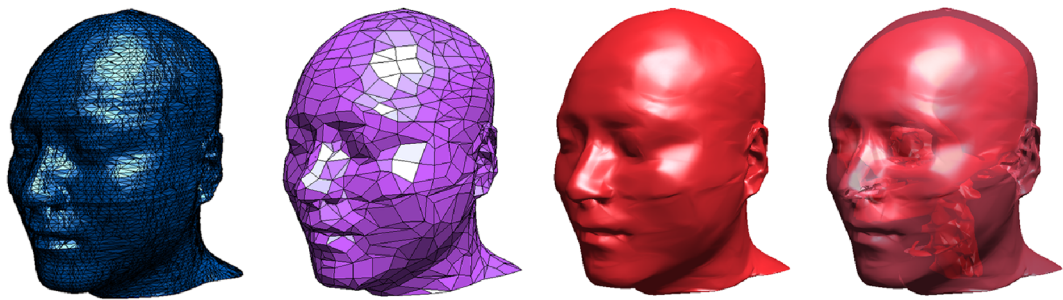


医用ラティス構造に基づく
Web3Dモデリング

WEB3D MODELING
BASED ON MEDICAL LATTICE
STRUCTURE



脇田 玲

AKIRA WAKITA

慶應義塾大学 大学院 政策・メディア研究科

GRADUATE SCHOOL OF MEDIA AND GOVERNANCE

KEIO UNIVERSITY

医用ラティス構造に基づく Web3D モデリング

コンピュータ支援外科においては、患部を可視化する目的から、CT スキャナや MRI によって取得される 2 次元画像を元に 3 次元再構成を行う。3 次元モデルを構築することにより、2 次元画像では困難であった直観的な形状把握と術前のシミュレーションが実現される。3 次元再構成において一般的に利用される形状表現は 3 角形メッシュである。3 角形メッシュは 3 角形平面の集合によって物体形状を表現する手法であり、データ構造が単純で扱い易いために、多くのスタンドアロン型のシステムで利用されている。

一方、インターネット上で 3 次元 CG を利用する Web3D 技術が大きな注目を集めている。コンピュータ支援外科において Web3D を利用することにより、遠隔地間での診察や手術シミュレーション、オンラインカルテなどの発展が期待されている。しかし、従来の 3 角形メッシュに基づく患部の 3 次元モデルはデータサイズが膨大であり、配信に多くの時間を必要とする。また、3 角形メッシュは極めて一般的な形状表現であり、医用に特化した形状表現ではない。そのため、手術シミュレーションの高度な用途に必要な、柔軟な変形操作や特徴領域の明示的な可視化も困難である。Web3D としての利用が可能であり、かつ医用に特化した形状表現とそのモデリング手法が必要とされている。

本研究では、Web3D 型の手術シミュレーションを実現するために、医用ラティス構造という新しい形状表現とそのモデリング手法を提案する。医用ラティス構造は自由曲面に基づく形状表現であり、少ない制御点でなめらかな形状を表現することが可能である。そのため、データサイズが非常に小さく、高速なデータ転送が可能である。更に、単純な多面体によって曲面形状を操作することが可能であるため、柔軟な変形操作を実現することができる。また、特徴領域の指定や精度の適応的な制御が可能であるために、手術シミュレーションに必須な切断線や組織境界線の明示的な可視化を実現している。本手法によって作成した複数のアプリケーションを評価し、手法の有効性を検討した。

キーワード

1. Web3D	2. コンピュータ支援外科
3. ラティス構造	4. 3 次元再構成
5. 幾何形状圧縮	

慶應義塾大学 大学院 政策・メディア研究科

脇田 玲

Web3D Modeling Based on Medical Lattice Structure

In Computer Aided Surgery, 3D models are often reconstructed to visualize patient's affected parts spatially. 3D Reconstructions mean building up 3D shapes from a set of CT/MRI images. This technique is used for surgical simulations, intuitive diagnosis, medical treatments and so on. The most popular shape representation used in 3D reconstructions are triangle meshes. Triangle meshes consist of planer triangles and therefore are easy to implement. Utilizing this advantage, many stand-alone systems are built based on triangle meshes.

Web3D is one of the most innovative topics in computer graphics. Taking advantage of Web3D, surgical simulations among physically apart doctors or online bookkeeping of 3D medical records can be realized. However, 3D models used in most of Web3D-based medical systems are also based on the same triangle meshes as stand-alone systems. Transferring triangle meshes requires quite a long time. In addition, reflecting character lines and controlling the accuracy adaptively, which are indispensable for medical use, are not supported. Effective shape representations for Web3D as well as for medical use are strongly required.

We present Medical Lattice Structure, a new shape representation for Web3d and for medicine. Because Medical Lattice Structure is based on free-form surfaces, the data size is kept small. This advantage realizes fast transmission of 3D data on the Internet. Moreover, intuitive free-form shape deformations using simple polygonal meshes are also possible. Character line inputs and adaptive controls of shape accuracy are also supported for advanced surgical simulations. We have developed web-based medical applications using Medical Lattice Structure for evaluations.

Key Words

1. Web3D	2. Computer Aided Surgery
3. Lattice Structure	4. 3D Reconstruction
5. 3D Geometry Compression	

Keio University Graduate School of Media and Governance

Akira Wakita

目 次

1. 序論	1
1.1 はじめに	2
1.2 本論文の目的・手法	3
1.3 本論文の社会的意義	5
1.4 本論文の構成	6
1.5 本論文の用語	6
2. コンピュータ支援外科	8
2.1 スタンドアロン環境におけるコンピュータ支援外科	9
2.1.1 CT/MRI の利用	9
2.1.2 3次元再構成	9
2.2 WWW におけるコンピュータ支援外科	10
2.2.1 WWW による人体断層画像の配信	10
2.2.2 VRML を用いた Web3D システム	11
3. Web3D のための形状表現	12
3.1 立体表現とデータ構造	13
3.2 曲面形状表現	18
3.2.1 3角形メッシュ	18
3.2.2 自由曲面	20
3.2.3 細分割曲面	23
3.3 曲面フィッティング	27
3.3.1 細分割曲面フィッティング	27
3.3.2 B-Spline 曲面フィッティング	28
3.4 第3章のまとめ	29
4. 医用ラティス構造	30
4.1 ラティス構造	32
4.1.1 ラティス曲面と初期多面体	32
4.1.2 丸め操作と逆丸め操作	32
4.1.3 重み付け	40
4.2 医用ラティス構造	42

4.2.1	特徴の反映	42
4.2.2	精度の適応的制御	44
4.3	CT 画像からの医用ラティス曲面再構成	45
4.3.1	CT 画像からの3角形メッシュ構築	45
4.3.2	3角形メッシュへのラティス曲面フィッティング	53
4.3.3	特徴と精度の入力	63
5.	Web3D アプリケーション	68
5.1	ソフトウェア構成	70
5.2	頭蓋縫合早期癒合症	72
5.3	下顎骨切断シミュレーション	73
5.4	口唇裂 3D テキストブック	76
5.5	CT/MRI ビューア	80
5.6	下顎骨運動ビューア	84
6.	曲面形状の構築例と考察	87
6.1	曲面形状の構築	88
6.2	精度の適応的制御と特徴線の反映	91
6.3	詳細度制御	100
6.4	変形操作	103
7.	結論と展望	107
7.1	結論	108
7.2	今後の展望	109
	付録	110
	参考文献	120
	発表論文	125
	謝辞	127

図 目 次

3.1	形状が一意に判断できないワイヤーフレームモデル	14
3.2	位相要素と幾何要素の関係	16
3.3	winged-edge 構造とハーフエッジ構造	17
3.4	3 角形メッシュ	18
3.5	3 角形メッシュによる区分線形近似	19
3.6	双 3 次 Gregory パッチ	20
3.7	自由曲面 (双 3 次 Gregory パッチ) によって表現されている 3 次元モデル	22
3.8	Doo-Sabin 細分割曲面	24
3.9	face point と edge point	24
3.10	Doo-Sabin 細分割のステップ	25
4.1	ラティス構造	33
4.2	ラティス曲面の頂点計算	35
4.3	ラティス曲面の曲線計算	36
4.4	双 3 次 Gregory パッチ間の接続	37
4.5	G^1 連続の条件	38
4.6	基礎パッチによる曲面の内挿	39
4.7	重み付け	41
4.8	ラティス構造による医用 3 次元モデル	42
4.9	特徴線の反映	43
4.10	精度の適応的制御	44
4.11	構成物質による CT 値の分布	46
4.12	CT 値による人体組織のめやす	47
4.13	CT 画像と 2 値化処理後の画像	48
4.14	マーチングキューブス法の 15 個のパターン	50
4.15	3 次元再構成再編集システムの概観	51
4.16	3 次元再構成例 (人体の頭蓋)	52
4.17	稜線消去	53
4.18	面の方程式	54
4.19	ラティス曲面フィッティングにおける QEM の評価点	57
4.20	頂点近傍の参照点の算出 - 1	58
4.21	頂点近傍の参照点の算出 - 2	59

4.22	頂点近傍の参照点の算出 - 3	60
4.23	3 角形同士のペアリング	65
4.24	1 つの評価関数だけでは検知できない 3 角形のペア	65
4.25	4 角形面と 3 角形面への双 3 次 Gregory パッチの内挿	66
4.26	ディスプレイタブレットによる入力インターフェイス	67
5.1	Web3D アプリケーションのソフトウェア構成	71
5.2	頭蓋縫合早期癒合症の手術前後ビジュアライゼーション	72
5.3	矢状骨きりと垂直骨きり	73
5.4	ディスプレイタブレットによる切断線の入力	74
5.5	切断シミュレーション後の下顎骨モデル	75
5.6	口唇裂 3D テキストブックのスナップショット	76
5.7	口唇裂 3D テキストブックのスナップショット (ウォークスルーモード)	77
5.8	口唇裂手術のデザインと切断	78
5.9	口唇裂手術の切断ステップ	79
5.10	CT/MRI ビューアのスナップショット	80
5.11	切断面の移動	81
5.12	CT 画像と MRI の切り替え	82
5.13	透明度の変更	83
5.14	下顎骨運動ビューアのスナップショット	84
5.15	ヘッドフレームとフェイスボウの装着	85
5.16	3 次元モデルへのマーキング	85
5.17	下顎骨運動の軌跡の表示	86
6.1	下顎骨 (jaw) の実行例	92
6.2	頭蓋骨 (skull) の実行例	93
6.3	頭部表皮 (skin) の実行例	94
6.4	舟状頭 1(scapo-pre) の実行例	95
6.5	舟状頭 2(scapo-aft) の実行例	96
6.6	舟状頭 3(scapo-face) の実行例	97
6.7	特徴線の反映例	98
6.8	特徴線指定時の誤差表示	99
6.9	分割数の変更による詳細度制御	101
6.10	視点の遠近に応じた詳細度制御	102

6.11	ラティス変形	104
6.12	医用ラティス曲面の変形例-口元の変形	105
6.13	医用ラティス曲面の変形例-顎の変形	106

表 目 次

1	位相要素	15
2	幾何要素	15
3	医用ラティス曲面生成の実行結果	89
4	特徴線の有無による曲面生成の誤差評価	91

第 1 章

序論

1.1 はじめに

コンピュータ支援外科 (Computer Aided Surgery, 以下 CAS)[15] は, コンピュータを用いた手術シミュレーション, 医用画像処理, 手術ナビゲーションなどの総称である. 近年のハードウェアの高速化とソフトウェアの高度化に伴い, 急速な発展を遂げている技術である. 手術シミュレーションをコンピュータ上で行うことにより, 事前に何回でも手術の疑似体験をすることが可能になり, 実際の手術における危険性を軽減することが可能である [30]. また, 医用画像処理を利用することで, 従来の診療や診断では発見することが困難な要因を直観的に可視化することも可能である. また, 近年では VR(Virtual Reality) 技術の発展により, 専用の医療デバイスとの併用による技術も進歩しており, 高齢社会において大きな役割を果たすことが期待されている.

CAS においては, 人体内部の断層画像を非侵襲に観察することができる医療機器が頻繁に用いられる. X 線 CT(Computed Tomography) を用いた CT スキャナが最初の断面撮影機器であり, 1971 年に臨床利用が開始されて以来, 現在まで急激な進歩を遂げている. また, 人体からの NMR (核磁気共鳴) 信号をデータ化する MRI(Magnetic Resonance Imaging) も臨床において広く利用されている. 骨などの硬い組織の撮影には CT が適しており, 臓器などの軟部組織には MRI が適している. これらの機器から得られる画像は, フィルムに現像することで診療や診察における重要な資料として利用されている. また, コンピュータに 2 次元画像として取り込み, データの閾値を変更することで, 様々な臓器の画像を強調, 抽出することが可能である.

一方で, これらの断面撮影装置から取得されるデータは, 2 次元画像であるために直観的かつ立体的な形状把握に熟練を要する. また, 手術シミュレーションなどの高度な用途への利用のためには, 3 次元可視化されていることが望ましい. このような理由から, CT/MRI 画像を 3 次元化する 3 次元再構成の研究が盛んに行われてきた. ボクセル法 [40, 36, 12] は最も盛んに研究されてきた手法である. 3 次元空間を X, Y, Z 軸方向の単位格子 (ボクセル) に分割し, 各単位格子が対象物の内部にあるか, 外部にあるかによって形状を表現する手法である. 断面撮影装置から取得される全ての CT スライスについて単位格子を当てはめ, 得られる格子群を直接レンダリングすることで 3 次元形状の可視化が実現される. ボクセル法は自動的な処理が可能のために, 容易に 3 次元再構成を実現することができる. しかし, 高い精度が必要である人体形状の可視化には膨大な数の単位格子が必要であるために, 計算負荷が高く, データサイズも大きくなってしまふ. そこで, 考案されたのがボクセルによって構成される 3 次元モデル (ポリウムモデル) を 3 角形平面によって構成される 3 次元モデル (3 角形メッシュ) に変換する手法である. この手法はマーチングキューブス法 [35] と

呼ばれており、自動的な処理が可能である。また、出力される 3 角形メッシュは物理モデルに基づく手術シミュレーションや、ラピッドプロトタイピングによる人工骨生成にも利用することができる。

一方で、WWW(World Wide Web) を利用した CAS システムへの期待が高まっている。これにより、遠隔地の複数の医師間で、それぞれの専門知識を統合した手術シミュレーションや症例検討を実現することができる。また、地方での過疎化により一層進んでいるであろう高齢社会において、高度で質の高い医療を提供することも期待されている。WWW を利用した CAS システムとしては、VRML(Virtual Reality Modeling Language) [42] を利用した Web3D システム [21, 3, 26, 27, 22] が挙げられる。VRML とは WWW ブラウザ上で 3 次元 CG を表示する Web3D 技術として最初に発表されたものである。VRML はデータ表現として 3 角形メッシュを採用していたために、3 次元再構成によって作成した 3 次元モデルをそのまま利用することができた。また、補間アニメーションやハイパーリンク機能を容易に実現することができたために、様々な先進的な CAS システムの開発が試みられた。

しかし、VRML を用いたシステムは普及せず、実用に至った Web3D 型の CAS システムはほとんど存在しない。これは、VRML が採用している 3 角形メッシュのデータサイズが非常に大きいためである。人体の形状は滑らかな曲面によって構成される場合が多いため、3 角形メッシュで形状を表現するには膨大な数の 3 角形が必要になる。その結果、データサイズが大きくなり、膨大な転送時間が必要になってしまう。スタンドアロン型の CAS システムではデータ転送の必要がないために、高速な表示が可能である 3 角形メッシュはかえって効果的であった。しかし、Web3D 型のシステムにおいては、転送時間が長いことは致命的である。Web3D 用途に適した、軽量で高品質な新しい形状表現が必要とされている。

1.2 本論文の目的・手法

本論文の目的は、Web3D 型の CAS システムに最適な形状表現とそのモデリング手法を確立することである。この目的を達成するために、本論文では医用ラティス構造という新しい形状表現を提案し、患者の CT/MRI 画像から作成される 3 角形メッシュを医用ラティス構造に変換する曲面再構築手法について論ずる。

医用ラティス構造は、自由曲面に基づく形状表現であり、滑らかな曲面形状を少ないデータサイズで表現することが可能である。そのため、高速なデータ転送が可能であり、Web3D 用途に適した形状表現である。また、この自由曲面データは、自身と同一のグラフ構造（位相）を保持する単純な多面体として管理されるため、曲面形状の直感的な変形操作が可能である。医用に着目すると、医用ラティス構造は 2 つの長所を持つ。第 1 に、特徴線の反映で

ある．医用においては 3 次元モデルの切断線や組織境界線などが明示的に可視化されていることが重要である．本論文では，このような線のことを特徴線と呼ぶ．3 角形メッシュにおいて，これらの特徴線を強調したり選択することは非常に困難であり，実用的な CAS システムを構築する上での妨げとなっている．本論文で提案する曲面再構築手法では，CT/MRI 画像から 3 次元モデルを生成する過程で入力される曲線列を，曲面境界線として反映することが可能である．第 2 の利点は，精度の適応的制御である．医用においては，3 次元モデルの患部以外の領域は，大まかな形状把握に用いるだけであるため，高い精度を必要としない．Web3D では，極力データサイズを小さくする必要があるため，3 次元モデルの領域に応じて，適応的に精度をコントロールする仕組みが必要である．本論文で提案する曲面再構築手法では，CT/MRI 画像から 3 次元モデルを生成する過程で領域ごとに指定した精度を保持した曲面再構築を可能にしている．

本論文における主たる提案は，3 角形メッシュから医用ラティス構造に基づく曲面モデルを再構築する手法である．入力となる 3 角形メッシュを CT/MRI 画像から作成する手法や，自由曲面を 3 角形メッシュから再構築する手法については様々なものが提案されているが，操作性を考慮した包括的なモデリング手法は提案されていない．そこで，本論文では，CT/MRI 画像を入力として，医用ラティス構造を出力とする包括的な医用 Web3D モデリング手法について提案する．本論文で提案する手法は 3 つのステップを持つ．第 1 ステップは，3 角形メッシュの再構築である．このステップでは一般的な 3 次元再構築手法であるマーチングキューブス法を用いる．このステップは自動的に実行することが可能であるが，最適なデータ生成のために，CT/MRI のスライスごとに容易に再編集可能な手法を提案する．第 2 ステップは，特徴線と精度の入力である．この段階では，手術シミュレーションにおける切断線などの用途に用いる特徴線と患部領域の精度の入力を行う．特徴線と高精度領域の入力は 3 次元ペイントのインターフェイスによって実現される．そのため，多くのユーザインプットは必要なく，実際に切断線などをモデルに描画する感覚で操作することができる．第 3 ステップは，3 角形メッシュへの自由曲面フィッティングである．この処理は自動的に行われるものであり，非常に高い精度での近似が可能である．この自由曲面フィッティング手法が本論文の最も重要な提案であり，平均 10 分の 1 以下にデータサイズを圧縮させることが可能な幾何形状圧縮手法として有用である．

本論文で提案する医用ラティス構造とその作成手法を評価する上で，アプリケーションの構築と運用は非常に必要な要素である．そこで，再構築された 3 次元モデルを用いた複数の Web3D アプリケーションを実装した．これらの Web3D アプリケーションは，医学における臨床，研究及び教育での利用を考慮して，それぞれに最適な症例を対象としている．こ

これらの Web3D アプリケーションにおいて、データは実際の患者の CT 画像を元に作成されており、WWW ブラウザ上で動作させることが可能になっている。

最後に、本論文では作成された 3 次元モデルの精度の数値的評価を行う。誤差評価には、オリジナルの 3 角形メッシュと変換後の自由曲面との平均 2 乗距離誤差、及び最大 2 乗距離誤差を用いる。更に、特徴線として指定した稜線列と精度を適応的に制御した領域についても同様の手法を適用し、医用での有用性を評価する。

本論文で述べる手法及び Web3D アプリケーションは、「サイバーワールドモデリングと医学への応用」プロジェクトによる研究成果をまとめたものであり、複数の研究を統合したものである。主たる提案である医用ラティス構造に基づく曲面再構築手法を除き、その他の手法及びアプリケーションはこのプロジェクトの他メンバーによる功績であることを補足しておく。

1.3 本論文の社会的意義

本論文の社会的意義の 1 つとして、医療教育の発展が挙げられる。3 次元で表現された患部のモデルを用いることで、直観的な理解が促進されるため、従来の 2 次元ベースの教科書と比較するとその学習効果は高い。また、写真などの 2 次元画像では、撮影することが困難である視点からでもモデルを閲覧することが可能である。更に、インターネットにより教材にアクセスすることが可能であるため、時間と空間の制約を超えた教育が期待できる。

2 つ目の意義は、医療の統合及び高度化である。現在は、医学の高度化と専門化が進んでおり、1 人の医師で全ての病状を診察、治療することは不可能である。もし、遠隔地に滞在している専門医間のコラボレーションが可能になれば、様々な専門知識を統合した高度な医療の提供が可能になる。患者の 3 次元モデルをインターネットを通して共有することにより、適切な診断や術前会議を行うことが可能になる。

3 つ目の意義は、ネットワーク上における手術シミュレーションの実現である。手術シミュレーションは、実際の手術における危険回避と患者の負担を減少させる点で非常に有意義である。しかし、現在の Web3D 技術では、切断などのシミュレーションを行うことは不可能であり、閉じたスタンドアロン環境でないとシミュレーションを実現することができない。医用ラティス構造を用いることで、ネットワーク上でのコラボレーションに基づく手術シミュレーションが期待できる。

4 つ目の意義は、医療 3D データベースの高度化である。現在の医療データベースはテキストや 2 次元画像によるものが一般的であり、ネットワークを介した利用も進んでいない。医用ラティス構造は高精度でデータサイズも小さいために、データの蓄積に適してい

る．また，医用ラティス構造は XML(eXtensible Markup Language) に基づいており，次世代 Web3D 標準規格の X3D [50, 51] の仕様にも採用されている．そのため，既存のデータベースとの相性もよく，効率的なシステム作成が期待される．

以上の将来像のプロトタイプを作成し，運用及び評価を行うことは非常に重要である．そのため，本論文では，上記の将来像のプロトタイプとなる Web3D アプリケーションを作成し，実際の医師による実験と評価を行っている．

1.4 本論文の構成

本論文は 7 つの章により構成される．

まず第 2 章では，コンピュータ支援外科について述べる．医用画像を用いた 3 次元再構成手法について，スタンドアロン型での研究例と Web3D 型での研究例についてそれぞれ概説し，その問題点を述べる．

第 3 章では，Web3D のための形状表現について述べる．形状モデリングで用いられている様々な形状表現手法について概説し，それぞれの問題点について述べる．更に，効率的な Web3D を実現するために必要な要素について考察する．

第 4 章では，医用ラティス構造について述べる．まず，Web3D 用の形状表現であるラティス構造について触れ，それを医用拡張した医用ラティス構造について解説する．また，2 次元の医用画像を入力として，医用ラティス構造に基づく 3 次元モデルを再構築する手法についても詳細に述べる．更に，3 次元再構築の過程で，医用に必須な要素である特徴線の反映と精度の適応的制御を実現する手法についても述べる．

第 5 章では，医用ラティス構造に基づく Web3D アプリケーションについて述べる．第 4 章で述べる医用ラティス構造とその作成手法が，実際の医用において有効であるか否かを評価するためには，アプリケーションの構築と運用は非常に必要な要素である．そのため，複数の症例をモデルケースに取り上げ，臨床，教育，研究の 3 つの観点から開発した教育アプリケーションや手術シミュレーションについて解説する．また，効率的な Web3D アプリケーションの実装方法とソフトウェア構成についても触れる．

第 6 章では，本論文で提案した手法の数値的評価を行う．具体的な 3 次元再構築例を紹介し，それぞれについて実行時間，データ圧縮率などの数値的，実用的評価を行う．

最後に第 7 章で本論文をまとめ，今後の展望について述べる．

1.5 本論文の用語

以下に，本論文で用いる用語及び表記について解説する．

- 形状モデル (\mathcal{M}) や点群 (\mathcal{X}) などの要素の集合を示す際は、カリグラフィック文字で表す。
- 頂点 (v)、稜線 (e)、面 (f) などの位相要素は斜体細字で表す。また、各要素について、これらの親となるものを右上に、自身の順番を右下に記す。例えば、モデル \mathcal{M} の i 番目の頂点は $v_i^{\mathcal{M}}$ のように表す。ただし、右上の添え字に、更に右上の添え字があるような場合（例えば v^{S^1} ）は、簡潔に v^1 と表すことがある。
- 頂点座標などの 3 次元ベクトルは \mathbf{v} のように正体太字で表す。
- 行列は \mathbf{A} のように大文字の太字で表す。
- スカラ量は k のように斜体の細字で表す。
- ベクトルの内積は $s \cdot t$ のように \cdot で表す。
- ベクトルのノルムは $|\mathbf{n}|$ のように表す。

第 2 章

コンピュータ支援外科

2.1 スタンドアロン環境におけるコンピュータ支援外科

2.1.1 CT/MRIの利用

コンピュータ支援外科 (Computer Aided Surgery, 以下 CAS) はコンピュータを用いた手術シミュレーション, 医用画像処理, 手術ナビゲーションなどの医療技術の総称である。計算機の高速化とソフトウェアの高度化に伴い急速な発展を遂げており, きたる高齢社会において不可欠な技術として大きな注目を集めている。例えば, 手術シミュレーションをコンピュータ上で行うことができれば, 事前に何回でも手術の疑似体験をすることが可能になる。その結果, 実際の手術において, 迅速で効率的な作業を行うことが可能になり, 患者の危険性を軽減することができる。また, 医用画像処理を利用することで, 従来の診療や診断では発見することが困難な病状を早期に発見することも可能である。更に, 近年ではVR(Virtual Reality) 技術やロボティックスの発展に伴い, 専用の医療デバイスとの併用による技術も進歩している。

CAS においては, 人体内部の断層画像を非侵襲に観察することができる医療機器が頻繁に用いられる。X 線 CT(Computed Tomography) を用いた CT スキャナは, 1971 年に臨床利用が開始されて以来, 現在まで急激な進歩を遂げている。また, 人体からの NMR (核磁気共鳴) 信号をデータ化する MRI(Magnetic Resonance Imaging) も臨床において広く利用されている。これらの機器から得られる画像は, 診療や診察における重要な資料として利用されている。実際には, コンピュータに 2 次元データとして取り込み, データの閾値を変更することで, 様々な臓器の画像を強調, 抽出することが一般的に行われている。

2.1.2 3 次元再構成

断面撮影装置から取得されるデータは 2 次元画像であるために, 直観的かつ立体的な形状把握に熟練を要する。また, 手術シミュレーションなどの高度な用途への利用のためには, 形状要素を直観的に把握でき, かつ操作を加えることができるよう, 3 次元化することが望ましい。このような理由から, CT/MRI 画像を 3 次元化する 3 次元再構成の研究が盛んに行われてきた。ボクセル法 [40, 36, 12] は最も盛んに研究されてきた手法である。3 次元空間を X, Y, Z 軸方向の単位格子 (ボクセル) に分割し, 各単位格子が対象物の内部にあるか, 外部にあるかによって形状を表現する手法である。断面撮影装置から取得される全ての CT スライスについて単位格子を当てはめ, 得られる格子群を直接レンダリング (ポリウムレンダリング) することで 3 次元形状の可視化が実現される。ボクセル法は自動的な処理が可能のために, 容易に 3 次元再構成を実現することが可能である。しかし, 高い精度が必要で

ある人体形状の可視化には膨大な数の単位格子が必要であるために、計算負荷が高く、データサイズも大きくなってしまふ。また、ボリュームを直接レンダリングする手法であり、形状要素として表現されるものではない。

そこで、考案されたのがボクセルによって構成される3次元モデル(ボリュームモデル)を、3角形平面によって構成される3次元モデル(3角形メッシュ)に変換する手法である。Lorenson と Cline [35] は、近隣8つの単位格子の内部にモデルが存在するか否かによってパターン化されたポリゴンデータを当てはめる手法(マーチングキューブス法)を提案した。マーチングキューブス法は自動的な処理が可能である。出力される3角形メッシュは物理モデルに基づく手術シミュレーションや、ラピッドプロトタイピングによる人工骨生成にも利用することができる。

また、ボクセルモデルを介さずにCTデータから直接3角形メッシュを作成する手法も提案されている。Geiger [19] は、CT画像から抽出された輪郭線列を元に3角形メッシュを再構築する手法を提案した。この手法では、手作業で輪郭線を抽出する必要がある一方で、精度の高い3角形メッシュを取得できるのが特徴である。また、佐藤ら [53] は、CADシステムが認識可能な自由曲面に基づくソリッドモデルを再構成する手法を提案している。

2.2 WWWにおけるコンピュータ支援外科

2.2.1 WWWによる人体断層画像の配信

WWW(World Wide Web)を利用したCASシステムへの期待が高まっている。これにより、遠隔地の複数の医師間で、それぞれの専門知識を統合した手術シミュレーションや症例検討を実現することができる。また、地方での過疎化がより一層進んでいるであろう高齢社会において、高度で質の高い医療を提供することも期待されている。

WWWを用いたCASプロジェクトとしては、米国 National Library of Medicine の Visible Human Project [39] が広く知られている。このプロジェクトでは、男女1体ずつの実際の遺体を非常に細かい幅でスライスし、その断層写真画像をインターネットにより配信している。このデータセットは人体の全ての部位の断層画像が入手可能という意味で、非常に価値の高いものである。このデータセットを利用して様々な医用シミュレータ [30, 41] が実装されており、CAS分野に与えた功績は大きい。

2.2.2 VRML を用いた Web3D システム

WWW を利用した CAS システムとしては、VRML¹ (Virtual Reality Modeling Language) [42] を利用した Web3D システムが挙げられる。VRML とは WWW ブラウザ上で 3 次元 CG を表示する Web3D 技術として最初に発表されたものである。VRML はデータ表現として 3 角形メッシュを採用しているために、3 次元再構成によって作成した 3 次元モデルをそのまま利用することができる。Hendin ら [21] と Behr ら [3] は、インターネット上でボリュームレンダリングを実現する手法を提案した。ボリュームレンダリングは一般的な可視化手法であるが、クライアントとなる計算機の計算負荷が高いために用途が限定されてしまう。John ら [26, 27] は、教育用途のための様々なシミュレータを VRML により開発、公開している。彼らのシミュレータは医学生と指導者とのコミュニケーションを支援するマルチユーザ環境を実現しており、トレーニングツールとして有効である。Holten-lund ら [22] は、PC 上で実現可能な Web3D 型 CAS システムのベンチマークを実際のシステム開発を通して報告している。彼らの報告では、シミュレーションに必要な機能の多くは JAVA プログラミングによって実現可能であるとしている。

以上の様に、VRML を用いた様々なシステムが開発されているが、実用に至った Web3D 型の CAS システムはほとんど存在しない。これは、VRML が採用している 3 角形メッシュのデータサイズが非常に大きいためである。人体の形状は滑らかな曲面によって構成される場合が多いため、3 角形メッシュで形状を表現するには膨大な数の 3 角形が必要になる。その結果、データサイズが大きくなり、膨大な転送時間が必要になってしまう。スタンドアロン型の CAS システムではデータ転送の必要がないために、高速な表示が可能である 3 角形メッシュはかえって効果的であった。しかし、Web3D 型のシステムにおいては、転送時間が長いことは致命的である。Web3D 用途に適した、軽量で高品質な新しい形状表現が必要とされている。

¹VRML97, ISO/IEC 14772-1:1997

第 3 章

Web3Dのための形状表現

計算機上で形状モデルを表現するために、様々な手法が提案されている。それぞれの手法には用途に応じて長所と短所があるが、本章では Web3D 用途に限定した上で、望ましい形状の表現方法について議論する。

3.1 節では、本研究に用いる位相的な立体表現とデータ構造について述べる。位相とは、形状の要素間のつながりのことである。まず、位相を表現するための立体表現法とデータ構造について解説する。本手法で用いた立体表現手法であるソリッドモデルについて解説し、ソリッドモデルの位相表現手法である境界表現について述べる。次に、境界表現において位相構造を管理するためのデータ構造について解説する。

3.2 節では、代表的な曲面形状の表現方法について解説する。まず、最も一般的に用いられている曲面表現である 3 角形メッシュを取り上げ、その長所と Web3D 利用での課題について述べる。次に、デザイン用途でよく用いられる自由曲面について取り上げ、その特徴と短所について解説する。また、本研究において利用している双 3 次 Gregory パッチの曲面表現式についても解説する。最後に、近年注目を集めている細分割曲面について解説し、本研究で利用している Doo-Sabin 細分割曲面の曲面表現式について解説する。

3.3 節では、3 角形メッシュを他の曲面表現へと変換する曲面フィッティング手法について解説する。細分割曲面へのフィッティングと B-Spline 曲面へのフィッティングについて、代表的な先行研究のアルゴリズムを簡潔に述べる。本研究で利用している Doo-Sabin 細分割曲面へのフィッティングについても概説するが、詳細は 4.3.2 項にて解説する。

3.1 立体表現とデータ構造

立体表現手法

3 次元 CAD/CG における立体の表現方法には、ワイヤーフレームモデル、サーフィスモデル、ソリッドモデル [37, 57, 56] がある。ワイヤーフレームモデルは頂点と稜線のみで 3 次元形状を表現する。頂点には一意の 3 次元座標値があり、稜線には始点と終点の 2 つの頂点がある。稜線は直線のみならず曲線も表現できる。データ構造が非常に単純であるために、高速な処理と容易な実装が可能であり、多くのシステムでサポートされている。しかし図 3.1 に示すように、モデルの形状を直感的に一意に判断することができないという欠点がある。

これを解決するのがサーフィスモデルである。サーフィスモデルはワイヤーフレームモデルに面のデータを付加したものである。表面のデータを保持するため、モデルの形状を一意に判断することが可能である。しかし、モデルの表面のデータを保持するだけで、中身は空の状態であるため、厳密な意味での正確なデータは保持していない。そのため、変形によ

り面の間に間隙が生じるなどの欠点がある。

それに対して、ソリッドモデルは立体を完全に表現できる。ソリッドモデルは中身の詰まった表現であり、立体の内部と外部を境界面により区別している。そのため、解析や変形等においても正しい形状の処理を自動的に行うことができる。

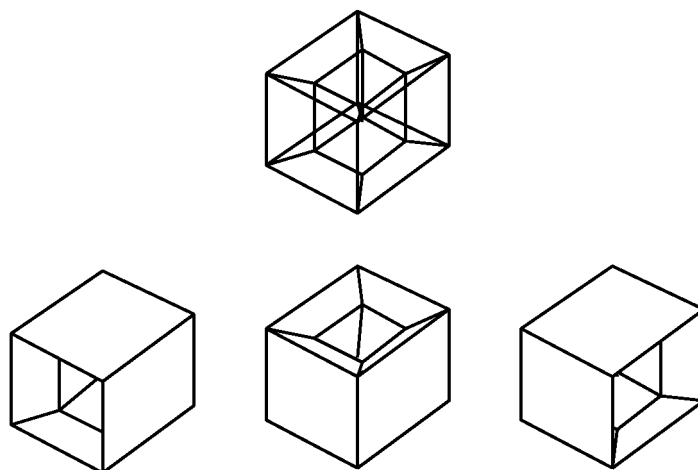


図 3.1: 形状が一意に判断できないワイヤースケルトンモデル

境界表現

ソリッドモデルにおける立体表現手法には、CSG(Constructive Solid Geometry) 表現と境界表現 (B-reps:Boundary Representations) がある。CSG 表現は、立方体、角錐、球、円柱、円錐、トーラス、楕円体などのプリミティブ立体により形状を表現する手法である。これらのプリミティブ立体の集合演算により複雑な形状を表現することが可能である。集合演算には立体を結合させる和演算、ある立体から別の立体を取り去る差演算、2つの立体の共通部分を生成する積演算がある。CSG 表現は、データ構造が単純であるが、曲面表現が困難であり、また表示にも時間がかかる。このような理由から、本手法では以下に述べる境界表現を採用している。境界表現とは、立体の内部と外部を区別することにより、形状を表現する手法である。立体の内外を区別する位相要素には表 1 に示すように、立体 (Body)、殻 (Shell)、面 (Face)、ループ (Loop)、稜線 (Edge)、頂点 (Vertex) などがある。これらの位相要素は立体の実際の座標値とは関係のないものであり、立体の任意のアフィン変換に関して不変な要素である。

表 1: 位相要素

立体 (Body)	1 つの立体を表現する要素 .
殻 (Shell)	3 次元空間内のある領域を表現する要素 . 1 つの立体は複数の殻の集合により表現される .
面 (Face)	立体の表面を表現する要素 . 1 つの殻は複数の面に囲まれることによって表現される .
ループ (Loop)	面の境界を表現する要素 . 1 つの面は , 穴がない場合 , 1 つのループによって領域を表現される .
稜線 (Edge)	2 つの面が交わる境界を表現する要素 . 1 つのループは複数の稜線によって表現される .
頂点 (Vertex)	立体の角を表現する要素 . 1 つの頂点は複数の稜線が交わることによって表現される .

表 2: 幾何要素

曲面 (Surface)	位相要素の面 (Face) に付随する要素であり , 3 次元空間での面の形状を表現する .
曲線 (Curve)	位相要素の稜線 (Edge) に付随する要素であり , 3 次元空間での線の形状を表現する .
点 (Point)	位相要素の頂点 (Vertex) に付随する要素であり , 3 次元空間での座標値を表現する .

一方 , この位相要素に対して実際の立体の形状 (幾何) を表現する幾何要素には , 表 2 に示すように , 曲面 (Surface) , 曲線 (Curve) , 点 (Point) などが挙げられる .

このように , 境界表現は位相要素と幾何要素から構成されるものであり , 図 3.2 のように相互にリンクされた関係を持っている . 境界表現では位相要素の面と幾何要素の曲面が一意に対応しており , 複雑な曲面構造も容易に扱うことが可能である . 境界表現において , ソリッドモデルに対して何らかの処理を行う場合には , 位相要素と幾何要素の両方を考慮して処理を進める必要がある . ソリッドモデルで使用する幾何要素はパラメトリックな曲線 , 曲面式で表現可能であるために , 制御点の座標値が決定すれば自ずと形状が定義される . そのため , ソリッドモデルにおいて効率的なデータの管理を行うためには , 主に位相要素に関するデータのデータ構造を考慮すれば良いことになる .

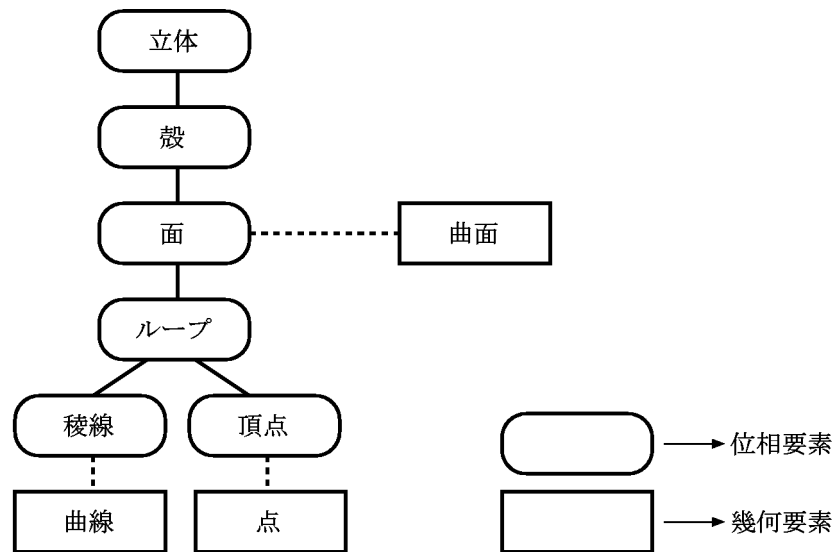
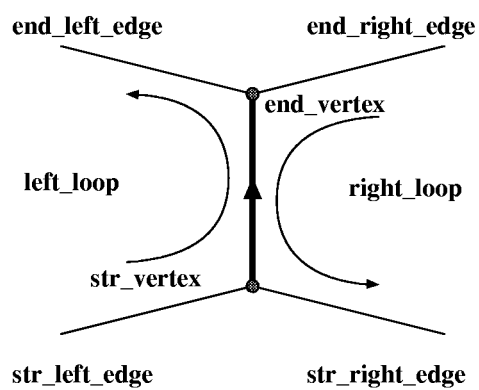


図 3.2: 位相要素と幾何要素の関係

境界表現のデータ構造

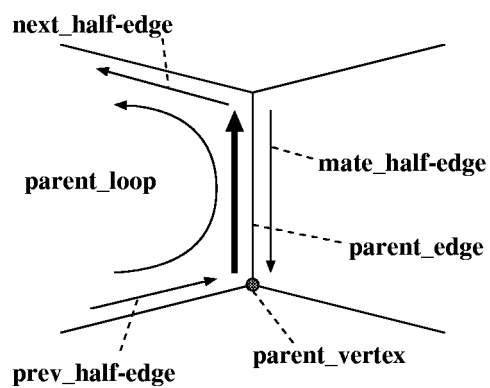
位相要素を管理するためのデータ構造としては，winged-edge 構造 [2] がよく用いられる．winged-edge 構造は，立体の稜線を中心に位相要素を管理するデータ構造である．全ての稜線において，データが固定長であるため，効率よくデータ管理ができる．しかし，ある頂点に接続する稜線を全て参照する場合など，非常に頻繁に用いられる操作でさえも，winged-edge 構造では簡潔に行うことが困難である．この問題を解決するのが，ハーフエッジ (half-edge) 構造 [37] である．ハーフエッジ構造では，図 3.3 に示すように 1 つの稜線 (edge) の両端に 2 つの有向稜線を定義している．この有向稜線をハーフエッジ (半稜線，half-edge) と呼ぶ．ハーフエッジ構造では，1 つの頂点 (parent-vertex) は自身から延びるハーフエッジ (half-edge) のうちの 1 つを保持している．1 つの稜線は自身の両端に接続する頂点情報を保持しており，この頂点情報から，自身の両端にある 2 つのハーフエッジにアクセスすることができる．1 つのハーフエッジは自身の始点にあたる 1 つの頂点と，その頂点回りにループを回ったときに自身の次にあたるハーフエッジ (next_half-edge)，及び自身の前にあたるハーフエッジ (prev_half-edge) を保持している．本研究ではハーフエッジ構造を採用している．これにより，ある頂点に接続する稜線を全て参照する場合などの操作を簡潔に行うことが可能になる他，3 次元空間内に 1 つの頂点のみが存在する状況なども表現することが可能になる．



edgeの持つ情報

left_loop	right_loop
str_vertex	end_vertex
str_left_edge	str_right_edge
end_left_edge	end_right_edge

(a)winged-edge構造



half-edge の持つ情報

parent_loop	
parent_edge	
parent_vertex	
next_half-edge	prev_half-edge
mate_half-edge	

(b)half-edge 構造

図 3.3: winged-edge 構造とハーフエッジ構造

3.2 曲面形状表現

3.2.1 3 角形メッシュ

3 角形メッシュは 3 次元 CG において最も一般的に利用されている形状表現である．図 3.4 に 3 角形メッシュを示す．3 角形メッシュは，面 f ，稜線 e ，頂点 v の 3 つの位相要素と，頂点の幾何情報を示す 3 次元空間上の座標群

$$\mathcal{X} = \{v_1, v_2, \dots, v_n\} \in \mathbf{R}^3 \quad (3.1)$$

及び，それらの位相要素間のグラフ構造によって構成される．ここで，位相間のグラフ構造は

$$\begin{aligned} e_1 &= \{v_1, v_2\} \\ e_2 &= \{v_2, v_3\} \\ f_1 &= \{v_1, v_2, v_3\} \\ f_2 &= \{v_2, v_6, v_3\} \end{aligned} \quad (3.2)$$

のようになり，ハーフエッジ構造によって管理されている．

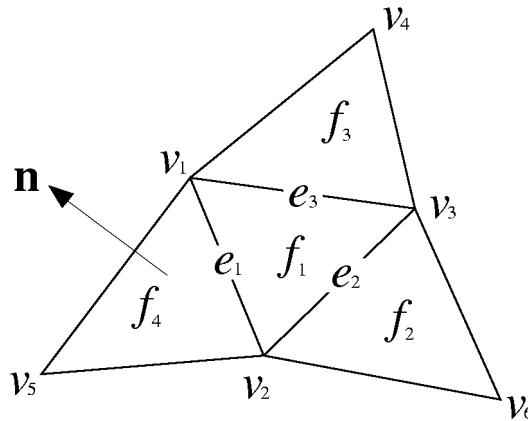


図 3.4: 3 角形メッシュ

3 角形平面の集合によって形状を構成するために，データ構造が単純で非常に扱いやすい．そのため，多くのハードウェアでサポートされており高速なレンダリングが可能である．一方，高品質な曲面形状を表現するためには，図 3.5 に示すような区分線形近似を行うために，数万ものポリゴン数が必要になる．そのため，データサイズが膨大になり，データ転送

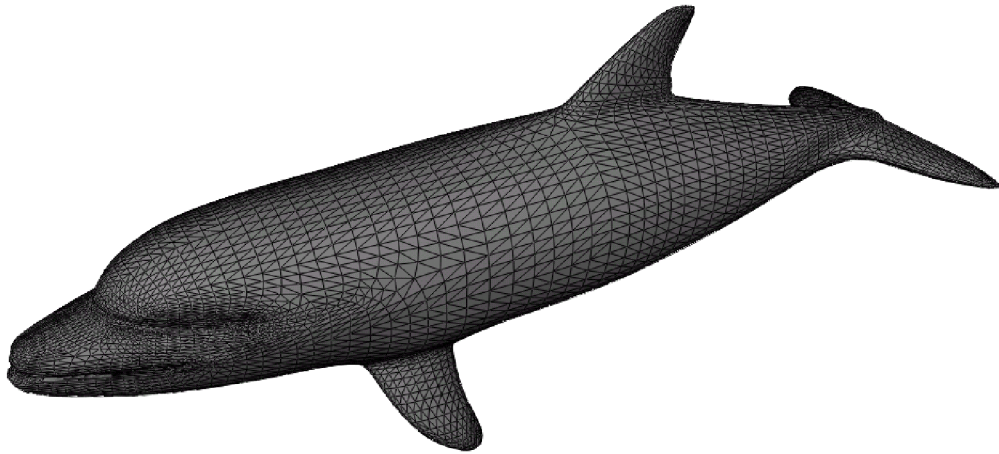


図 3.5: 3 角形メッシュによる区分線形近似

時間の増大やハードディスク容量の圧迫など様々な問題を引き起こす。更に、データを受信する計算機の処理能力が低い場合は、モデルを構成する膨大なポリゴンをインタラクティブに処理することができず、ユーザに大きなストレスを与えることになる。VRML が Web3D データフォーマットの標準として提案されながらも普及しなかった最も大きな原因はここにある。

このような現状の中で、3 角形メッシュに様々な属性を付加することで、これらの問題を解決しようとする研究が盛んである。Hoppe [25] は、エネルギー関数の最小化により複雑なメッシュ形状を単純化し、その単純化の過程を階層的に管理することによって、メッシュの解像度を自由に調節する手法を提案した。この手法により、詳細な 3 角形メッシュが転送された場合でも、受信側の計算機の処理能力に応じて形状の詳細度を調節し、インタラクティブに形状を表示することが可能になる。Abadjev ら [1] は Hoppe の手法を拡張し、幾何情報に加えてテクスチャ情報も階層的に管理することで、3D データの効果的なストリーミングを実現するフレームワークを提案した。これらの手法を用いることにより、インタラクティブな表示制御は可能であるが、データサイズ自体を縮小させることはできない。更に、階層的なデータ構造に変換するため、通常の 3 角形メッシュの約 2 倍のデータサイズが必要であると言われている。3 角形メッシュのデータサイズを削減するためには面の数を削減せざるを得ず、その結果形状に歪みが生じ、使用用途が著しく限定されてしまう。よって、Web3D の形状表現として 3 角形メッシュを採用するのは実用的ではない。

3.2.2 自由曲面

自由曲面は，少数の制御点によって広域的で滑らかな形状を表現する手法である．一般に多項式パラメトリック曲面を用いることが多く，デザインの分野を中心に広く利用されている．Coons [5] は，計算機上でインタラクティブに曲面デザインを行う表現手法として Coons 曲面を開発した．Bézier [4] は，インタラクティブな自動車デザインのツールとして Bézier 曲面を開発した．これらの2つの曲面表現はともに，4 辺形の領域を持つパッチを生成する手法である．これらの曲面式に，有理化による拡張や一般化を行うことにより，様々な曲面表現式が提案されている．Gordon と Riesenfeld [17] は，Bézier 曲面間の接続を滑らかにする B-Spline 曲面を提案した．B-Spline 曲面を有理化し，複数の曲面セグメントによる1つの曲面表現を可能にした NURBS(Non-Uniform B-Spline Surfaces) は，曲面モデリングのデファクトスタンダードな表現式となっている [14] ．

自由曲面は厳密な形状の定義が可能であり，3 角形メッシュに比べてデータサイズも小さい．しかし，制御点を媒介とした変形操作には非常に熟練を要するため，意図した通りの変形が困難である．また，位相の変化を伴う変形を行った場合に，連続性を保つことも難しい．

双 3 次 Gregory パッチ

医用ラティス構造において，自由曲面は双 3 次 Gregory パッチ [7] により表現される．双 3 次 Gregory パッチは，一般 Coons 曲面のツイストベクトルを境界曲線の横断ベクトルとして表した Gregory による拡張 [18] を双 3 次 Bézier パッチに適用したものである．

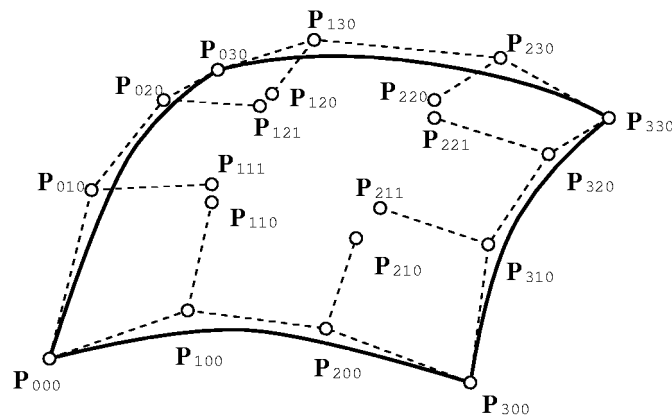


図 3.6: 双 3 次 Gregory パッチ

図 3.6 に示すように，双 3 次 Gregory パッチは 20 個の制御点 $\mathbf{P}_{ijk}(i = 0, \dots, 3; j = 0, \dots, 3; k = 0, 1)$ により構成される．双 3 次 Gregory パッチの曲面表現式は以下のようになる．

$$\mathbf{S}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_i^3(u) B_j^3(v) \mathbf{Q}_{ij}(u, v) \quad (3.3)$$

$$(0 \leq u \leq 1, 0 \leq v \leq 1)$$

ここで $B_i^3(u)$ と $B_j^3(v)$ は Bernstein 基底関数 [14] である．また，制御点 $\mathbf{P}_{ijk}(i = 0, \dots, 3; j = 0, \dots, 3; k = 0, 1)$ と $\mathbf{Q}_{ij}(i = 0, \dots, 3; j = 0, \dots, 3)$ には次のような関係がある．

$i \neq 1, 2$ または $j \neq 1, 2$ の時

$$\mathbf{Q}_{ij}(u, v) = \mathbf{P}_{ij0} \quad (3.4)$$

$i = 1, 2$ または $j = 1, 2$ の時

$$\begin{aligned} \mathbf{Q}_{11}(u, v) &= \frac{u\mathbf{P}_{110} + v\mathbf{P}_{111}}{u + v}, \\ \mathbf{Q}_{12}(u, v) &= \frac{u\mathbf{P}_{120} + (1-v)\mathbf{P}_{121}}{u + (1-v)}, \\ \mathbf{Q}_{21}(u, v) &= \frac{(1-u)\mathbf{P}_{210} + v\mathbf{P}_{211}}{(1-u) + v}, \\ \mathbf{Q}_{22}(u, v) &= \frac{(1-u)\mathbf{P}_{220} + (1-v)\mathbf{P}_{221}}{(1-u) + (1-v)} \end{aligned} \quad (3.5)$$

ただし $0 \leq u \leq 1, 0 \leq v \leq 1$ である．

境界曲線は 3 次 Bézier 曲線によって定義され，双 3 次 Bézier パッチの内部制御点を二重にした構造を持っている．また，流れベクトル関数 (Cross Boundary Derivative) を u, v のパラメータごと独立に定義することができる．そのため，輪郭曲線列さえ確定すれば，曲面間の G^1 連続を容易に確保することが可能である．双 3 次 Gregory パッチは連続性に優れた曲面表現であり，NURBS にも変換可能 [44] である．この特徴を利用することにより，ラティス構造で表現される形状データを CAD/CAM システムで利用することが可能になる．図 3.7 に双 3 次 Gregory パッチによって表現されている 3 次元モデルの例を示す．



図 3.7: 自由曲面（双 3 次 Gregory パッチ）によって表現されている 3 次元モデル

3.2.3 細分割曲面

細分割曲面 (Subdivision Surfaces) は、単純な多面体形状に対して、分割と重み付けの操作を繰り返し適用することで、滑らかな曲面形状を生成する手法のことである。多面体形状を初期メッシュ、生成される曲面形状を極限曲面 (Limit Surface) と呼ぶ。Doo と Sabin [11] 及び Catmull と Clark [6] によって基礎理論が構築され、Loop [33] が三角形メッシュを対象とした手法を提案したことにより、広く用いられるようになった。Doo と Sabin による手法 (Doo-Sabin 細分割) においては、極限曲面は双 2 次 B-Spline 曲面を近似する形状になる。Catmull と Clark による手法 (Catmull-Clark 細分割) においては、極限曲面は双 3 次 B-Spline 曲面を近似する形状になる。そのため、細分割曲面は B-Spline 曲面の一般化手法として捉えられている。

Zorin ら [52] は Loop の細分割手法を局所的に適用することを可能にした。これにより、精度が不要な部分は荒く分割することが可能になり、細分割後の最終形状でもデータサイズをコンパクトにすることが可能になった。DeRose ら [10] は、Catmull と Clark の手法をアニメーション分野で現実的に応用することを試みた。彼らは複数の曲面形状をブレンドすることで、様々な曲面形状の作成を可能にしている。Lee ら [32] は複雑な表面を持つ形状を、Loop 細分割曲面の初期メッシュとスカラー値の集合に分解して保存することにより大幅にデータを削減する手法を提案した。細分割曲面は、任意の位相を持つポリゴン形状に適用することが可能であるため汎用性が高い。また、データサイズの小さい初期形状から複雑な曲面形状を生成することが可能であるため、インターネット利用に適した形状表現である。一方で、最終的な生成形状を確認するためには複数のフィルタリングが必要である。また、既存の CAD/CG システムが認識可能な NURBS などの曲面制御点情報を明示的に保持していないという問題がある。

Doo-Sabin 細分割曲面

Doo-Sabin 細分割曲面は、図 3.8 に示すように、任意位相の初期メッシュに細分割処理を繰り返し適用することで、双 2 次の B-Spline 曲面を近似する極限曲面を生成する手法である。本研究の曲面生成手法では、Doo-Sabin 細分割曲面の分割手法を利用しているため、以下にその詳細を解説する。

まず、初期メッシュ M^0 の各面について、図 3.9 に示すような edge point, face point という 2 つの要素を定義する。edge point は面を構成する稜線の中点のことであり、face point は面の重心のことである。

以下に、細分割のステップについて述べる。

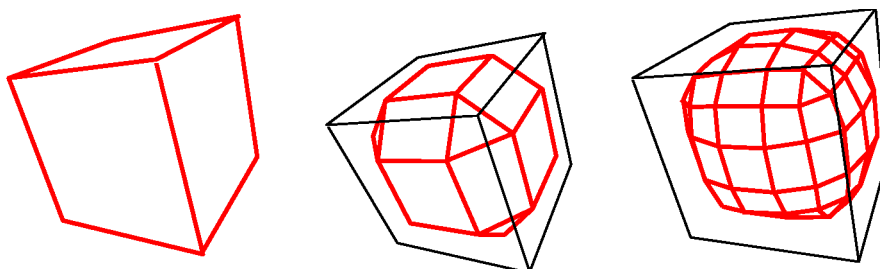


図 3.8: Doo-Sabin 細分割曲面

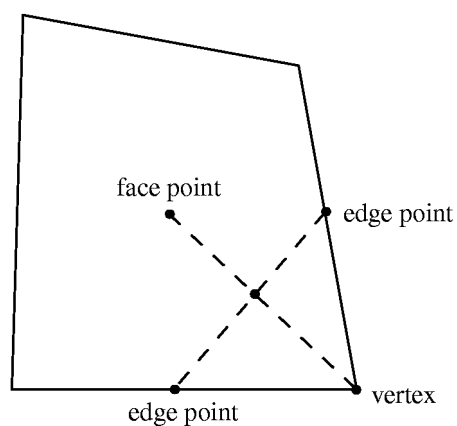


図 3.9: face point と edge point

1. 初期メッシュ M^0 の全て頂点 p_i について、新しい頂点 p'_i を求める。 p'_i は、 p_i 、2つの edge point、face point の平均点である（図 3.10 上）
2. 全ての面において、新しく生成された頂点 p'_i を接続する（図 3.10 中）
3. 全ての稜線について、自身に接続する面上で作成された新しい頂点同士を接続する。（図 3.10 下）
4. 以上のステップで作成されたメッシュを、次の細分割ステップの初期メッシュとして、細分割を繰り返す。

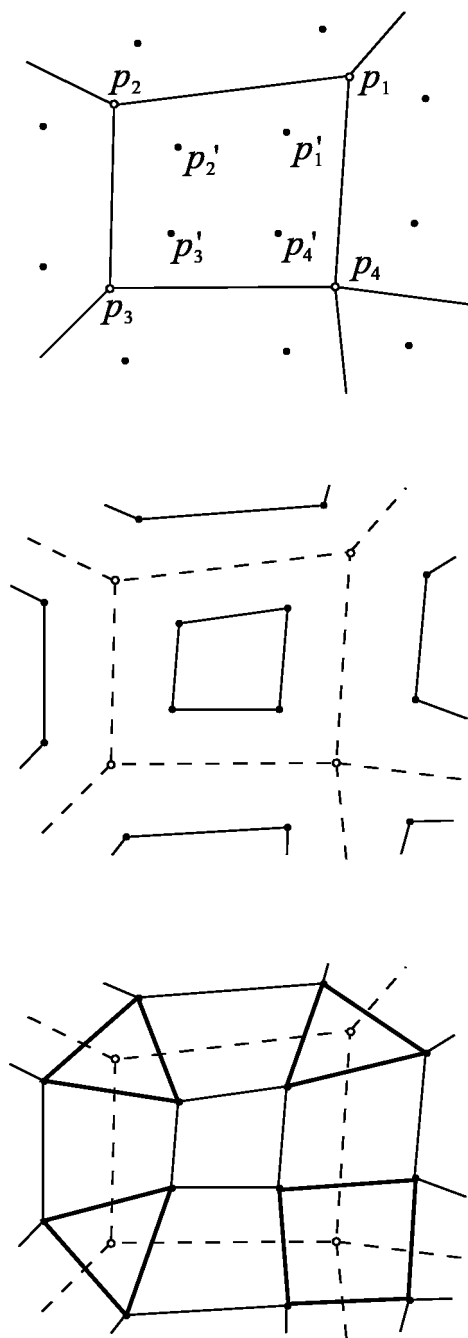


図 3.10: Doo-Sabin 細分割のステップ

以上のように，頂点位置計算ステップは1つの操作によって行われる．また，新しく生成される頂点の価数（接続稜線数）は必ず4になる．ここで，face point を g とすると，新しい頂点 p_i' を求める細分割ステップは以下のように記述することができる．

$$p_i' = \frac{1}{4}g + \frac{1}{8}p_{i-1} + \frac{1}{2}p_i + \frac{1}{8}p_{i+1} \quad (3.6)$$

3.3 曲面フィッティング

本節では，曲面フィッティングの先行研究について解説する．曲面フィッティングとは，3 角形メッシュに基づく 3 次元モデルを，自由曲面や細分割曲面に基づく 3 次元モデルに変換する手法のことである．3.1 節で述べたように，3 角形メッシュによる曲面表現は Web3D での利用に適していない．しかし，3 次元再構成によって生成される 3 次元モデルは 3 角形メッシュであるために，他の曲面表現へのフィッティングが必要である．曲面フィッティングには多くの先行研究があるが，本節では，本研究において利用している細分割曲面へのフィッティング手法と，不規則点群から B-Spline 曲面を生成する手法について概説する．

3.3.1 細分割曲面フィッティング

Loop 細分割曲面フィッティング

Hoppe ら [24] は，3 次元レーザースキャナによって測定された点群データに，Loop 細分割曲面をフィッティングする手法を提案した．Hoppe らのアルゴリズムを以下に概説する．

1. 位相構造と初期形状の構築

不規則点群 \mathcal{X} から，高密度な 3 角形メッシュ \mathcal{M}^I を構築する．この段階で，極限曲面における位相と初期の幾何形状が決定される．

2. メッシュの最適化

\mathcal{M}^I の 3 角形数を削減する．削減の過程で幾何形状が初期の不規則点群 \mathcal{X} により近づくように処理が行われ，最適化メッシュ \mathcal{M}^O が作成される．

3. 細分割曲面のフィッティング

\mathcal{M}^O を初期メッシュとして作成される極限曲面を，区分的に \mathcal{X} に対してフィッティングする．この際，以下のエネルギー関数を最小化することによって鋭角領域を保存した区分線形補間が可能になる．

$$E(K, L, V) = E_{dist}(K, L, V) + c_{rep}m + c_{sharp}e \quad (3.7)$$

ここで， K は頂点，稜線，面の接続性を表す位相， L は位相 K における稜線の部分集合， V はメッシュの形状を定義する頂点座標の集合を表している．また， E_{dist} は不規則点群 \mathcal{X} から極限曲面までの距離の 2 乗を表している． $c_{rep}m$ は頂点数 m に対する制約を， $c_{sharp}e$ は鋭角稜線数 e に対する制約をそれぞれ表している．

彼らの手法では、鋭角領域を極限曲面に反映することが可能であるために、工業製品などの曲面化にも利用することができる。一方で、計算コストが非常に高いためにインタラクティブな処理が困難である。

Doo-Sabin 細分割曲面フィッティング

Takeuchi ら [45] は、3 角形メッシュに Doo-Sabin 細分割曲面をフィッティングさせる手法を提案した。彼らの手法は QEM(Quadric Error Metrics)[16] に基づく 3 角形メッシュの簡略化操作を利用しており、細分割曲面の初期メッシュが直接出力されるのが大きな特徴である。本研究の曲面フィッティング手法は、Takeuchi らの手法を改良したものである。詳細は 4.3.2 項で述べるため、ここでは概要を記す。

QEM に基づく 3 角形メッシュの簡略化は、3 角形メッシュに属する全ての稜線にスカラー値を付加し、その値の小さい稜線から順番に削除することで、簡略化を実現している。QEM とはこの際の評価関数のことであり、オリジナルメッシュ \mathcal{M}^0 と、簡略化メッシュ \mathcal{M}^i の誤差を評価している。Takeuchi らの手法では、オリジナルメッシュ \mathcal{M}^0 と、簡略化メッシュ \mathcal{M}^i を初期メッシュと見立てた際に生成される極限曲面との誤差を評価している。これにより、簡略化されたメッシュは、Doo-Sabin 細分割曲面の初期メッシュとなる。

3.3.2 B-Spline 曲面フィッティング

Eck ら [13] は、Hoppe ら [23] の 3 次元再構成手法によって生成された 3 角形メッシュに対して、B-Spline 曲面を自動的にフィッティングする手法を提案した。Eck らのアルゴリズムを以下に概説する。

1. 不規則点群からの 3 角形メッシュ作成

Hoppe らの手法 [23] によって、不規則点群 $\mathcal{X} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ から密な 3 角形メッシュ \mathcal{M}^0 を構築する。

2. 3 角形領域複体 K の作成

密な 3 角形メッシュ \mathcal{M}^0 に対して、ボロノイ図とドロネー 3 角形分割を用いて領域分割を行う。この領域 K を 3 角形領域複対と呼び、曲面のパッチネットワークのガイドラインとなる。 K 内に含まれる全ての点群に対して、調和写像により K 内でのパラメタリゼーションを行う。

3. K をペアリングした 4 辺形領域複対 K の作成

B-Spline 曲面パッチは 4 辺形であるために、2 つの 3 角形領域複対をペアリングして

4 辺形領域複対 K を作成する .

4. B-Spline 曲面の生成

不規則点群から B-Spline 曲面 S への距離の 2 乗和を表すエネルギー関数 $E_{dist}(S)$ を定義し , それを最小化することによって B-Spline 曲面を生成する .

$$E_{dist}(S) = \sum_{i=1}^n d^2(\mathbf{p}_i, S) \quad (3.8)$$

彼らの手法では , 連続性に優れた高品質な B-Spline 曲面を生成することが可能である . しかし全ての処理が自動的に実行されてしまうため , ユーザの意図する特徴線をパッチの輪郭に反映することができないという問題がある .

3.4 第 3 章のまとめ

以上のように , 現存する形状表現にはそれぞれ一長一短があり , Web3D に最適なものは存在しない . これまでの議論を通して , Web3D に特化した形状表現を作り上げるには , 以下の 2 点を満たすことが望ましいと考えられる .

- 軽量かつ高品質であること
- データの用途を限定しないこと

前者の条件は , 高速な転送を実現するために必須な要素である . また , データの精度を損なわずにサイズを縮小できるものでなくてはならない . 後者の条件は , ブラウザ上で形状を表示するだけでなく , 変形操作やラビッドプロトタイピングなどの様々な用途での利用のために必須な条件である .

これらの条件を満たすためには , 自由曲面のデータ軽量性及び高品質性を利用するのが最も効果的である . しかし , 自由曲面には変形操作における柔軟性が欠けているために , より容易に自由曲面を扱う枠組みが望まれている .

第 4 章

医用ラティス構造

本章では、Web3D 型の CAS 用途にふさわしい形状表現である医用ラティス構造と、そのモデリング手法について解説する。

4.1 節では、医用ラティス構造の元になる形状表現であるラティス構造について述べる。まず、4.1.1 項にて、ラティス構造を定義づける構成要素について述べる。また、類似する形状表現である細分割曲面との比較を通して、その特徴についても解説する。次に、4.1.2 項にて、曲面形状の構築方法について述べる。曲面形状を構成する曲面と曲線の数学的定義について解説し、曲面形状から単純な多面体へ変換する手法についても述べる。最後に、4.1.3 項にて、曲面モデルを構成する位相要素に重み付けを行うことで、多様性のある形状定義を実現する手法について述べる。

4.2 節では、ラティス構造に医用拡張を加えた医用ラティス構造の特徴について解説する。まず、4.2.1 項にて、第 1 の特徴である特徴線をモデルに反映することの重要性について解説する。4.2.2 項では、第 2 の特徴であるモデルの精度を領域ごとに適応的に制御することの重要性について解説する。

4.3 節では、CT 画像から医用ラティス構造に基づく 3 次元モデルを構築する手法について述べる。まず、4.3.1 項では、CT 画像を入力として、3 角形メッシュに基づく 3 次元モデルを構築する手法について解説する。4.3.2 項では、3 角形メッシュに自由曲面をフィッティングし、ラティス構造へと変換する手法について述べる。まず、過去の曲面フィッティング手法について概説し、特にラティス構造と形状表現が似ている細分割曲面へのフィッティング手法に着目し、その概要について解説する。次に、細分割曲面へのフィッティング手法を拡張し、ラティス構造へのフィッティングを行う詳細について解説する。4.3.3 項では、特徴線を曲面輪郭線としてモデルに反映させる手法と、領域ごとに精度を制御する手法について述べる。

4.1 ラティス構造

4.1.1 ラティス曲面と初期多面体

ラティス構造 [47, 48] は, Web3D 用途において自由曲面データを効果的に表現することができる新しいフレームワークである。ラティス構造において, 形状データは自由曲面とポリゴンの両方の特性を持つものとして扱われる。データ転送の際には, 自由曲面形状の特徴を反映した単純なポリゴン (以後, 初期多面体) として扱うことが可能である。そのためデータサイズはコンパクトになり, 高速なデータ転送が実現される。データを受信した計算機上では, 転送されてきた初期多面体から自由曲面 (以後, ラティス曲面) を構築し, 高品質な自由曲面形状を表示することができる。このように, 1 つの形状データを用途に応じて自由曲面とポリゴンの 2 つの形式で扱うことができる枠組みをラティス構造と呼ぶ。図 4.1 にラティス構造を示す。右がラティス曲面であり, 左がそれに対応する初期多面体である。初期多面体はラティス曲面と同一位相であり, 頂点, 稜線, 面, 重み付け情報によって構成されている。初期多角形はラティス曲面より再構築することが可能であり, 相互に変換することが可能である。

ラティス曲面は, 双 3 次 Gregory パッチ [7] に基づく自由曲面データである。このことは, データ転送後に CAD/CAM システムへ曲面制御点データを入力し, 再利用可能であることを意味する。細分割曲面でも曲面形状を表現することができるが, 曲面制御点を明示的に保持していないため, 細分割後の密なメッシュデータを入力する必要がある。また, ラティス構造では, 初期多角形を変形することで内部の曲面形状を変形することができるため, 変形アニメーションにおいて有効である。自由曲面の変形操作としては, 曲面制御ポリゴンによる変形が一般的である。しかし, この操作は曲面間の連続性を保存した状態での変形が困難であり, 意図する変形を実現するためには熟練を要する。6.4 節で詳細を述べるが, 初期多面体を変形媒体とすることで, 常に G^1 連続を保持した状態での変形が可能である。また, ラティス曲面と初期多面体の位相が同一であるために, 直感的な操作が可能である。更に, ラティス構造では, 曲面の分割数を変更することで, 用途に応じた詳細度制御をすることができる。転送先の計算機の処理能力が低い場合や, モデルを遠くの視点から閲覧する場合には, 少ない分割数を利用することで効率的な描画が実現できる。

4.1.2 丸め操作と逆丸め操作

本項では, 初期多面体からラティス曲面を生成する手法及び, 生成されたラティス曲面から初期多面体を再構成する手法について述べる。双 3 次 Gregory パッチの特徴を利用す

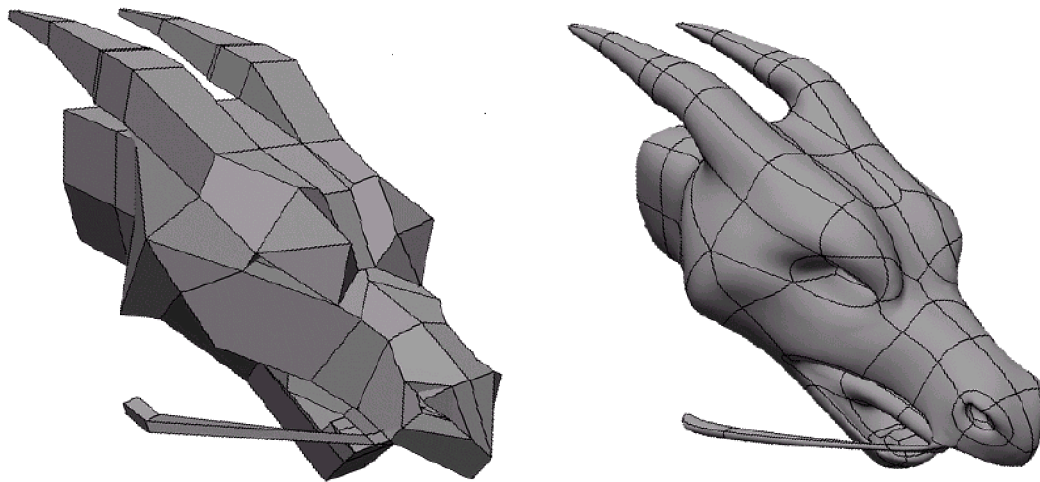


図 4.1: ラティス構造

ることにより，初期多面体と位相的に等しいラティス曲面を生成する曲面生成手法について解説する．ラティス曲面は，細分割曲面のようにポリゴンとして得られるものではなく，自由曲面の制御点データとして取得できる点が大きな特徴である．ラティス曲面を構築する曲面生成手法は初期多面体の線形変換として定義できるため，逆変換を求めることによりラティス曲面から初期多面体を再構築することが可能である．

ラティス曲面生成

ラティス曲面は丸め操作 (Rounding Operations) [8, 20, 34] によって生成される．丸め操作とは，単純なポリゴン形状を入力として滑らかな曲面形状を作成するモデリング手法のことである．従来の丸め手法は位相が著しく変化してしまい，生成された曲面形状と丸め前の格子形状の対応付けが困難であった．そのため，格子形状と丸め形状の連携が必要である直感的な変形操作や，格子形状の再構築などが困難であった．本手法の丸め操作は格子形状と位相的に等しい丸め形状を作成する．また，丸め形状から格子形状を再構築する（以後，逆丸め）ことが可能である．

本丸め手法は以下の 3 つの手順により構成される。

STEP1 初期多面体の各頂点に対応するラティス曲面の頂点を求める。

STEP2 初期多面体の各稜線に対応するラティス曲面の曲線を求める。

STEP3 曲線によって囲まれた領域に曲面を内挿する。

以下、各手順の詳細について述べる。

STEP1：ラティス曲面の頂点作成

まず、図 4.2(a) に示すように初期多面体 \mathcal{M}^0 の各頂点 $v_i (i = 0 \dots n)$ とする。初期多面体 \mathcal{M}^0 に Doo-Sabin 細分割を 1 回施し、図 4.2(b) に示すようなメッシュ \mathcal{M}^1 を得る。 \mathcal{M}^1 を構成する頂点 p_k^i は、 \mathcal{M}^0 の頂点 $v_i (i = 0 \dots n)$ を親に持つ。よって、同一の親を持つ頂点 p_k^i によって構成される面 $f_i (i = 0 \dots n)$ は、 \mathcal{M}^0 の各頂点 $v_i (i = 0 \dots n)$ に対応するものである (図 4.2(c))。これらの頂点の重心 $v'_i (i = 0 \dots n)$ は、Doo-Sabin 細分割において極限曲面上に乗る点であり、かつ今後の細分割ステップにおいて移動しない点である。この事実に着目し、これらの頂点 $v'_i (i = 0 \dots n)$ をラティス曲面の頂点とする (図 4.2(d))。なお、頂点 $v'_i (i = 0 \dots n)$ によって構成されるグラフ構造 (位相) は、初期多面体 \mathcal{M}^0 と等しくなる。これにより、初期多面体とラティス曲面の位相を 1 対 1 に対応させることが可能になる。

STEP2：ラティス曲面の曲線作成

初期多面体の線分に対応するラティス曲面の曲線を作成する方法について述べる。ラティス曲面の曲線データは 3 次 Bézier 曲線として表現する。これはラティス曲面の曲面表現式に双 3 次 Gregory パッチを用いているためである。図 4.3 (a) に示すように、初期多面体の線分 $v_0 v_1$ に対応する曲線を求めるとする。3 次 Bézier 曲線は 4 つの制御点によって構成されており、両端の制御点は曲線の終点及び始点と等しい (図 4.3 (b))。よって、図 4.3 (a) に示すように、曲線の始点と終点は、STEP1 で算出した c_0, c_3 にそれぞれ設定する。3 次 Bézier 曲線を定義するには、内側の 2 つの制御点 c_1, c_2 を求めればよい。まず、図 4.3 (c) に示すように、STEP1 により作成された f_0 に属する稜線の中点 q と $p_0 (= c_0)$ からベクトル $q - c_0$ を定義する。このベクトルを、3 次 Bézier 曲線の始点における接線ベクトルとする。同様に、終点における接線ベクトルを r と c_3 によって定義する。制御点 c_1, c_2 は、以下の式を満たす値として定義する。

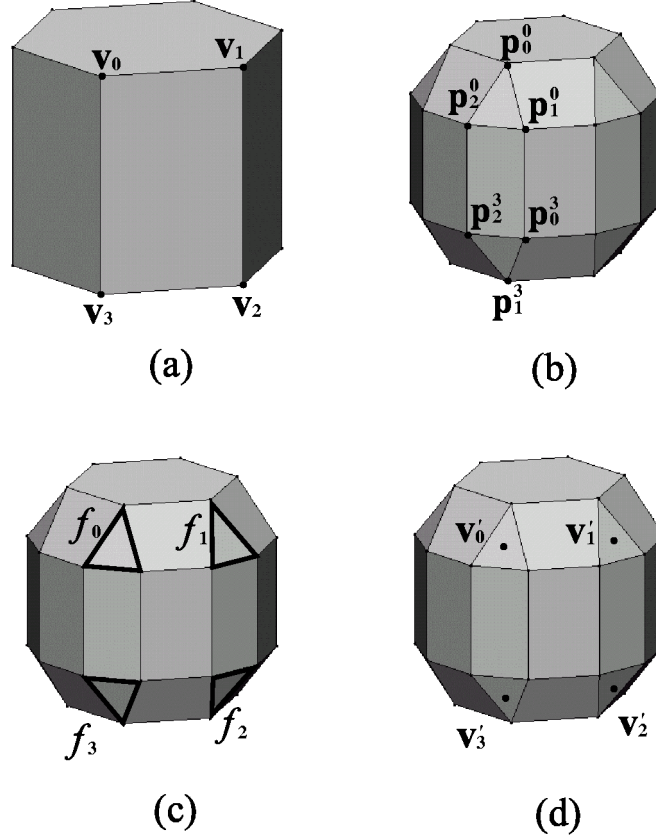


図 4.2: ラティス曲面の頂点計算 .

$$|\vec{c_0c_1}| = \frac{4}{3}|\vec{c_0q}| \quad |\vec{c_3c_2}| = \frac{4}{3}|\vec{c_3r}| \quad (4.9)$$

ここで, $\vec{c_0c_1}$ を $\vec{c_0q}$ の $4/3$ 倍の長さのベクトルとして定義する理由は, ラティス曲面が初期多面体の内部に収まるようにするためである. 3 次 Bézier 曲線が左右対称であると仮定すると, 図 4.3 (d) に示すように, 制御点も左右対称に配置される. パラメータ値 0.5 における曲線上の点を計算するには, 制御ポリゴンの各稜線をパラメータ値 0.5 で分割し, 新たに生成された頂点を接続するという操作を再帰的に行えばよい. 曲線上の点のがのる稜線を左右に延長し, 元の制御ポリゴンと交わる点を s_i, s_{i+1} とすると, 曲線の最小の凸包を構成する $c_0s_i s_{i+1}c_3$ を得る. この凸包が初期多面体であると仮定すると, $|\vec{c_0c_1}| = \frac{4}{3}|\vec{c_0s_1}|$, $|\vec{c_3c_2}| = \frac{4}{3}|\vec{c_3s_{i+1}}|$ になる. s_i, s_{i+1} をそれぞれ q, r とみなすと, 初期多面体に収まるラティス曲面を生成するには, $|\vec{c_0c_1}|$ を $|\vec{c_0q}|$ の $4/3$ 倍のベクトルとして定義するのが望ましいこと

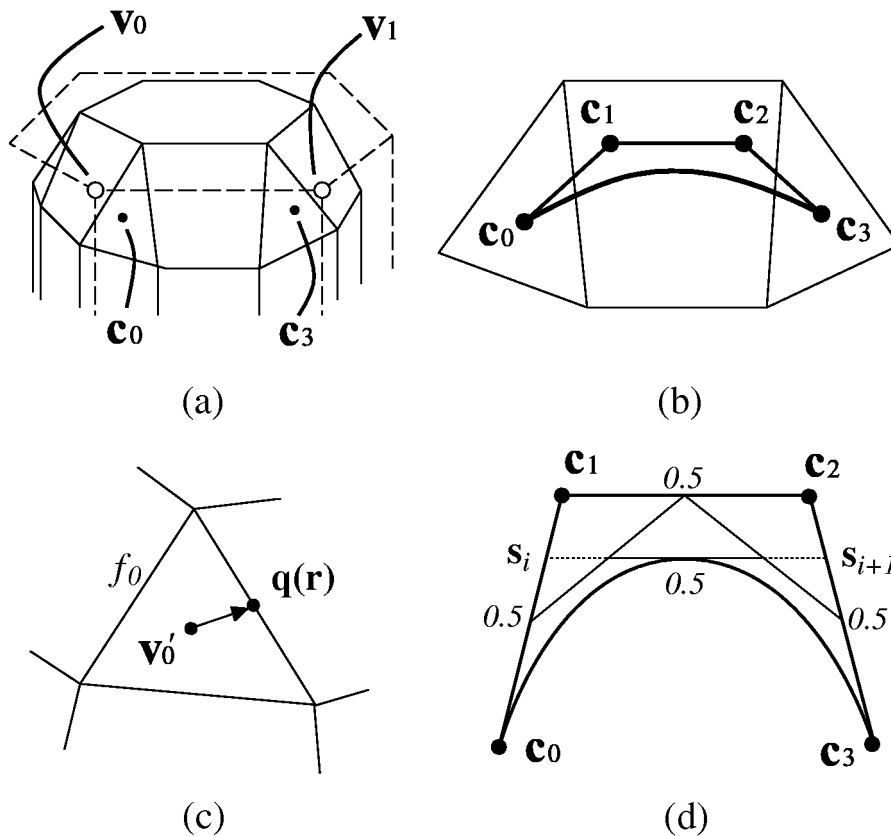


図 4.3: ラティス曲面の曲線計算 .

が分かる．初期多面体に極端な凹凸や非多様体に分類される部位が存在する場合は，Bézier 曲線同士が交差したり破断することがありうる．また，3 次 Bézier 曲線が左右対称であるという仮定の妥当性も問題になる．しかし，初期多面体を用意するのがユーザであるという一般的な条件下で使用する限り，これらの問題は致命的なことではない．

STEP3：曲面の内挿

曲線によって囲まれた領域に曲面を内挿する方法について述べる．STEP2 までの手順により，全ての領域は 3 次 Bézier 曲線により囲まれている．この領域に Chiyokura と Kimura[7] の手法を用いて双 3 次 Gregory パッチを内挿する．双 3 次 Gregory パッチの 20 個の制御点の中で境界に位置する 12 個の制御点は，STEP2 において求めた境界曲線の制御点に等しい．そのため，ここで求めるべきものは内部の 8 つの制御点である．ここで曲面間の連続性

を確保するために、彼らの手法では、基礎パッチ法 [55] と呼ばれる手法を用いている。この手法は、接続する曲面間の境界曲線上に仮想的なパッチを想定し、そのパッチとそれぞれの曲面を接続する手法である。以下に、基礎パッチ法による曲面制御点の内挿方法について述べる。

図 4.4 にあるように 2 枚の双 3 次 Gregory パッチ S^1 と S^2 が境界曲線を境に接しているとする。

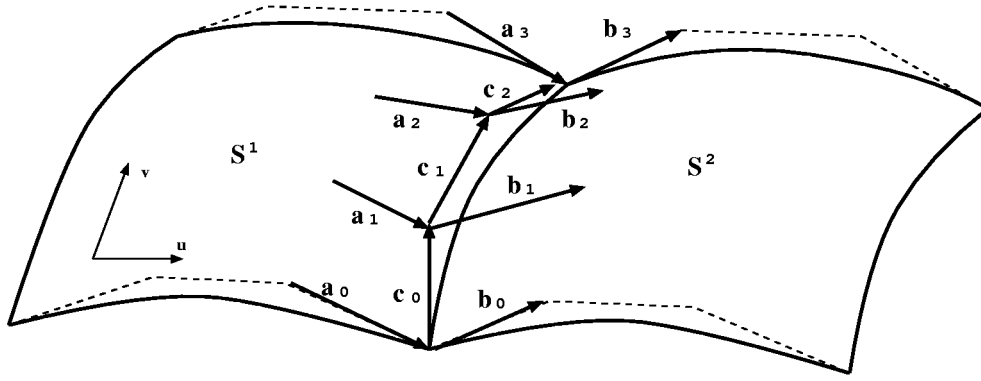


図 4.4: 双 3 次 Gregory パッチ間の接続

この時の境界に隣接する制御点間ベクトルを $a_i (i = 0, \dots, 3)$, $b_i (i = 0, \dots, 3)$, $c_i (i = 0, \dots, 2)$ とすると, S^1 と S^2 が G^1 連続で接続されるための条件は以下の式で表される。

$$\frac{\partial S^2(0, v)}{\partial u} = k(v) \frac{\partial S^1(1, v)}{\partial u} + h(v) \frac{\partial S^2(0, v)}{\partial v} \quad (4.10)$$

ここで, $k(v)$, $h(v)$ は任意のスカラー関数であり, 以下のような 1 次式であると設定する。

$$\begin{aligned} k(v) &= k_0(1 - v) + k_1v, \\ h(v) &= h_0(1 - v) + h_1v \end{aligned} \quad (4.11)$$

ただし, k_0 , k_1 , h_0 , h_1 は実数である。

式 (4.10) を制御点間ベクトルを使って表すと

$$\sum_{i=0}^3 B_i^3(v) \mathbf{b}_i = k(v) \sum_{i=0}^3 B_i^3(v) \mathbf{a}_i + h(v) \sum_{i=0}^2 B_i^2(v) \mathbf{c}_i \quad (4.12)$$

となる．ここで， $B_i^n(v)$ は Bernstein 基底関数である．この式から分かるように，内部制御点を表すベクトル \mathbf{b}_1 と \mathbf{b}_2 は $\mathbf{a}_i (i = 0, \dots, 3)$ と $\mathbf{c}_i (i = 0, \dots, 2)$ の多項式で表すことができる． $\mathbf{a}_0, \mathbf{a}_3$ と $\mathbf{c}_i (i = 0, \dots, 2)$ は既知の値であるので， $\mathbf{a}_1, \mathbf{a}_2$ の値が分かれば，式 (4.12) を解くことができる．

千代倉ら [55] は基礎パッチ法という手法を用いてこの問題を解決した．基礎パッチ法とは，曲面 S^1 と S^2 の境界上に仮想的なパッチを想定し，そのパッチと S^1, S^2 をそれぞれ接続する手法である．この際の仮想的なパッチは境界曲線状の流れベクトル関数を表している．そのため未知数である $\mathbf{a}_1, \mathbf{a}_2$ は基礎パッチによって既知数となり，式 (4.12) を解くことができる．図 4.5 にあるように，境界曲線の端点 V_0, V_1 における制御点間ベクトルを，それぞれ $\mathbf{p}_0, \mathbf{q}_0, \mathbf{r}_0$ と $\mathbf{p}_1, \mathbf{q}_1, \mathbf{r}_1$ とする．これらのベクトルは V_0, V_1 に接続する曲線の接ベクトルである．この境界曲線において，二つの曲面が G^1 連続で接続されるためには， $\mathbf{p}_0, \mathbf{q}_0, \mathbf{r}_0$ が同一平面にあり，かつ $\mathbf{p}_1, \mathbf{q}_1, \mathbf{r}_1$ が同一平面にあることが必要である．

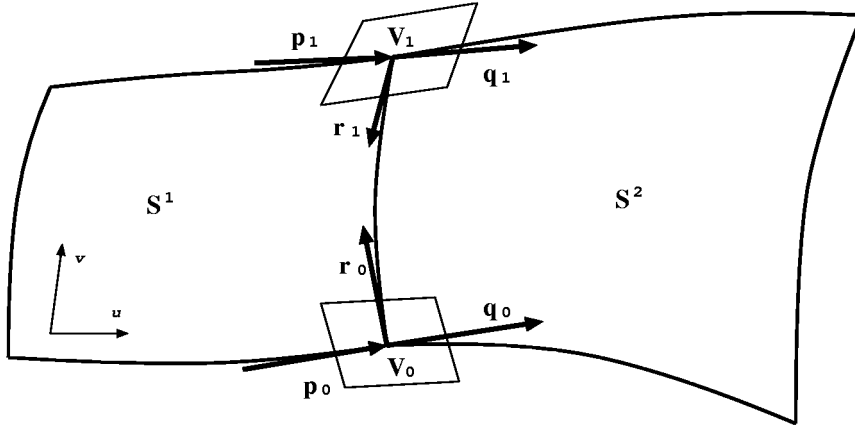


図 4.5: G^1 連続の条件

このとき，流れベクトル関数の要素 $d_i (i = 0, 1, 2)$ は以下ようになる．

$$\begin{aligned} d_0 &= \frac{\mathbf{p}_0 + \mathbf{q}_0}{|\mathbf{p}_0 + \mathbf{p}_0|}, \\ d_2 &= \frac{\mathbf{p}_1 + \mathbf{q}_1}{|\mathbf{p}_1 + \mathbf{p}_1|}, \\ d_1 &= \frac{d_0 + d_2}{2} \end{aligned} \quad (4.13)$$

このことから分かるように，流れベクトル関数の端点の要素 d_0, d_2 は境界曲線の端点における制御点ベクトルから定義され， d_1 は d_0, d_2 の線形補間により定義される．図 4.6 に基礎パッチによる曲面の内挿を示す．

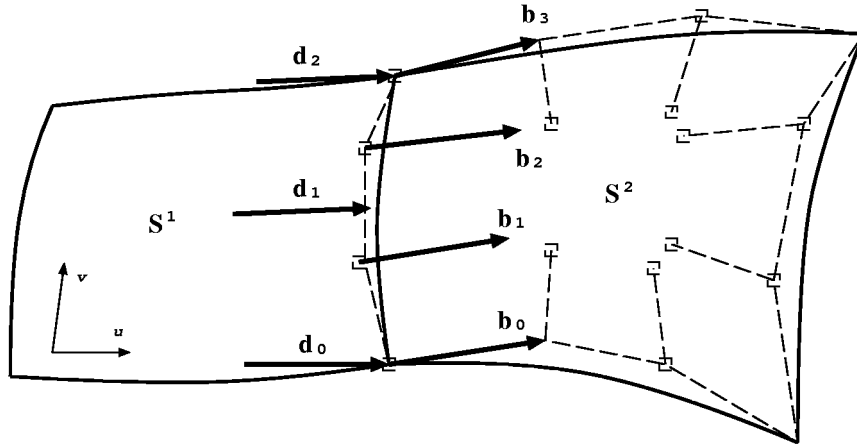


図 4.6: 基礎パッチによる曲面の内挿

以上の結果から，曲面の内部制御点 b_1, b_2 は以下のように計算できる．

$$\begin{aligned} b_1 &= \frac{2k_0}{3}d_1 + \frac{k_1}{3}d_0 + \frac{2h_0}{3}c_1 + \frac{h_1}{3}c_0, \\ b_2 &= \frac{k_0}{3}d_2 + \frac{2k_1}{3}d_1 + \frac{h_0}{3}c_2 + \frac{2h_1}{3}c_1 \end{aligned} \quad (4.14)$$

以上の手法により，曲面間が G^1 連続を保つように曲面を内挿することができる．

初期多面体の再構築

ラティス曲面を生成するには，双 3 次 Gregory パッチの境界曲線を構成する 3 次 Bézier 曲線の制御点を求めればよいことを述べた．3 次 Bézier 曲線の制御点は初期多面体 \mathcal{M}^0 を構成する頂点 v_i の線形結合によって取得することができる．そのため，ラティス曲面上のある頂点を v'_0 とし，それに対応する初期多面体上の頂点を v_i とすると，以下のような関係が成り立つ．

$$v'_0 = \sum k_i v_i \quad (4.15)$$

ただし， k_i は実数である．これはラティス曲面の制御点が初期多面体の頂点の線形結合で表せることを示している²．線形結合には，一般に逆変換が存在するため，初期多面体の頂点 v_0 は

$$v_0 = \sum l_i v'_i \quad (4.16)$$

として求めることができる．ただし， l_i は実数である．これらの関係は，ラティス曲面から初期多面体が再構築可能であることを意味する．ラティス曲面の各頂点に対応する初期多面体の各頂点を求めた後に，ラティス曲面の各曲線を直線に変換することで初期多面体が再構築される．

以下，この逆変換操作のことを逆丸め操作と呼ぶ．逆丸め操作により初期多面体を再構築することの意義は，変形操作における有効性にある．初期多面体は曲面形状の特徴を反映した単純な多面体であるために，変形操作における制御格子として用いることができる．初期多面体を変形媒体とすることで，曲面間の連続性を確保した状態での変形が容易になる．

4.1.3 重み付け

初期多面体の要素に重み付けをすることにより，ラティス曲面の形状を局所的に変形することが可能である．重みを付けることができるのは，初期多面体の頂点と稜線である．重み値を大きくすれば，生成されるラティス曲面の形状は初期多面体の形状に近づく．

稜線に重みを付けるには，図 4.3 (b) における 3 次 Bézier 曲線 $c_0c_1c_2c_3$ とそれに対応する初期多面体の線分 v_0v_1 に着目する．曲線の内部の制御点 c_1, c_2 を，それぞれ v_0, v_1 に近づけることによって稜線の重み付けがなされる．頂点に重みを付けるには，3 次 Bézier 曲線の両端の制御点 c_0, c_3 を，それぞれ v_0, v_1 に近づける．稜線，頂点ともに重みを最大に付け

²詳しい数学的な解説は 4.3.2 項にて述べる．

ると、3 次 Bézier 曲線 $c_0c_1c_2c_3$ は線分 v_0v_1 と等しくなり、生成される曲面形状も初期多面体と等しくなる。図 4.7 に重み付けを適用した例を示す。

更に、本手法では始点丸めベクトルと終点丸めベクトルという概念を用いる。これは、生成される 3 次 Bézier 曲線の始点と終点における接平面の法線ベクトルのことである。これらの要素を自由に調節することにより、より柔軟な曲面形状生成が実現される。

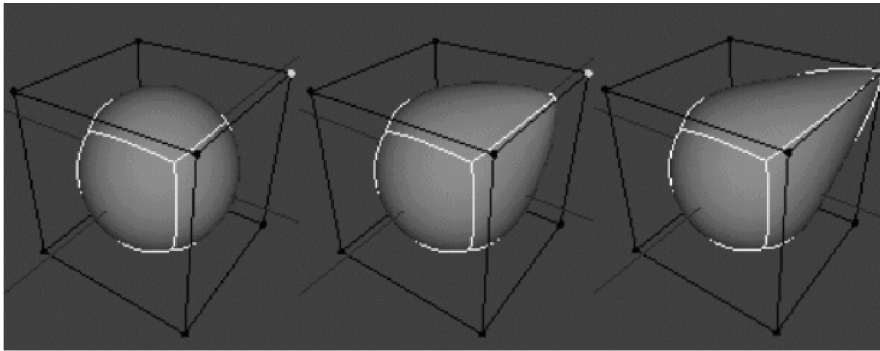


図 4.7: 重み付け

4.2 医用ラティス構造

本節では，医用ラティス構造について述べる．医用ラティス構造は，ラティス構造に以下の2点の改善を加えたものである．

1. 特徴線の反映
2. 精度の適応的制御

図 4.8 に示すような，ラティス構造のみによって3次元の医用モデルを表現する際には，いくつかの致命的な不具合が生じる．この不具合を改善するために，本論文では上記2点の拡張をラティス構造に加え，さらにそのモデリング手法についても解説していく．以下に，具体的に生じる不具合と，それを解決するにあたっての，上記の拡張の有効性について議論する．

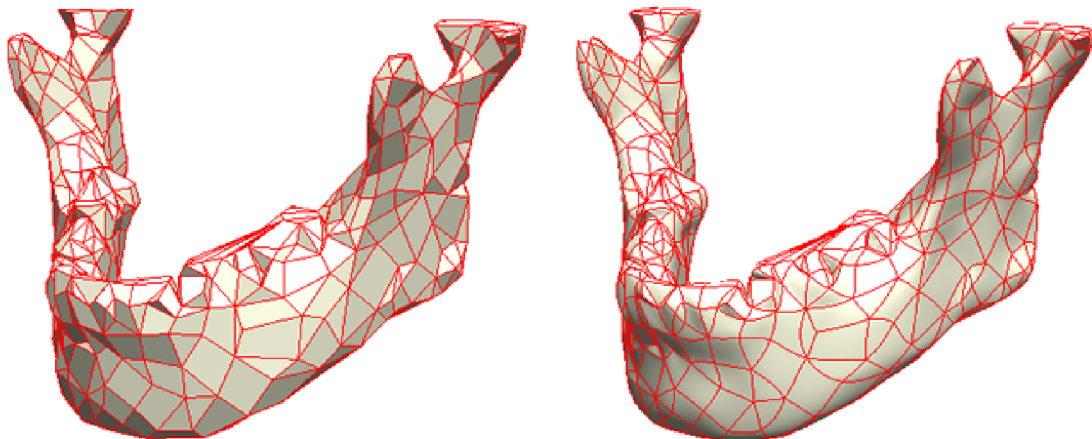


図 4.8: ラティス構造による医用3次元モデル

4.2.1 特徴の反映

図 4.8 において，左が初期多面体で右がラティス曲面である．この3次元データを形状の閲覧のみに使用するのであれば大きな問題はない．しかし，手術シミュレーションなどの高度な用途に利用する際には，医師が入力した切断線などの要素を明示的に保持する必要

がある．また，生体組織の境界線などのように，医学の知識を背景として初めてその存在を把握できる要素を保持することも必要である．手術シミュレーションとして切断操作を行う場合には，モデルを構成する位相要素が操作対象となる．図 4.8 において，パッチ間の輪郭線はランダムに作成されているものであり，切断部位にあたる部分に稜線列が存在していない．そのため，このモデルを利用して切断シミュレーションを実現することは困難である．

特徴線は，医師の手によって 3 次元モデルに直接入力されるべきものであり，ラティス曲面のパッチ間の境界線として明示的にモデルに反映されるべき要素である．高度な手術シミュレーションを実現するためには，特徴線を反映した位相構造をもつ 3 次元モデルを構築することが必須である．図 4.9 に特徴線を反映したモデルと反映していないモデルの比較を示す．左のモデルと比較して，右のモデルは切断線を明確に把握することができる．右のモデルであれば，切断線に当たる部分に稜線列が存在しているため，位相要素を参照した様々な変形操作を実現することが可能である．特徴線を反映した曲面モデルの生成手法については，4.3.3 項にて詳細を述べる．

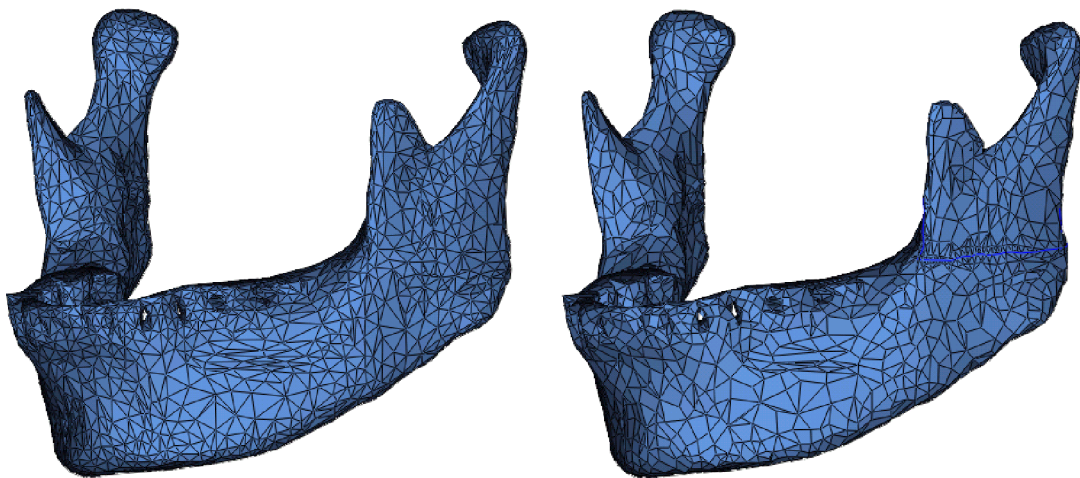


図 4.9: 特徴線の反映

4.2.2 精度の適応的制御

図 4.8 において，全てのパッチは任意の大きさを持っており，モデルの重要な部位とそうでない部位による差別化は行っていない．一般に，パッチの面積が大きい場合には，少ない制御点で広い面積を表現する必要があるため，特に詳細な形状を表現する場合には，小さい場合と比較すると誤差は大きくなる．モデルの患部にあたる領域では，高い精度が必要なために，パッチの面積が小さい状態であることが望ましい．それにより高い精度が保たれ，シミュレーションによる計測でも誤差を抑えることが可能になる．一方，患部以外の部分は，全体の形状把握のために利用されるので，高い精度は必要としない．そのため，これらの部位においてはパッチの面積を大きくすることで，全体のデータサイズを抑えることが可能になる．

部位ごとに精度を適応的に制御することで，高い精度を保持した状態でデータサイズを抑えることが可能になる．医用ラティス構造の作成プロセスでは，医師の手による高精度領域の入力が可能であり，それを反映したパッチの構成を実現している．図 4.10 に適応的な精度制御を実現した曲面モデルの例を示す．適応的な精度制御を実現した曲面モデルの生成手法については，4.3.3 項にて詳細を述べる．

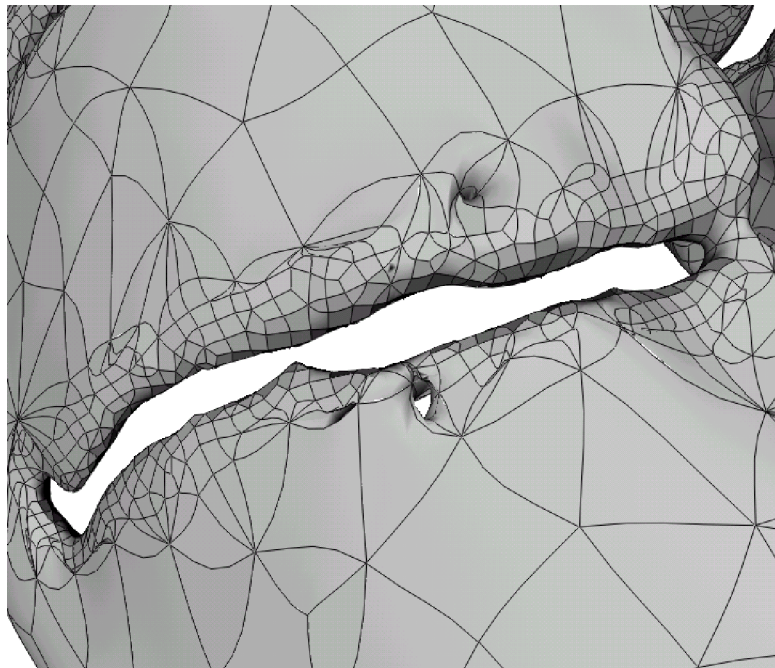


図 4.10: 精度の適応的制御

4.3 CT 画像からの医用ラティス曲面再構成

本節では，CT 画像から医用ラティス曲面モデルを再構成する手法について詳細に述べる．本再構築手法は以下に示す 3 つのステップから構成される．

1. CT 画像からの 3 角形メッシュ構築
2. 特徴線と精度の入力
3. 3 角形メッシュへの曲面フィッティング

以下，各ステップについて述べるが，説明の都合上，第 3 ステップを第 2 ステップより先に述べる．

4.3.1 CT 画像からの 3 角形メッシュ構築

CT 画像の 2 値化

CT 画像による 3 次元再構築の前処理として，CT データの 2 値化を行う．CT スキャナから取得される CT 画像は，X 線の減衰率から計算される値を濃淡情報に置き換えたグレースケール画像である．この濃度値は CT 値と呼ばれ，各組織によって異なる X 線吸収係数の差を濃淡情報に変換している．CT 値には，Hounsfield 単位と EMI 単位の 2 種類が存在するが，以下の解説では前者を用いる．この CT 値の違いにより，人体組織を区別することが可能である．生体の構成物質による CT 値の分布を図 4.11 に，CT 値による組織（臓器）のめやすを図 4.12 に示す．

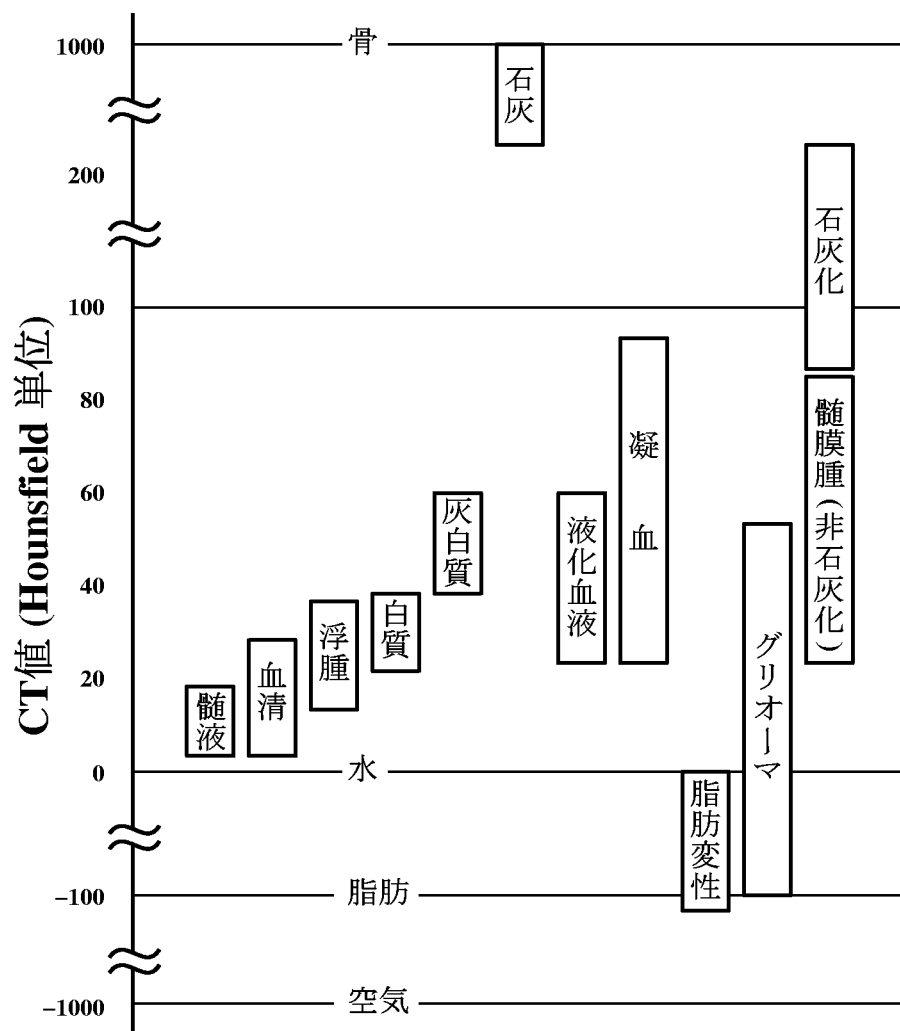


図 4.11: 構成物質による CT 値の分布

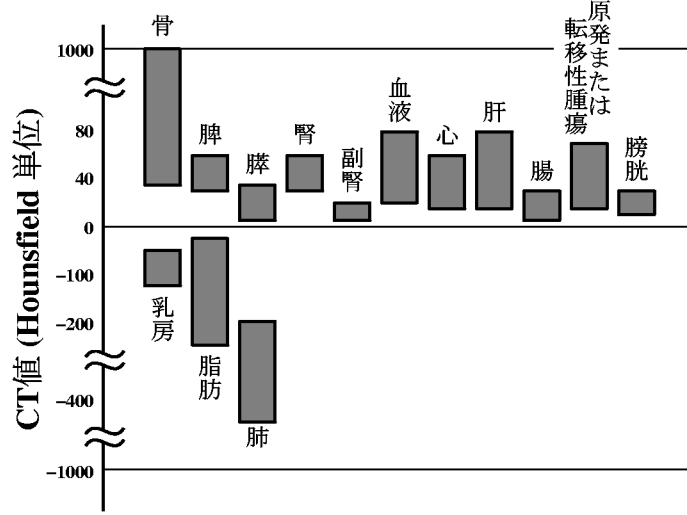


図 4.12: CT 値による人体組織のめやす

このグレースケール画像において，3次元再構成の対象となる組織を，他の組織から分離もしくは強調表示する必要がある．これを実現する代表的な手法が画像の2値化である．2値化とは，グレースケール画像などの多値画像から，白と黒のみによって構成される2値画像を得るための処理のことで，閾値処理 (Thresholding) により実現される．図 4.13 に CT 画像と，2値化処理後の CT 画像を示す．本研究の閾値処理は，対象組織に該当するピクセル値を 100，対象外組織のピクセル値を -100 として 2 値化を行っている．また，組織の CT 値はある程度の範囲に渡って分布するため，中央値と範囲を考慮して 2 値化を行う必要がある．座標 (x, y) における CT 値を $f(x, y)$ とし，中央値を c ，範囲を s とした場合の，閾値処理された値 $f_{c,s}(x, y)$ は，以下のように表すことができる．

$$f_{c,s}(x, y) = \begin{cases} 100 & c - s \leq f(x, y) \leq c + s \text{ の時} \\ -100 & f(x, y) < c - s, \ c + s < f(x, y) \text{ の時} \end{cases} \quad (4.17)$$

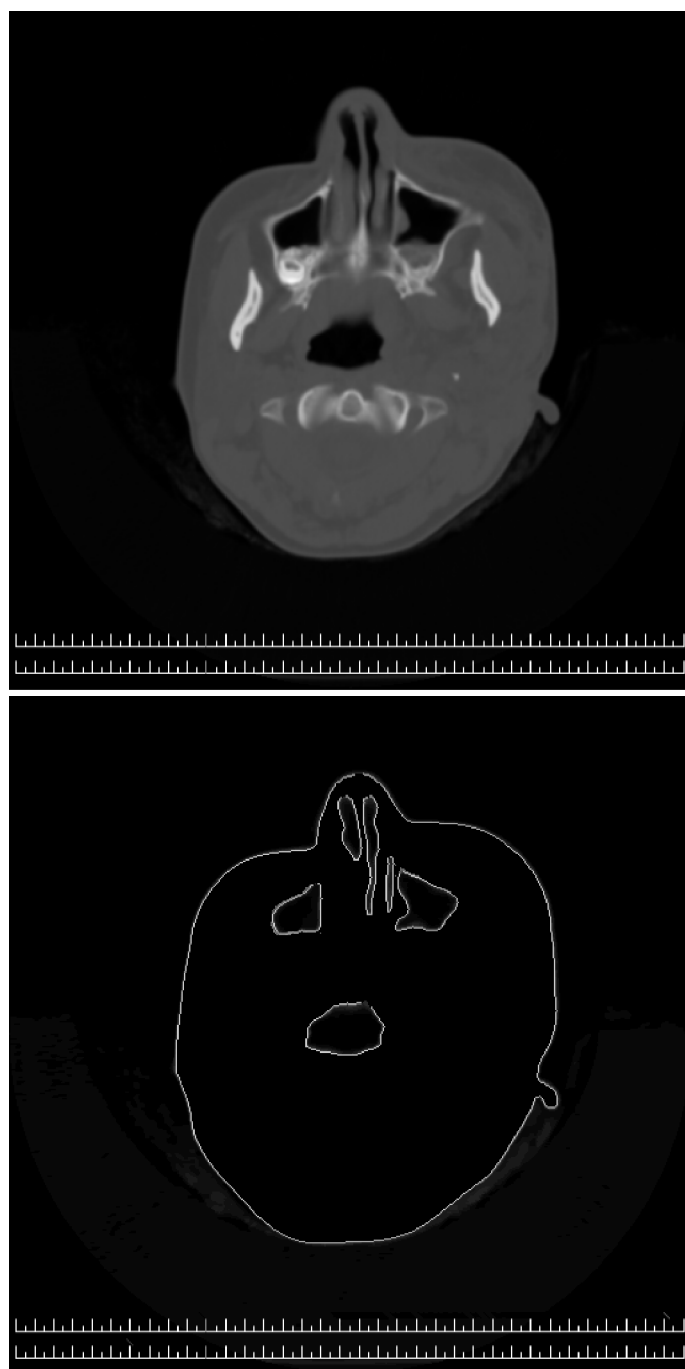


図 4.13: CT 画像と 2 値化処理後の画像

マーチングキューブス法

2 値化処理された CT 画像から，3 次元の 3 角形ポリゴンメッシュを構築する際には，マーチングキューブス法 [35, 29] を利用する．マーチングキューブス法は，3 次元空間を細かい単位格子によって分割し，近隣 8 つの単位格子の内部にモデルが存在するか否かによってパターン化されたポリゴンデータを当てはめる手法である．

具体的には，直方体を形成するような近隣の参照点 8 個の集合（セル）において，各参照点の値が 0 か 1 かを調べ，そのパターンに応じてポリゴンデータを生成する．この操作を全てのセルに適用することで，全体形状をポリゴンデータの集合として表現することができる．近隣の 8 つの参照点には 1 か 0 の値が割り振られているので，1 つのセルについては， $2^8 = 256$ パターンの配置が存在することになる．この 256 パターンについて，幾何学的な対象形状を考慮すると，最終的には 15 パターンの配置によって代用することが可能になる．なお，幾何学的な対象操作とは，回転形，反射形，0 と 1 を反転させた形のことである．この 15 パターンのそれぞれについて，3 角形面の配置方法を定義し，全てのセルについてパターン化された 3 角形面を当てはめると，結果として全体のポリゴン形状の集合を取得することができる．図 4.14 に，15 パターンの 3 角形面の配置について示す．図において，頂点が表示されている参照点が 1 の値を持ち，頂点が表示されていない参照点は 0 の値を持つことを表している．

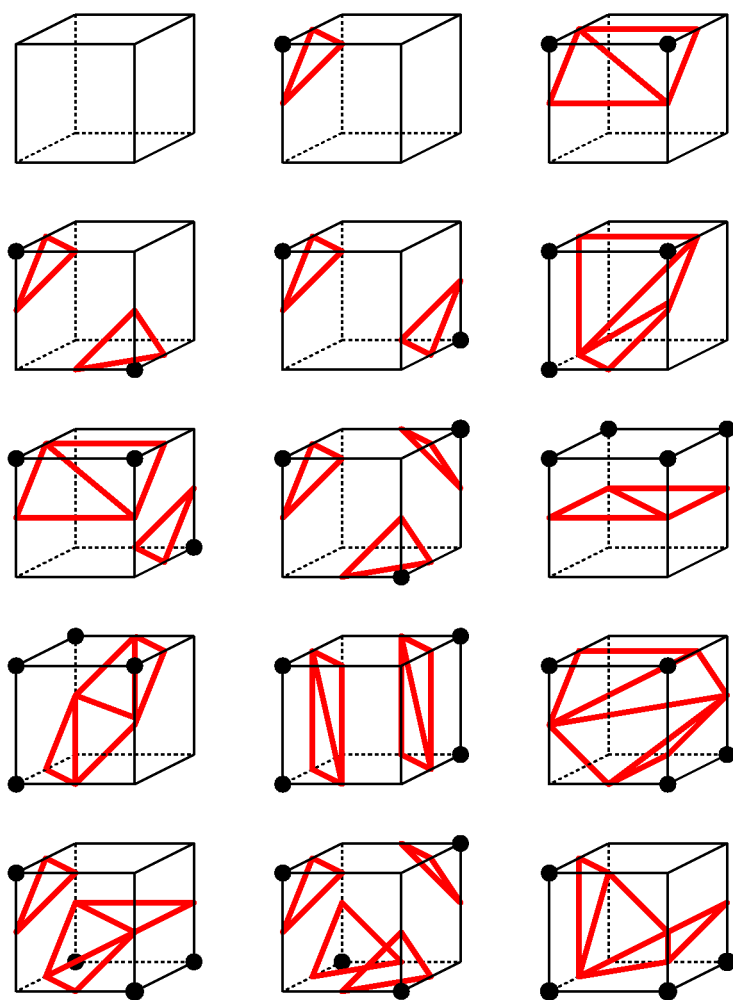


図 4.14: マーチングキューブス法の 15 個のパターン

3 角形メッシュの編集

マーチングキューブス法によって3角形メッシュを自動的に再構成することが可能であるが、2値化の際の閾値処理では、3次元再構成の対象組織以外の画像も含まれてしまうことがある。これらの余分な組織モデルを除去するために、CTスライスごとに閾値を再編集し、各CTスライスに対応する断面を再編集するシステムを実装した。この操作は即時に3角形メッシュに反映することが可能であり、インタラクティブな3次元再構成を実現することができる。図4.15にシステムの概観を、図4.16に3次元再構築例として頭蓋モデルを示す。

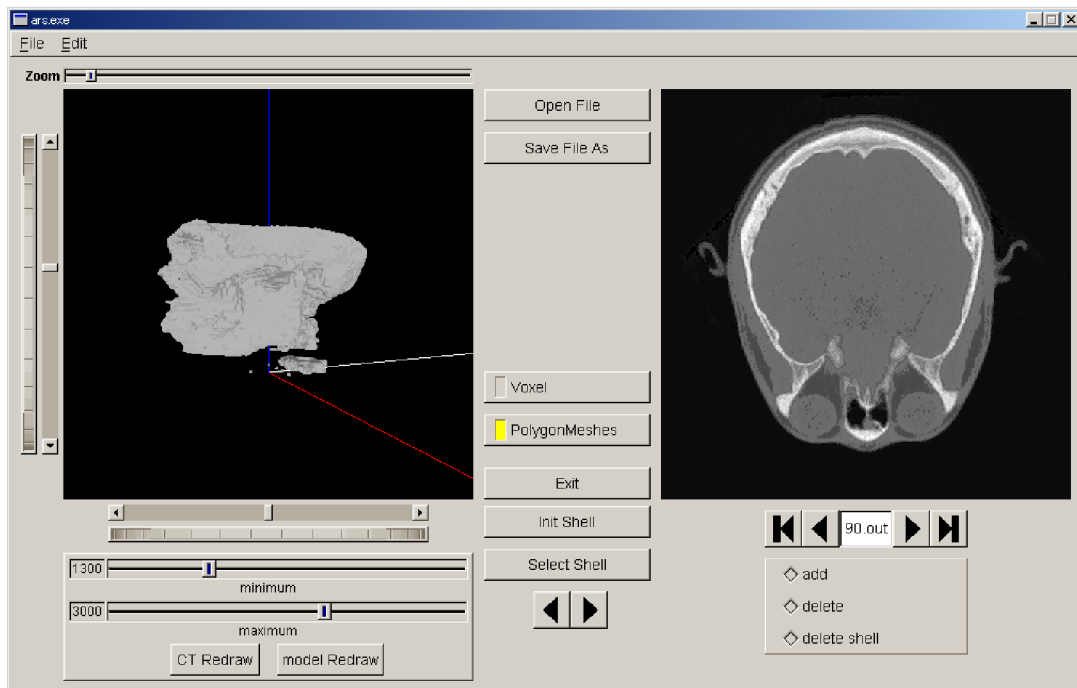


図 4.15: 3次元再構成再編集システムの概観

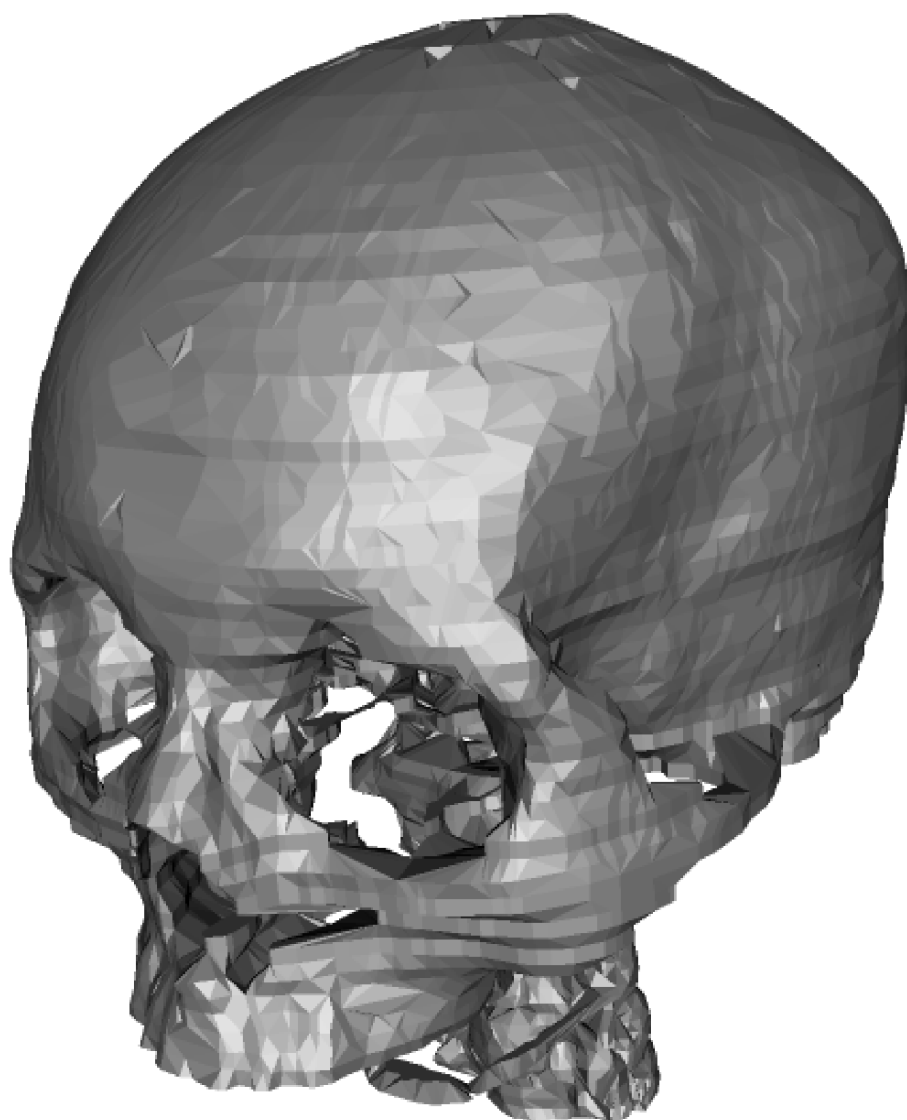


図 4.16: 3次元再構成例（人体の頭蓋）

4.3.2 3 角形メッシュへのラティス曲面フィッティング

3次元再構成された3角形メッシュに対して、ラティス曲面をフィッティングし、ラティス構造への変換を行う。この処理による出力は、ラティス構造に基づく3次元モデルであり、医用ラティス構造に基づく3次元モデルではない。説明を分かりやすくするために、医用ラティス構造への対応については4.3.3項にて詳しく述べることにし、本節ではラティス構造への変換手法について述べる。まず、ラティス曲面フィッティングの基礎となる、QEM(Quadric Error Metrics) メッシュ簡略化 [16] について解説する。次に、QEM メッシュ簡略化を用いた Doo-Sabin 細分割曲面へのフィッティング手法について解説し、それを拡張することによって、ラティス曲面へのフィッティングを行う手法について解説する。

QEM による 3 角形メッシュの簡略化

Garland らは QEM を用いたメッシュ簡略化手法を提案している。この手法は、QEM と呼ばれる評価関数を用いることで、3 角形メッシュの各稜線に重みを付け、重みの少ない稜線から順番に、稜線消去 (Edge Collapse) という位相操作を繰り返すことによって実現されている。図 4.17 に示すように、稜線消去は稜線の両端の頂点 v_1, v_2 を頂点 \bar{v} に統合することによって、1 つの稜線 e を消去する操作のことである。1 つの稜線 e が削除されると同時に、2 つの稜線と 2 つの面、1 つの頂点も消去される。

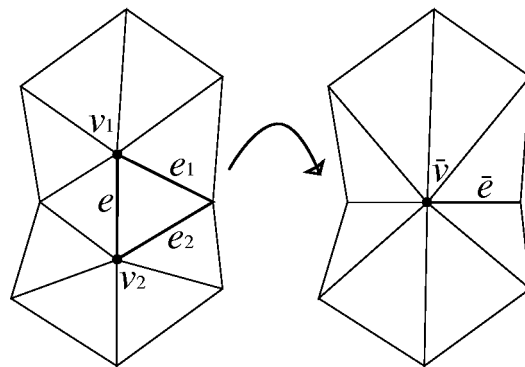


図 4.17: 稜線消去

以下に，QEM の定義について述べる．3 角形メッシュ M の各面 f に対して，2 次関数 $Q^f(\mathbf{x})$ を，ある頂点 \mathbf{x} から面 f を含む平面への距離の 2 乗として定義する．図 4.18 に示すように，ここで \mathbf{n} は面 f の法線ベクトルであり， d は原点から平面までの符号付スカラーである．

$$Q^f(\mathbf{x}) = (\mathbf{n}^T \mathbf{x} + d)^2 = \mathbf{x}^T (\mathbf{n} \mathbf{n}^T) \mathbf{x} + 2d \mathbf{n}^T \mathbf{x} + d^2 \quad (4.18)$$

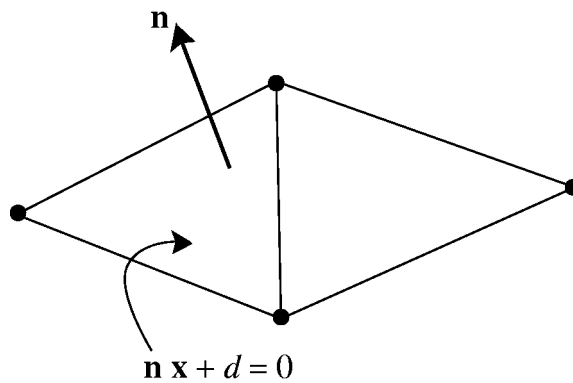


図 4.18: 面の方程式

次に，3 角形メッシュ M の各頂点 v に対して，その頂点に接続している面の $Q^f(\mathbf{x})$ の面積重み付け和を割り当てる．式 (4.19) において， \mathbf{A} は 3×3 の対称行列， \mathbf{b} は列ベクトル， c はスカラーである． $area(f)$ は三角形 f の面積を示す．面積による重み付けは，3 角形の大きさによって QEM の値が変化するのを防ぐために行う．

$$Q^v(\mathbf{v}) = \sum_{f \ni v} area(f) \cdot Q^f(\mathbf{v}) = \mathbf{v}^T \mathbf{A} \mathbf{v} + 2\mathbf{b}^T \mathbf{v} + c \quad (4.19)$$

これにより，各頂点 v には，

$$Q^v = (\mathbf{A}, \mathbf{b}, c) \quad (4.20)$$

を構成する $6 + 3 + 1 = 10$ 個の浮動小数点が蓄積されることになる．

3 角形メッシュ M の各稜線 e に対して，稜線消去 ($\{v_1, v_2\} \rightarrow \bar{v}$) 実行後の \bar{v} の頂点座標値を以下の方法により求める．まず， v_1, v_2 が保持している QEM の和 ($Q^{\bar{v}} = Q^{v_1} + Q^{v_2}$) を最小にするような頂点座標 \mathbf{v}^{min} を求める．これは， $Q^v(\mathbf{v})$ の勾配 $\nabla Q^v(\mathbf{v})$ が 0 になる座

標位置であると考え、すると、稜線消去後の頂点座標 \mathbf{v}^{min} は、以下の線形方程式を解くことにより求めることができる。

$$\mathbf{A}\mathbf{v}^{min} = -\mathbf{b} \quad (4.21)$$

$Q^{\bar{v}}(\mathbf{v}^{min})$ を稜線 e のコストと呼ぶ。3 角形メッシュ M の各稜線 e に対して、コストを計算し、コストの小さい稜線から順番に稜線消去を実行する。稜線消去を 1 回実行した後は、 \bar{v} に接続する稜線についてのみ、コストを再計算する。コストの再計算は高速に行うことができるため、非常に高速な 3 角形メッシュ削減が実現される。以下に、簡略化のアルゴリズムを示す。

1. 初期メッシュの全ての頂点 \mathbf{v} に対して $Q^v = (\mathbf{A}, \mathbf{b}, c)$ を計算する。
2. 初期メッシュの全ての稜線に対して、稜線 (v_1, v_2) を削除した時の新しい頂点 \bar{v} の最適座標 $\bar{\mathbf{v}}$ を計算する。誤差関数 $\bar{\mathbf{v}}^T(Q^v(\mathbf{v}_1) + Q^v(\mathbf{v}_2))\bar{\mathbf{v}}$ が稜線 (v_1, v_2) のコストになる。
3. 全ての稜線を要素とするヒープを構築する。この際、コストの最小の稜線が先頭になる昇順とする。
4. コストの小さい稜線から稜線消去 $(\mathbf{v}_1, \mathbf{v}_2 \rightarrow \bar{\mathbf{v}})$ を実行し、 $Q^{\bar{v}}(\bar{\mathbf{v}})$ を $Q^{\bar{v}} = Q_1 + Q_2$ とする。また、 $\bar{\mathbf{v}}$ に接続する全ての稜線の QEM を更新する。
5. 終了条件になるまで、3~4 の操作を繰り返す。

QEM による細分割曲面のフィッティング

3 角形メッシュの QEM は, 3 角形を削減する度に, 現在のメッシュ M^i とオリジナルメッシュ M^0 との誤差を評価する. これにより, 簡略化メッシュはオリジナルメッシュを近似する形状となる.

一方, Takeuchi ら [45] は, 3 角形を削減する度に, 現在のメッシュ M^i に Doo-Sabin 細分割を適用した極限曲面 S とオリジナルメッシュ M^0 との誤差を評価する. これにより, 簡略化メッシュは細分割曲面における初期メッシュになり, 初期メッシュに細分割操作を適用することで生成される極限曲面が, オリジナルメッシュを近似する形状となる. このことは, 3 角形メッシュにより構成される 3 次元モデルを, 細分割曲面により構成される 3 次元モデルへと変換することを意味する. 一般に, このような変換を曲面フィッティングと呼び, Takeuchi らの手法は Doo-Sabin 細分割曲面へのフィッティングと考えることができる.

また, 金井 [54] は Takeuchi らの手法を拡張し, 対象となる頂点の近傍頂点についても評価を行い, より正確な頂点座標値の計算を可能にしている. Garland らの手法においては, QEM は頂点ごとに管理されている. そして, 稜線消去を施した際に, 新たに計算が必要な頂点だけについて再評価を行う. 金井による拡張では, QEM の管理は頂点と稜線の両方で行う. 簡略化段階のメッシュ M^i の各稜線 e において, 以下の式を持つことになる.

$$Q^e = Q^{f_l} + Q^{f_r} \quad (4.22)$$

ここで, f_l, f_r は e に隣接する面を示す. アルゴリズムの開始時に各稜線 e について Q^e を計算する. アルゴリズム途中の各稜線消去において, 消去される稜線 e の持つ QEM は, 新しく生成される頂点 \bar{v} に格納される. つまり, 稜線消去を繰り返していく過程で頂点に QEM が蓄積していくことになる. これを一般化すると, 図 4.17 において,

$$\begin{cases} Q^{\bar{v}} = Q^e + Q^{v_1} + Q^{v_2} \\ Q^{\bar{e}} = Q^{e_1} + Q^{e_2} \end{cases} \quad (4.23)$$

という QEM の更新を行う.

QEM によるラティス曲面のフィッティング

本研究では金井の手法に以下のような改良を加えた．本手法において，簡略化メッシュはラティス構造における初期多面体になり，初期多面体に丸め操作を適用することで生成されるラティス曲面が，オリジナルメッシュを近似する形状となる．そのために，3 角形を削減する度に，現在のメッシュ M^i に丸め操作を適用したラティス曲面 S とオリジナルメッシュ M^0 との誤差を評価する．ラティス構造において，初期多面体とラティス曲面の位相要素は 1 対 1 に対応するため，図 4.19 に示すように，初期多面体の頂点 v に対応するラティス曲面上の頂点 v' を評価する．また，金井の手法と同様に，頂点のみならず，近傍頂点についても評価を行う．近傍頂点の QEM については，ラティス曲面の輪郭曲線を構成する各 3 次 Bézier 曲線のパラメータ 0.5 における点 s を評価する．

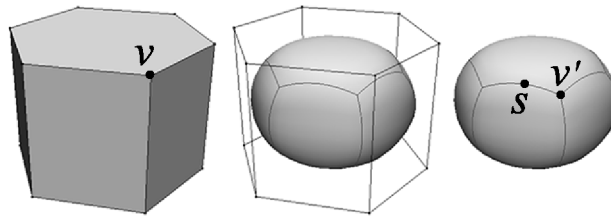


図 4.19: ラティス曲面フィッティングにおける QEM の評価点

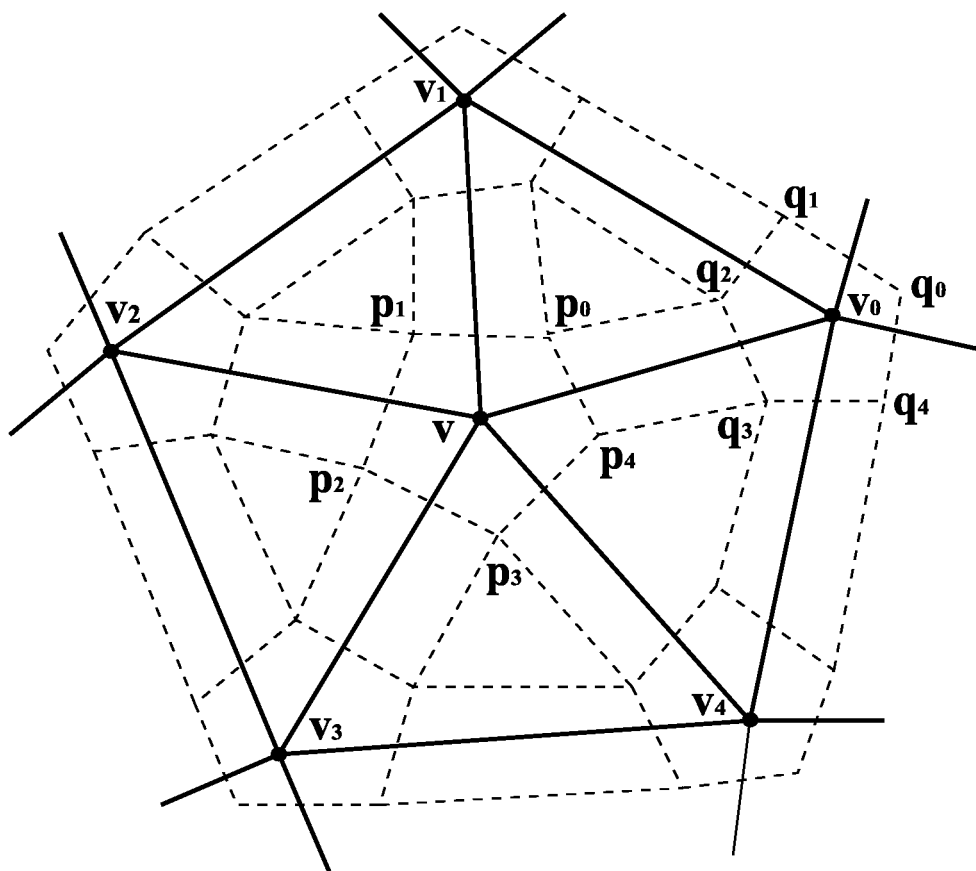


図 4.20: 頂点近傍の参照点の算出 - 1

図 4.20 において，実線で表されているメッシュはオリジナルメッシュ \mathcal{M}^0 を示す．また，破線で表されているメッシュは，オリジナルメッシュに対して Doo-Sabin 細分割を 1 回施したメッシュ \mathcal{M}^1 を示している．頂点 $v_j (j = 0 \dots k)$ はオリジナルメッシュにおいて，頂点 v に隣接している頂点である．ここで， j 及び k は実数である．頂点 $p_j (j = 0 \dots k)$, $q_j (j = 0 \dots k)$ は，それぞれオリジナルメッシュに対して Doo-Sabin 細分割を 1 回施した時点の頂点での v, v_0 に対応する頂点である．

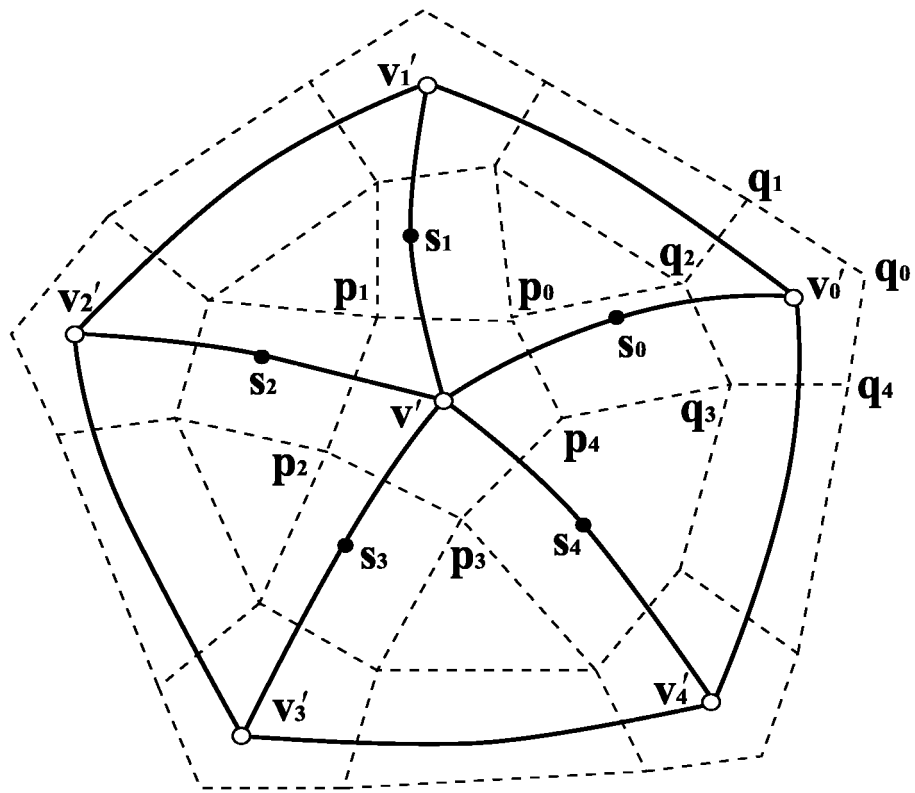


図 4.21: 頂点近傍の参照点の算出 - 2

図 4.21 において、破線で表されているメッシュは、オリジナルメッシュに対して Doo-Sabin 細分割を 1 回施したメッシュ \mathcal{M}^1 を示している。実線で表されている曲面では、 \mathcal{M}^1 を元に算出したラティス曲面を表している。 v' 及び $v'_j (j = 0 \dots k)$ は、 v, v_0 に対応する丸め操作後のラティス曲面上の点である。 v' と $v'_j (j = 0 \dots k)$ を結ぶ曲線はラティス曲面の輪郭線であり、それぞれ 3 次 Bézier 曲線によって表現されている。 $s_j (j = 0 \dots k)$ は 3 次 Bézier 曲線における、パラメータ値 0.5 に対応する点である。

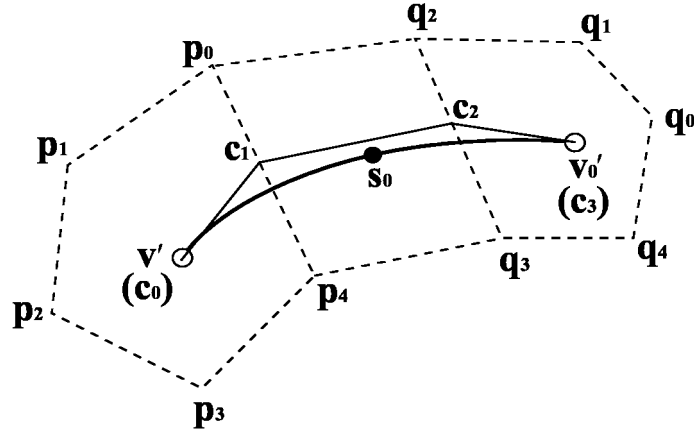


図 4.22: 頂点近傍の参照点の算出 - 3

図 4.22 は図 4.21 の v', v'_0 付近を拡大したものである．3 次 Bézier 曲線 $v'v'_0$ は 4 つの制御点 $c_0c_1c_2c_3$ によって制御されている．このうち，両端の制御点 c_0, c_3 はそれぞれ v', v'_0 と等しい．

以上の前提のもとで，本手法における評価関数について詳細を述べる．本手法においては，曲面上の点 v' と， v' に接続する 3 次 Bézier 曲線の中点 s_j を QEM で評価する．そのため，評価関数は以下のように記述することができる．

$$Q^v(v) = Q^v(v') + \sum_{j=1}^k Q^f(s_j) \quad (4.24)$$

式 (4.24) の右辺の第 1 項は，頂点 v 付近の形状を評価したものであり， v のもつ QEM が使用される．第 2 項は， v に隣接する稜線付近の形状を評価したものであり，稜線の持つ QEM が使用される．最適化計算に関しては， v だけを変数として考える．つまり， $Q^v(v)$ が最小となるような v を求める．そのために，以下に v に注目した式の書き下しを行う．

式 (4.24) における v', s_j は，以下のようにそれぞれ v の 1 次関数の形で表すことができる．

$$v' = \alpha v + e \quad (4.25)$$

$$s_j = \alpha_j v + e_j \quad (4.26)$$

ここで, α, α_j は \mathbf{v} の係数であり, e, e_j は残りのベクトル全てを集めたものである. 式 (4.25), (4.26) を用いて式 (4.24) を展開し整理すると, 以下のようにまとめることができる.

$$Q^v(\mathbf{v}) = \mathbf{v}^T A \mathbf{v} + 2b^T \mathbf{v} + c \quad (4.27)$$

これは \mathbf{v} に関する 2 次関数となる. $Q^v(\mathbf{v})$ を最小にする頂点座標 \mathbf{v}^{min} は他の QEM 同様 $\nabla Q^v(\mathbf{v}) = 0$, すなわち以下の式を解くことで得ることができる.

$$A \mathbf{v}^{min} = -b \quad (4.28)$$

次に補足として、式 (4.24) における \mathbf{v}' と \mathbf{s}_j を \mathbf{v} によって算出する方法について述べる。頂点 \mathbf{v} の丸め操作後の頂点 \mathbf{v}' は、オリジナルメッシュを Doo-Sabin 細分割した後の、頂点 \mathbf{v} に対応する面の重心である。よって \mathbf{v}' は以下の式によって求めることができる。

$$\mathbf{v}' = \frac{1}{k} \cdot \sum_{j=1}^k \mathbf{p}_j \quad (4.29)$$

ここで、 k は \mathbf{v} に接続する稜線数である。頂点 $\mathbf{v}, \mathbf{v}_j, \mathbf{v}_{j+1}$ によって構成される面の重心を \mathbf{t}_j とすると、式 (3.6) より、

$$\begin{aligned} \mathbf{p}_j &= \frac{1}{2}\mathbf{v} + \frac{1}{4}\mathbf{t}_j + \frac{1}{8}\mathbf{v}_j + \frac{1}{8}\mathbf{v}_{j+1} \\ &= \frac{7}{12}\mathbf{v} + \frac{5}{24}\mathbf{v}_j + \frac{5}{24}\mathbf{v}_{j+1} \end{aligned} \quad (4.30)$$

式 (4.29) に式 (4.30) を代入すると、以下のように \mathbf{v} で書き下すことができる。

$$\mathbf{v}' = \frac{7}{12}\mathbf{v} + \frac{5}{12} \sum_{j=1}^k \mathbf{v}_j \quad (4.31)$$

次に頂点 \mathbf{v}' に接続する 3 次 Bézier 曲線の中点 \mathbf{s}_j の算出方法について述べる。3 次 Bézier 曲線の制御点を $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$ とすると、一般的にパラメータ 0.5 における点 \mathbf{s} は以下の式によって求められることが知られている。

$$\mathbf{s} = \frac{1}{8}\mathbf{c}_0 + \frac{3}{8}\mathbf{c}_1 + \frac{3}{8}\mathbf{c}_2 + \frac{1}{8}\mathbf{c}_3 \quad (4.32)$$

図 4.22 に示すように、頂点 \mathbf{v}' と \mathbf{v}'_j に接続する 3 次 Bézier 曲線の \mathbf{c}_0 は \mathbf{v}' 、 \mathbf{c}_3 は \mathbf{v}'_j である。式 (4.30)、式 (4.31)、式 (4.32) を用いて、 \mathbf{s}_j を \mathbf{v} とその近傍頂点で書き下すと以下のようになる。

$$\mathbf{s}_j = \frac{3}{8}\mathbf{v} + \frac{1}{8}\mathbf{v}_{j-1} + \frac{3}{8}\mathbf{v}_j + \frac{1}{8}\mathbf{v}_{j+1} \quad (4.33)$$

4.3.3 特徴と精度の入力

4.3.2 項の QEM に対して故意的なコントロールを行うことで、医用ラティス曲面の構築に必要な特徴線の反映と精度の適応的制御を行う手法について述べる。

特徴線をモデルに反映させる手法については、Lee ら [31] や Khodakovsky ら [28] が報告している。Lee らの手法は、特徴線になる稜線列を選択し、メッシュ簡略化の際のリストから削除してしまうことで、結果的に特徴線として残存させるものである。この手法では、特徴線列を位相要素として残すことが可能である。それに対して、Khodakovsky らの手法は、特徴線列はオリジナルメッシュの位相に関係なく描画することが可能である。描画された特徴線列を法線ベクトルのフィルタ情報とし、極限曲面上の頂点を移動させることで特徴線を浮き上がらせる手法である。本手法では、位相要素である稜線列として特徴線をモデルに反映する必要があるため、Lee らと同様のアプローチを取る。

精度の適応的制御については、Zorin ら [52] が Loop 細分割曲面に適用させる手法について報告している。彼らの手法では、部位に応じて細分割を適応的に適用することで、パッチの大きさと精度をコントロールしている。本手法では、入力となる 3 角形メッシュにおいて、精度が不要な領域のみを簡略化することで精度の適応的制御を実現している。以下、本手法の詳細について述べる。

精度の適応的制御

4.3.2 項により、3 角形メッシュにラティス曲面をフィッティングする手法について解説した。ラティス曲面のフィッティングでは、QEM を用いて 3 角形メッシュの簡略化を行っているが、この際には QEM の小さい稜線から削減操作を適用している。つまり、QEM の高い要素は形状における特徴的な部分を示すことになり、削減される順番は遅くなることになる。このことに着目し、本手法では高精度を要求する領域に対して、高い QEM を意図的に付加する。これにより、その部位の削減操作が実行される頻度は少なくなり、高い精度が保たれる。

高い精度を付加されるべき領域を \mathcal{X} とし、 \mathcal{X} に含まれる頂点を \mathbf{v}_j とする。このとき、 \mathbf{v}_j のコスト $Q^v(\mathbf{v}_j)$ は制約値を U とした場合、以下の式によって求めることとする。

$$Q^v(\mathbf{v}) = Q^v(\mathbf{v}^{min}) \cdot U \quad (4.34)$$

制約値 U が大きいほど、領域 \mathcal{X} は高い精度が保たれることになる。メッシュを簡略にすればする程、全ての稜線の QEM は大きくなっていくので、制約値 U も大きくする必要がある。

以上の処理により，患部における高い精度の制御が可能になるが，患部以外の領域は精度を下げて概形のみを表示することが求められる．これらの領域には高い QEM が与えられていないために，メッシュ簡略化の過程で稜線消去が実行される．本手法では，これらの稜線について 4 角形化を行うことで更なる簡略化を行う．4 角形化は以下のステップにより実行される．

1. 簡略化メッシュ M^n について，稜線の QEM をキーとしたヒープを構築する．
2. 最も QEM が小さい稜線から順番に参照し，その稜線に隣接する 2 つの 3 角形が長方形に近い場合には，その稜線を削除し 4 角形化を行う．
3. 作成された 4 角形に属する稜線をヒープから削除する．
4. 終了条件になるまで，2. ～ 3. の操作を繰り返す．

上記の 2. において，4 角形が長方形に近いか否かを評価する理由は，長方形に近い 4 角形面ほど双 3 次 Gregory パッチが滑らかに内挿できるからである．このための評価関数は以下のものである．

$$|F(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2)| + |F(\mathbf{p}_0, \mathbf{p}_2, \mathbf{p}_3)| < S_1 \quad (4.35)$$

$$| (|F(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2) + F(\mathbf{p}_0, \mathbf{p}_2, \mathbf{p}_3)| - |F(\mathbf{p}_1, \mathbf{p}_0, \mathbf{p}_2) + F(\mathbf{p}_0, \mathbf{p}_3, \mathbf{p}_2)|) | < S_2 \quad (4.36)$$

ただし

$$F(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = \frac{\mathbf{p}_i - \mathbf{p}_k}{|\mathbf{p}_i - \mathbf{p}_k|} \cdot \frac{\mathbf{p}_j - \mathbf{p}_k}{|\mathbf{p}_j - \mathbf{p}_k|} \quad (4.37)$$

式 (4.35) は図 4.23 に示すように e を対角稜線として向かい合う 2 つの角について，それぞれの角度を直角に近づけるためのものである． S_1 は実数であり 0 に近づくほど精度が高くなる．式 (4.36) は e の両端において，それぞれの端点に接続する角度の和を求め，それらの差を最小にするためのものである． S_2 は実数であり 0 に近づくほど精度が高くなる．

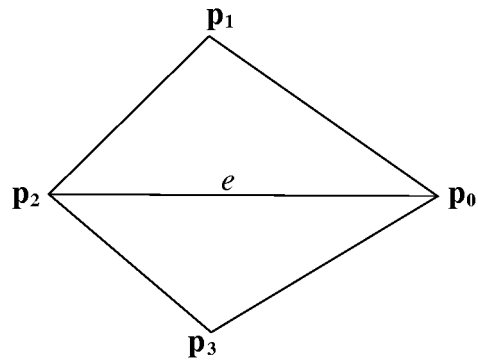


図 4.23: 3 角形同士のペアリング

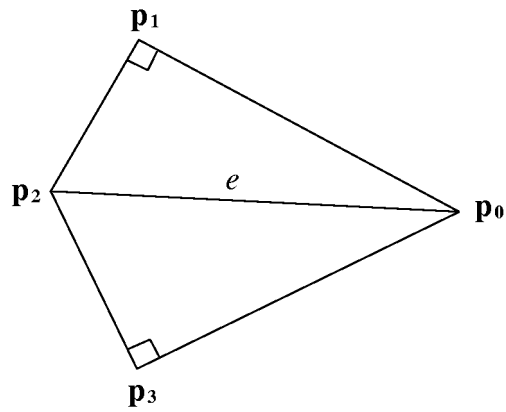


図 4.24: 1 つの評価関数だけでは検知できない 3 角形のペア

式 (4.35) を満たすだけでは, 図 4.24 に示すように, e に対して向き合う角度はほぼ直角であるが鋭角部が残る可能性が残る. この状態を回避するために式 (4.36) が必要である. 我々の実験結果によると, S_1 及び S_2 のそれぞれが 1.0 以下であれば, 滑らかな双 3 次 Gregory パッチの内挿が可能であることが分かっている.

また、4 角形化は計算負荷とデータサイズを小さくする上でも有効である。図 4.25 に示すように、双 3 次 Gregory パッチにより 3 角形面を内挿する場合は、3 つのパッチによって分割（リパッチ）する必要がある。そのため、4 角形面の場合と比べて約 3 倍の計算負荷が必要になり、データサイズも大きくなる。4 角形化によりラティス構造に基づく 3 次元モデルの表示を大幅に効率化することが可能である。

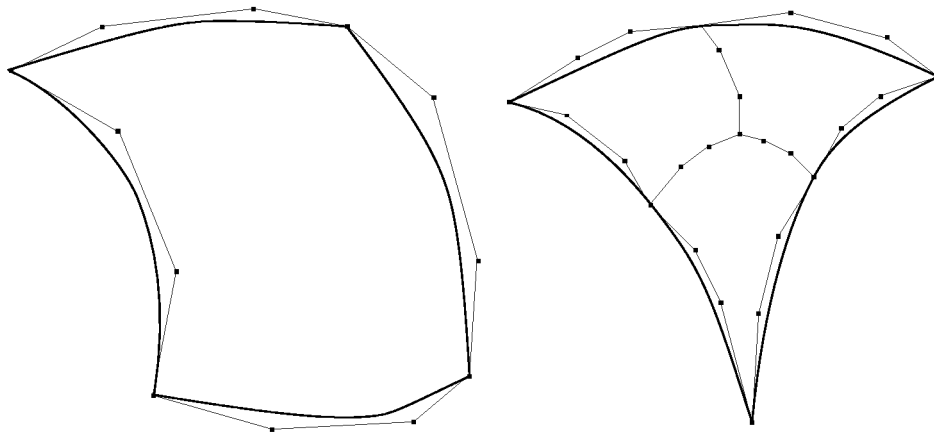


図 4.25: 4 角形面と 3 角形面への双 3 次 Gregory パッチの内挿

特徴線の反映

特徴線に対応する稜線についても、同様に式 (4.34) を適用することで、簡略化メッシュにおいて特徴線を削除せずに残すことが可能になる。しかし、簡略化において非常に多くの稜線消去を実行する場合は、特徴線として指定した稜線も削減される可能性がある。そのため、削減する要素数が多い場合は、特徴線として指定した稜線群を稜線削減のリストから除いた上で簡略化を行う。

本研究は、計算機の専門知識をもたない医師が容易に 3 次元モデルを構築することを目的としているため、特徴線や精度のインプットは直感的な GUI によって実現される必要がある。そのため、本研究では、図 4.26 のようなディスプレイタブレットを用いたシステムを開発し、3 次元モデルにペイントする感覚で要素の入力を行うことを実現した。



図 4.26: ディスプレイタブレットによる入力インターフェイス

第 5 章

Web3D アプリケーション

本章では、医用ラティス構造に基づく3次元モデルを利用したWeb3Dアプリケーションについて述べる。

5.1節では、Web3Dアプリケーションのソフトウェア構成について述べる。本研究で提案している手法によって再構築された3次元モデルに対して、ECMA Scriptによってアニメーションとイベントの付加を行う手法について解説する。

5.2節では、幼児期に診られる症例である頭蓋縫合早期癒合症を取り上げ、術前と術後の患部形状を比較して閲覧できるアプリケーションについて述べる。精度の適応的制御を効果的に利用したモデル作成例についても紹介する。

5.3節では、下顎骨の切断シミュレーションについて述べる。ディスプレイタブレットを利用した直感的インターフェイスを用いた効果的なシミュレーションシステムについて紹介する。実際の医師によって実現された切断シミュレーションの切断例についても紹介する。

5.4節では、先天的に上口唇が裂けている症例である口唇裂について取り上げ、術式を学習するための3Dテキストブックについて紹介する。口唇周辺の構造を理解するための解剖学テキスト機能についても触れる。

5.5節では、CTスキャナやMRIによって取得された2次元医用画像を、3次元モデルの切断面として表示するシステムについて紹介する。また、インタラクティブな操作によるモデルへのマーキングや、軌跡の表示機能によって効果的な診療、診察を実現するための機能についても述べる。

5.6節では、下顎骨の動きをデバイスによって取得し、3次元モデルに反映するアニメーションシステムについて紹介する。皮膚と頭蓋の2つのモデルの属性値を変更することで、2次元画像上の生体組織を容易に把握させる機能についても紹介する。

5.1 ソフトウェア構成

本章で紹介する Web3D アプリケーションは、ラティス・テクノロジー社の XVL Player [58] 上に実装されている。XVL Player は Microsoft 社の ActiveX コンポーネントの 1 つで、同社の WWW ブラウザである Microsoft Internet Explorer のプラグインとして利用することが可能である。XVL Player を用いることにより、WWW ブラウザ上で XVL データを閲覧することが可能になる。XVL(eXtensible Virtual-world description Language) とは、ラティス・テクノロジー社の提案する Web3D フォーマットであり、XML によって記述されている。XVL データはラティス構造を扱うことが可能であるために非常に軽量であり、Javascript や JScript などの ECMA Script によって容易にプログラミングすることが可能である。

XVL Player では、XVL Object Model と ECMA Script によって、3 次元シーンの各構成ノードにアクセスすることができる。XVL Object Model は 3 次元シーンオーサリングのためのプログラミングインターフェイスである。VRML における EAI(External Authoring Interface) [49] のような外部インターフェイスの役割と、一般的な SAI(Scene Authoring Interface) のような内部インターフェイスの役割の双方を利用することができる。実際のプログラミングにおいては、XVL Object Model と DHTML(Dynamic HTML) [38] を利用する。DHTML は、HTML、CSS(Cascading Style Sheets)、DOM の集合のことであり、ECMA Script によってプログラミングを行う。DHTML と ECMA Script を利用する形態は、Web コンテンツのプログラミングにおいては一般的な形態であるために汎用性が高い。

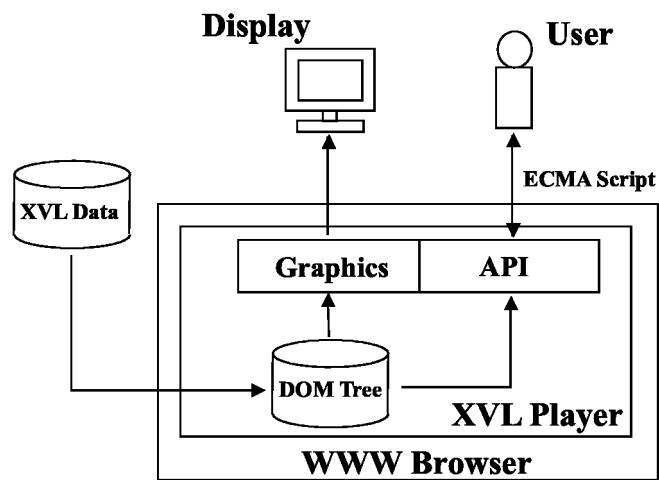


図 5.1: Web3D アプリケーションのソフトウェア構成

図 5.1 に、ソフトウェアの構成図を示す。本手法によって作成された XVL データは XVL Player に読み込まれ、DOM Tree として再構築される。プログラマ (User) は、XVL Player の API(Application Programming Interface) を介して DOM Tree の各ノードにアクセスし、ECMA スクリプトによってプログラミングを行う。その結果は Graphics モジュールを介して Display に出力される。

XVL Player における HTML 例とプログラミング例を付録 C に記す。

5.2 頭蓋縫合早期癒合症

頭蓋縫合早期癒合症 (Craniosynostosis) は、頭蓋骨のつなぎ目が成人になる前に癒合してしまうことで、頭蓋骨形状の異常を示す病気である。形態異常には、塔状頭、三角頭、舟状頭、斜頭といったものがあり、頭蓋内圧の改善と形態の改善を目的とした手術が行われる。手術では、頭蓋骨を切断し、意図した形態になるように拡大することで頭蓋の変型を改善し、脳の成長に影響が及ぶのを避ける。

本アプリケーションでは、頭蓋骨が縦長に変形した周状頭の症例を元に、術前と術後の頭蓋の成長過程を可視化する。図 5.2 において、左が術前の周状頭、右が術後の頭蓋を示している。術前のモデルにおいては、切断線となる部分は高い精度が必要であるために、細かいパッチを割り当てることで精度のコントロールを行っている。術後のモデルでは、切断領域の成長過程を観察することが非常に重要であるために、切断領域は同様に高い精度が割り当てられてる。

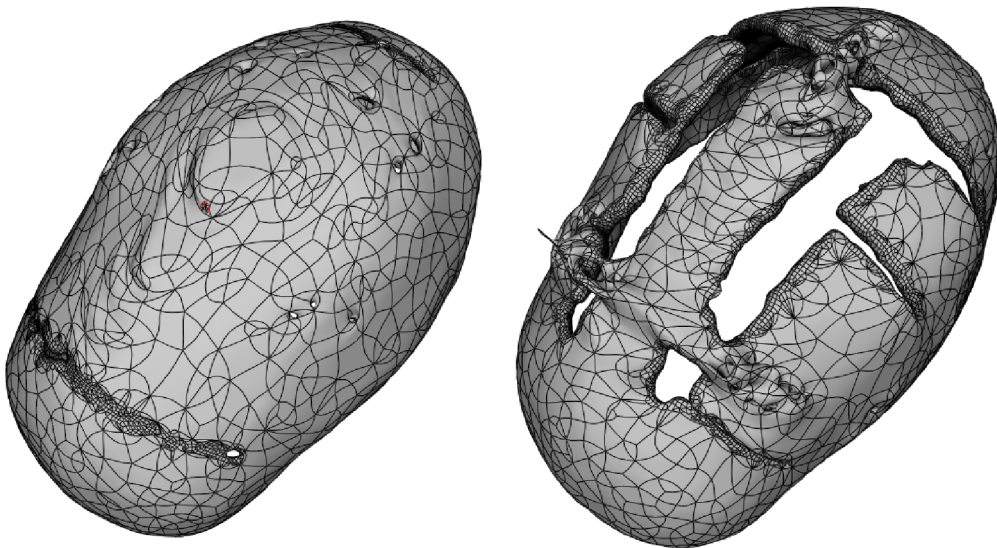


図 5.2: 頭蓋縫合早期癒合症の手術前後ビジュアライゼーション

5.3 下顎骨切断シミュレーション

下顎骨切断手術は，下顎が左右非対称であるような先天性の変形を対象疾患とする手術である．その際に，下顎骨切り術と下顎延長手術という2つの手術を施すことで，正常な下顎の再建を行う．下顎骨きり術は，矢状骨切り (Sagittal Osteotomy) (図 5.3 左) と垂直骨切り (Vertical Osteotomy) (図 5.3 右) という2種類の切断方法がある．垂直骨切りは垂直に近い曲面で下顎を切断する手法であり，比較的理解し易い．しかし，矢状骨切りは，非常に複雑な曲面によって下顎を切断する手法であり，シミュレーションも困難である．

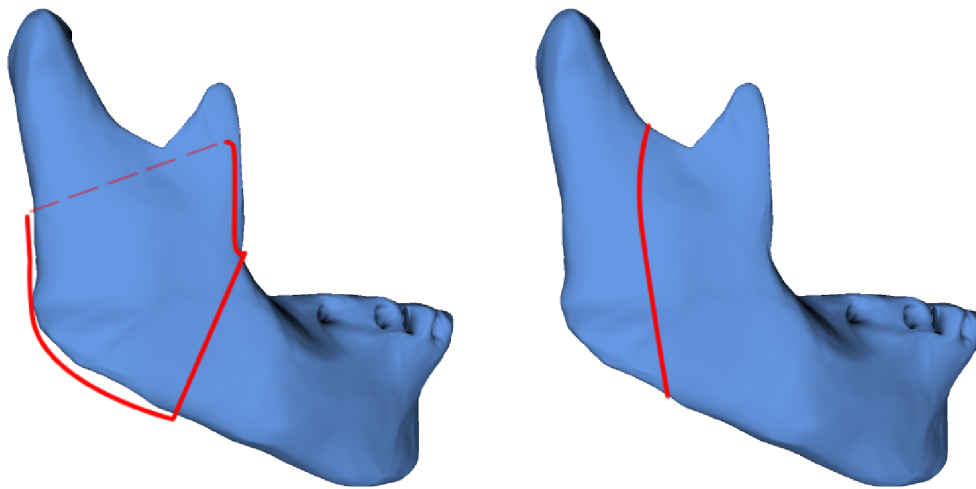


図 5.3: 矢状骨きりと垂直骨きり

本研究では，従来困難であった矢状骨きりの手術シミュレーションを，ディスプレイタブレットを利用したシステムを開発することで実現した．ディスプレイタブレットは，計算機のディスプレイに対して直接ペン入力が可能であるため，直感的なポインティングデバイスである．シミュレーションにあたっては，慶應義塾大学看護医療学部的小林正弘助教授にご協力を頂いた．図 5.4 に示すように，ユーザ（医師）はディスプレイタブレットを利用して，下顎の3次元モデル（3角形メッシュ）に対して，直接切断線を描画することができる．切断線は3Dモデルの表面にペイントされ，ユーザの切断線入力の試行錯誤が終了後，新たな稜線列としてモデルに作成される．この際作成される稜線列は，必ず3角形メッシュの3角形面上，もしくは既存の稜線上に作成されるため，形状を変化させることはない．作成された稜線列を特徴線とみなして高いQEMを付加し，医用ラティス曲面へと変換する．作成された医用ラティス曲面は，パッチ間の輪郭線として切断線を保持している．切断線に当た

る稜線列は自動的に選択することが可能であり，容易に切断操作による骨きり術と下顎骨延長術のシミュレーションを実現することができる．図 5.5 に，切断シミュレーションの実行例を示す．

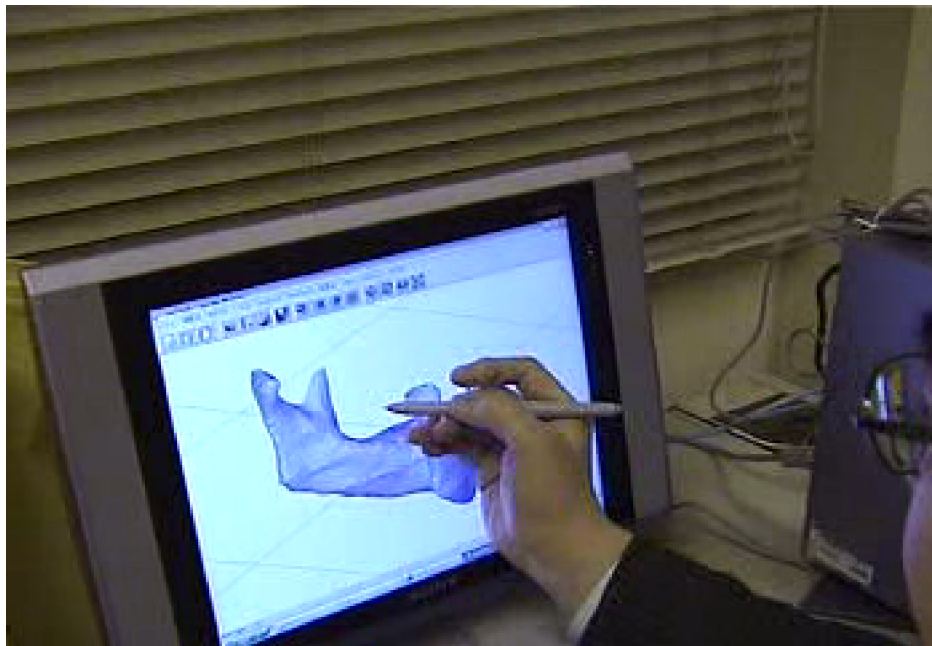


図 5.4: ディスプレイタブレットによる切断線の入力

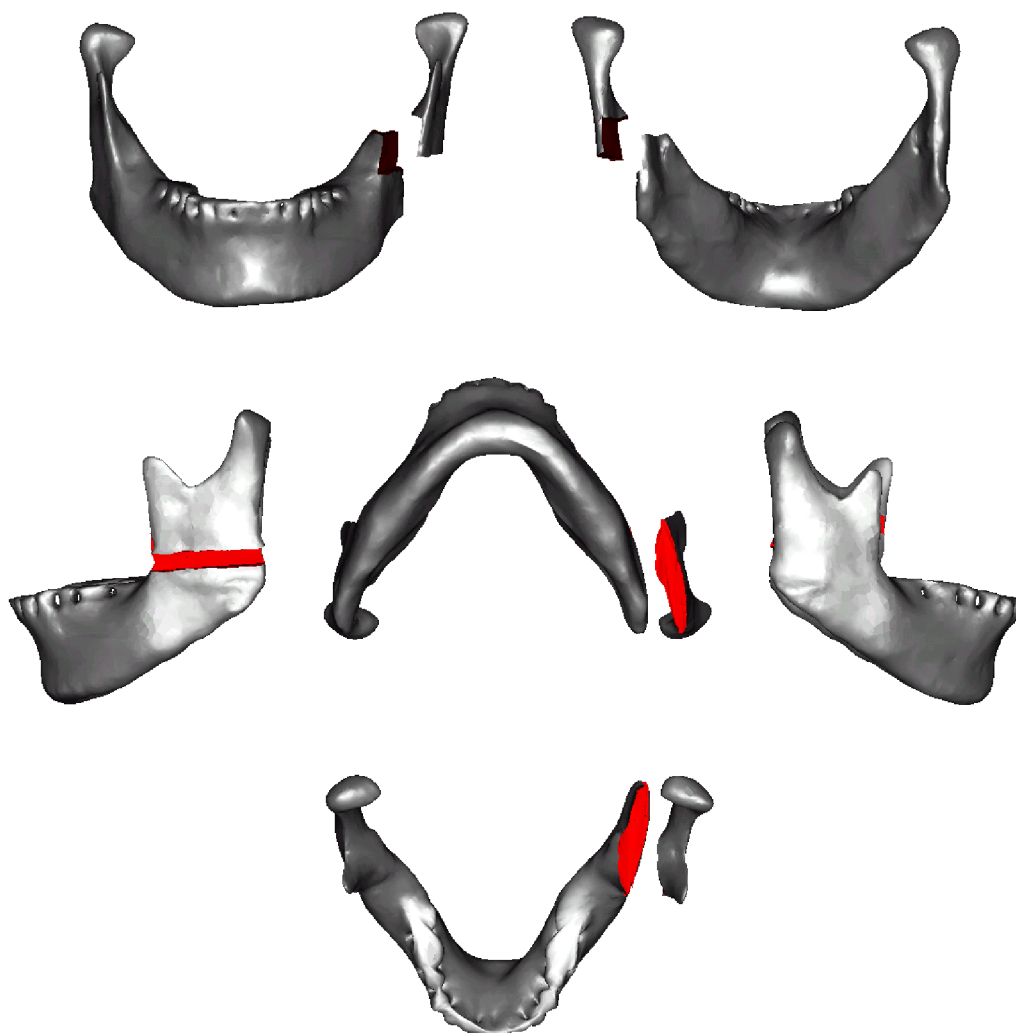


図 5.5: 切断シミュレーション後の下顎骨モデル

5.4 口唇裂3Dテキストブック

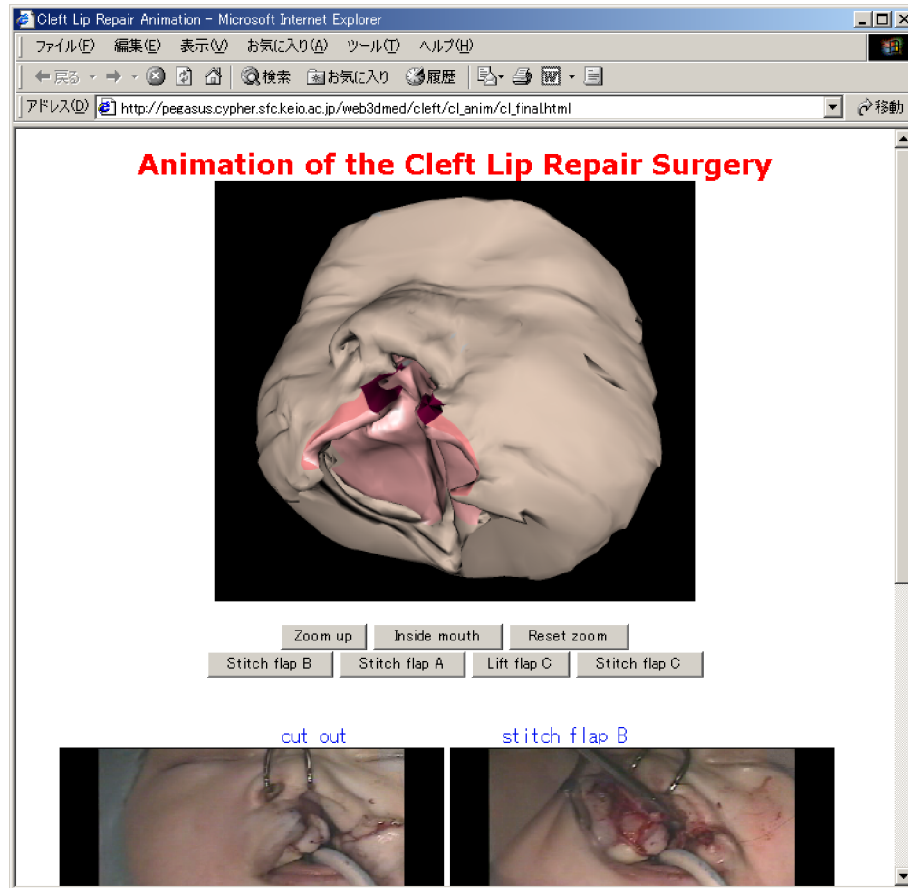


図 5.6: 口唇裂 3D テキストブックのスナップショット

口唇裂 3D テキストブックは、口唇裂の術式を 3 次元 CG のアニメーションにより表示するアプリケーションである。図 5.6 にスナップショットを示す。

口唇裂とは、先天的に上口唇の皮膚に割れ目が生じる症状を示す。割れ目の程度は様々で、口唇の皮膚に軽い筋と窪みが生じる不完全唇裂や、歯茎にまで割れ目が及ぶ完全唇裂などがある。口唇口蓋裂児が生まれる確率は 500～600 人に 1 人であるといわれており、形成外科における主要な治療対象の 1 つとなっている。口唇裂の手術は、生後すぐに行われることが多く、遅くとも生後 1 ヶ月の間に行われるのが一般的である。

口唇裂手術の成功と術後の万全な治療のためには、形成外科だけではなく、歯科口腔外科、小児科、耳鼻咽喉科、麻酔科など口唇口蓋に関連する様々な専門医との密な連携が必須

である．そのため，3次元モデルによって手術計画を行い，更にネットワークで配信することが可能になれば，これらの専門医間のコラボレーションを活かした高度な医療の提供が期待できる．また，口唇裂手術は小さい患部に対して複雑な縫い合わせが必要であるため，長年の修練による熟練した技術が要求される．一方で，研修医や医学生が実際の術式を見学して学ぶ機会は限られているため，Web3D技術を用いた効果的な教材も求められている．

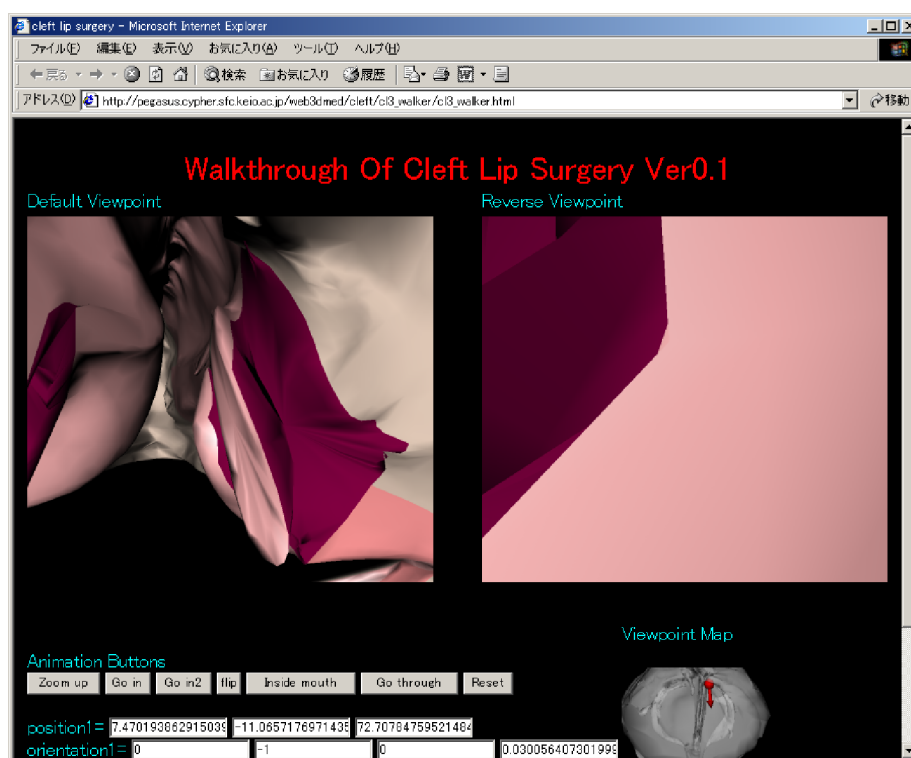


図 5.7: 口唇裂 3D テキストブックのスナップショット（ウォークスルーモード）

口唇裂 3D テキストブックには，術式を学ぶためのアニメーションモード（図 5.6）と口唇口蓋の解剖学のためのウォークスルーモード（図 5.7）が用意されている．アニメーションモードでは，術式の各ステップの 3D アニメーションを自由な角度から閲覧することが可能である．視点のズームも可能であり，現実では不可能な視点から術式を学ぶことを可能にしている．ユーザは，図 5.8 に示すように，切断線のデザインと実際の切断方法について，模範例を閲覧することができる．図 5.8 上は術前，中は切断線のデザイン，下は実際の切断を示している．また，図 5.9 に示すように，切断後の縫合についても，各ステップごとにアニメーションを実行することが可能である．

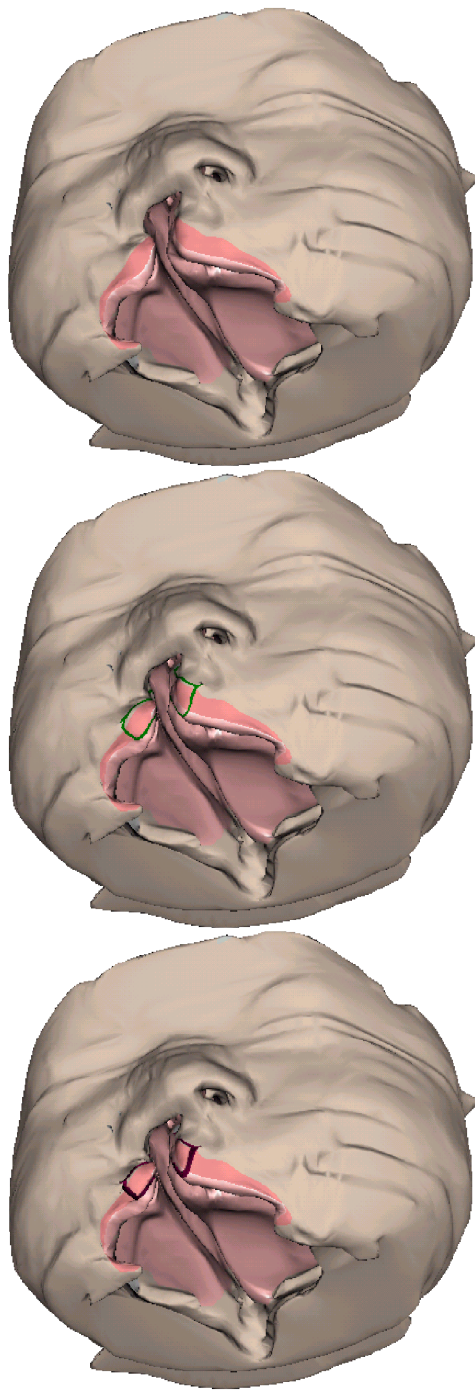


図 5.8: 口唇裂手術のデザインと切断



図 5.9: 口唇裂手術の切断ステップ

5.5 CT/MRI ビューア

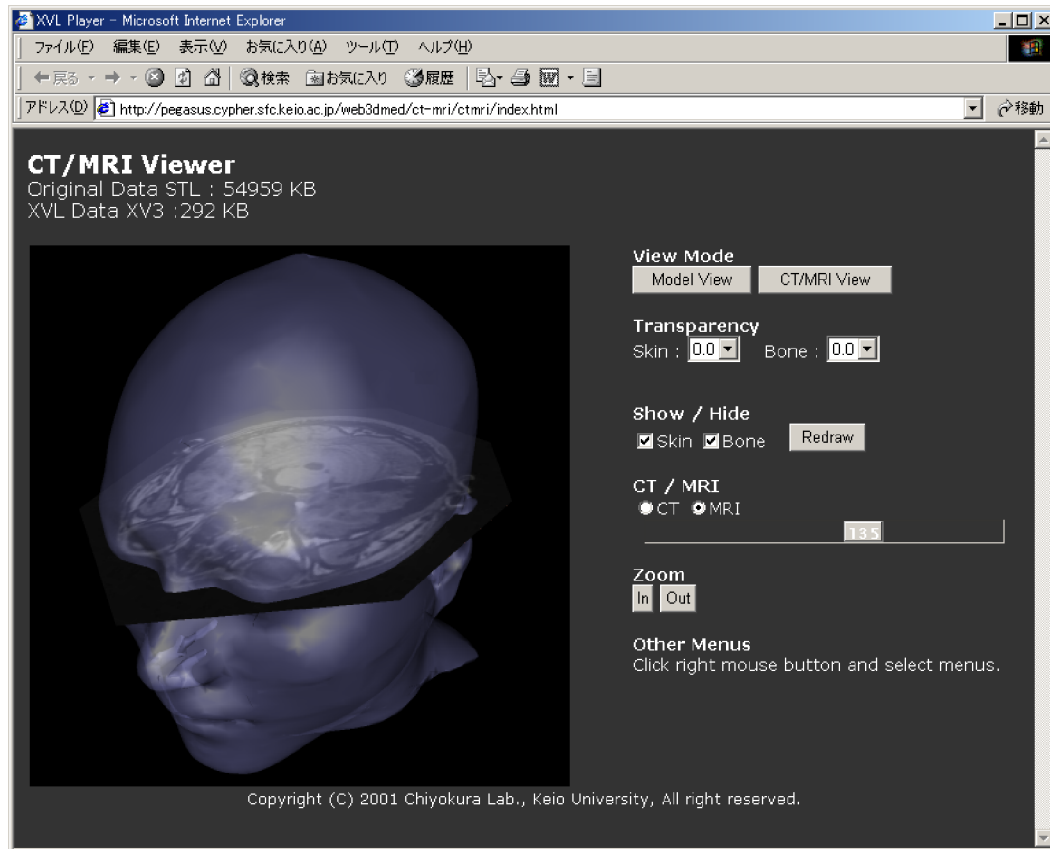


図 5.10: CT/MRI ビューアのスナップショット

CT/MRI ビューアは、3次元モデルの切断面としてCT画像とMRI画像を表示するアプリケーションである。研修医や医者とコラボレーションをするエンジニアなど、医用画像の閲覧方法に慣れていないユーザーに対する教育を目的としている。図 5.10 に本アプリケーションのスナップショットを示す。

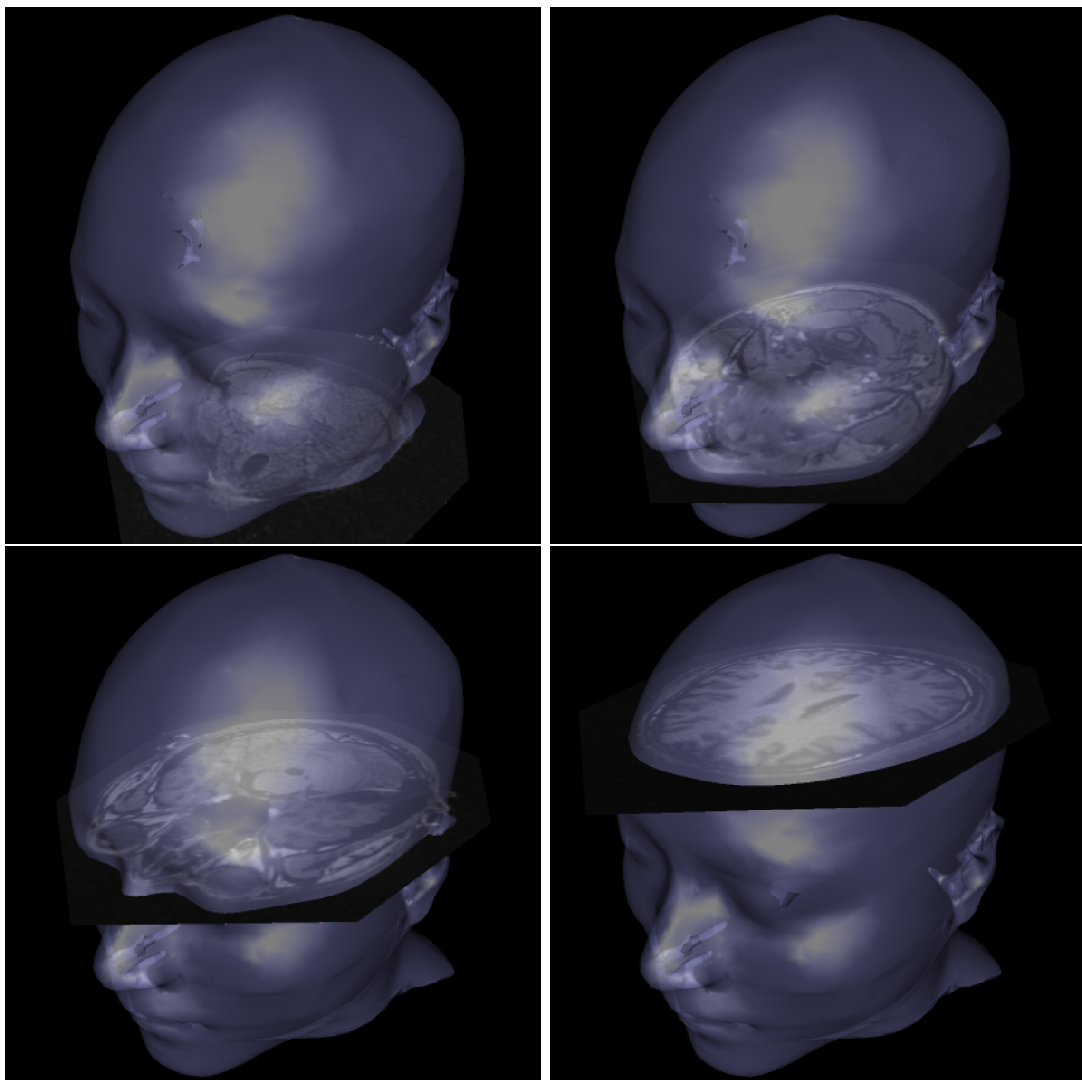


図 5.11: 切断面の移動

ユーザは，図 5.11 に示すように，3 次元モデルの任意の場所に切断面を移動させ，その部分の医用画像を閲覧することができる．切断面では，図 5.12 に示すように，CT 画像と MRI 画像の表示を切り替えることが可能である．また，皮膚と頭蓋の両方の 3 次元モデルを作成しているため，透明度を変更することで，図 5.13 のように様々な組み合わせの閲覧が可能である．

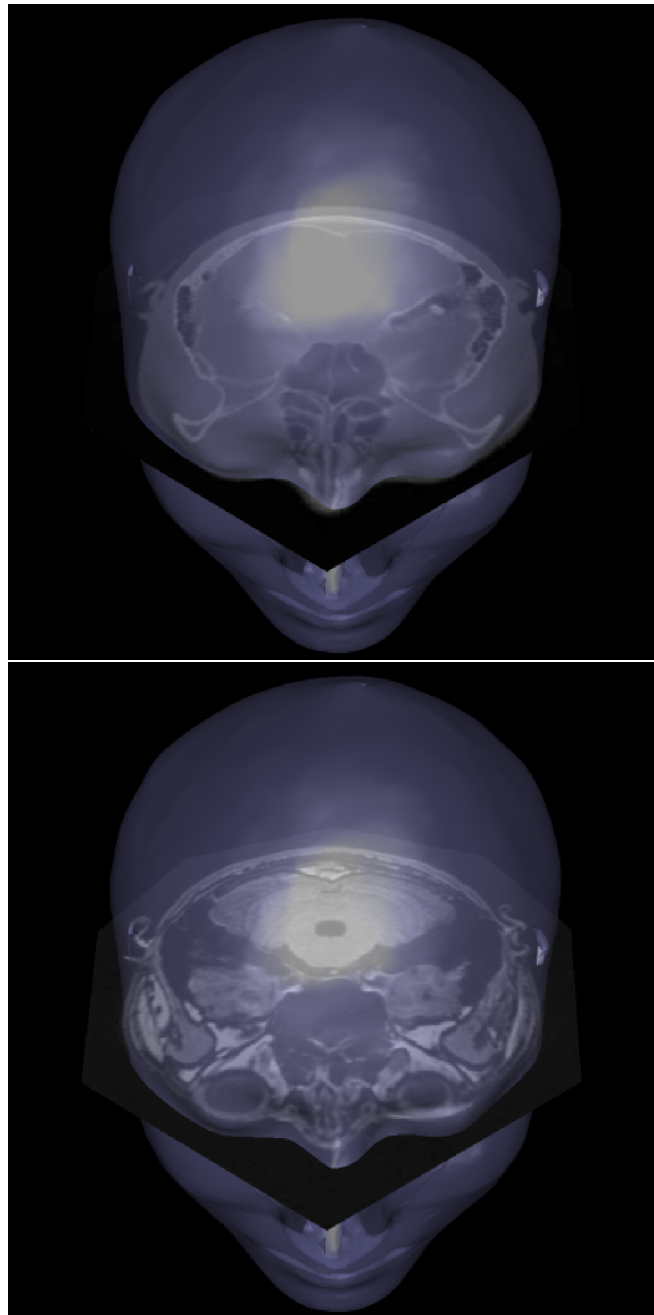


図 5.12: CT 画像と MRI の切り替え

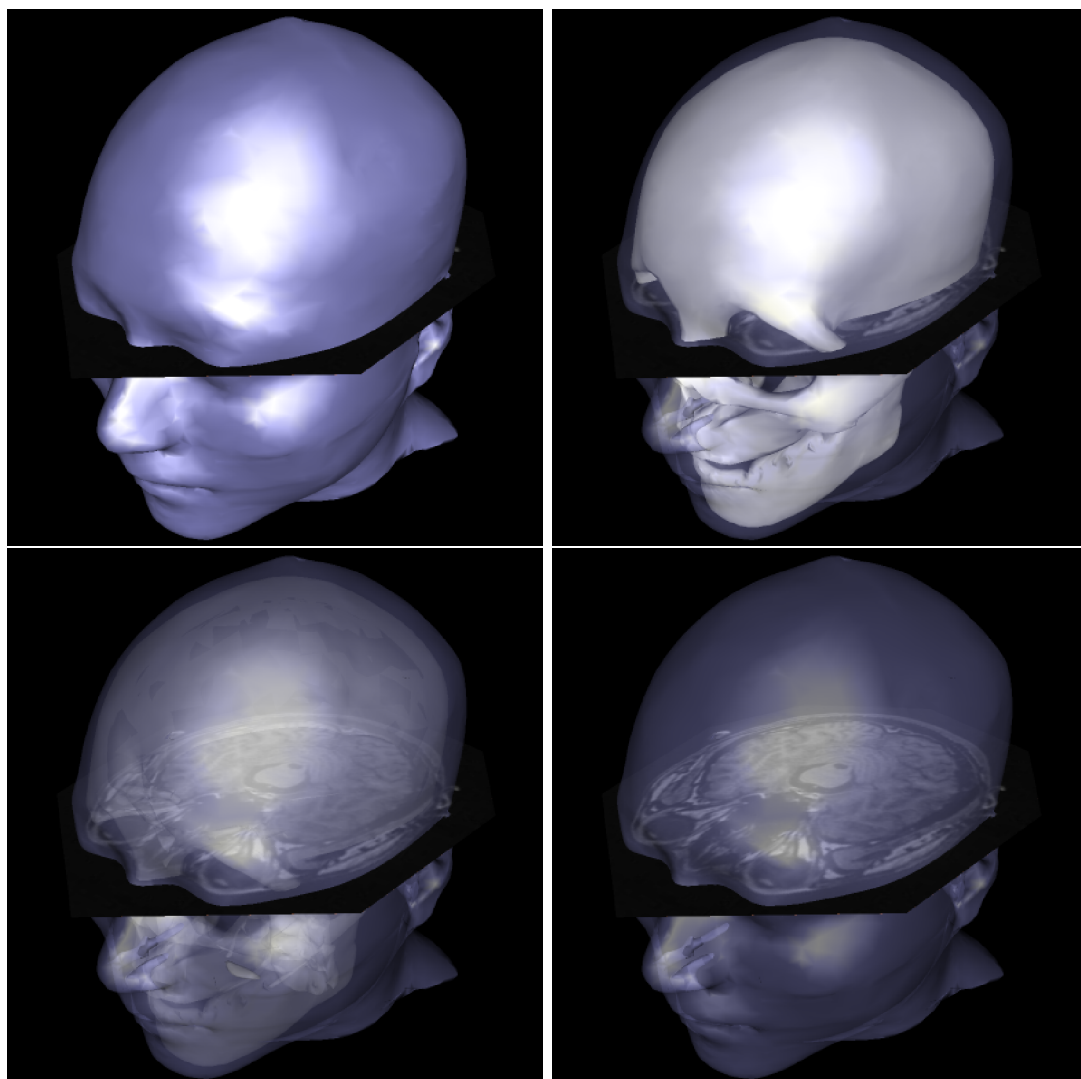


図 5.13: 透明度の変更

5.6 下顎骨運動ビューア

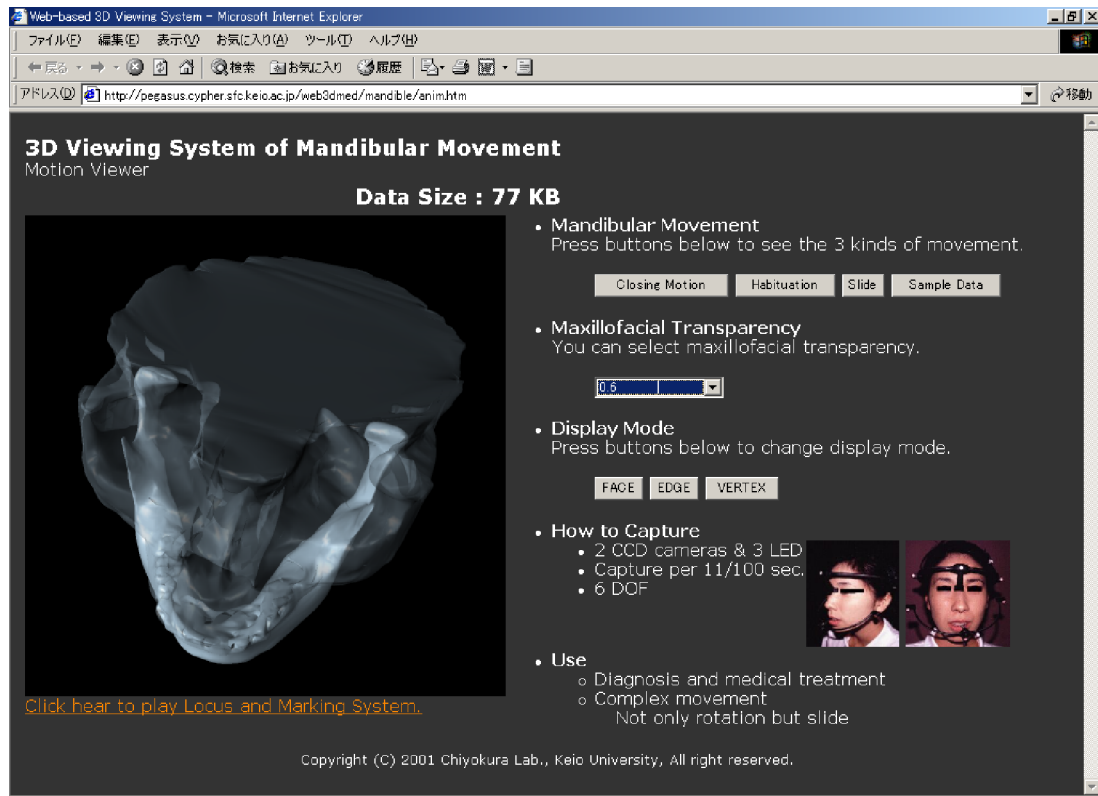


図 5.14: 下顎骨運動ビューアのスナップショット

下顎骨運動ビューアは、下顎骨の運動を3次元可視化するシステムであり、臨床における診断、診察において効果的に利用されることを目的としている。図 5.14 にスナップショットを示す。

人体の顎の動きは、回転運動のみによって成り立つものではなく、回転と平行移動の組み合わせによる複雑な動作をしている。術前に顎骨の動きを把握することは非常に重要であるが、軟部組織を切断しない限り、的確な動きを把握することは困難である。下顎骨運動ビューアは、ナソヘキサグラフ（ONO SOKKI 社製）を用いて顎骨運動を測定し、患者の CT データより作成した3次元頭蓋モデルにアニメーションを適用した。測定は、図 5.15 に示すようなヘッドフレームとフェイスボウを装着して行う。ヘッドフレームとフェイスボウには、それぞれ LED が3箇所に装備されており、被験者の前方にある2台の CCD カメラにより、約 11/100 秒毎に測定される。

ユーザは顎骨運動を WWW ブラウザ上で自由な角度から閲覧することができる。また，図 5.16 に示すように，3 次元モデルにインタラクティブにマーキングを行うことや，図 5.17 に示すように，運動の軌跡を 3 次元空間で可視化する機能も保持している。これらの機能は，術前会議などによる医師同士の手術計画の補佐や，患者に対してのインフォームドコンセントにおいて，有効に利用することができる。

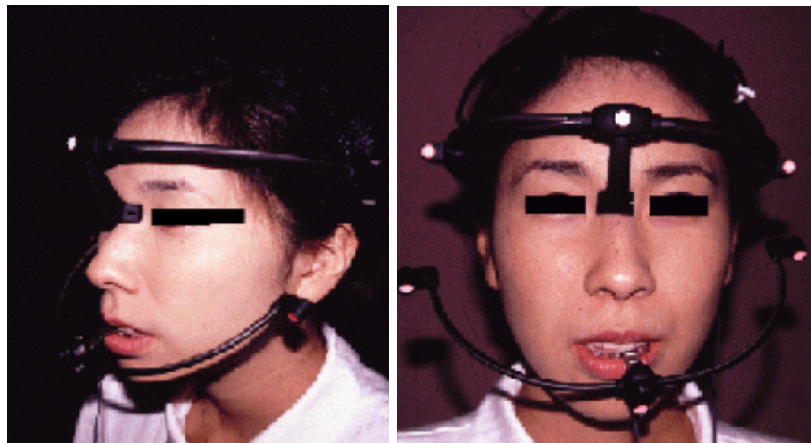


図 5.15: ヘッドフレームとフェイスボウの装着

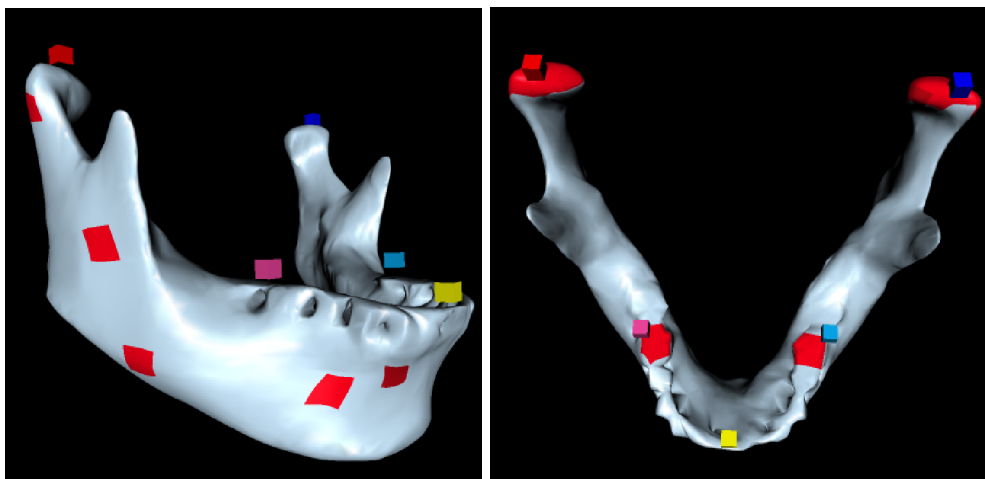


図 5.16: 3 次元モデルへのマーキング

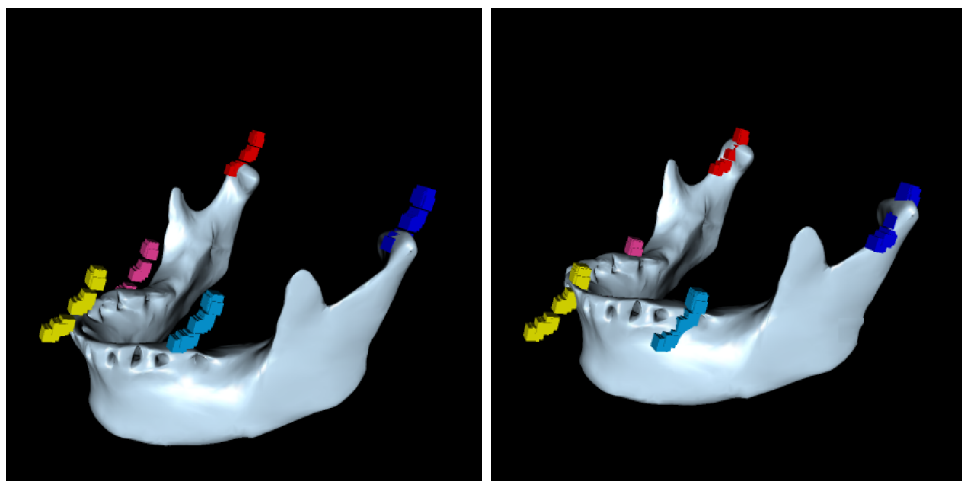


図 5.17: 下顎骨運動の軌跡の表示

第 6 章

曲面形状の構築例と考察

本章では，医用ラティス曲面モデルの構築例を示し，本手法の評価を行う．

6.1 節では，本手法によって作成した医用ラティス曲面の例について，オリジナル 3 角形メッシュ，初期多面体，医用ラティス曲面の 3 つの過程の比較を行う．曲面化においては，平均 2 乗距離誤差と計算時間の評価を行う．

6.2 節では，精度の適応的制御と特徴線入力についての実行例を示し，評価を行う．この際，平均 2 乗距離誤差だけではなく，最大 2 乗距離誤差，最小 2 乗距離誤差についても評価を行い，特徴入力操作を行わない場合との比較を行う．

6.3 節では，医用ラティス曲面の分割数を変更することにより，詳細度制御を行った実行例を示す．視点座標値に応じて詳細度を変更する場合についても評価を行う．

6.4 節では，初期多面体を制御媒体とした自由曲面の変形例を示す．

6.1 曲面形状の構築

本節では，4.3 節で述べた手法によって，以下の 6 つの例について医用ラティス曲面化を行った．

1. 下顎骨 (jaw)
2. 頭蓋骨 (skull)
3. 頭部表皮 (skin)
4. 舟状頭 1(scapo-pre)
5. 舟状頭 2(scapo-aft)
6. 舟状頭 3(scapo-face)

1 つ目のデータである下顎骨 (jaw) は，5.6 節及び 5.3 節にて利用されているものある．2 つ目のデータである頭蓋骨 (skull) と 3 つ目のデータである頭部表皮 (skin) は，5.5 節にて利用されている人体頭部の皮膚のモデルである．4 つ目のデータである舟状頭 1(scapo-pre)，及び 5 つ目と 6 つ目のデータである舟状頭 2(scapo-aft)，舟状頭 3(scapo-face) は，5.2 節にて用いられている症例データである．舟状頭 1(scapo-pre) は頭蓋骨切り術の術前データであり，舟状頭 2(scapo-aft) は術後のデータである．舟状頭 3(scapo-face) は頭蓋骨の頭部以外のデータである．

表 3: 医用ラティス曲面生成の実行結果

Data	V_{org}	F_{org}	Size _{org} (KB)	$V_{aft.}$	$F_{aft.}$	Size _{aft.} (KB)	Time(sec.)	Error(%)
jaw	8980	17098	8590	1000	2036	93.1	40	0.175
	8980	17098	8590	500	1024	47.1	45	0.264
skull	20684	41399	14533	2000	2031	191.0	114	0.135
	20684	41399	14533	1000	1031	96.9	128	0.234
skin	13270	26547	9056	1000	2007	95.6	70	0.123
	13270	26547	9056	500	1007	49.7	75	0.275

近似精度の制御

曲面形状の精度について, jaw, skull, skin の3つのデータを用いて検証した. 表3は, これらの3つのデータについての曲面生成時間と平均2乗距離誤差を示すものである. 表3において, Dataは測定データ, V_{org} はオリジナルメッシュ M^0 の頂点数, F_{org} は M^0 の面数を示す. また, Size_{org}(KB)はオリジナルメッシュ M^0 をVRML形式によって表現した際のデータサイズを示す. 更に, $V_{aft.}$ は簡略化メッシュ(初期多面体) M^n における頂点数, $F_{aft.}$ は M^n における面数を示す. また, Size_{aft.}(KB)は初期多面体 M^n をVRML形式によって表現した際のデータサイズを示す. Time(sec.)はメッシュ簡略化が終了しレンダリングされるまでの実行時間を示す. Error(%)はオリジナルメッシュ M^0 と, 簡略化メッシュ M^n に丸め操作を適用したラティス曲面との平均2乗距離誤差を示す. 平均2乗距離誤差(Mean Square Error)は, Cignoniら[9]が開発した誤差評価手法ツールであるIRI-CNR Metro Toolを利用した.

図6.1はjawの曲面フィッティング結果を示している. オリジナルメッシュ(上)の頂点数は8980, 面数は17098である. このデータの曲面フィッティングに際しては, 頂点数を1000まで削減したものと, 500まで削減したものを用意し, それぞれについての医用ラティス曲面(右)と初期多面体(左)を示してある.

図6.2はskullの曲面フィッティング結果を示している. オリジナルメッシュ(上)の頂点数は20684, 面数は41399である. このデータの曲面フィッティングに際しては, 頂点数を2000まで削減のものと, 1000まで削減したものを用意し, それぞれについての医用ラティス曲面(右)と初期多面体(左)を示してある.

図6.3はskinの曲面フィッティング結果を示している. オリジナルメッシュ(上)の頂点数は13270, 面数は26547である. このデータの曲面フィッティングに際しては, 頂点数を1000まで削減したものと, 500まで削減したものを用意し, それぞれについての医用ラ

ティス曲面（右）と初期多面体（左）を示してある．

図 6.1，図 6.2，図 6.3 に示すように，頂点数を少なくした場合の方が，細かい特徴を反映することができず，医用ラティス曲面の近似精度が低くなることが見て取れる．また，表 3 においても，平均 2 乗距離誤差は頂点数が少なくなるに従い，単調増加しているのが見て取れる．これが QEM の特徴であり，評価値の小さい稜線から消去を行うアルゴリズムの効果が現れている．

計算時間

表 3 により，計算時間に関する評価を行う．表 3 の実行時間 (Time(sec.)) はメッシュ簡略化が終了し，モデルがレンダリングされるまでの時間を示している．jaw について，頂点数 1000 の場合と頂点数 500 の場合とでの実行時間の差は 5 であるが，この差は全体の計算時間に比べると非常に短い．skull，skin についても同様であり，頂点数の違いによる計算時間の差は小さいことが分かる．これは，レンダリングにかかる時間が実行時間の大半を占めており，メッシュ簡略化自体は非常に高速に実行できていることを示す．つまり，モデルの規模に関係なく簡略化は高速に行えることになり，本手法の有効性を確認することができる．

データサイズ

表 3 により，データサイズに関する評価を行う．オリジナルメッシュのデータサイズ ($Size_{org}(\text{KB})$) と初期多面体のデータサイズ ($Size_{aft.}(\text{KB})$) を比較すると，100 分の 1 弱のデータ圧縮が実現されていることが分かる．また，頂点数が半数のものと比較すると，データサイズは初期多面体の頂点数に単調に依存することが分かる．このデータ圧縮率の高さは本手法の Web3D データ表現としての有効性を示している．

表 4: 特徴線の有無による曲面生成の誤差評価

Data	E_{max}	E_{mean}	E_{ms}
jaw 特徴線有り	0.2652	0.05197	0.06252
jaw 特徴線無し	0.3209	0.06084	0.07412

6.2 精度の適応的制御と特徴線の反映

図 6.4 及び図 6.5 は，頭蓋縫合早期癒合症の術前術後のモデルである．切断された断面に着目すると，その周辺領域のパッチサイズは非常に小さくなっており，精度が高く保たれていることが分かる．また，周辺領域のパッチサイズは大きく，4 辺形のものが多く見受けられる．これは精度を考慮したデータの最適化の役割のみならず，実行時の消費メモリ削減にも効果を発揮している．図 6.6 は，頭蓋縫合早期癒合症の頭部から下の形状である．このモデルについては特徴線の指定や精度の調節は行っていないが，非常に複雑な頭蓋内部の形状に対しても曲面化が実現されていることを示すためのよいサンプルである．

図 6.7 は，jaw に特徴線の指定を行ったもの (右) と指定を行っていないもの (左) との比較，及びその拡大表示 (下) である．特徴線周辺のパッチは，そうでない領域のパッチに比べて面積が小さくなっている．このことは，この領域に高い精度が割り振られていることを意味する．特徴線には，非常に高い QEM が割り当てられるので，その周辺領域も簡略化されずに残されている．切断などのシミュレーションにおいては，特徴線周辺は高い精度が必要であるため，周辺パッチの精度が高いことは非常に有効である．

図 6.8 及び表 4 は，図 6.7 における 2 つのモデルの精度を計測したものである．表 4 では，最大距離誤差 (E_{max})，平均距離誤差 (E_{mean})，平均 2 乗距離誤差 (E_{ms}) を示している．平均 2 乗距離誤差 (E_{ms}) を誤差によって色付けしたものが図 6.8 であり，青が誤差が低い領域，赤が誤差が高い領域を示している．図 6.8 において，特徴線を指定した領域は青を中心とした色づけがされており，その領域の近似精度が高いことを示す．また，表 4 においても，全ての誤差において高い近似を示していることが見て取れる．

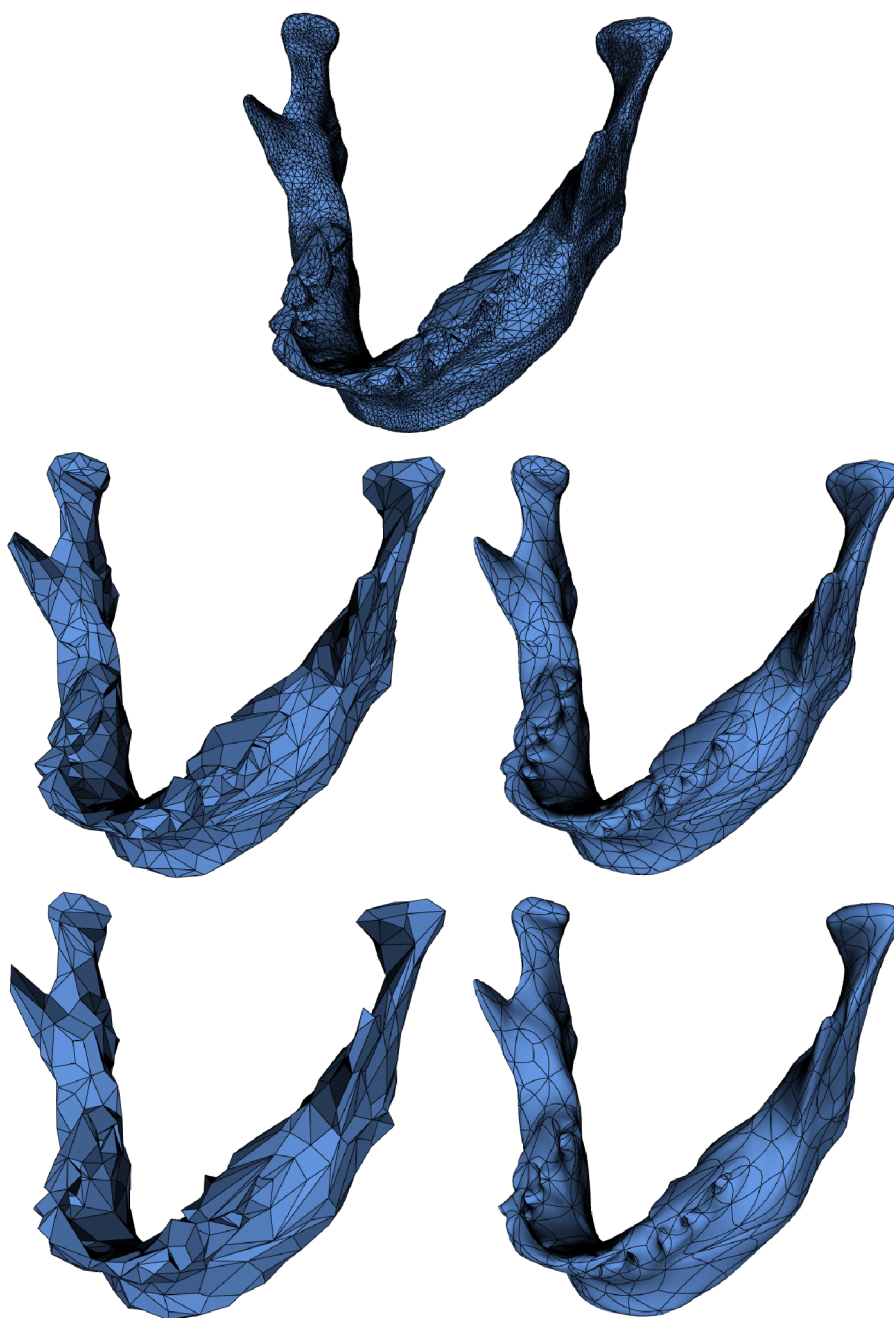


図 6.1: 下顎骨 (jaw) の実行例

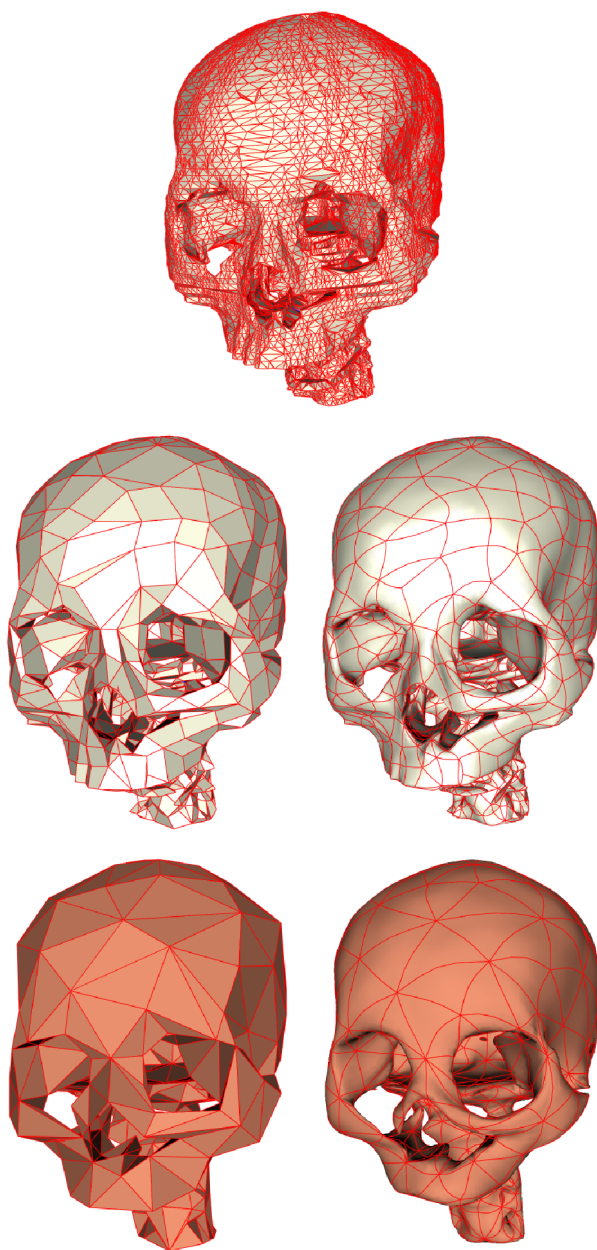


図 6.2: 頭蓋骨 (skull) の実行例

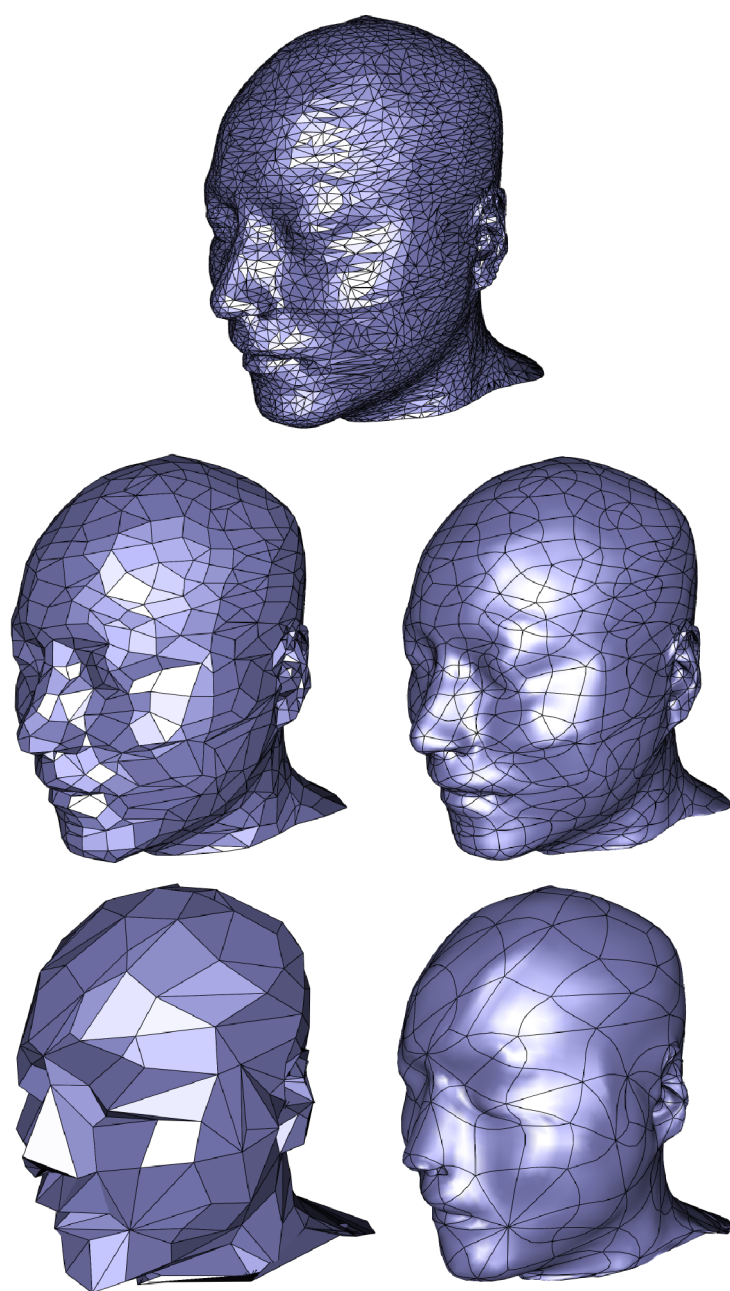


図 6.3: 頭部表皮 (skin) の実行例



図 6.4: 舟状頭 1(scapo-pre) の実行例

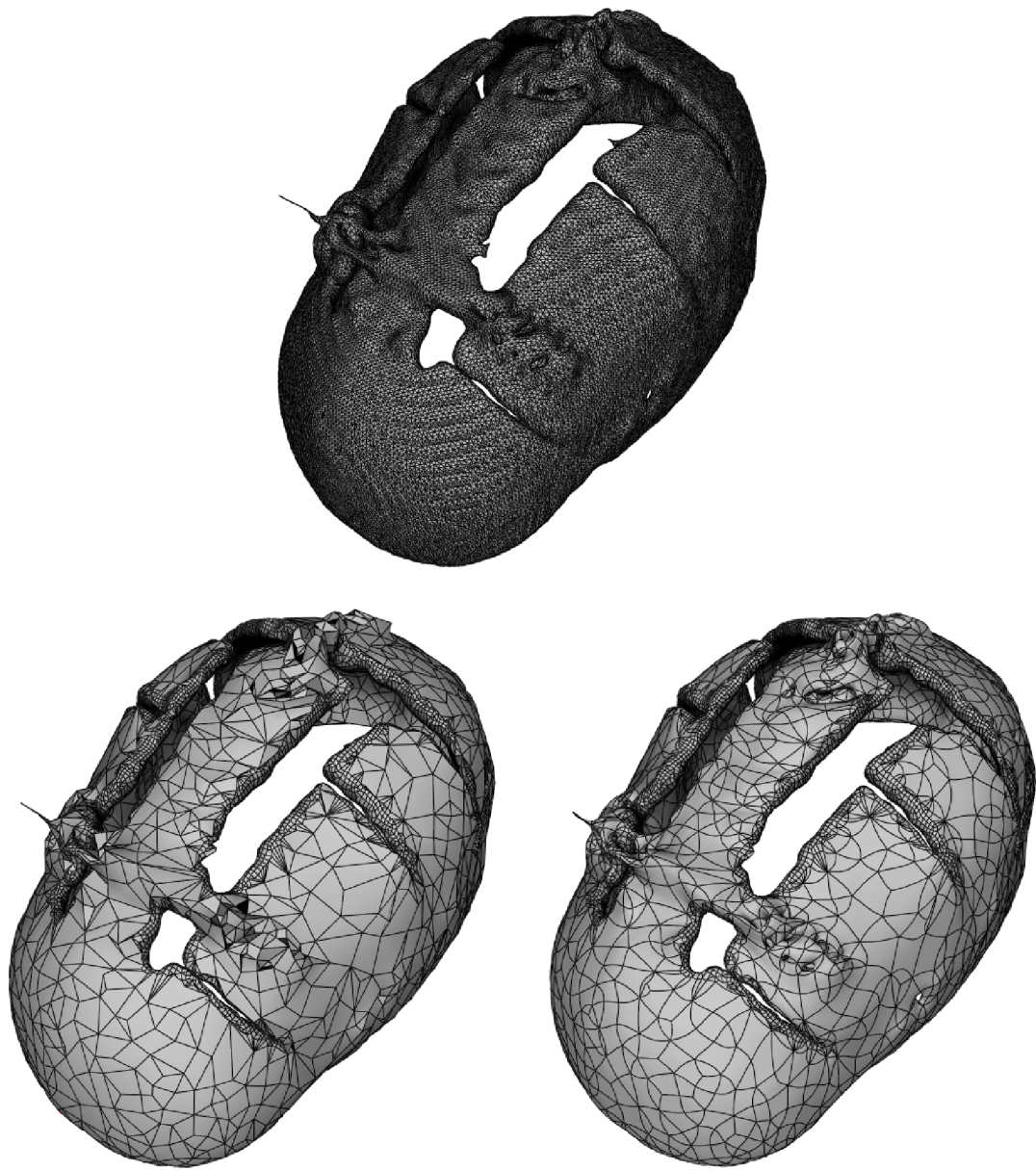


図 6.5: 舟状頭 2(scapo-aft) の実行例



図 6.6: 舟状頭 3(scapo-face) の実行例

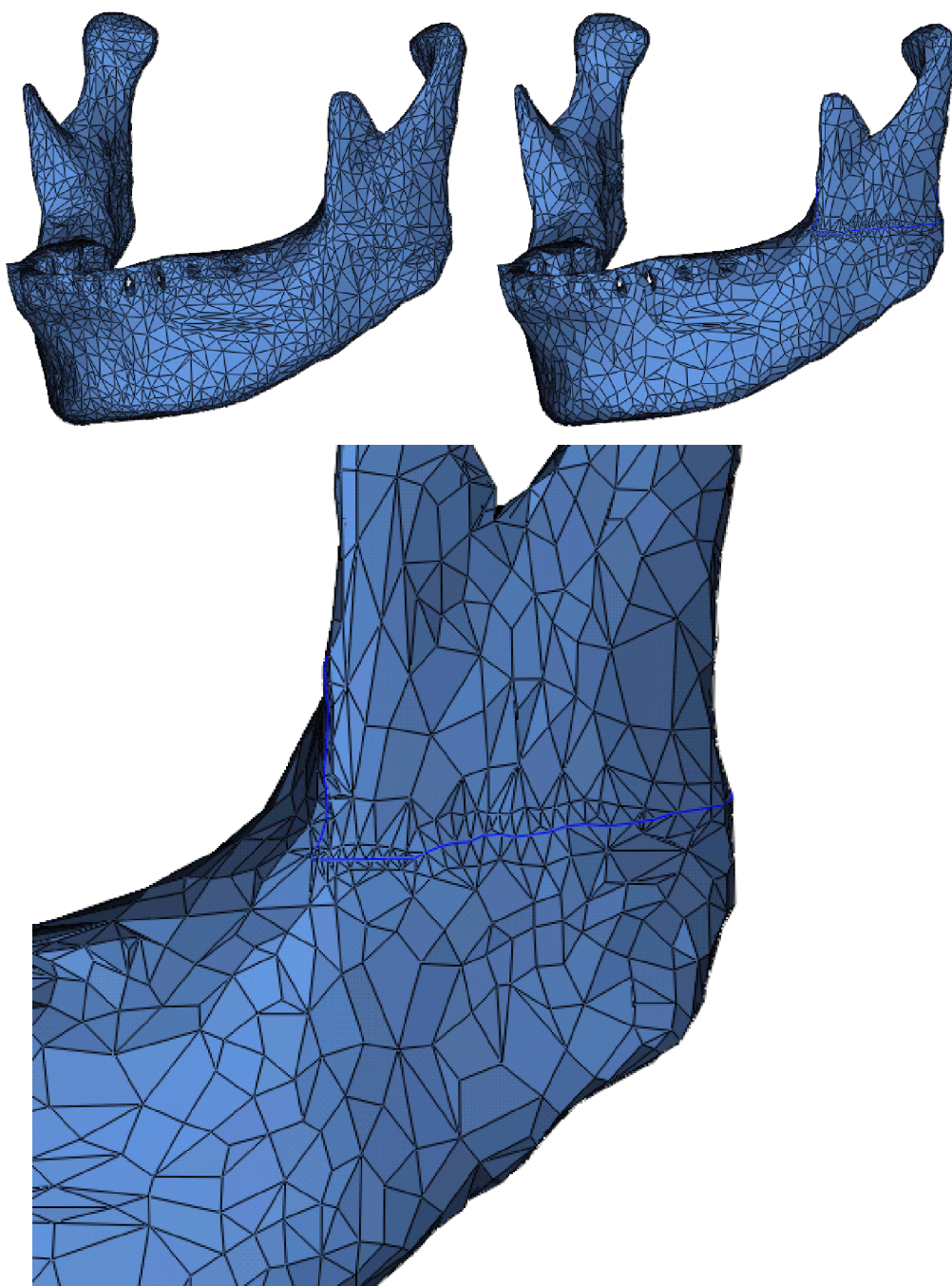


図 6.7: 特徴線の反映例

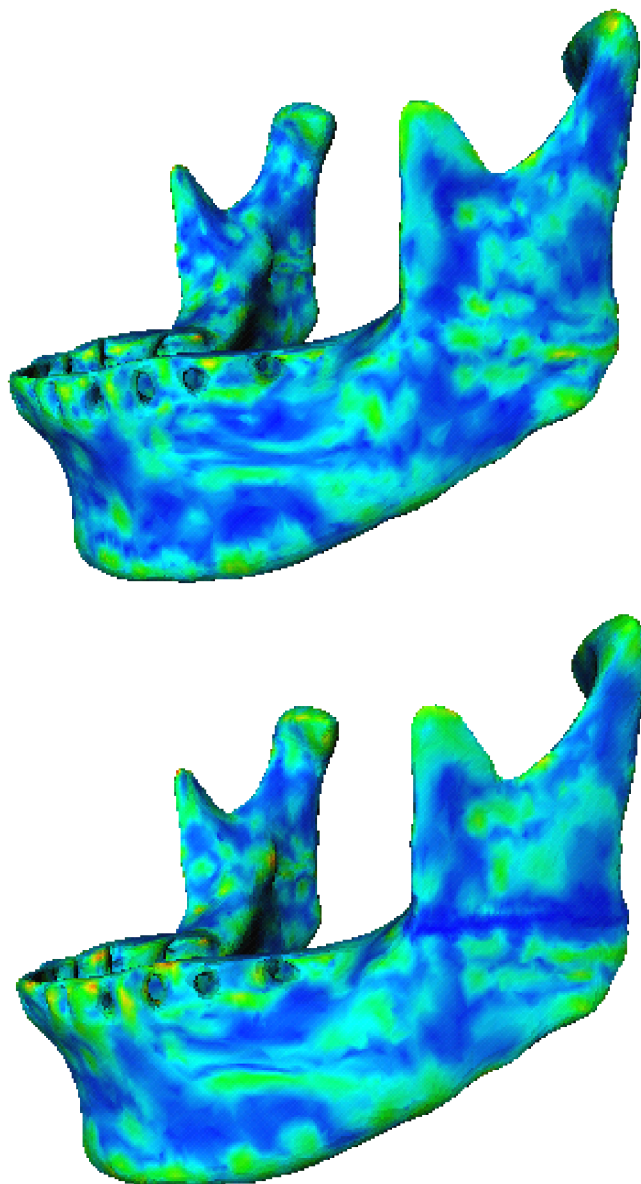


図 6.8: 特徴線指定時の誤差表示

6.3 詳細度制御

本手法の応用事例として詳細度制御の評価を行う。詳細度制御 (LOD, Level of Details) とは、目的に応じて3次元モデルの解像度を自由に調節する技術のことである。医用ラティス曲面は自由曲面モデルであるために、自由曲面の分割数を変更するだけで様々な解像度の表示を実現することができる。Web3D用途において、データ転送先の計算機の処理能力が低い場合は、解像度を下げて表示すると効果的である。また、3次元シーンにおいて視点が描画対象に近い場合は高い解像度で表示を行い、視点が遠ざかった際には低い解像度で表示を行えば、最適化された表示切替が実現される。図6.9に分割数を変更することによって、詳細度制御を行っている例を示す。図6.9において、左上は初期多面体、右上は分割数1、左下は分割数4、右下は分割数8の場合をそれぞれ示している。分割数が増えるにしたがって解像度は増加し、細部を反映した形状を表示することができる。視点の遠近に応じてモデルの解像度を切り替えた例を図6.10に示す。図6.10において、左から（小さいものから）順番に、初期多面体、分割数1、分割数2、分割数8の場合の解像度である。この例において、分割数の違いによる画像の劣化は人間の目では検出できないほど小さい。初期多面体による表示は、自由曲面の分割操作を必要としないために、高速に行うことができる。そのため、低解像度時の表示において計算負荷を最小限にすることが可能であり、非常に効果的である。更に、視点が近づくのに応じて解像度を無限大に増加することができるのは、自由曲面データを扱っていることの大きな利点であることが分かる。

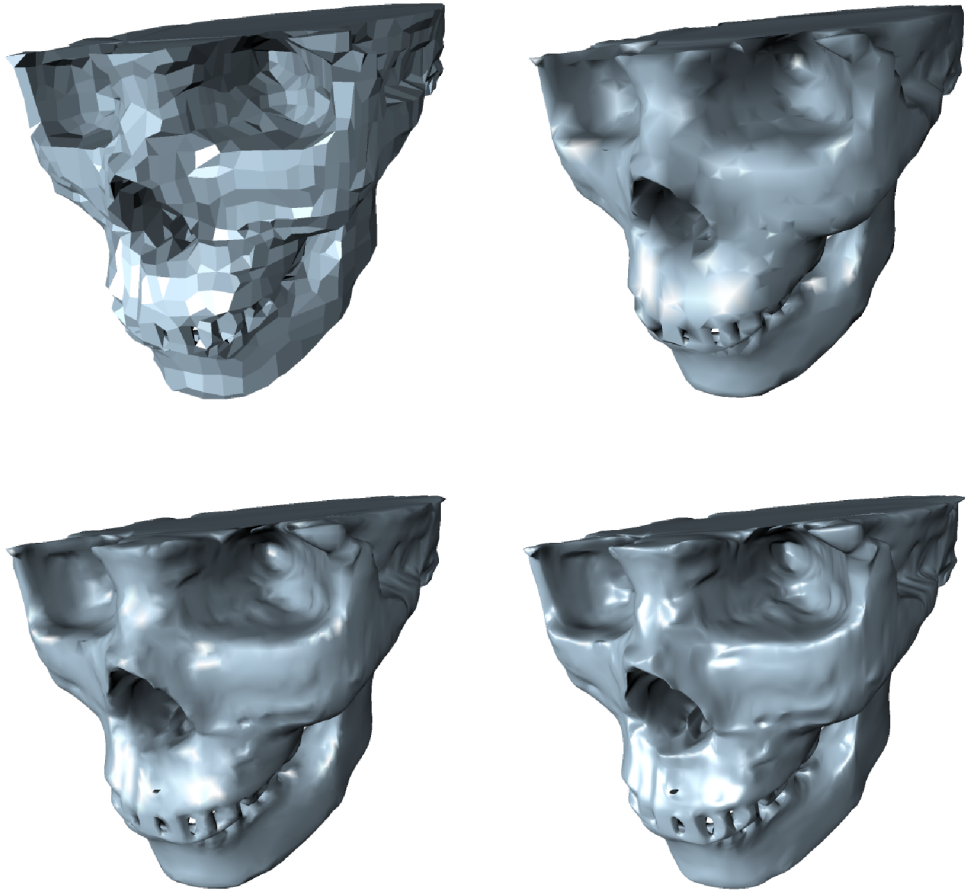


図 6.9: 分割数の変更による詳細度制御

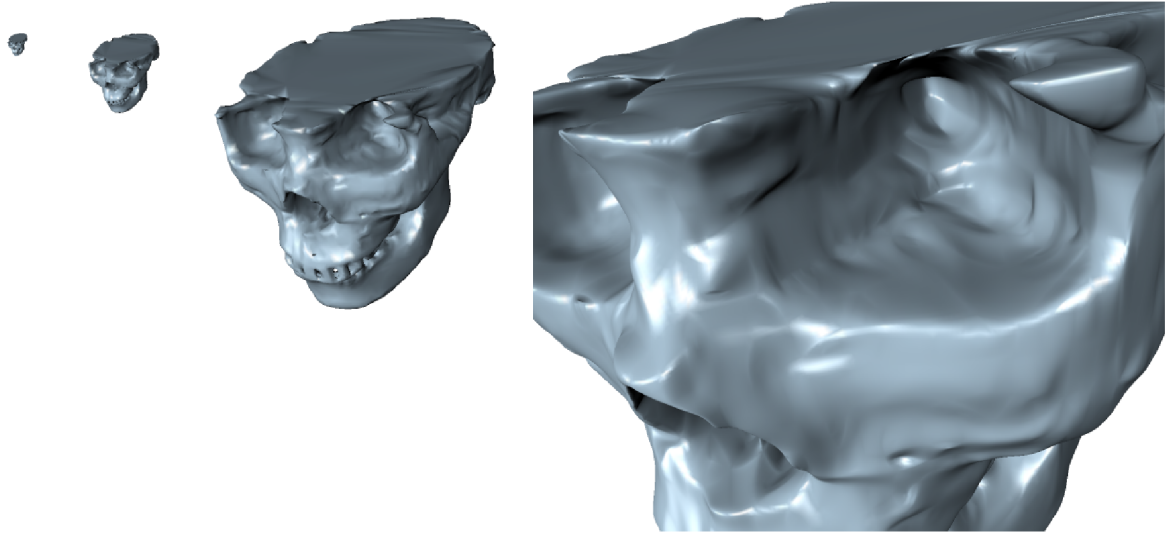


図 6.10: 視点の遠近に応じた詳細度制御

6.4 変形操作

本手法の応用例として、初期多面体を変形媒体とした自由曲面変形の実行例を示す。自由曲面の変形には、FFD(Free Form Deformation, 自由形状変形)[43]などの手法が考案されており、制御点を直接移動することは一般に行わない。それは、制御点を直接移動することによって意図した変形を実現することがとても困難であるためである。FFDによる変形では、変形対象となるモデルに対して変形媒体を定義する必要があるため、対象となる自由曲面とは別の制御ポリゴンモデリングする必要がある。制御ポリゴンは、自由曲面から自動的に作成することはできないために、一般的には専門のツールが必要になる。医用ラティス構造では、初期多面体をFFDにおける制御ポリゴンとみなすことで、直感的な変形が可能になる。それは、初期多面体と医用ラティス曲面の位相が1対1に対応しているためである。初期多面体に属するある稜線に移動操作を加えれば、それに対応する医用ラティス曲面の曲線が動的に移動される。変形操作後には、医用ラティス曲面から初期多面体を即時に再構築することが可能であるため、すぐに次の変形操作を実現することができる。このような初期多面体を利用した曲面変形操作のことを、ラティス変形と呼ぶ。

図 6.11 にラティス変形の手順を示す。まず、初期多面体において変形対象となる部位を選択する(図 6.11 左上)。次に、選択領域に変形操作を加える(図 6.11 右上)。この操作は普通のポリゴンメッシュの移動であるため、実時間での実行が可能である。選択要素の変形が終了した時点で、初期多面体に丸め操作を適用し医用ラティス曲面を内挿する(図 6.11 下)。以上の操作でラティス変形が実現される。丸め操作を行う最終ステップでは、初期多面体の全ての面に対して丸め操作を適用するのではなく、選択要素の影響領域だけに対して丸め操作を実行する。これにより動的に自由曲面の変形操作を実現することができる。ラティス変形は粘土をこねるような操作感で自由曲面のモデリングが可能であるため、一般的な3次元CGのモデリングツールとしての利用も可能である[59]。

FFDなどの変形手法と比較した場合のラティス変形の優位性は、曲面間の連続性の保持にある。FFDにおいて制御ポリゴンの変化量が多い場合などは、変形対象立体の連続性が崩壊することが度々ある。これは変形媒体と変形対象立体が無関係に作成されているからである。ラティス変形においては、変形媒体(初期多面体)から変形対象立体を生成するために、2つの立体において必ず対称性が確保されている。丸め操作において自由曲面を内挿する際は、4.1.2項で述べたように、曲面間の G^1 連続が保証されるために、崩れた形状が生成されることはない。

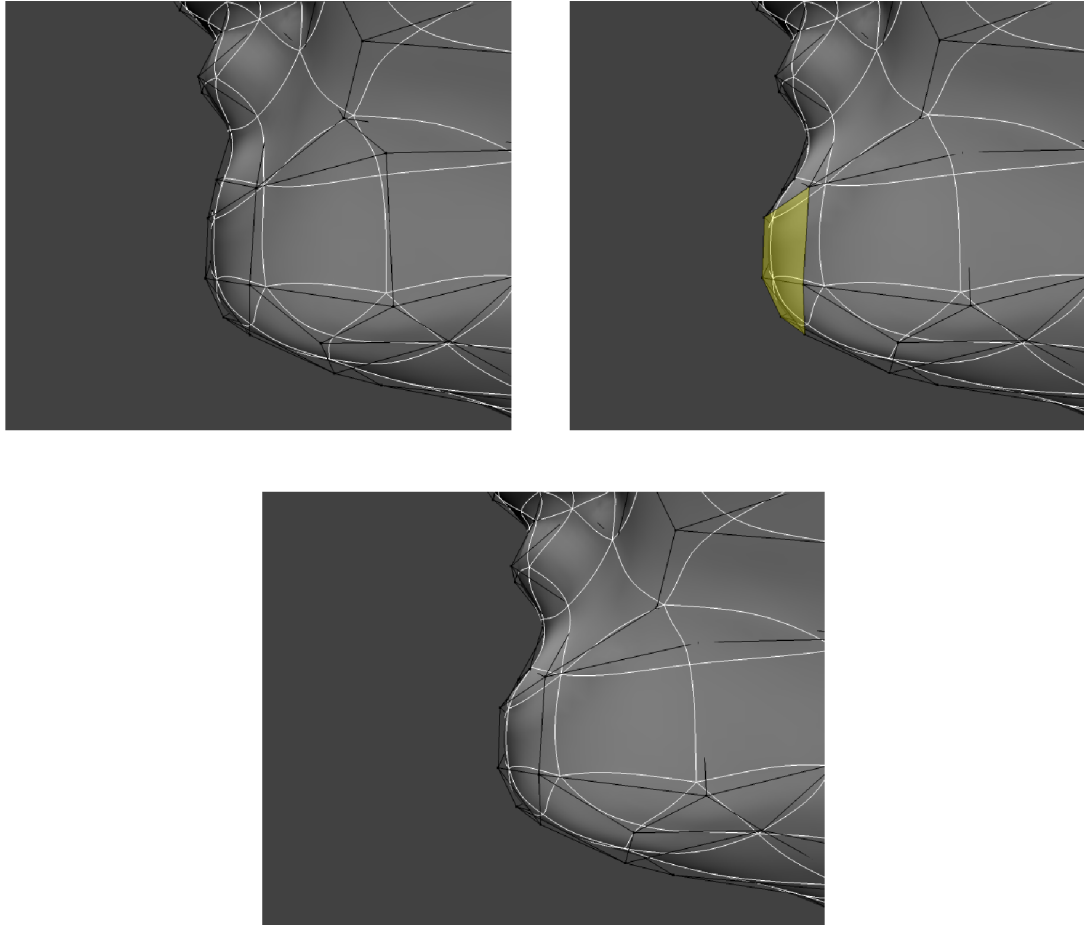


図 6.11: ラティス変形

図 6.12 と図 6.13 は、医用ラティス曲面の変形例である。図 6.12 は口元の変形例を示している。このような部位の変形操作は、顔面神経麻痺における表情筋の再建シミュレーションなどへの応用が期待できる。図 6.13 は顎の変形例を示している。5.3 節で述べた顎骨切断シミュレーションと組み合わせることで、下顎前突症などの下顎障害への応用が期待できる。

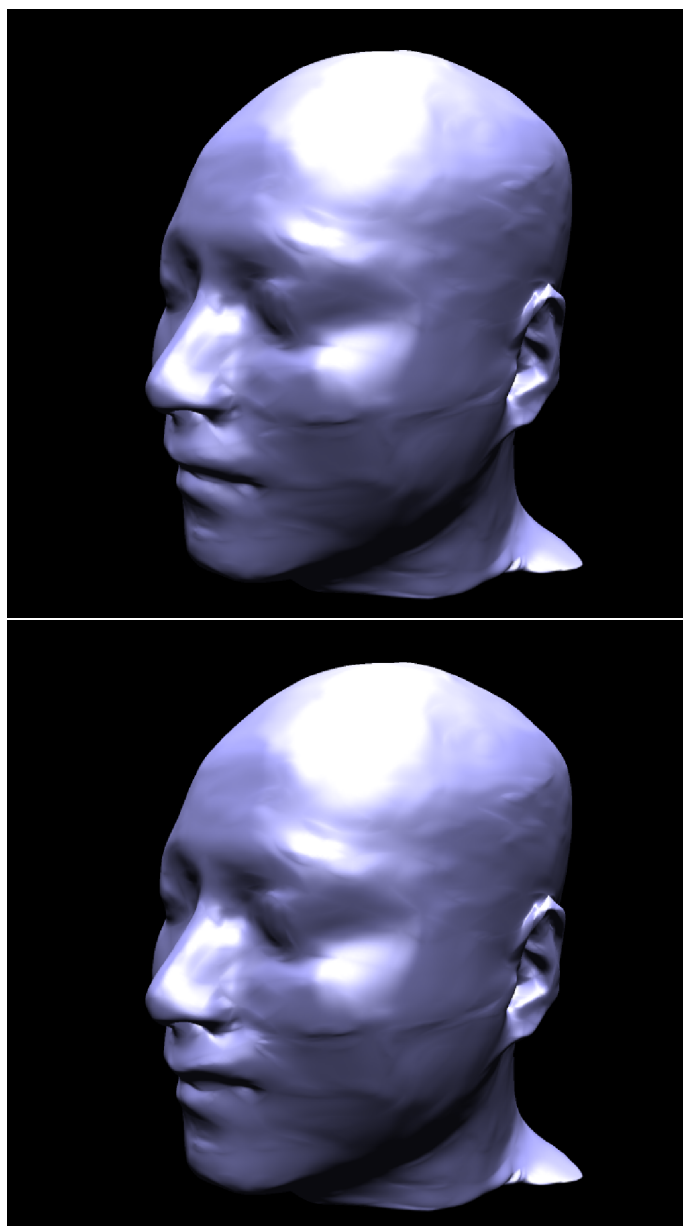


図 6.12: 医用ラティス曲面の変形例-口元の変形

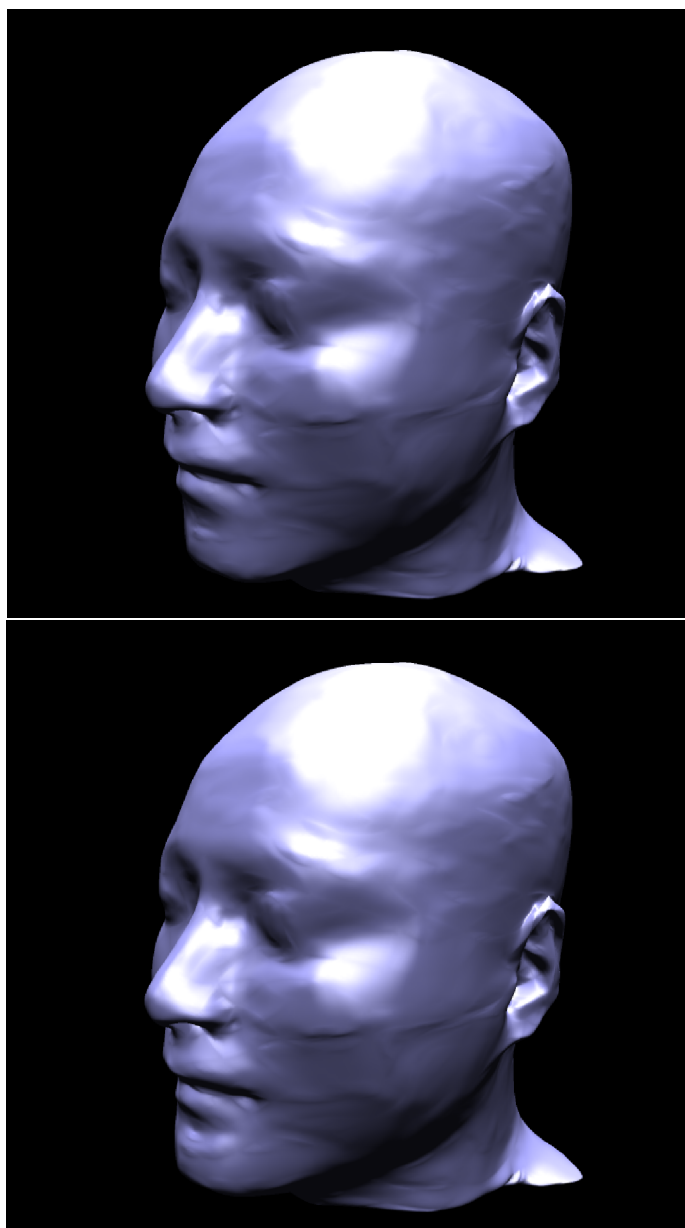


図 6.13: 医用ラティス曲面の変形例-顎の変形

第 7 章

結論と展望

7.1 結論

本論文では、Web3D 型の CAS システムを実現するために、医用ラティス構造という新しい形状表現とそのモデリング手法を提案した。

医用ラティス構造は、自由曲面に基づく形状表現であり、滑らかな曲面形状を少ないデータサイズで表現することが可能であることを示した。この特徴を利用することで高速なデータ転送が可能であり、Web3D 用途での効果的な利用が期待できる。更に、医用ラティス構造は、医用における 2 つの重要な利点を持つことを示した。第 1 は、特徴線の反映であり、切断線や生体組織の境界線などを明示的に可視化することが可能である。第 2 は、精度の適応的制御であり、患部のみを高精度にした最適化モデルの作成が可能である。

医用ラティス曲面を、患者の CT/MRI 画像から生成する手法を示した。まず、マーチングキューブ法を用いて、2 次元画像である CT/MRI 画像から 3 角形メッシュに基づく 3 次元モデルを再構築する。この過程において、スライス毎に 3 次元モデルをインタラクティブに編集できるオーサリングツールを実装し、医師による容易な 3 次元モデル作成環境を提供した。次に、3 角形メッシュに基づく 3 次元モデルを医用ラティス曲面に変換する曲面フィッティング手法を適用する。この曲面フィッティング手法は、3 角形メッシュ簡略化アルゴリズムを拡張したものであり、医用ラティス曲面の初期多面体を簡略化の結果として出力する手法である。出力された簡略化メッシュを初期多面体とみなして、丸め操作を行うことで、オリジナルな 3 角形メッシュを近似する曲面モデルを作成することができる。また、曲面フィッティングの過程で入力される特徴線と高精度領域を、出力される曲面モデルの輪郭線に反映する手法も提案した。これにより、患部領域に高い精度を割り当てることや、切断線を明示的に可視化することが可能になる。

本研究の有効性を検討するために、医用ラティス曲面に基づく 3 次元モデルを利用した Web3D アプリケーションについて紹介した。複数の症例をモデルケースに取り上げ、臨床、教育、研究の 3 つの観点から開発した教育アプリケーションや手術シミュレーションについて解説した。医用ラティス曲面の特徴である軽量性、特徴線の反映、精度の適応的制御が Web3D アプリケーションにおいて有効であることを示した。特に、ディスプレイタブレットを利用した下顎骨の切断シミュレーションでは、医師の手による直感的な手術シミュレーションが可能であることが実証された。

3 次元幾何形状圧縮手法としての本研究の有効性を検証するために、数値的な評価を行った。患者の CT/MRI 画像から作成される 3 角形メッシュと医用ラティス曲面との平均 2 乗距離誤差を測定し、提案した曲面フィッティング手法が高い精度での近似を実現していることを示した。また、特徴線が入力されたモデルや精度を適応的に制御したモデルについても

同様の手法で評価を行い，医用における有効性を示した．

応用事例として，曲面分割数による詳細度制御と初期多面体を利用した変形操作について述べた．詳細度制御への応用では，医用ラティス構造を構築する双 3 次 Gregory パッチの分割数を変化させることで，詳細度制御手法として利用可能であることを示した．また，初期多面体を最も詳細度の低い立体として表示することが有効であることも示した．初期多面体を利用した変形操作においては，医用ラティス曲面と初期多面体の位相が 1 対 1 に対応する利点を利用することで，直感的な自由曲面変形が可能であることを示した．また，自由曲面の内挿方法の特徴により，変形後においてもパッチ間の G^1 連続性を確保することが可能であることを示した．

7.2 今後の展望

今後の課題としては，第 1 に誤差入力による曲面フィッティングが挙げられる．本論文で提案した曲面フィッティング手法では，出力メッシュの頂点数を入力することで意図するモデルの生成を実現している．そのため，作成されるモデルの頂点数は確実に予想できるが，モデルの精度については予測することが困難である．より柔軟な曲面モデル生成のためには，出力モデルの最大誤差を入力し，それを終了条件とした 3 角形メッシュの簡略化操作を実現する必要がある．

第 2 に，臨床における使用を前提とした評価基準の設定である．手法の評価には平均 2 乗距離誤差を用いたが，これが臨床における評価基準を満たしているという科学的な保証はない．そのために，本論文で示した Web3D アプリケーションは，主に教育用途としての利用が中心になっている．臨床での使用を前提とした場合，小さな誤差は予想以上の影響を与える場合が少なくないため，医学的な根拠に基づく具体的な評価方法と基準値の検討が必要である．

第 3 に，FEM(Finite Element Method，有限要素法)に基づく物理シミュレーションの実現である．応用例として軟部組織の変形操作を示したが，正確なシミュレーションのためには骨と軟部組織の階層モデルに基づく物理的解析が必要である．医用ラティス曲面に FEM を適用可能にするための形状表現の拡張を進めていきたい．

付録

付録 A VRML97 によるラティス構造の記述

```
1:    #VRML V2.0 utf8
2:    ##### 重み付け情報ノード
3:    PROTO XVL_EDGE [
4:        field SFFloat round_val 0
5:        field SFVec3f round_str 0 0 0
6:        field SFVec3f round_end 0 0 0
7:    ]
8:    { Text { string [ "weight of edge " ] } }
9:    ##### 丸めフラグノード
10:   PROTO XVL_STATUS [
11:       field SFString status "XVL_LATTICE"
12:   ]
13:   { Text { string [ "status of shape" ] } }
14:   Group{
15:       children [
16:           ##### 初期多角形情報
17:           Group {
18:               children [
19:                   Shape {
20:                       geometry IndexedFaceSet {
21:                           #####
22:                           ## 初期多角形の幾何，位相情報（省略）
23:                           #####
24:                       }]],
25:           ##### ラティス曲面の重み付け情報
26:           Switch {
27:               choice [
28:                   XVL_STATUS {
29:                       status "XVL_GREGORY"
30:                   }
```

```

31:          XVL_EDGE {
32:              round_val 0.5
33:              round_str 0 0 0
34:              round_end 0 0 0
35:          }
36:          IndexedLineSet {
37:              coordIndex [ 24 103 ]
38:          }
39:      ]
40:      whichChoice -1
41:  }]
42:  }

```

この記述は VRML97 によるラティス構造の記述例である．初期多面体をポリゴンとして扱うことにより，ラティス構造をあらゆる VRML ブラウザで読み込むことが可能になる．もしブラウザが XVL をサポートするものであれば，初期多面体からラティス曲面を構築し，滑らかな曲面形状が描画される．このサンプルにあるように，XVL は Group ノードと Switch ノードの 2 つのパートから構成される．Group ノードには初期多面体の幾何情報が保存される．Switch ノードにはラティス曲面を構築するために必要な属性情報が保存される．Switch ノードの whichChoice フィールドが -1 になっているために，XVL を認識できないブラウザはこの情報を無視することができる．この場合は初期多面体の形状が描画されることになる．XVL を認識する場合は，choice フィールドから必要な情報を取得し，ラティス曲面を生成する．XVL ファイルをラティス曲面として表示するか，もしくは初期多面体として表示するかは XVL_STATUS ノードによって決定される．status フィールドが XVL_GREGORY であればラティス曲面を，XVL_LATTICE であれば初期多面体を描画する．このサンプルの場合はラティス曲面を描画する．曲面を生成するには，頂点と稜線に重み付けをすることで様々な形状を生成することができる．重み付け情報である重み値 (round_val)，始点丸めベクトル (round_str)，終点丸めベクトル (round_end) は XVL_EDGE ノード及び XVL_VERTEX ノードに保存される．このサンプルの場合はある 1 つの稜線について重み付けがされることになる．稜線は IndexedLineSet ノードにより記述される．座標値は初期多面体情報が記述されている Group ノードに用いたものを使用し，稜線を構成する 2 つの頂点インデックスのペアを記述することで稜線情報を記述する．このサンプルの場合は，初期多面体で用いられている 24 番目と 103 番目の頂点により構成される稜線に重み付けをすることになる．

付録 B X3D(VRML2002) によるラティス構造の記述

```
1:  <!ENTITY % Xv1ShellTypes " CS_POLYLN | CS_BEZIER
2:    | SS_POLYGON | SS_LATTICE | SS_GREGORY " >
3:  <!ELEMENT  Xv1Shell  (XYZ, UV, Faces*, Edges*, Vertices*)>
4:  <!ATTLIST  Xv1Shell
5:    Type      ( %Xv1ShellTypes; ) #REQUIRED
6:    DEF      ID      \#IMPLIED
7:    USE      IDREF   \#IMPLIED
8:  >
9:  <!ELEMENT  Faces  (Material? , Texture? ,Face+) >
10: <!ELEMENT  Edges  (Material? , Edge+) >
11: <!ELEMENT  Vertices  (Material? , Vertex+) >
12: <!ELEMENT  Face  (Material? , Texture? %Xv1G1T1FaceChild;) >
13: <!ATTLIST  Face
14:    ix      %MInt32; #REQUIRED
15:    tx      %MInt32; #IMPLIED
16:  >
17: <!ELEMENT  Edge  (Material? %Xv1G1T1EdgeChild; ) >
18: <!ATTLIST  Edge
19:    ix      %MInt32; #REQUIRED
20:    vec      %MInt32; #IMPLIED
21:    WRnd    %SFFloat; #IMPLIED
22:  >
23: <!ELEMENT  Vertex  (Material?) >
24: <!ATTLIST  Vertex
25:    ix      %MInt32; #REQUIRED
26:    VRnd    %SFFloat; #IMPLIED
27:  >
28: <!ELEMENT  Texture  (EMPTY) >
29: <!ATTLIST  Texture
30:    url      CDATA      \#IMPLIED
```

```

31:      repeatS  %SFBool;  "true"
32:      repeatT  %SFBool;  "true"
33:      bin      IDREF      \#IMPLIED
34:      DEF      ID         \#IMPLIED
35:      USE      IDREF      \#IMPLIED
36:  >
37:  <!ELEMENT  XYZ  \#PCDATA >
38:  <!ELEMENT  UV   \#PCDATA >

```

XML(eXtensible Markup Language)にはDTD(Document Type Definition)というタグ仕様の宣言ファイルが必要になる。XMLパーザはDTDを参照しながらファイルの構文チェックを行う。このサンプルは、ラティス構造を実現するデータ構造を定義し、それをDTDとしてまとめたものである。このDTDは次世代Web3D言語使用であるX3D(eXtensible 3D, VRML2002)の拡張ノードとしてWeb3D Consortiumに提案されている[51]。

CAD/CAM分野での利用も前提にするためには、データ構造はソリッドモデルを扱うことができなくてはならない。我々は位相要素として、シェル(XvShell)、面集合(Faces)、稜線集合(Edges)、頂点集合(Vertices)、面(Face)、稜線(Edge)、頂点(Vertex)を扱う。幾何要素としては、座標値(XYZ)、テクスチャ座標値(UV)を扱う。材質要素としては、テクスチャ(Texture)、マテリアル(Material)を扱う。更に、レンダリングはワイヤフレーム表示とシェーディング表示を制御する。ワイヤフレーム表示では、ポリライン(CS-POLYLN)、曲線(CS-BEZIER)の表示が可能である。シェーディング表示では、初期多角形表示(SS-LATTICE)、ラティス曲面表示(SS-GREGORY)、ポリゴンメッシュ表示(SS-POLYGON)が可能である。これらの表示モードの情報はXvShellTypesタグで制御される。シェル要素(XvShell)は自身の属性値(アトリビュート)として、レンダリングモード(Type)、自身のID(DEF)、他のシェルへのポインタ(USE)を保持している。面要素(Face)は属性値として、頂点座標のインデックス(ix)、テクスチャ座標のインデックス(tx)を保持している。稜線要素(Edge)は属性値として、頂点座標のインデックス(ix)、始点と終点の丸めベクトル(vec)、重み値(WRnd)を保持している。頂点要素(Vertex)は属性値として、頂点座標のインデックス(ix)、重み値(VRnd)を保持している。テクスチャ要素(Texture)は属性値として、テクスチャ画像ファイルのURL(url)、テクスチャの反復とクランプの情報(repeatS,repeatT)を保持している。

付録 C XVL Object Model によるスクリプトプログラミング

XVL Object Model では、Javascript や JScript などの ECMA Script によってプログラミングを行う。プログラミングは HTML ファイルの中に以下のような SCRIPT タグを挿入することで実現する。

```
<HTML>
  <HEAD>
    <SCRIPT language="Javascript">
      .
      .
      .
      .
    </SCRIPT>
  </HEAD>
  <BODY>
    <OBJECT width="450" height="450"
      classid="CLSID:C0975880-11DF-11D5-8084-0040B4116A31" id="player">
      <PARAM name="FileName" value="anim.xv3">
      <PARAM name="Behavior" value="viewer">
      <PARAM name="DivNum" value="4">
      <PARAM name="DrawBack" value="off">
      <OBJECT Id="player" cl>
        .
        .
      </OBJECT>
    </BODY>
  </HTML>
```

OBJECT タグ内の width と height は XVL Player の埋め込まれたウインドウの幅と高さを表す。classid に続く数列は XVL Player に割り振られたグローバルにユニークな ID である。また、id という属性には XVL Player への識別子を指定する。この識別子はどのような名前でもよいが、以後の解説では、player と指定したことにする。PARAM タグは、XVL Player のデフォルト属性を指定するものである。このサンプルの場合は、anim.xv3 という

ファイルをビューアモードで読み込み，その際の解像度は4，裏面表示は行わない，というオプションが与えられていることになる．

以下の解説では，SCRIPT タグを省略し，内部の記述だけを記す．ただし，特別なタグ定義が必要なイベント関連の解説においては，タグを省略せずに記す．

カメラアニメーション

カメラアニメーションを実現するためには，camera オブジェクトのメソッドとプロパティを利用する．メソッドとはオブジェクトのメンバ関数のことであり，プロパティとはオブジェクトのメンバ変数のことである．

```
1:    // カレントカメラオブジェクトを取得
2:    var camera1 = player.model.getCurrentCamera();
3:
4:    // 3次元ベクトルオブジェクトを生成
5:    var vec1 = player.createVec3();
6:    // 回転オブジェクトを生成
7:    var rot1 = player.createRotation();
8:
9:    // 3次元ベクトルオブジェクトに(0,0,-100)をセット
10:   vec1.setValue(0,0,-100);
11:   // 回転オブジェクトにY軸中心で180度回転をセット
12:   rot1.setValue(0,1,0,3.141592);
13:
14:   // カレントカメラに座標値と回転角を反映
15:   camera1.setAttitude(vec1, rot1);
```

上記のサンプルにあるように，全てのオブジェクトはplayer オブジェクトから参照する．また，3次元シーンに存在しないものは，player オブジェクトのメソッドで生成することができる．

座標系アニメーション

```
1:    // 3次元シーンから G_1 という名前のグループオブジェクトを取得
2:    var grp1 = player.model.getGroup("G_1");
3:
4:    // Y軸を中心に約 180 度回転しながら (100,100,100) に移動
5:    grp1.moveTo(100, 100, 100);
6:    grp1.rotate(0,1,0,3.1415);
```

上記の例にあるように、座標変換によるアニメーションは逐次実行されるのではなく、同時に実行される。つまり、全てのアニメーションを行う 4×4 の行列を作成し対象オブジェクトに乗算することになる。

```
1:    // 3次元シーンから G_1 という名前のグループオブジェクトを取得
2:    var grp1 = player.model.getGroup("G_1");
3:
4:    // 2倍に拡大しながら、現在位置から (100,100,100) 離れた場所に移動
5:    grp1.move(100, 100, 100);
6:    grp1.scale(2,2,2);
```

座標系アニメーションメソッドには、絶対的なものと相対的なものがある。上記の場合、`move(x,y,z)` が原点を中心とした絶対的な座標系アニメーションであるのに対し、`moveTo(x,y,z)` が現在位置を中心とした相対的なアニメーションになる。

マテリアルの管理

```
1:    // M1 という名前のマテリアルオブジェクトを取得
2:    var mat1 = player.model.getMaterial("M1");
3:    // マテリアルの透過度を 0.5 に設定する .
3:    mat1.transparency = 0.5;
```

上記の例は、マテリアルオブジェクトを取得後、そのプロパティに直接値を代入している。このように、メソッドを用いずにプロパティ参照によるアニメーションを実行することも可能になっている。

```
1:    // M1 という名前のマテリアルオブジェクトを取得
2:    var mat1 = player.model.getMaterial("M1");
3:
4:    // F2 という名前の面オブジェクトを取得
5:    var face2 = player.model.getFace("F2");
6:
7:    // F2 にマテリアル M1 を割り当てる
8:    face2.assignMaterial(mat1);
```

上記の例は、マテリアルオブジェクトを取得後、割り当て対象となるオブジェクトのメソッドを用いて代入を行っている。XVL Object Model では、ほとんどの操作がメソッドとプロパティの両方で実現可能になっている。

イベント駆動

```
<OBJECT ID="XVLPlayer" ... >.....</OBJECT>
<SCRIPT FOR="XVLPlayer" EVENT="OnXvlKeyDown(keycode)" LANGUAGE="JavaScript">
    alert("Key " + keycode + " is down.");
</SCRIPT>
```

上記のサンプルは、キーボードイベントによって XVL アニメーションを呼び出すための、HTML の記述方法である。イベント駆動でアニメーションを起動する場合は、イベントの種類ごとに SCRIPT タグを新たに用意する必要がある。この場合はキーボードが押された場合のキーボード番号をアラートする例になっているが、キーボードごとに様々な 3D アニメーションを駆動させることも可能である。

```
<OBJECT ID="player" ... >.....</OBJECT>
<SCRIPT FOR="player" EVENT="OnXvlVertexPick(object, name)"
LANGUAGE="JavaScript">
    if(object.id == "v_1") player.animStop();
</SCRIPT>
```

このサンプルは、頂点ピックをイベントとしてアニメーションを実行する例である。v_1 という名前の頂点がクリックされるとアニメーションが終了するようにプログラミングされている。EVENT 属性の引数には name という引数があり、イベントそのものに名前を付けることが可能になっている。

```
<OBJECT ID="player" ... >.....</OBJECT>
<SCRIPT FOR="player" EVENT="OnXvlLoad" LANGUAGE="JavaScript">
    alert("XVL Player starts");
</SCRIPT>
```

このサンプルは、XVL Player の起動時にアニメーションが実行されるようになっている。HTML の読み込み時とは別に、もう 1 つの読み込みイベントを持つことが可能になる。

アニメーションの属性制御

```
1:    // XVL Player の背景画像を back1.jpg に変更
2:    player.backgroundImage = "back1.jpg";
3:
4:    // XVL Player をビューアモードに変更
5:    player.behavior = "walker";
6:
7:    // anim_1 という名前のアニメーションを起動
8:    player.animStart("anim_1");
```

XVL Player そのものの属性を変更するためには、player オブジェクトを利用する。player オブジェクトは、OBJECT タグにて指定した ID でアクセスすることができる。player オブジェクトも他のオブジェクト同様、メソッドとプロパティによる双方のプログラミングが可能である。player オブジェクトのプロパティの多くのものは、OBJECT タグ内の PARAM タグにて指定することが可能である。

参考文献

- [1] A. Abadjev, M. del. Rosario, A. Lebedev, A. Migdal, and V. Paskhaver, MetaStream, *VRML'99*, pp.53–62, ACM Press, 1999.
- [2] B. G. Baumgart, Geometric modeling for computer vision, Technical report, Report STAN-CS-74-463, Stanford University: Standord Artificial Intelligence Laboratory, 1974.
- [3] J. Behr and M. Alexa, Volume rendering in VRML, *Web3D 2001*, ACM Press, pp.23–27, 2001.
- [4] P. Bézier, Definition numerique des courbes ed surfaces, *Automatisme*, 1966.
- [5] S. A. Coons, Surfaces for computer-aided design of space figures, MIT, 1964.
- [6] E. Catmull and J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, *Computer Aided Design*, pp.350–355, 1978.
- [7] H. Chiyokura and F. Kimura, Design of solids with free-form surfaces, *Computer Graphics (Proc. SIGGRAPH 85)*, pp.289–298, ACM Press, New York, 1983.
- [8] H. Chiyokura, An extended rounding operation for modeling solids with free-form surfaces, *IEEE Computer Graphics and Applications*, 1987.
- [9] P. Cignoni, C. Montani, R. Scopigno, Metro: measuring error on simplified surfaces, *Computer Graphics Forum*, Blackwell Publishers, vol. 17(2), pp.167–174, 1998.
- [10] T. DeRose, M. Kass, T. Truong, Subdivision surfaces in character animation, *Computer Graphics (Proc. SIGGRAPH 98)*, pp.85–94, ACM Press, 1998.
- [11] D. Doo and M. Sabin, Behaviour of recursive division surfaces near extraordinary points, *Computer Aided Design*, 1978.
- [12] R. A. Drebin, L. Carpenter and P. Hanrahan, Volume rendering, *Computer Graphics(SIGGRAPH'88 Proceedings)*, pp.65–74, 1988.
- [13] M. Eck and H.Hoppe, Automatic reconstruction of B-Spline surfaces of arbitrary topological type, *Computer Graphics(SIGGRAPH'96 Proceedings)*, pp.325–334, 1996.

- [14] G. Farin, Curves and surfaces for computer aided design, Academic Press, 1992.
- [15] T. Fujino, Simulation and computer-aided surgery, Wiley, 1994.
- [16] M. Garland and P. S. Heckbert, Surface simplification using quadric error metrics, Computer Graphics *Computer Graphics (Proc. SIGGRAPH 97)*, ACM Press, pp.209-216, 1997.
- [17] W. J. Gordon and R. F. Riesenfeld, Bernstein-Bézier methods for the computer aided design of free-form curves and surfaces, *Journal of the ACM*, 21:293-310, 1974.
- [18] J. A. Gregory, Smooth interpolation without twist constraints, *Computer Aided Geometric Design*, R.E.Barnhill and R.F.Riesenfeld, ed., Academic Press, 1974.
- [19] R. Geiger, Three-dimensional modeling of human and its application to diagnosis and surgical planning, Technical Report 2105, Institut National de Recherche en Informatique et Automatique, (France), 1993.
- [20] T. Harada, H. Toriya and H. Chiyokura, An enhanced rounding operation between curved surfaces in solid modeling, *CG International 90*, pp.563-588, 1990.
- [21] O. Hendin, N. John and O. Shochet, Medical volume rendering on the WWW using JAVA and VRML, *MMVR 98*, IOS Press, pp34-40, 1998.
- [22] H. Holten-Lund, M. Hvidtfeldt, J. Madsen and S. Pedersen, VRML visualization in a surgery planning and diagnostic applications, *Web3D-VRML 2000*, ACM Press, pp.111-118, 2000.
- [23] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, Surface reconstruction from unorganized points, *Computer Graphics(SIGGRAPH'92 Proceedings)*, pp.71-78, 1992.
- [24] H. Hoppe, T. DeRose, T. Duchamp, H. Jin, M. Halstead, J. McDonald, J. Schweitzer, and W. Stuetzle, Piecewise smooth surface reconstruction, *Computer Graphics(SIGGRAPH 94 Proceedings)*, pp.295-302, 1994.
- [25] H. Hoppe, Progressive meshes, *Computer Graphics (Proc. SIGGRAPH 96)*, pp99-108, ACM Press New York, 1996.

- [26] N. John, N. Phillips, R. Vawda and J. Perrin, A VRML simulator for ventricular catheterization, Eurographics UK, 1999.
- [27] N. John and N. Phillips, Surgical simulators using the WWW, *MMVR 2000*, IOS Press, pp.146-152, 2000.
- [28] A. Khodakovsky and P. Schröder, Fine level feature editing for subdivision surfaces, *Solid Modeling 99*, pp.203-211, 1999.
- [29] L. Kobbelt, M. Botsch, U. Schwaner and H. P. Seidel, Feature sensitive surface extraction from volume data, *Computer Graphics (Proc. SIGGRAPH 01)*, pp.57-66, ACM Press, New York, 2001.
- [30] R. M. Koch, M. H. Gross, F. R. Carls, D. F. Von Büren, G. Fankhauser and Y. I. H. Parish, Simulating facial surgery using finite element models, *Computer Graphics (SIGGRAPH'96 Proceedings)*, pp.421-428, ACM Press, 1996.
- [31] A. Lee, W. Sweldens, P. Schroeder, L. Cowsar and D. Dobkin, MAPS: Multiresolution adaptive parameterization of surfaces, *Computer Graphics (SIGGRAPH'98 Proceedings)*, pp.95-104, 1998.
- [32] A. Lee, H. Moreton and H. Hoppe, Displaced subdivision surfaces, *Computer Graphics (Proc. SIGGRAPH 2000)*, pp.85-94, ACM Press, New York, 2000.
- [33] C. Loop, Smooth subdivision surfaces based on triangles, Master's thesis, University of Utah, Department of Mathematics, 1987.
- [34] C. Loop, Smooth spline surfaces over irregular meshes, *Computer Graphics (Proc. SIGGRAPH 94)*, pp.303-310, ACM Press, New York, 1994.
- [35] W. E. Lorensen and H. E. Cline, Marching cubes : a high resolution 3D surface construction algorithm, *Computer Graphics (Proc. SIGGRAPH 87)*, ACM Press, New York, 1987.
- [36] M. Levoy, A hybrid ray tracer for rendering polygon and volume data, *IEEE Computer Graphics and Applications*, 10:2, pp.33-40, 1990.
- [37] M. Mäntylä, *An Introduction to Solid Modeling*, Computer Science Press, 1988.

- [38] Microsoft Corporation, DHTML Reference, <http://msdn.microsoft.com/workshop/-author/dhtml/reference/dhtmlrefs.asp>, 2002.
- [39] National Library of Medicine, The Visible Human Project, http://www.nlm.nih.gov/-research/visible/visible_human.html, 1998.
- [40] D. R. Ney, E. K. Fishman, D. Magid and R. A. Drebin, Volumetric rendering of computed tomographic data: principles and techniques, *IEEE Computer Graphics and Applications*, 10(2):pp.24-32, 1990.
- [41] Oak Ridge National Laboratory, Virtual human project, <http://www.ornl.gov/rwd/newVH.html>, 1999.
- [42] M. Pesce, VRML-browsing and building cyberspace, New Riders Publishing, 1995.
- [43] T. W. Sederberg and S. R. Parry, Free-form deformation of solid geometric models, *Computer Graphics(SIGGRAPH'86 Proceedings)*, pp.151-159, 1986.
- [44] T. Takamura, M. Ohta, H. Toriya and H. Chiyokura, A method to convert a gregory patch and a rational boundary gregory patch to a rational bezier patch and its application, *CG International 90*, pp.543-562, Springer-Verlag, Tokyo 1990.
- [45] S. Takeuchi, T. Kanai, H. Suzuki, K. Shimada and F. Kimura, Subdivision surface fitting with QEM-based mesh simplification and reconstruction of approximated b-spline surfaces, *Pacific Graphics 2000*, pp.202-212, 2000.
- [46] M. Vannier, J. Marsh and L. Warren, Three-dimensional computer graphics for craniofacial surgery planning and evaluation, *Computer Graphics(SIGGRAPH'83 Proceedings)*, ACM Press, 1983.
- [47] A. Wakita, M. Yahima, T. Harada, H. Toriya, H. Chiyokura, XVL: A compact and qualified 3D representation with lattice mesh and surface for the internet, *ACM Web3D-VRML2000*, pp.45-51, ACM Press, New York ,2000.
- [48] A. Wakita, T. Hayashi, T. Kanai, H. Chiyokura, Using lattice for web-based medical applications, *ACM Web3D 2001*, pp.29-34, ACM Press, New York 2001.
- [49] Web3D Consortium External Authoring Interface Working Group, EAI : External authoring interface, <http://www.web3d.org/WorkingGroups/vrml-eai/EAI>, 2002.

- [50] Web3D Consortium, X3D (Extensible 3D) : New-generation open web3D standard, <http://www.web3d.org/x3d/>, 2002.
- [51] Lattice Technology, Lattice extension for X3D, <http://www.web3d.org/TaskGroups/x3d/lattice/LatticeProposal.html>, 2002.
- [52] D. Zorin, P. Schröder and W. Sweldens, Interactive multiresolution mesh editing, *Computer Graphics (Proc. SIGGRAPH 97)*, pp.259–268, ACM Press, New York, 1997.
- [53] 佐藤淳, 三原円美, 千代倉弘明, 小林正弘, 中嶋英雄, 藤野豊美, CAD 技術の外科手術への応用, 第 7 回 NICOGRAPH 論文コンテスト, pp.320-328, 1991.
- [54] 金井 崇, 細分割曲面の連続的多重解像度表現, 情報処理学会 グラフィックスと CAD 研究会第 14 回夏の集中研究集会「インターネット時代の CG と CAD」(情報処理学会研究報告, Vol.2000, No.78, 99-CG-100, pp. 19-24), 山梨, 9 月, 2000.
- [55] 千代倉弘明: ソリッドモデリング, 工業調査会, 1985.
- [56] 千代倉弘明, グラフィックス機能論, 産業図書, 1994.
- [57] 鳥谷浩志, 千代倉弘明, 3 次元 CAD の基礎と応用, 共立出版, 1991.
- [58] ラティス・テクノロジー株式会社, XVL Player, <http://www.xvl3d.com/j/support/-player/main.htm>, 2002.
- [59] ラティス・テクノロジー株式会社, Lattice Designer, <http://www.xvl3d.com>, 2002.

発表論文

投稿論文他（査読付）

1. 脇田玲，矢島誠，原田毅士，鳥谷浩志，千代倉弘明，”ラティス構造に基づく軽量で高品質な Web3D データ表現”，情報処理学会論文誌，第 42 巻 第 5 号，2001．
2. Mitsuaki Saito, Takamichi Hayashi, Akira Wakita, Hiroaki Chiyokura, ”Feature Preserving Refinement of Surfaces for Web-based Surgical Simulation”, MMVR 2002(Medicine Meets Virtual Reality 2002), to appear.
3. Akira Wakita, Takamichi Hayashi, Takashi Kanai, Hiroaki Chiyokura, ”Using Lattice for Web-based Medical Applications”, Web3D 2001, pp.29-34, ACM-Press, 2001.
4. Akira Wakita, Makoto Yajima, Tsuyoshi Harada, Hiroshi Toriya, Hiroaki Chiyokura, ”XVL : A Compact and Qualified 3D Representation with Lattice Mesh and Surface for the Internet”, Web3D-VRML 2000, pp.45-51, ACM-Press, 2000.
5. Akira Wakita, Tsuyoshi Harada, Hiroaki Chiyokura, ”Surface Fitting to Polygon Meshes with Bicubic Gregory Patch”, Sixth SIAM Conference on Geometric Design, Albuquerque, New Mexico, 1999.

講演論文他

1. 脇田玲，千代倉弘明，三浦憲二郎，金井崇，”慶應 SFC におけるモデリング教育”，情報処理学会 第 106 回 グラフィックスと CAD 研究会 (発表予定)．
2. 金井崇，脇田玲，千代倉弘明，”慶應 SFC における CG 教育”，情報処理学会 第 106 回 グラフィックスと CAD 研究会 (発表予定)．
3. 武部佳文，脇田玲，千代倉弘明，”タブレットインターフェイスを用いたラティス構造のリモデリング”，第 17 回 Nicograph/Multimedia 論文コンテスト，2001．
4. 斎藤満昭，林隆道，脇田玲，千代倉弘明，”医用曲面モデル生成システムの構築”，第 17 回 Nicograph/Multimedia 論文コンテスト，2001．
5. 脇田玲，千代倉弘明，”Web3D の現状と今後の動向”，第 17 回 Nicograph/Multimedia 論文コンテスト，2001．

6. 林隆道，齋藤満昭，脇田玲，金井崇，千代倉弘明，”CT・MRI データからのラティス構造に基づく医用曲面モデル自動再構成”，3次元画像コンファレンス 2001，東京，7月，2001．
7. 脇田玲，矢島誠，原田毅士，鳥谷浩志，千代倉弘明，”XVL：ラティス構造に基づく軽量で高品質な Web3D データ表現”，情報処理学会 グラフィックスと CAD 研究会 第 14 回夏の集中研究集会「インターネット時代の CG と CAD」山梨，9月，2000．
8. 斎藤佳奈子，脇田玲，金井崇，千代倉弘明，”WWW における三次元キャラクターデザインシステムの開発”，第 16 回 NICOGRAPH/MULTIMEDIA 論文コンテスト，論文集 pp.175-182，2000．
9. 齋藤満昭，林隆道，脇田玲，小林正弘，金井崇，千代倉弘明，”医用曲面モデル自動再構成システムの構築”，第 11 回日本コンピュータ支援外科学会，東京，9月，2001．
10. 武部佳文，斎藤佳奈子，脇田玲，千代倉弘明，”CoDA - Coal-mine Digital Archive”，情報処理学会 第 62 回 (平成 13 年前期) 全国大会，特別トラック (2) 講演論文集 コンピュータグラフィックスとビジュアルコンテンツ，2001．
11. 脇田玲，渡邊大地，千代倉弘明，”キャラクターラインを考慮したポリゴンデータの最適化”，電子情報通信学会 1998 年度総合大会シンポジウム講演論文集，1998．
12. 渡邊大地，脇田玲，千代倉弘明，”三角形メッシュ最適化における複数手法の連携の実現”，電子情報通信学会 1998 年度総合大会シンポジウム講演論文集，1998．

解説記事

1. 脇田玲，”Web3D の最前線”（解説記事），「情報処理」vol.42，No.2，pp167-172，情報処理学会，2001．
2. 脇田玲，”サイバースペース構築技術について”，「node - 20 世紀の建築と 21 世紀の建築（新建築 創刊 75 周年記念 2000 年 11 月臨時増刊）」，pp82-83，2000．
3. 鳥谷浩志，脇田玲，千代倉弘明，”3D コミュニケーション社会に必要なデータ流通環境とは？：インダストリーからホビーまでカバーする新しいモデリングカーネルの提案”，「日経 CG」1999 年 1 月号，pp176-179，1999．

謝辞

本論文を締めくくるにあたり，日頃より懇切にご指導頂き，本研究を進める上で多くの適切な助言を頂きました本塾大学院政策・メディア研究科の千代倉弘明教授に心より感謝申し上げます．

本塾看護医療学部的小林正弘助教授には，CT データのご提供や医学的側面からのご指導を頂きました．また，手術シミュレーションの際には医師の立場としてご参加頂き，多くの有用なご意見を頂きました．

本塾 環境情報学部の金井崇専任講師には，本研究を進めるに上での非常に重要な議論をして頂きました．また，論文の執筆方法から海外での学会発表に至るまで，研究活動を進める上での大変有意義な知識をご教授して頂きました．

システム開発及び論文執筆にあたり惜しみないご協力を頂いた「サイバーワールドモデリングと医学への応用プロジェクト」の諸氏に感謝いたします．特に，政策・メディア研究科修士課程2年の武部佳文氏には，本手法の有効性を評価する上で非常に重要であるディスプレイタブレットによる切断シミュレーションを実装して頂きました．また，環境情報学部4年の斎藤満昭氏，同学部3年の豊島周氏には，論文執筆のためのたくさんの資料を提供して頂きました．

システムの開発環境を提供して頂いたラティス・テクノロジー株式会社インターネットグラフィックス事業部の皆様に感謝申し上げます．田中浩司氏には，効率的な開発を行うための上位ライブラリ群を提供して頂きました．矢島誠氏には，本研究の数学的背景について熱心に相談に乗って頂きました．

本塾 大学院政策・メディア研究科の 奥出直人教授，武藤佳恭教授，古川康一教授には本稿作成の上での貴重なご意見を頂きました．

最後に，日頃より暖かく励まし支えてくださった家族と友人に深く感謝致します．