不規則点群からの再構成手法[5][6][7]により不規則点群から3次元モデルを生成することが可能だが、再構成した3次元モデルが密で細かなメッシュなためメッシュ修正が難しい。大久保[8]の特徴稜線を反映した再構成手法では、形状の局所的な特徴やアウトラインを保持した頂点数の削減が可能になり、頂点数が少ないながらも入力形状と形状がほぼ変わらない曲面モデルを再構成することが

可能である。しかし、メッシュの細かさや稜線方向は頂点数削減処理の結果に左右してしまうため、やはり修正を施す必要がある。

メッシュの再構築を行う再メッシュ化手法として Surazhsky ら [9] は入力形状のメッシュを正 3 角形メッシュに再構築する手法を提案した。入力形状より綺麗なメッシュを再構築できるものの、メッシュの細かさが均一であり、入力形状によっては再構築することで頂点数が増大する恐れもある。金井ら [10] はユーザーが入力形状から整列化する部分領域やメッシュの整列パターンを指定することで自動的にメッシュの再構築を行った。この手法は広い範囲で領域内のメッシュの再構築を施せるが、局所的なメッシュの再構築を行うにはユーザーが細かな領域分割を指定しなくてはならない。また、再構築するメッシュは用意した整列パターンに依存するため、ユーザーが求めるメッシュの細かさや稜線方向を構成しない恐れがある。森田ら [11][12] の 3 次元モデルの表面の特徴パラメータを利用した再メッシュ化手法では、Lai ら [13] や Pottmann ら [14] の Feature Sensitive 計量を応用したポロノイ分割によって自動的に最適な領域分割を可能としたが、やはり再メッシュ化するメッシュは整列パターンに依存し、メッシュの細かさや稜線方向の制御は限られている。

このように、3D スキャナデータをメッシュの修正が容易になるように再構成する手法において、ユーザーがメッシュの細かさや稜線方向を任意に制御することはできない。そのため、ユーザーが求めるメッシュの細かさや稜線方向を反映した修正が容易なメッシュを再構築することができず、再構築後も大幅なメッシュの修正を強いられる恐れがある。以上のことから、3D スキャナが 3DCG コンテンツに普及しつつあるとはいえ、膨大なメッシュの修正時間を費やすことからまだ実用的であるとは言い難く、ユーザーが任意に制御したメッシュの細かさや稜線方向を反映した修正が容易なメッシュの再構築手法が必要となっている。

本研究は、3D スキャナによって生成した 3 次元モデルからユーザーの求めるメッシュの細かさや稜線方向を反映した修正が容易なメッシュの再構築手法の確立を目的とした。ここでいう修正が容易なメッシュとは、少ない頂点数で形状の特徴

を表し、特徴的な部位はメッシュが細かくそれ以外の部位はメッシュが粗く、かつ筋肉や骨の流れに沿った稜線方向を持つ4角形メッシュのことである。既存手法として、不規則点群からの3次元モデル再構成手法と再メッシュ化手法がある。3次元モデル再構成手法では、頂点数の削減のみを考慮しており、メッシュの細かさや稜線方向は頂点数削減処理の結果に左右してしまう。再メッシュ化手法では、入力形状に比べメッシュの細かさや稜線方向は修正しやすくなったものの、制御可能なメッシュの細かさや稜線方向のパターンが限られている。

本研究ではメッシュの細かさや稜線方向をユーザーが任意に制御することに着目し、3D スキャナデータからユーザーが制御したメッシュの細かさや稜線方向を反映した修正が容易なメッシュを再構築することで、メッシュの修正時間を短縮し3D スキャナデータをより実用的なものとした。本研究で想定する入力形状は、3D スキャナなどで測定した不規則点群から再構成した顔のポリゴンモデルとした。メッシュの細かさや稜線方向を制御可能なメッシュ生成手法として、平方充填バブルメッシュ法 [15][16][17] がある。この手法では、領域内に分子間力を仮想した球状物体を配置し、球状物体とベクトル場から与えられた隣接点間で分子間力を働かせ最密に平方充填する。そして、充填した球状物体の中心点を結合することで良質な4角形メッシュを生成する。球状物体の大きさ(以下要素サイズ)と、複数の単位ベクトルからなるベクトル場をユーザーが任意に指定することでメッシュの細かさと稜線方向を制御できる。

本手法では、入力形状からデザインにおいて重要な要素であるキャラクターラインを特徴稜線により近似して指定し、特徴稜線を適用した平方充填バブルメッシュ法を施すことで再メッシュ化を行った。その際に、特徴稜線上の頂点の要素サイズとベクトル場をユーザーが任意に指定することで、ユーザーが求めるメッシュの細かさと稜線方向を反映した修正が容易なメッシュを再構築できる。また、ユーザーが手入力で行っていた特徴稜線上の頂点への要素サイズ指定を近似ガウス曲率から自動的に算出することによって、形状の特徴の重要度に応じたメッシュの細かさを効率的に指定できる。以上から、3D スキャナデータからユーザーが求

める修正が容易なメッシュを効率的に再構築することが可能とした。本手法により、従来はCADのみで利用していた3D スキャナを用いたモデリング手法をビデオゲームや3DCG アニメーションなどの3DCG コンテンツにおいても有効に活用することが可能となる。

検証実験として、3D スキャナで生成した顔のポリゴンモデルに対し本手法を適用した。その結果、要素サイズをユーザーが直接、もしくは近似ガウス曲率から自動的に指定することで、特徴的な部位はメッシュが細かくそれ以外の部位はメッシュが粗くなり、形状の特徴に応じたメッシュの細かさを制御することができた。顔の筋肉や骨の流れに沿ったベクトル場を入力形状に指定することで、再構築したメッシュの稜線方向を筋肉や骨の流れに沿うことを実現できた。また、メッシュの細かさを制御することで、再構築したメッシュの頂点数を入力形状の頂点数より少なくすることができた。

本手法で再構築したメッシュの検証として、入力形状との誤差判定とモデリングソフトでのメッシュの修正を行った。誤差判定は再構築したメッシュの頂点とその頂点に一番近い入力形状の頂点との距離で判定を行った。その結果、再構築したメッシュの全ての頂点において僅かな誤差で形状を形成しているため、入力形状とほぼ変わらぬ形状を生成できている。本手法で再構築したメッシュが修正が容易かどうかを検証するため、3D スキャナデータと本手法を適用した3次元モデルそれぞれをモデリングソフトで読み込みメッシュの修正を施した。3D スキャナデータへのメッシュ修正は、頂点数の削減とメッシュの細かさ、稜線方向の調整を行うため、頂点の結合や移動、面の分割を幾度と繰り返す必要があり、修正に膨大な時間を費やした。一方、本手法を適用した3次元モデルへのメッシュ修正は、再構築する際にメッシュの細かさや稜線方向を制御していたため、3D スキャナデータへの形状修正に比べ少ない時間で3DCG コンテンツに適するモデルへと修正ができた。また、メッシュの細かさの制御によって頂点数を削減することができたため、手作業による頂点数削減は数回程度で済み、メッシュ修正も容易にできた。

以上の結果から、本手法を適用することで3D スキャナデータからユーザーが任意に制御したメッシュの細かさや稜線方向を反映した修正が容易なメッシュを再構築することができ、メッシュの修正を施す時間を短縮することが可能となった。

1.2 論文構成

本論文の構成は5章からなる。その内訳は以下の通りである。2章では、本手法が目的とする修正が容易なメッシュの定義と、形状の特徴を表すキャラクターラインと特徴稜線について述べる。3章では、特徴稜線を適用した平方充填バブルメッシュ法による3D スキャナデータの再メッシュ化手法について述べる。4章では、本研究で実装したツールを基に本手法の実行結果と評価について述べ、手法の課題について考察する。最後に5章では、本論文のまとめと実行結果を踏まえた手法の今後の展望について述べる。

1.3 論文内の表記

以下に本論文で用いる表記について述べる。本論文における数式表現は、基本的にはSpanierの著書[18]に準ずるものとする。

- ベクトルの内積は $a \cdot b$ 、外積は $a \times b$ のように表す。
- $|v|$ をベクトル v のノルムとする。

第 2 章

修正が容易なメッシュを持つ3次元モデル

本章では、3DCG コンテンツの3次元モデルのメッシュ構成の説明や本研究で目的とする修正が容易なメッシュの特徴を説明し、定義を明確にする。また、修正が容易なメッシュを構築する際には形状の特徴を考慮する必要がある。本研究では、主にデザイナー用語として概念的に用いられることが多いキャラクターラインと渡辺 [19] が定義した特徴稜線に着目した。3次元形状の特徴を表す概念であるキャラクターラインを特徴稜線として再メッシュ化に反映することができれば、形状の特徴を考慮したメッシュの再構築が可能である。2.1節では3DCG コンテンツに適するメッシュの説明と修正が容易なメッシュの定義について述べる。2.2節ではキャラクターラインと特徴稜線について述べる。まず2.2.1項では、主にデザイナー用語として概念的に用いられることが多いキャラクターラインについて詳細を述べ、定義を行う。次に2.2.2項では、キャラクターラインを近似した特徴稜線について詳細を述べ、定義を行う。

2.1 修正が容易なメッシュ

ビデオゲームや3DCGアニメーションなどの3DCGコンテンツで利用する3次元モデルは、アニメーションすることを想定したメッシュ構成となっている。このメッシュ構成の特徴は大きく分けて4つある。

- 頂点数が少ない。
- 部位によってメッシュの細かさが異なる。
- 稜線方向が筋肉や骨の流れに沿っている。
- 4角形メッシュで構成している。

第1の特徴は、少ない頂点数で形状の特徴を表現していることである。3DCGコンテンツではコンテンツ内で使用できるデータサイズが限られている。そのため、3次元モデルの多くは使用可能なデータサイズの制限を設けている。3次元モデルのデータサイズはポリゴン数、テクスチャの枚数と解像度など複数の要素で決定

する。その中でも、ポリゴン数は3次元モデルのデータサイズの大半を占める重要な要素である。ポリゴン数を多く使用すれば滑らかな曲面を表現することができるため、精度の高い3次元モデルを生成できる。しかし、不必要にポリゴン数を増やすとデータサイズが大きくなり3DCGコンテンツで利用できなくなる。一方、ポリゴン数を少なくすると3次元モデルのデータサイズが軽くなるが、形状の特徴を失う恐れがあり精度の低い3次元モデルになってしまう。そこで、形状の特徴的な部位や曲線を多用した部位では、ポリゴン数を多くすることで形状の特徴や曲線を保っている。そして、それ以外の部位ではポリゴン数を抑えることでデータサイズの削減を行っており、形状の特徴によってポリゴン数を使い分けている。また、頂点数を少なくすることで形状に対する編集も容易にできる。

第2の特徴は、特徴的な部位はメッシュが細かく、それ以外の部位はメッシュが粗くなっていることである。アニメーションを行うためには3次元モデルにいくつかの設定をする必要がある。その中でウェイト調整があり、これは人の関節に当たるボーンが移動や回転をした場合、頂点がどの程度追従するかをパラメーターで決定することである。このウェイト調整は頂点1つ1つに対して行う。特に肩や腰などの関節部分、目や口など頻繁に動作する部分においてはウェイト調整を緻密に設定する必要がある。そのため、これらの部分ではメッシュを細かくし頂点数を増やすことで緻密なウェイト調整を設定できるようにしている。

第3の特徴は、メッシュを構成する稜線が筋肉や骨の流れに沿っていることである。モデリングにおいて稜線方向は形状を表現するのに重要な要素である。例えば、人の腕の筋肉は捻じれている。これを理解せずに単純な均等で規則的な稜線でモデリングすると、形状に不自然な凹凸が現れやすく滑らかな形状を形成できない恐れがある。肩甲骨、腰骨、膝、肘などの肉より盛り上がっている骨のライン、あるいは大胸筋、大円筋などといった筋肉の流れに沿って稜線方向を決定することで人体学に基づいたモデルが形成でき、形状に対する編集も容易にできる。また、稜線方向はアニメーションにおいても重要な要素となっている。格子状の稜線を持つモデルに対し走る、前屈みになるなどの動作を設定すると、肩回りや

脚回りなど大きく動く関節部分においてメッシュ同士の干渉、重なりといった形状の崩れが生じてしまう。一方、筋肉や骨の流れに沿った稜線を持つモデルに対し同様の動作を設定すると、形状の崩れが起きにくく形状を保った動作が可能となる。このように稜線方向を形状の流れに沿うことで形状の破綻が起きにくくなり、さらに前述したウェイト調整が綺麗にできる利点がある。

第4の特徴は4角形メッシュで構成していることである。4角形メッシュは3角形メッシュに比べ、メッシュの細かさや稜線方向が把握しやすいため、形状編集が容易である。3角形メッシュによって生じやすい形状の皺がほぼなく、滑らかな形状を表現できる。また、3角形メッシュに比べポリゴン数が少ないため、データサイズの削減においても有効である。

一方、3D スキャナデータのメッシュ構成は均一で稜線方向が格子状になっており、さらに3角形メッシュを膨大な頂点数で構成している。3D スキャナデータは対象物の形状を正確に表現できているが、3DCG コンテンツに不向きなメッシュ構成のためメッシュの修正が必要になってくる。メッシュを修正するには、頂点数の削減をし、形状の特徴の重要度に応じたメッシュの細かさ、筋肉や骨の流れに沿った稜線へと調整する必要がある。この調整は、頂点や面を1つ1つ選択し移動や削除などすることで行う。現状の修正手段として、まず3D スキャナデータをモデリングソフトで読み込み、特徴が消えないようにある程度までポリゴン数の削減を行う。次に、3D スキャナデータを腕や足など各パーツごとに分け、さらに各パーツごとにポリゴン数の削減とメッシュの細かさや稜線方向を調整する。そして最後に、各パーツを統合しテクスチャ編集やアニメーションの設定を行う。この作業は、一部モデリングソフトで自動的に行える工程もあるがほぼユーザーが手作業で行っている。これにはユーザーのスキルや経験が問われ、さらに頂点数が膨大なことからメッシュの修正時間が大幅にかかってしまう。

これらを踏まえ、本研究では修正が容易なメッシュを以下のように定義する。

- 少ない頂点数で形状の特徴を表現している。

- 特徴的な部位はメッシュが細かくそれ以外の部位はメッシュが粗くなっている。
- 筋肉や骨の流れに沿った稜線方向である。
- 4角形メッシュで構成している。

3D スキャナデータのメッシュを上記定義の修正が容易なメッシュへと再構築することで、修正しやすい3次元モデルが生成しメッシュ修正時間の短縮を施す。

2.2 キャラクターラインと特徴稜線

2.1 節で述べたように、修正が容易なメッシュを構築する際には形状の特徴を考慮する必要がある。本研究では、キャラクターラインと渡辺が定義した特徴稜線に着目した。大久保は入力形状に対してキャラクターラインを近似した特徴稜線を反映したメッシュ最適化を施すことで、形状の特徴を保った3次元モデル再構成を可能とした。本研究では、3D スキャナデータからキャラクターラインを近似した特徴稜線を指定し平方充填バブルメッシュ法に適用することができれば、形状の特徴を考慮したメッシュの再構築が可能であると考えた。本節では、主にCADのデザイナー用語として概念的に用いられることが多いキャラクターラインについて詳細を述べ、定義を行う。さらに、キャラクターラインを近似した特徴稜線について詳細を述べ、定義を行う。

2.2.1 キャラクターライン

キャラクターラインとは一般的にCADのデザイナー用語として使われており、主に2つの意味がある。1つは物体の形状の特徴を表す面の流れや曲線という概念的な使い方をする。特に自動車のデザインを評価する際に用い、車体のフォルムや細部における曲面の流れが綺麗に表れた曲線や、車体の特徴部を表す曲線を指してキャラクターラインと呼ぶ。キャラクターラインは主に自動車などのごく一部の業界におけるデザインで用いられることが多く、車体の印象を決定付け、スピー

ド感や躍動感、力強さなどを演出する要素として重要視されている。近年は、一般的な製品や3DCGのキャラクターモデルなどのデザインにおいてもキャラクターラインを用いることが多くなってきている。そのため、主に自動車のデザイン評価において利用してきたキャラクターラインを、一般的な製品やキャラクターモデルなどの評価に用いるケースも増えてきている。

もう1つは、モデリングにおける曲面形状のアウトラインや細部の特徴を表す曲線の意味で用いる。モデリングの過程の多くは、まずデザインを参考に芯やアウトラインとなるキャラクターラインを作成する。次に、モデルの全体的な形状の概要を作成し、それに基づきポリゴンや曲面を生成していく。その後、細部における特徴部を表すキャラクターラインの追加や、面の分割などを行い精密なモデルを作成していく。モデリングにおけるキャラクターラインは、デザイナーがポリゴンや曲面の形状を定義する際に用いる曲線のことを指し、3次元形状を形作る重要な要素となる。

これらのことを踏まえ、大久保 [8] の定義したキャラクターラインを以下に示す。

- デザイニングにおけるデザインの評価の対象となる曲面の流れや曲線。
- モデリングにおける形状のアウトラインや意匠的特徴を表す曲線。

本研究でも上記のようにキャラクターラインを定義する。

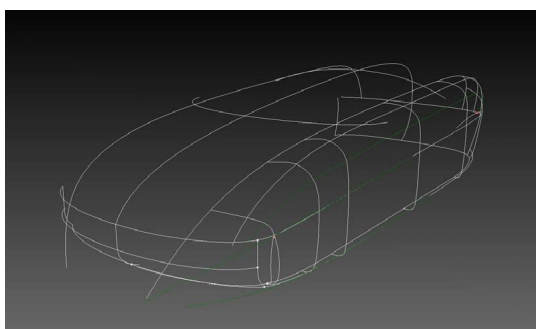


図 2.1: 自動車のキャラクターライン

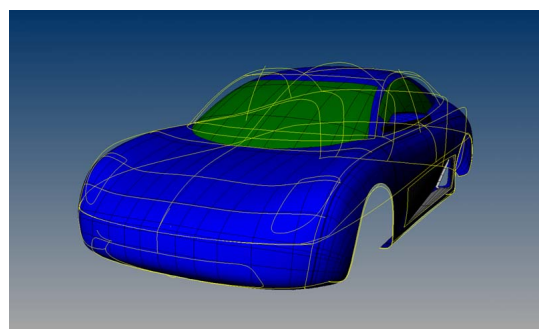


図 2.2: 自動車の3次元モデル

図 2.1 は自動車のキャラクターライン、図 2.2 はキャラクターラインからモデルリ

ングした自動車の3次元モデルである。自動車の特徴であるフォルムやボディーの曲面の流れがキャラクターラインとなっており、それを基にモデリングをしていることが分かる。

2.2.2 特徴稜線

特徴稜線とは、3次元モデルを構成する稜線の中で形状の特徴を表す稜線を指す。点群から再構成した密なポリゴンモデルより、幾何学的、数学的計算によって算出することができる。以下に渡辺による特徴稜線の定義を述べる。

- 初期メッシュが表現する形状において各部位の境界を表す稜線。
- 曲率や接ベクトルなどの幾何学的数値やベクトルが設定する基準を満たした稜線。
- 曲面の流れベクトルに沿った稜線。
- 形状の意匠的特徴部とそれ以外の境界を表す稜線。

大久保は初期メッシュから上記定義によって抽出した特徴稜線群をキャラクターラインとして、曲面モデルを生成する際に反映させた。これによって、形状の特徴部を考慮した曲面モデルを生成している。本研究では上記定義を満たす稜線とユーザーが直接指定した稜線を特徴稜線群とする。そして、特徴稜線群をキャラクターラインとして再メッシュ化に反映することで形状の特徴部を考慮したメッシュを生成する。

第 3 章

平方充填バブルメッシュ法を用いた再メッシュ化

本章では、特徴稜線を適用した平方充填バブルメッシュ法による 3D スキャナデータの再メッシュ化手法について述べる。3.1 節では、手法の概要と再メッシュ化の手順について述べる。3.2 節では、3D スキャナデータからの特徴稜線抽出について述べる。既存の特徴稜線の抽出方法と本手法での特徴稜線の抽出方法について解説する。以降の節では特徴稜線を適用した平方充填バブルメッシュ法について述べる。まず 3.3 節にて、3D スキャナデータの頂点へのバブル配置について述べる。バブルの大きさを決める要素サイズの算出と要素サイズを分布する補間形状生成について解説する。次に 3.4 節にて、バブルの平方充填について述べる。バブルの配列を制御するベクトル場とバブル間力を働かせる隣接点、バブルに引力を働かせるバブル間力について説明する。また、バブルの移動と消滅条件について解説する。次に 3.5 節にて、充填したバブルからの 4 角形メッシュ生成について述べる。バブルの中心点を結び良質な 4 角形メッシュを生成する手法について解説する。

3.1 提案手法の流れ

本手法では、2つのステップによって 3D スキャナデータから修正が容易なメッシュを再構築する。まず、3D スキャナデータからキャラクターラインを近似した特徴稜線をユーザーが指定する。次に、指定した特徴稜線を平方充填バブルメッシュ法に適用し、形状の特徴を考慮したメッシュを再構築する。具体的な手順としては以下の通りである。

1. 3D スキャナデータからの特徴稜線抽出
2. 特徴稜線を適用した平方充填バブルメッシュ法
 - (a) 3D スキャナデータへのバブル配置
 - (b) バブルの平方充填
 - (c) バブルからの 4 角形メッシュ生成

第1のステップにおいて、キャラクターラインを近似した特徴稜線を3D スキャナデータからユーザーが指定する。次のステップは、特徴稜線を適用した平方充填バブルメッシュ法のメッシュ再構築の手順となる。まず、3D スキャナデータの頂点に対しバブルの配置を行う。特徴稜線上の頂点に要素サイズをユーザーが直接、もしくは近似ガウス曲率を用いた算出方法から自動的に指定する。そして、指定した要素サイズを基に要素サイズ補間形状を生成し、指定していない頂点に対し要素サイズを与え、要素サイズを持つ全ての頂点に対しバブルを配置する。メッシュの細かさは要素サイズと要素サイズ補間形状によって制御が可能である。次に、配置したバブルを平方充填する。ユーザーが任意に指定したベクトル場から全てのバブルに対し4つの隣接点を求める。そして、あるバブルとそれ以外のバブルの隣接点との間にバブル間力が働くことによって、バブルを平方に最密充填する。メッシュ化した際の稜線方向はベクトル場を指定することで制御可能である。最後に、バブルの中心点から4角形メッシュを生成する。以下に各ステップの詳細について述べる。

3.2 3D スキャナデータからの特徴稜線抽出

この節では、特徴稜線の抽出方法について述べる。多くの特徴稜線抽出手法[19][20][21]は、位相距離を考慮した部分領域の湾曲を検出する関数や稜線に隣接する面のなす角度を検出する関数など、複数の評価関数を用いた。そして、判別する稜線に対し特徴稜線を否かを決定するしきい値を設定し、しきい値を満たす稜線を特徴稜線とした。しきい値は対話的な操作によって設定でき、さらに特性の異なる複数の評価関数を用いているため、柔軟な特徴稜線抽出が可能である。しかし、3D スキャナデータは形状を的確に表現することのみを考慮しておりメッシュ構成までは考慮していないため、キャラクターラインとなる特徴稜線を持たない恐れがある。また、形状の特徴部ではメッシュを細かくするため、特徴稜線を細かく指定し頂点数を多くする必要があり、幾何学的、数学的な抽出方法では特徴稜線がユーザーの求めるキャラクターラインに近似しない可能性も十二分にある。

そこで、本手法ではキャラクターラインとなる特徴稜線をユーザーが直接指定することで特徴稜線の抽出を行う。3D スキャナデータはキャラクターラインとなる特徴稜線がない恐れがあるため、稜線を直接指定するのではなく特徴稜線を形成する2つの頂点を指定する。そして、指定した2つの頂点から稜線を生成し、これを特徴稜線とする。既存手法に比べ抽出に時間はかかるものの、ユーザーが求めるキャラクターラインを直感的かつ的確に指定することが可能である。形状の特徴部においてユーザーが特徴稜線を密に指定することで、細かなメッシュを制御することができる。また、頂点を指定することで新たに稜線を生成するため、キャラクターラインとなる特徴稜線を持たないモデルに対しても特徴稜線を指定することができ、細かな特徴稜線の指定も可能である。以下に特徴稜線を抽出したモデルを示す。図 3.1 は既存手法によって抽出した特徴稜線、図 3.2 は本手法によって抽出した特徴稜線を示す。

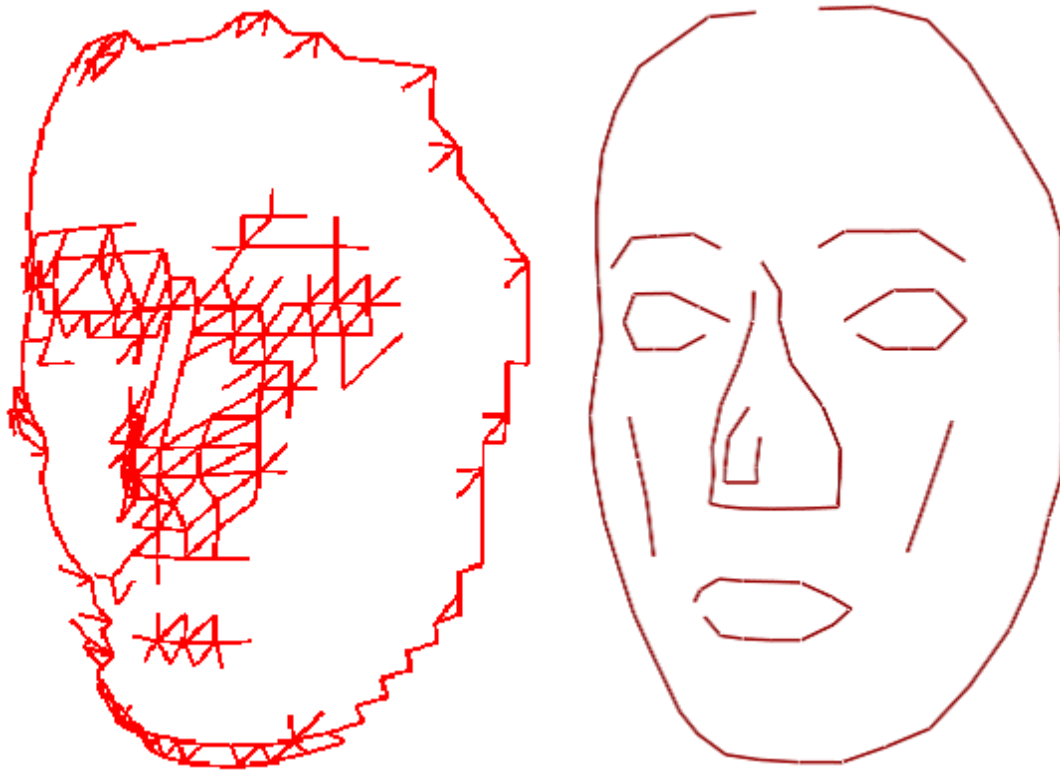


図 3.1: 既存手法にて抽出した特徴稜線 図 3.2: 本手法によって抽出した特徴稜線

3.3 3D スキャナデータへのバブル配置

入力形状に対しバブルの配置を行うには、入力形状の全ての頂点に対し要素サイズを指定する必要がある。本手法では、抽出した特徴稜線上にある全ての頂点に対し直接ユーザーが指定するか近似ガウス曲率を用いた算出方法から要素サイズを指定する。そして、指定した要素サイズから要素サイズ補間形状を生成し要素サイズを指定していない頂点に対し補間形状から要素サイズを自動的に指定する。最後に、要素サイズを持つ全ての頂点にバブルを配置する。本節では、近似ガウス曲率からの要素サイズの算出と要素サイズ補間形状生成について述べる。

3.3.1 特徴稜線上の頂点への要素サイズ算出

要素サイズとはバブルの大きさのことであり、要素サイズが小さいバブルでは細かいメッシュ、大きいバブルでは粗いメッシュを生成するため、メッシュの細かさを制御する重要な要素である。Shimadaら [15] は入力領域の頂点上にユーザーが直接指定することで要素サイズを決定した。本研究では、特徴稜線上の頂点に要素サイズを指定する。Shimadaらと同様に頂点上にユーザーが直接指定する方法では、ユーザーが求めるメッシュの細かさを直接指定できるものの、3D スキャナデータから抽出した特徴稜線上の頂点数が膨大なため、要素サイズを指定する時間が大幅にかかってしまう。さらに、どの頂点にどの程度の要素サイズを指定すればよいかはユーザーのモデリングに関する知識や経験が問われてしまい、適切な要素サイズを指定できない恐れがある。そのため、形状の特徴の重要度に応じた要素サイズの自動指定方法が必要になってくる。

一般的に、形状の特徴を判別するには曲率や面のなす角度を用いる。特に、キャラクターラインが表す顔の目や口などの局所的な特徴部は鋭角な傾向にある [8][21]。形状を構成する任意の頂点において鋭角であるか否かを判定する値として近似ガウス曲率がある。近似ガウス曲率とは、曲面を近似した多面体の頂点におけるガウス曲率の近似値である。近似ガウス曲率の絶対値が大きい頂点では形状が尖っており特徴的な部位の傾向があるため、近似ガウス曲率を用いることで形状の鋭角な部位の判別が可能である。また、3 角形メッシュの面積が均一であれば正確に判別することができ、本研究で対象としている 3D スキャナデータは 3 角形メッシュの面積が均一な場合が多いため近似ガウス曲率は有効である。この近似ガウス曲率を用いることで形状の特徴の重要度に応じた値を得られ、これを用いて要素サイズを算出することで形状の特徴の重要度に応じた要素サイズの自動指定が可能である。

そこで本研究では、特徴稜線上にある頂点の近似ガウス曲率から自動的に要素サイズを算出し指定する方法を設けた。特徴的な部位ではメッシュを細かくする

ため、バブルの要素サイズは小さくする必要がある。近似ガウス曲率の絶対値は特徴的な部位では大きい傾向があるため、その逆数を用いることで要素サイズを決定する。これらを踏まえ、本手法での近似ガウス曲率から算出する要素サイズは以下のようにして求める。まず、特徴稜線上の頂点 V に対して近似ガウス曲率 K を求める。本手法では、Calladine[22] が提案した算出式によって近似ガウス曲率 K を求める。

$$K = \frac{a}{S} \quad (3.1)$$

a 、 S は以下のようにして求まる。また、図 3.3 は a のなす角を示す。

$$\begin{cases} a = 2\pi - (\text{頂点 } V \text{ に集まる面の隅の角度の総和}) \\ S = (\text{頂点 } V \text{ に集まる面の面積の総和})/3 \end{cases} \quad (3.2)$$

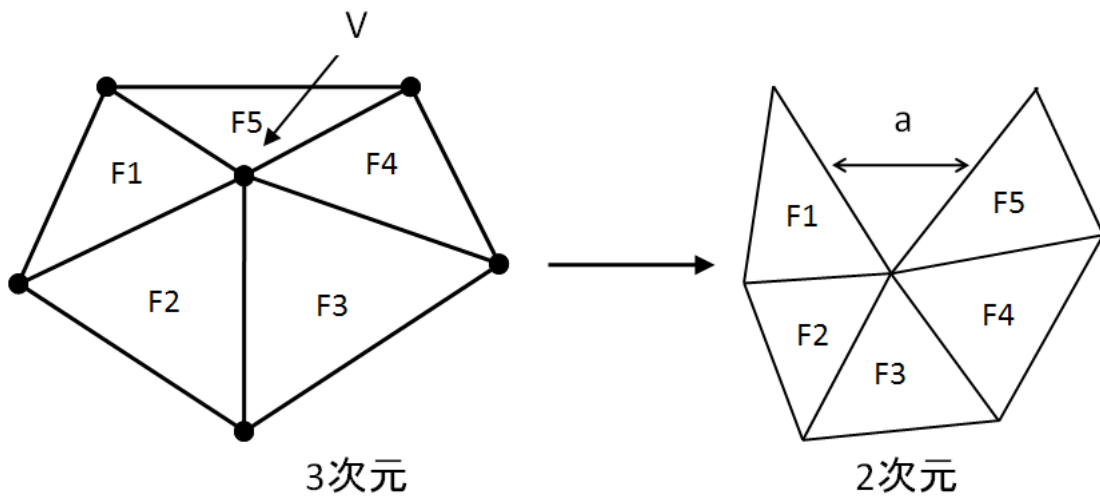


図 3.3: 頂点に集まる面の隅の角度の総和

式 (3.1) から要素サイズ E は以下のようにして求める。

$$E = \frac{1}{\sqrt[3]{K}} \quad (3.3)$$

以下に入力形状と指定した要素サイズを示す。図 3.4 は入力形状である顔の 3D スキャナデータと特徴稜線、図 3.5 は特徴稜線上の全ての頂点に要素サイズを指定

した結果である。図 3.4 において、赤い稜線が特徴稜線を示す。図 3.5 において、頂点から法線方向へ伸びた線の長さがその頂点での要素サイズを表している。

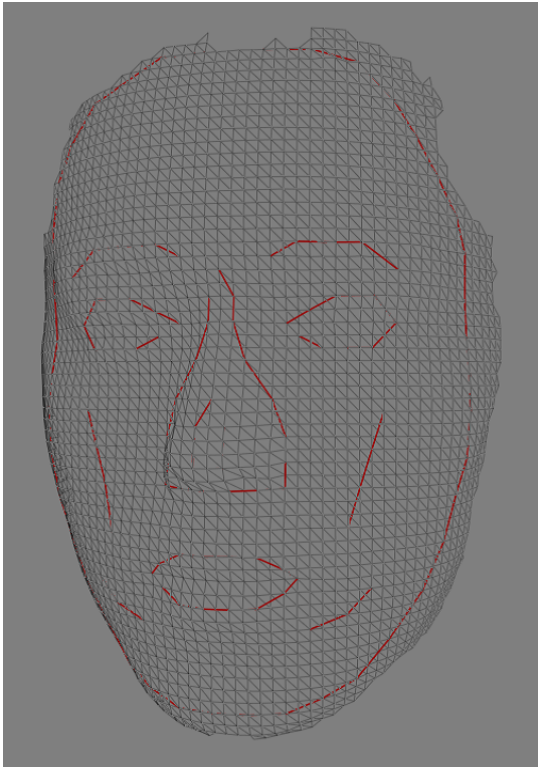


図 3.4: 入力形状と特徴稜線

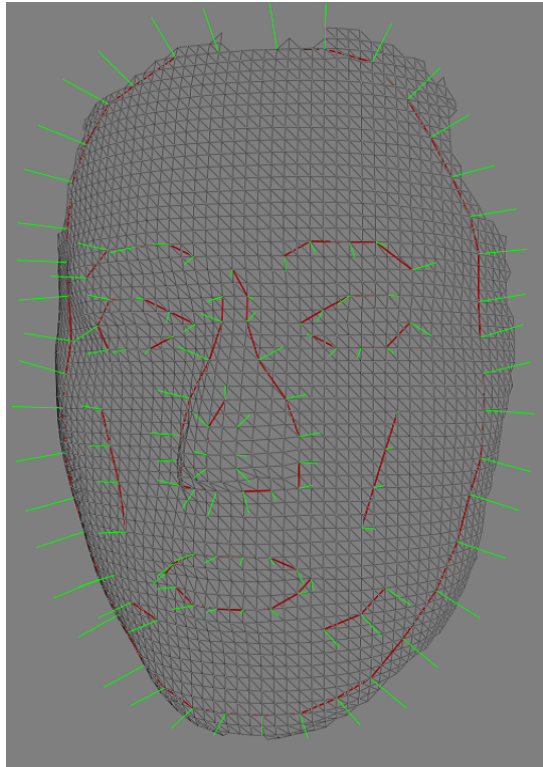


図 3.5: 要素サイズ

3.3.2 要素サイズ補間形状の生成

要素サイズを指定していない頂点に対する要素サイズ指定方法として、山田ら [16] が提案した要素サイズ分布関数がある。要素サイズ分布関数とは、領域のどの部分にどれくらいの大きさの要素サイズを生成するかを領域全体に対して規定する関数のことである。この分布関数は以下の 4 つの要件を満たす。

- 任意形状の領域に対して分布関数を生成できる。
- 領域上の任意の場所で指定した要素サイズを満たす分布関数である。
- 要素サイズが急激に変化せず連続的に変化する分布関数である。

- 少ない入力データで生成できる分布関数である。

山田らは上記4つの要件を満たす分布関数を、指定した要素サイズと入力形状、薄板の曲げモデルの歪みエネルギーを最小化する手法 [23] を用いて自動的に生成した。一般に要素サイズが急激に変化する箇所では極端に歪んだメッシュを生成してしまう。山田らの手法では、要素サイズを指定していない箇所においても要素サイズが連続的に変化するため、指定した要素サイズを考慮しながら領域全体に良質なメッシュを生成することが可能となった。

本研究では、この分布関数によって生成する形状に着目した。要素サイズで指定する点を滑らかに補間する形状を生成することで、要素サイズの分布が可能であると考えた。本手法では、指定した要素サイズと特徴稜線、3角形メッシュ生成手法である Delaunay3 角形分割 [24] を用いて要素サイズを分布する補間形状を生成する。まず、補間形状を形成する頂点群を求める。特徴稜線上の要素サイズを指定した頂点に対し、法線方向へ長さが要素サイズである線分を引き頂点を置く。この頂点を補間形状を形成する頂点とする。次に、求めた頂点群から Delaunay3 角形分割によってメッシュ化することで補間形状を生成する。Delaunay3 角形分割は、任意の3つの頂点からなる3角形の外接円に他の頂点が含まない場合において3角形で分割する性質がある。図 3.6 に Delaunay3 角形分割の条件を満たす3角形、図 3.7 に Delaunay3 角形分割の条件を満たさない3角形を示す。

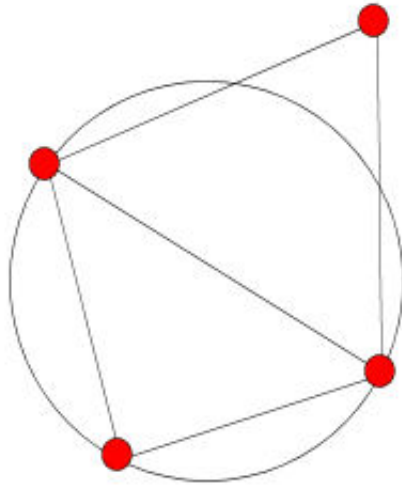


図 3.6: Delaunay 条件を満たす 3 角形分割

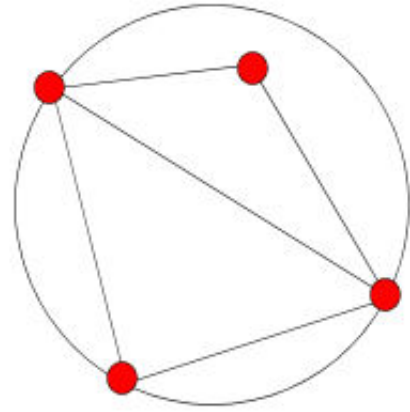


図 3.7: Delaunay 条件を満たさない 3 角形分割

本手法によって生成した補間形状を以下に示す。

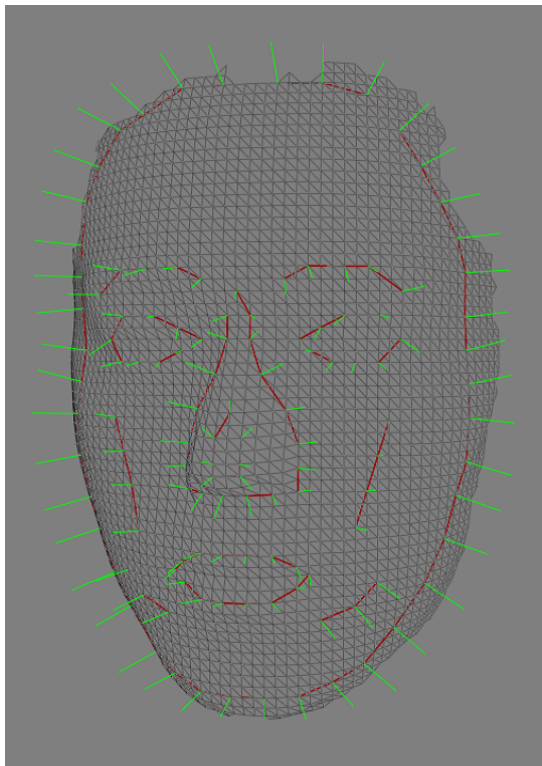


図 3.8: 入力形状と要素サイズ

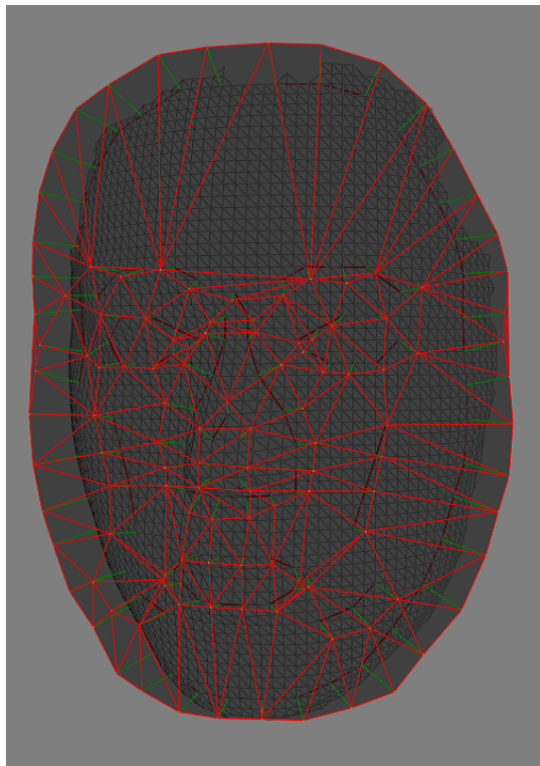


図 3.9: 生成した補間形状

図 3.8 は入力形状から指定した特徴稜線と要素サイズ、図 3.9 は生成した補間形状である。最後に、要素サイズを指定していない頂点に対し Delaunay 角形分割によって生成した補間形状から要素サイズを求める。要素サイズを指定していない頂点 P から法線方向へ直線を引き、直線と補間形状との最短交点 V までの長さを要素サイズとする。 V は次のように求める。まず頂点 P から法線方向へ引いた直線上に点 Q を置き、直線を実数 t を用いて以下のようにおく。

$$(1 - t)P + tQ \tag{3.4}$$

次に補間形状のある面 L の 3 点 ABC に対し、 L 上の任意の点 V は実数 u 、 v を用いて次のように表す。

$$V = A + uL + vM \quad (L = B - A, M = C - A) \tag{3.5}$$

式 (3.4)、式 (3.5) からベクトル方程式を導くと次式になる。また、図 3.10 は直線と面の位置関係を示す。

$$u\mathbf{L} + v\mathbf{M} + t(\mathbf{P} - \mathbf{Q}) = \mathbf{P} - \mathbf{A} \quad (3.6)$$

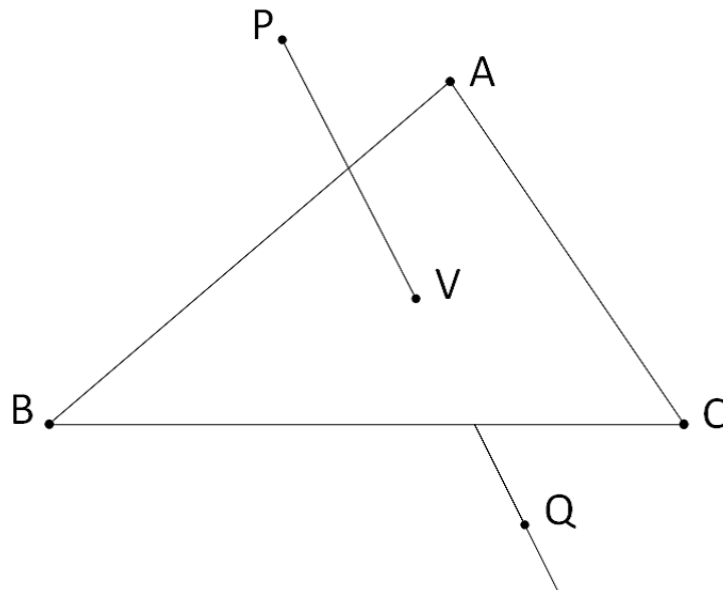


図 3.10: 補間形状からの要素サイズ指定

これを行列式で表すと次式になる。

$$\begin{pmatrix} L_x & M_x & P_x - Q_x \\ L_y & M_y & P_y - Q_y \\ L_z & M_z & P_z - Q_z \end{pmatrix} \begin{pmatrix} u \\ v \\ t \end{pmatrix} = \begin{pmatrix} P_x - A_x \\ P_y - A_y \\ P_z - A_z \end{pmatrix} \quad (3.7)$$

この行列が正則でない場合、平面と直線は平行である。正則な場合は、逆行列を求めてから u 、 v 、 t を求める。 u 、 v 、 t が次式の条件を満たすとき、 V は L の内部に存在し補間形状との最短交点となる。

$$\begin{cases} 0 \leq u \leq 1 \\ 0 \leq v \leq 1 \\ 0 \leq t \leq 1 \\ u + v \leq 1 \end{cases} \quad (3.8)$$

頂点 P に対し、補間形状の全ての面を参照し式 (3.8) を満たす最短交点 V を求め、 P と V の距離を P の要素サイズとする。全ての頂点に対し要素サイズを指定したら、要素サイズを基に全ての頂点上にバブルを配置していく。

3.4 バブルの平方充填

本節では、配置したバブルの平方充填について解説する。バブルの平方充填をする手法として伊藤ら [17] の手法がある。伊藤らの手法では、入力形状全体に稜線方向を制御するベクトル場を指定し、ベクトル場から全てのバブルに対して 4 つの隣接点を与えた。そしてあるバブルとそれ以外のバブルの隣接点間にバブル間力を働かせることでバブルの平方充填を可能とした。本研究でも同様に入力形状に対しベクトル場を設定し、全てのバブルにベクトル場から求まる 4 つの隣接点を与えることでバブルの平方充填を行う。以降にベクトル場とバブルの隣接点、バブルと隣接点間で働くバブル間力についての詳細について述べる。また、バブル間力によるバブルの移動と消滅について解説する。

3.4.1 ベクトル場と隣接点

伊藤らは平方充填するものとして金属結合に着目した。金属結合では、近接した分子が電子を共有すると分子周辺のポテンシャルに異方性が生じ、ポテンシャルが極小となる方向に別の分子を引きつける。この金属結合の過程におけるポテンシャル場と同様に、バブルに分子間力の影響範囲を表すポテンシャル場を仮定し、分子間力の働く方向を隣接バブルの中心方向に限定せずポテンシャルが極小となる隣接点の方向に適応的に仮定することでバブルを平方に充填することが可能である。

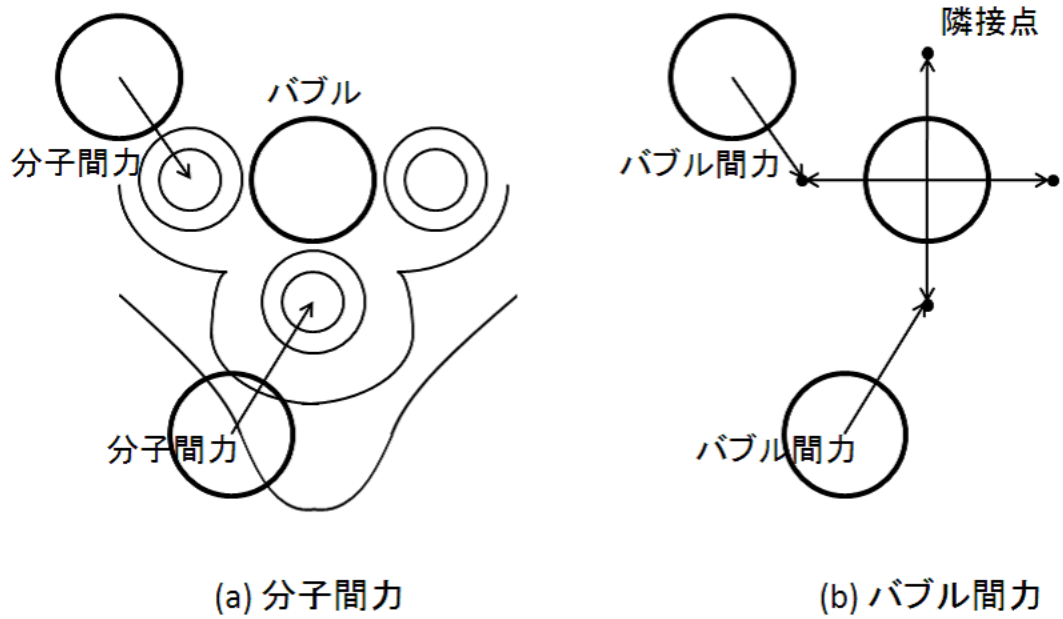


図 3.11: バブル間力のポテンシャル場

図 3.11(a) は金属結合の分子間力のポテンシャル場、図 3.11(b) はバブル間力のポテンシャル場である。図 3.11(b) の黒丸に示した隣接点の方向に近隣バブルにバブル間力を与えている。この隣接点をバブルに与えるため、伊藤らは入力形状全体に稜線方向を指示するベクトル場を与えた。ポテンシャルが極小となる隣接点は、ベクトルに平行、および垂直な方向に与えられる。

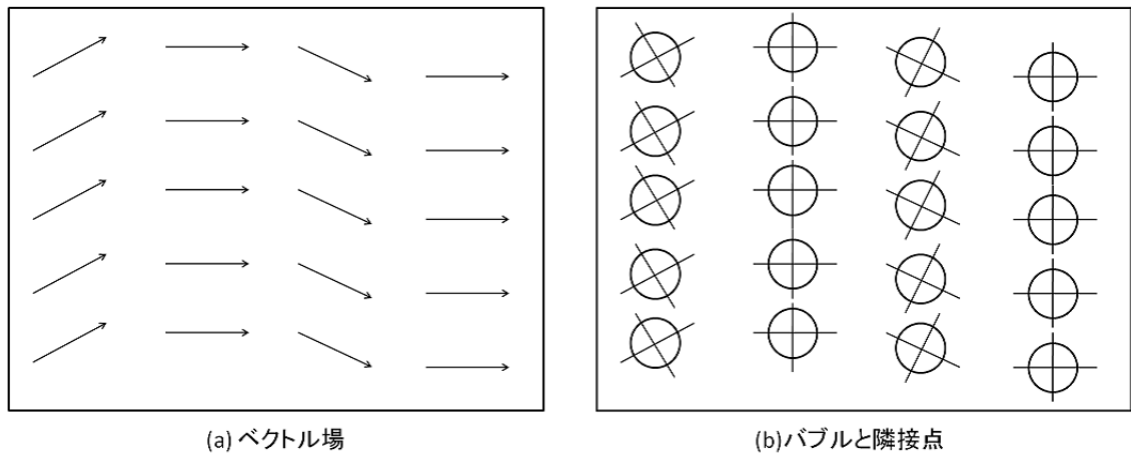


図 3.12: ベクトル場と隣接点

図 3.12(a) はベクトル場、図 3.12(b) はバブルと隣接点を示す。入力したベクトル場に沿って隣接点が配置されていることが分かる。本研究でも同様にポテンシャル場を仮定し、ポテンシャルが極小となる隣接点をバブルに配置することでバブルを平方充填する。まず、入力形状に対して稜線方向を指示するためのベクトル場を設定する。ベクトル場は複数の単位ベクトルからなるもので領域内に対し方向を示す。本研究では対象とするモデルを顔としているため、顔の筋肉や骨の流れに沿ったベクトル場を入力形状の表面に設定する。以下に設定したベクトル場を示す。

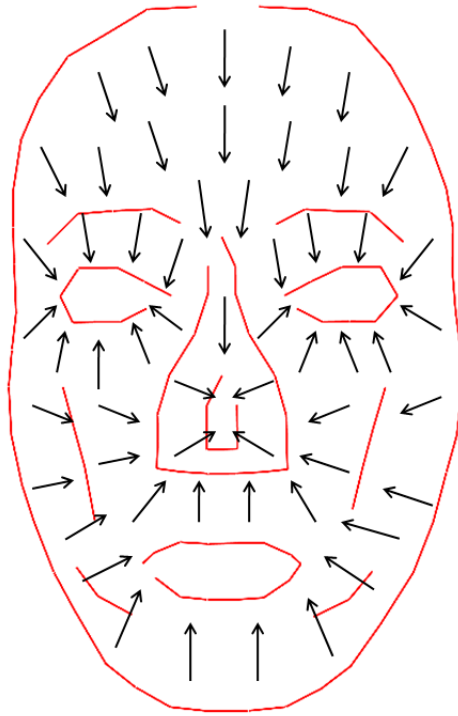


図 3.13: 顔に適したベクトル場

次に、設定したベクトル場からバブルの隣接点を求める。本手法ではバブル間力の方向を任意のバブルの中心とその他のバブルの隣接点とする。あるバブル A の中心位置を \mathbf{x}_A 、半径を r_A 、中心位置におけるベクトル場の方向を \mathbf{v}_A とし、バブルを配置した頂点の法線ベクトル \mathbf{N} とベクトル \mathbf{v}_A の外積から得るベクトルを \mathbf{v}_B とする。また、 \mathbf{v}_A 、 \mathbf{v}_B は単位ベクトルとする。バブル A に隣接する 4 点 \mathbf{x}_{A1} 、 \mathbf{x}_{A2} 、 \mathbf{x}_{A3} 、 \mathbf{x}_{A4} は以下のように求める。

$$\begin{aligned} \mathbf{x}_{A1} &= \mathbf{x}_A + 2r_A\mathbf{v}_A, \quad \mathbf{x}_{A2} = \mathbf{x}_A - 2r_A\mathbf{v}_A \\ \mathbf{x}_{A3} &= \mathbf{x}_A + 2r_A\mathbf{v}_B, \quad \mathbf{x}_{A4} = \mathbf{x}_A - 2r_A\mathbf{v}_B \end{aligned} \quad (3.9)$$

この隣接点を入力形状の表面に配置することによって、ベクトル場に沿ったバブルの整列制御が可能である。また、平方充填においてバブルが形状の表面を移動することができ、入力形状と再構築したメッシュの形状誤差を小さくすること

ができる。

3.4.2 バブル間力の算出

あるバブルとそれ以外のバブルの隣接点間に働くバブル間力は、あるバブルの中心点とそれ以外のバブルの隣接点が接しているときに力が働かず、離れているときに引力が働く。3.4.1項で求めた隣接点から、バブルAがバブルBに与えるバブル間力 F_{AB} を求める。まず、バブルAの隣接点からバブルBに最も距離が近い隣接点 x_{Ac} を選び、その距離 d_c を算出する。そしてバブルBの中心位置を x_B 、半径を r_B とし、以下の式からバブル間力 F_{AB} を算出する。図3.14は本手法での分子間力の方向と分子間力のグラフを示す。

$$F_{AB} = |f(x)| \frac{\mathbf{x}_{Ac} - \mathbf{x}_B}{|\mathbf{x}_{Ac} - \mathbf{x}_B|} \quad (3.10)$$

$$x = \frac{d_c}{r_A + r_B} \quad (3.11)$$

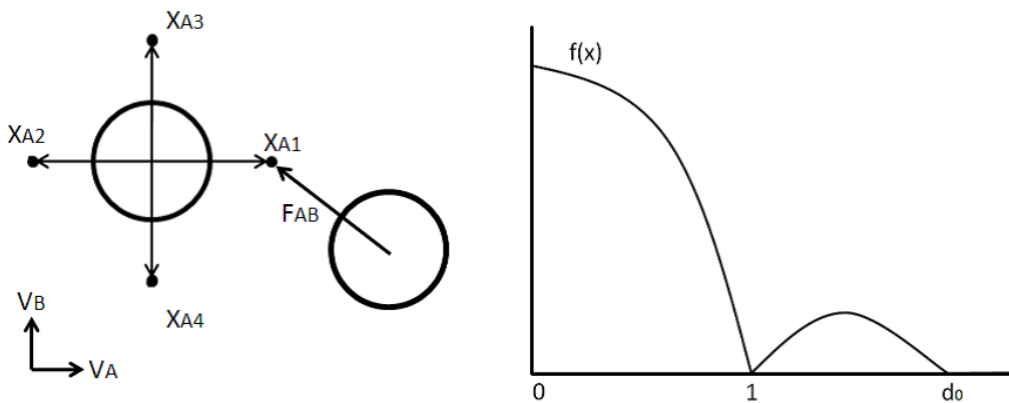


図 3.14: バブル間力の方向

パラメータ x は d_c とバブルが接した状態の距離 $r_A + r_B$ の比である。 $f(x)$ はバブル間力の大きさを表す距離の関数で以下の式となる。

$$f(x) = Ax^3 + Bx^2 + Cx + D \quad (3.12)$$

$$\begin{cases} f(1) & = 0.0 \\ f(d_0) & = 0.0 \\ f'(0) & = 0.0 \\ f'(-1) & = \frac{k_0}{r_0} \end{cases} \quad (3.13)$$

A 、 B 、 C 、 D は式 (3.13) に与えられた 4 つの境界条件より求める定数である。 d_0 はバブル間力が働く最大の距離、 k_0 はバブルが接した状態の距離で生じるバブル間力を線形バネに置き換えたときのバネ定数に相当するものである。

3.4.3 バブルの移動と消滅

バブルの移動はあるバブルに働くバブル間力の総和から算出する。バブル A は次の運動方程式によって移動する。

$$F_A = m_A \frac{d^2 \mathbf{x}_A}{dt^2} + c \frac{d\mathbf{x}_A}{dt} \quad (3.14)$$

F_A はバブル A に働くバブル間力の総和、 m_A はバブルの質量、 \mathbf{x}_A はバブルの位置、 c は粘性を表す定数である。これを微小時間 τ で離散化したとき、 τ 後のバブルの位置 $\mathbf{x}_A(t+\tau)$ は次式で与えられる。

$$\mathbf{x}_A(t+\tau) = \frac{F_A \tau^2 + (2m_A + c\tau)\mathbf{x}_A(t) - m_A \mathbf{x}_A(t-\tau)}{m_A + c\tau} \quad (3.15)$$

1 つのバブルが移動すると他の全てのバブルの移動に影響を与える。そのため、1 つのバブルの移動量を計算して移動すると最密充填に達するまでの計算量が膨大になる。したがって、配置した全てのバブルに対して移動量の計算を独立に行い、全てのバブルを同時に移動することで計算量を軽減する。

このバブルの移動過程でバブルの密度が極度に高い場合、バブルを消滅する。消滅条件は、任意のバブル A (半径 r_A) に対して、バブル A から B へのバブル間距離 d_{AB} が $d_{AB} < r_A$ を満たすときとする。また、バブルの消滅を微小時間 τ ごとに行うと、バブルが移動しきる前に消滅をする恐れがある。そこで、本研究ではバブルの消滅を 5τ ごとに行う。

3.5 バブルからの4角形メッシュ生成

本手法では、バブルの中心点から Delaunay3 角形分割を用いて 3 角形メッシュを生成し、3 角形メッシュの 4 角形化を行うことで 4 角形メッシュを生成する。3 次元モデルを 3 角形メッシュで生成するか 4 角形メッシュで生成するかは 3DCG コンテンツによって異なる。本研究においては、メッシュの細かさや稜線方向が把握しやすい 4 角形メッシュにて再メッシュ化を行う。Delaunay3 角形分割は 3.3.2 項にて解説を行ったので詳細は割愛し、本節では脇田ら [25] が提案した 3 角形メッシュの 4 角形化について解説する。4 角形化は先に述べた様にメッシュの細かさや稜線方向を把握しやすく、ポリゴン数とデータサイズの削減においても有効である。

Delaunay3 角形分割にて生成した 3 角形メッシュを形成する稜線は、図 3.15 に示すように 2 つの 3 角形面で構成している。任意の稜線 e に接続する 2 つの 3 角形面によって構成する 4 角形面が長方形に近い場合、その稜線を削除し 4 角形化を行う。4 角形化は 4 角形面が長方形に近いか否かを評価する関数としきい値を用いて判定を行う。稜線 e に接続する 2 つの 3 角形面の頂点をそれぞれ \mathbf{p}_0 、 \mathbf{p}_1 、 \mathbf{p}_2 と \mathbf{p}_0 、 \mathbf{p}_2 、 \mathbf{p}_3 とし、用いる評価関数を以下に示す。

$$|F(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)| + |F(\mathbf{p}_0, \mathbf{p}_2, \mathbf{p}_3)| < S_1 \quad (3.16)$$

$$|F(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)| + |F(\mathbf{p}_0, \mathbf{p}_2, \mathbf{p}_3)| - |F(\mathbf{p}_1, \mathbf{p}_0, \mathbf{p}_2)| + |F(\mathbf{p}_0, \mathbf{p}_3, \mathbf{p}_2)| < S_2 \quad (3.17)$$

ただし、 $|F(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)|$ は 3 角形メッシュ内のベクトルの内積で稜線の対頂角を表し、次式によって求めることができる。

$$|F(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)| = \frac{\mathbf{p}_i - \mathbf{p}_k \cdot \mathbf{p}_j - \mathbf{p}_k}{|\mathbf{p}_i - \mathbf{p}_k| |\mathbf{p}_j - \mathbf{p}_k|} \quad (3.18)$$

式 (3.16) は、稜線 e を挟んだ面の対頂角についてそれぞれが直角に近いか否かを判定する関数であり、 S_1 を実数のしきい値とし 0 に近いほど直角に近く精度を高めることができる。式 (3.17) は、稜線 e の両端における角度の和の差を判定する関数であり、 S_2 を実数のしきい値として 0 に近づくほど差が小さくなり、より

長方形に近くなるため精度を高めることができる。この2つの関数を用いることで長方形に近い4角形を判定することが可能となる。

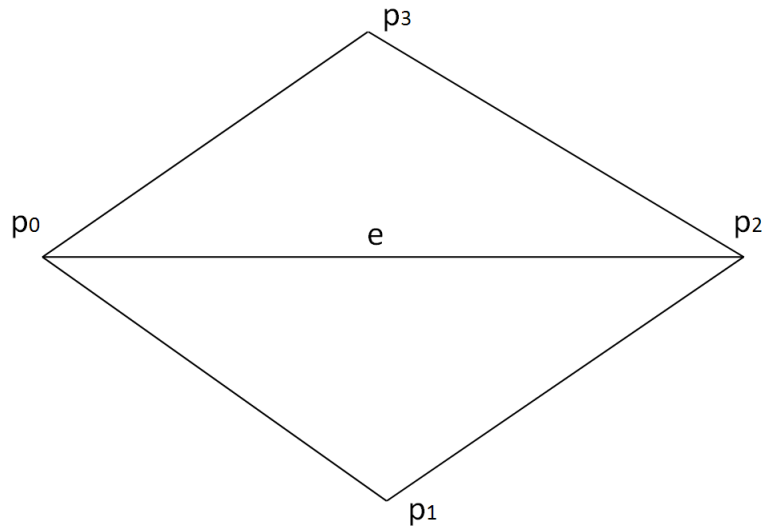


図 3.15: 3 角形メッシュの 4 角形化

第 4 章

再構成結果

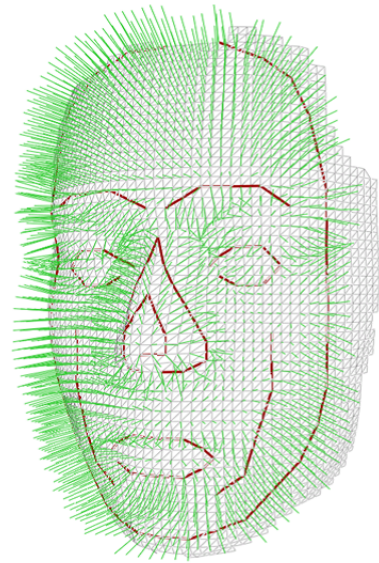
本章では、前章で提案した平方充填バブルメッシュ法を用いた 3D スキャナデータの再メッシュ化手法をツールで実装し得た結果を示す。3D スキャナデータから直接特徴稜線を指定し、特徴稜線を適用した平方充填バブルメッシュ法にてメッシュの再構築を施し、その結果について考察する。実装においては C++ 言語を用い、グラフィックスライブラリには OpenGL の上位ライブラリである渡辺らの FK システム [26] を用いてツールを作成した。

4.1 適用実験

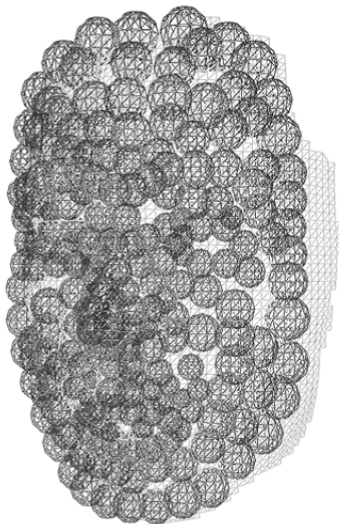
本手法の有効性を検証するため、3D スキャナで生成した顔の 3 次元モデルを再メッシュ化したモデルを作成した。以下に実行結果を示す。



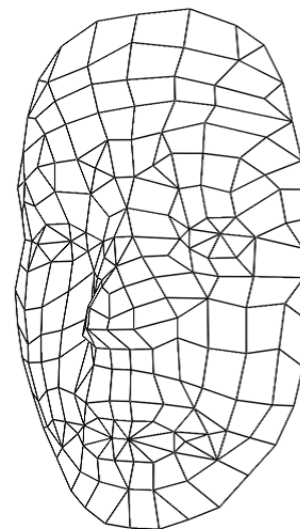
(1) 入力形状(7000ポリゴン)



(2) 特徴稜線と要素サイズ指定



(3) バブル充填



(4) メッシュ生成(230ポリゴン)

図 4.1: 3D スキャナデータに対する本手法の適用

図 4.1 は、3D スキャナで生成した顔の 3 次元モデルに対して本手法を適用した結果である。図 4.1(a) は入力形状を表し、図 4.1(b) は指定した特徴稜線と要素サイズを示す。要素サイズの指定は、近似ガウス曲率を用いた指定方法で行った。図 4.1(c) は指定、および分布した要素サイズを基に配置し平方充填したバブルを表

し、図 4.1(d) はバブルの中心点から 4 角形メッシュを生成したモデルを示す。本手法で再構築したメッシュの検証として、入力形状と本手法を適用した 3 次元モデルとの誤差判定と、修正が容易かどうかを検証するためのモデリングソフトでのメッシュ修正を行った。

入力形状との誤差判定は、まず本手法を適用した 3 次元モデルの頂点とその頂点に一番近い入力形状の頂点それぞれの原点からの距離を求める。そして、入力形状の頂点から原点までの距離を実測値 T 、本手法を適用した 3 次元モデルの頂点から原点までの距離を計算値 M とし、誤差率 ϵ を求め形状誤差を検証した。誤差率 ϵ は以下の式から求める。

$$\epsilon = \frac{|M - T|}{T} \times 100 \quad (4.1)$$

以下に入力形状と本手法を適用した 3 次元モデルとの形状誤差の判定結果を示す。顔の大きさは縦 24cm、横 15cm、奥 14cm と実際に測定した顔の大きさに設定した。一般的な形状誤差の判定において誤差率の許容範囲は 5%以下となっており、平均誤差率が許容範囲内であれば入力形状との誤差は少ない。また、特徴稜線上の頂点はバブルの平方充填において移動せず誤差がないため判定には含んでいない。判定した頂点数は 70 点である。

表 4.1: 入力形状との誤差判定

	実測値 (cm)	計算値 (cm)	誤差 (cm)	誤差率 (%)
誤差率最小	2.95	2.96	0.01	0.04
誤差率最大	1.15	1.03	0.12	10.43
誤差率平均				2.03

表 4.1 は入力形状と本手法を適用した 3 次元モデルとの形状誤差の判定結果である。実測値、計算値、誤差については、誤差率が最小と最大であった頂点の値を示している。一部、口の領域内の頂点にて大きい誤差がでたものの、それ以外の頂点に関しては僅かな誤差が生じた程度である。平均誤差率は 2.03%となっており許容範囲内の誤差率を示す結果となった。この結果から、本手法を適用した 3 次

元モデルは入力形状とほぼ同様の形状を表現できていると言える。

次に、本手法で再構築したメッシュが修正が容易かどうかを検証するため、3D スキャナデータと本手法を適用した3次元モデルそれぞれをモデリングソフトで読み込み、手作業でメッシュの修正を施した。モデリングソフトはMetasequoia[27]を使用した。3D スキャナデータは頂点数が膨大で、かつ均一で稜線方向が規則的なメッシュ構成をしている。メッシュの修正を施すにあたり、まず頂点数の削減を行い、その後メッシュの細かさや稜線方向に対する修正を施した。頂点数は複数の頂点を選択し結合することで削減を行い、メッシュの細かさや稜線方向は頂点の移動や結合、面の分割を施すことで行った。このように、3D スキャナデータのメッシュ修正には複数の操作、設定を幾度となく繰り返す必要があり、修正に約1、2時間を費やすことになった。一方、本手法を適用した3次元モデルは、頂点数が少なく形状の特徴に応じたメッシュの細かさで、かつ筋肉や骨の流れに沿った稜線を持つメッシュ構成をしている。

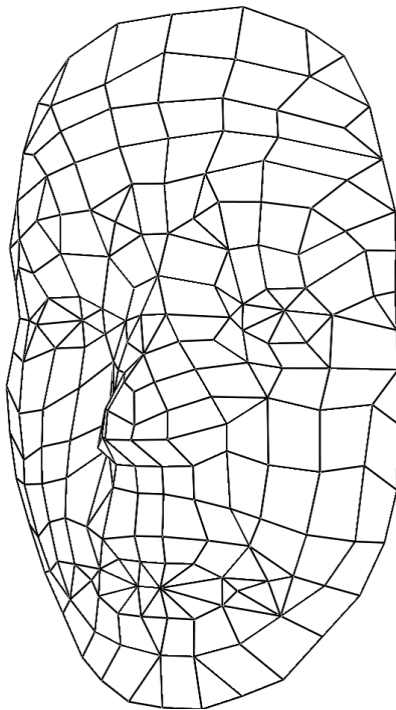


図 4.2: 本手法を適用したモデル

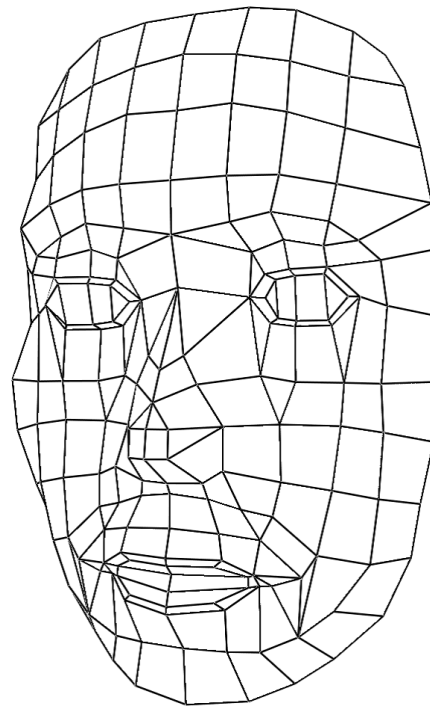


図 4.3: メッシュ修正を施したモデル

図 4.2 は本手法を適用した 3 次元モデル、図 4.3 は図 4.2 の 3 次元モデルに対しメッシュ修正を施した 3 次元モデルである。メッシュの細かさを制御したことによって頂点数を削減することができ、手作業による頂点数削減をする必要はほぼなく、メッシュ修正に必要な操作や設定も容易にできた。メッシュの細かさや稜線方向に対する修正は、一部頂点の結合をする必要があったものの 3D スキャナデータに比べ少ない操作で済ませることができた。頂点の結合を施した後は、頂点の位置を微調整するのみでメッシュの細かさや稜線方向の修正を施すことができ、3D スキャナデータよりも少ない時間で修正が可能だった。以上のことから、本手法を適用した 3 次元モデルは 3D スキャナデータに比べ修正を施すための操作が容易にでき、少ない操作と修正時間で 3DCG コンテンツに適したモデルへと修正できた。

4.2 考察

本手法により、入力形状からキャラクターラインを特徴稜線として指定し、特徴稜線を適用した平方充填バブルメッシュ法にて再メッシュ化することは可能であり、ユーザーの求めるメッシュの細かさや稜線方向を反映した修正が容易なメッシュを構築することができた。要素サイズ、ベクトル場をユーザーが任意に指定することで、メッシュの細かさや稜線方向の制御を可能とした。構築したメッシュは特徴的な部位はメッシュが細かくそれ以外の部位はメッシュが粗くなっており、稜線方向は筋肉や骨の流れに沿っている。さらに、4 角形メッシュで構成することでメッシュの細かさ、稜線方向が把握しやすく形状の修正が容易になっている。

本手法の問題点として、第 1 に特徴稜線抽出がある。現状、特徴稜線は 2 つの頂点を選択することで指定しており、ユーザーが求めるキャラクターラインを直感的かつ的確に指定可能である。しかし、頂点数が膨大な 3D スキャナデータから頂点を選択して特徴稜線を指定するには、大幅な時間がかかってしまい非効率的である。また、大きさが均一なメッシュのため形状の特徴が把握しづらく、現状の特徴稜線指定にはユーザーのモデリングに関する知識や経験が問われることになる。この問題は、既存手法を基により良い評価関数による特徴稜線の自動抽出

手法を提案するか、武部 [28] が提案したタブレットインターフェースなどの直感的なデバイスを用いた特徴稜線指定により対応が可能であると考え。また、顔の特徴をパターン化し 3D スキャナデータに適用しこれを特徴稜線とすることで、自動的に適切な特徴稜線を指定することも可能と考える。

第 2 に要素サイズの補間形状生成がある。本手法では、指定した要素サイズから得る頂点群と Delaunay3 角形分割を用いて補間形状を生成した。これにより、指定した要素サイズを基に領域全体に要素サイズを分布することが可能となった。しかし、現状の補間形状では曲線を多用した 3 次元モデルや曲率が大きい部位などでは周囲と大きく異なる要素サイズを指定してしまう可能性がある。これは歪んだメッシュを生成する要因になってしまうため、要素サイズの補間形状生成手法の改善が必要である。ポリゴンではなく曲面によって入力形状を覆う滑らかな補間形状を生成するか、山田らが用いた薄板の曲げモデルの歪みエネルギーを最小化する手法を拡張するなどの対処法が挙げられる。

本研究の検証実験では、3D スキャナデータと本手法を適用した 3 次元モデルに対し、形状の誤差判定とメッシュ修正の容易さの検証を行った。しかし、今回の実験においては実際の 3DCG コンテンツ制作において本手法がモデリングの制作時間や制作の手間の削減を行ったかどうかの検証をしていない。また、テクスチャ、アニメーションといったモデリング後の作業工程において本手法を適用した 3 次元モデルが有効であるか否かの検証も行っていない。そのため、実際の 3DCG コンテンツ制作における実証実験が必要である。

第 5 章

おわりに

本論文では、3D スキャナデータからユーザーが任意に制御したメッシュの細かさや稜線方向を反映した修正が容易なメッシュの再構築手法の提案をした。従来の手法では、メッシュの細かさや稜線方向は頂点数削減処理の結果に左右する、限られた制御しかできないなど、ユーザーが求めるメッシュを構築することが困難であった。そのため、再構築後もユーザーの求めるメッシュへと修正を施す必要があるため課題となっていた。本研究では、3D スキャナなど実物体を測定し生成した3次元モデルからユーザーがメッシュの細かさや稜線方向を任意に制御し、これらを反映した修正が容易なメッシュを再構築することでメッシュ修正の時間を削減し3D スキャナデータをより実用的なものにすることを目的とした。3次元形状の特徴を表す概念であるキャラクターラインを特徴稜線として指定し、特徴稜線を適用した平方充填バブルメッシュ法によって再メッシュ化を図った。特徴稜線を指定することで、形状の特徴を考慮した再メッシュ化が可能になった。また、平方充填バブルメッシュ法を用いることで、ユーザーがメッシュの細かさや稜線方向を任意に制御することが可能になり、ユーザーが求める修正が容易なメッシュの再構築を実現した。

検証実験として、3D スキャナで生成した顔のポリゴンモデルに対し本手法を適用した。その結果、指定した要素サイズとベクトル場によってメッシュの細かさや稜線方向を制御でき、再構築したメッシュは本研究で定義した修正が容易なメッシュの要件を満たすことができた。特徴稜線を再メッシュ化に適用することで、形状の特徴を保持したメッシュを再構築できた。また、本手法で再構築したメッシュに対し入力形状との誤差判定とメッシュの修正のしやすさを検証した。入力形状との誤差判定は、再構築したメッシュの頂点と入力形状の頂点の距離で判定を行った。その結果、多くの頂点では僅かな誤差が出る程度であったため、本手法を適用した3次元モデルは入力形状とほぼ変わらぬ形状であることが実証できた。次に、本手法を適用した3次元モデルが修正が容易かどうかを検証するため、モデリングソフトで読み込み手作業にてメッシュの修正を施した。本手法を適用した3次元モデルは、3D スキャナデータに比べメッシュの細かさや稜線方向の修正が容

易であり、僅かな時間で3DCGコンテンツに適したモデルへと修正できた。メッシュの細かさの制御によって頂点数を削減することができ、メッシュを修正するための操作も容易にできた。これらの結果から、本手法を適用することにより3Dスキャナデータに対するメッシュの修正時間を短縮することができ、3Dスキャナデータがより実用的になりモデリングの作業効率を向上することが可能となった。

本手法の課題として、第1により良い特徴稜線の抽出がある。本手法では、3Dスキャナデータの頂点群から2つの頂点を指定し、頂点間に稜線を生成しこれを特徴稜線とした。ユーザーの求めるキャラクターラインを直感的に指定できるものの、特徴稜線を形成する頂点を選択するには3Dスキャナデータの頂点数が膨大なため大幅な時間がかかってしまう。また、CADや3DCGでも通じるキャラクターラインの明確な定義を行う必要もある。次に要素サイズの補間形状の生成がある。本手法では、初期入力である要素サイズから得る頂点群とDelauny三角形分割を用いて補間形状を生成した。少ない入力で要素サイズの分布が可能となったが、曲率の大きい部位では周囲とは大きく異なる要素サイズを指定してしまう恐れがある。これを解決するには、得た頂点群から入力形状を覆う曲面の補間形状を生成することで連続的に変化する要素サイズを指定可能と考える。これらの課題を解決することによって、3DCGコンテンツ制作において3Dスキャナデータをより有効に活用することが可能となるだろう。

謝辭

本研究を締めくくるにあたり、学部時代から引き続きまして研究の方針から開発の手法、論文の執筆と幅広いご指導ご教授を頂きました、本校メディア学部 渡辺大地講師に深く感謝致します。また、副査をお引き受け下さり適切な助言を頂きました、本校メディア学部 宮岡伸一郎教授、三上浩司講師に深く感謝致します。

そして、3年間プログラムの実装方法や研究方針など、様々なご協力を頂いた竹内亮太先輩を始めとするゲームサイエンスプロジェクトの諸氏、ならびに本校大学院メディアサイエンス専攻の学友諸氏に深く感謝致します。

最後に、本研究にご協力頂きました全ての皆様と、この論文に目を通して下さった全ての方々に、厚くお礼を申し上げます。

参考文献

- [1] Autodesk, "Maya", <http://www.autodesk.co.jp/>.
- [2] NewTek, "3dsMAX", <http://newtek.com/>.
- [3] "CGWORLD Vol.98", WORKS CORPORATION, pp.50-51, 2007.
- [4] "CGWORLD Vol.106", WORKS CORPORATION, pp.44-47.64-67, 2007.
- [5] T.J.H.Hoppe, T.DeRose and W.Stuetzle, "Surface reconstruction from unorganized points", SIGGRAPH Proceedings, 1992.
- [6] J.D.Boissonnat, "Geometric structures for three-dimensional shape representation", ACM Transactions on Graphics, 1984.
- [7] 芹田 陽一郎, "ポリューミング手法を用いた不規則点群からのデータからのポリゴンモデル自動再構成", 慶應義塾大学博士論文, 2001.
- [8] 大久保 隆, "不規則点群データによるポリゴンモデルの形状編集が容易な曲面モデルへの再構成手法", 東京工科大学修士論文, 2005.
- [9] V.Surazhsky, "Esplicit Surface Remeshing", Eurographics Symposium, 2003.
- [10] 金井 崇, "対話的なメッシュの均一再メッシュ化手法", グラフィクスとCAD/VisualComputing 合同シンポジウム 2001, pp.91-96, 2001.
- [11] 森田 伊織, "3D 物体表面の特徴パラメータを利用した再メッシュ化手法", 電子情報通信学会技術研究報告 Vol106 No.229, pp.9-16, 2006.
- [12] 森田 伊織, "3次元表面の特徴に基づく等方構造への再メッシュ化", 情報処理学会研究報告 Vol2.2007No96, pp.67-72, 2007.
- [13] Y.Lai, Zhou, S.Hu, J.Qallner, H.Pottman, "Feature sensitive geometry processing", IEEE Trans, Vis.Comp.Graphics, 2006.

- [14] H.Pottmann, T.Steiner, M.Hofer, C.Haider, A.Hanbry, "The isophotic metric and its applications to feature sensitive morphology on surfaces", Computer Vision -ECCV 2004 Part 6, 2004.
- [15] K.Shimada, "Physically-Based Mesh Generation:Automated Triangulation of Surfaces and Volumes via Bubble Packing", Massachusetts Institute of Technology, 1993.
- [16] 山田 敦, "バブルメッシュ分割法のための要素サイズの制御方法", テレビジョン学会 Vol.20No.27, pp.1-6, 1996.
- [17] 伊藤 貴之, "平方充填バブルメッシュ法による自動四角メッシュ生成", 情報処理学会グラフィクスと CAD 研究会 Vol.97No.98, pp.7-12, 1997.
- [18] E.H.Spanier, "Algebraic topology", 1996.
- [19] 渡辺 大地, "任意三角形メッシュからの特徴稜線抽出", 電子情報通信学会論文誌 D-II Vol.J83-D-IINo.5, pp.1344-1352, 2000.
- [20] 野村 優, "細線化処理を利用した三次元メッシュ形状の特徴稜線抽出", 電子情報通信学会 Vol.100No.443, pp.49-56, 2000.
- [21] 早野 勝之, "特徴抽出と稜線操作によるポリゴンメッシュの簡単化", 情報処理学会グラフィクスと CAD 研究会報告 Vol98No16, pp.37-42, 1998.
- [22] C.R.Calladine, "Gausslan curvature and shellstructures", Oxford University Press, pp.179-196, 1986.
- [23] O.C.Zienkiewicz, "The Finite Element Method Third Edition", McGraw-Hill, United Kingdom, 1977.
- [24] M. ドバーグ, 浅野 哲夫, "コンピュータ・ジオメトリ -計算幾何学:アルゴリズムと応用", 近代科学社, 2001.

- [25] 脇田 玲, ”医用ラティス構造に基づく web3d モデリング”, 慶応義塾大学修士論文, 2001.
- [26] 渡辺大地, ”FK Tool Kit System”, <http://fktoolkit.sourceforge.jp/>.
- [27] Mizuno Osamu, ”Metasequoia Ver2.4.9”, <http://www.metaseq.net/metaseq/index.html>.
- [28] 武部 佳文, ”タブレットインターフェースを用いたラティス構造のモデリング”, 慶応義塾大学修士論文, 2001.