

心筋線維走向モデリングのためのスケッチインタフェース A Sketch-Based Interface for Modeling Myocardial Fiber Orientation

高山 健志[†] 五十嵐 健夫^{†,‡}

原口 亮* 中沢 一雄*

Kenshi TAKAYAMA[†] Takeo IGARASHI^{†,‡}

Ryo HARAGUCHI* and Kazuo NAKAZAWA*

[†] 東京大学

[†] The University of Tokyo

[‡] JST PRESTO

*国立循環器病センター *National Cardiovascular Center Research Institute

E-mail: [†] kenshi@ui.is.s.u-tokyo.ac.jp, [‡] takeo@acm.org,

*{haraguch, nakazawa}@ri.ncvc.go.jp

概要

本稿では、心臓電気生理シミュレーションにおいて重要な要素である心筋線維走向データをモデリングするためのスケッチインタフェースを提案する。提案手法は心筋線維走向が心臓表面において常にその接面に沿うという観察に基づいており、ユーザは心臓表面にストロークを描くことで心臓内部全体を満たす線維走向データを直感的に作成することができる。システムはユーザの入力を基に表面のメッシュ上と内部のボクセル上の2段階で Laplacian smoothing を行い、心臓内部全体の筋線維走向を計算する。一人の医師を対象に初期段階のユーザスタディを行い、本手法の有用性および課題を確認した。

1. はじめに

日本国内で毎年約9万人が亡くなる心臓突然死の原因の80~90%は致死性不整脈であり、従って不整脈対策は現在、国家的重要課題となっている。これに対するアプローチとして、計算機上に3次元的な心臓モデルを構築し電気生理シミュレーションを行うという手法がある[6](図1a)。これは3次元格子上に定義された仮想的な心筋ユニットに対して、その振る舞いを電気生理学的な数理モデルに基づいてシミュレートすることにより、心臓全体の電気生理現象を再現しようとする取り組みであり、最終的には致死性不整脈メカニズムの解明や、臨床における予防・診断・治療に役立てることを目標としている。

このシミュレーションには様々なデータやパラメータが必要とされるが、心筋線維(図1b)の走向データ

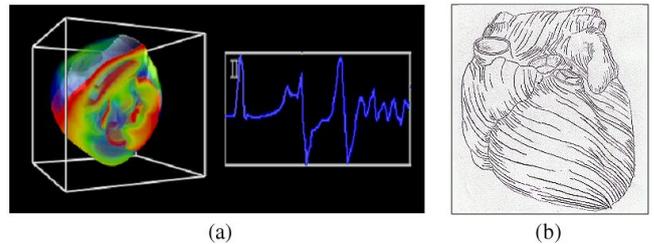


図 1: (a)心臓電気生理シミュレーション (b)心筋線維の概観

はその中でも重要な要素の一つである[1]。3次元ボリューム上のベクトル場として表現される心筋線維走向データは、心臓形状データ等に比べ計測が難しく、従ってシミュレーションのためには、医師が知識と経験に基づきある程度恣意的にモデリングする必要がある。

3次元ボリューム上のベクトル場の解析と可視化に関する研究[5]や、2次元平面上あるいは3次元形状の表面上のベクトル場デザインに関する研究[3]はこれまで多くなされているが、3次元ボリューム上のベクトル場デザインに関する研究は殆どなされていない。そのため心臓電気生理シミュレーションの分野では従来、個別のスライスごとに線維走向を指定していくといった非常に煩雑な手法が取られており、複雑な心筋線維走向をモデリングすることが極めて困難であった。

そこで本稿では、心筋線維走向モデルを表す3次元ボリューム上のベクトル場を、スケッチにより直感的に作成できるインタフェースを提案する。提案手法は心筋線維走向が心臓表面において常にその接面方向に沿うという観察に基づいており、ユーザのスケッチを入力として、表面のメッシュ上および内部のボクセル上の2段階で補間を行うことにより心臓内部全体の心

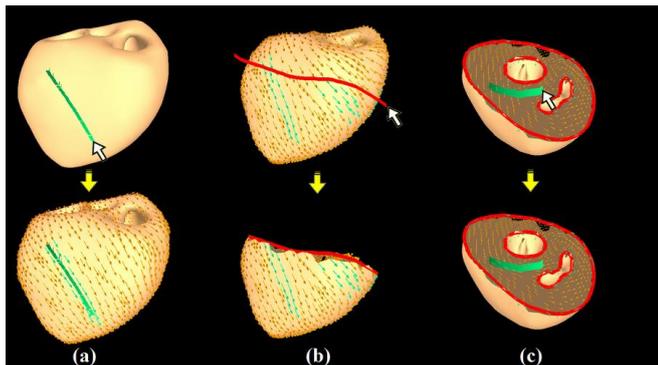


図 2: プロトタイプシステムのスナップショット
(a)表面上にストロークを描く (b)断面の生成 (c)断面上にストロークを描く

筋線維走向を計算することが可能である。

図 2 はプロトタイプシステムのスナップショットである。ユーザはペンやマウス等を用いてモデル上にストロークを描くことができる。システムはユーザが描いたストロークを基に補間を行い、ボリューム上のベクトル場を得る。補間アルゴリズムには Laplacian smoothing を用いているため、インタラクティブなモデリングが可能である。

初期段階のユーザスタディとして、心臓電気生理シミュレーションの研究を行っている一人の医師に提案システムを試用してもらい、インタビューを行った。提案システムでは心筋線維走向を手早く直感的にモデリングすることが可能であり、実用的なツールとなり得るとの評価を得るとともに、今後の課題となる点が明らかになった。

2. ユーザインタフェース

最初にシステムはユーザ指定の 3 次元ポリゴンモデルを読み込む。いくつかの前処理(内部ボクセルの計算, Laplacian 行列の計算等)の完了後、ユーザはスケッチによりボリューム上のベクトル場をデザインすることができる。

まずユーザはモデルの表面にストロークを描くことで表面のローカルな線維走向を指定する(図 2a)。ユーザは満足がいくまで好きなだけストロークを描くことができる。また、モデルを横切るストロークを描くことでモデルを切断し断面を生成することができる(図 2b)。ストロークは視線方向に押し出され、モデルのうちストロークの左側にある部分が非表示になる。ユーザはこの断面上にもストロークを描くことができ(図 2c)、内部の線維走向を指定することができる。また切断は心室の内壁面にストロークを描く際にも有効である(図 3)。

これらのストロークを入力として受け取ると、シ

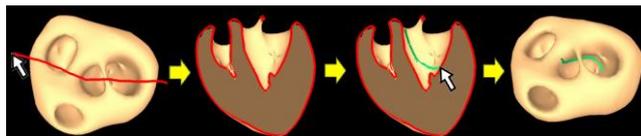


図 3: 切断により心室内壁面にストロークを描く

ステムは表面と内部の 2 段階で Laplacian smoothing を行い、ボリューム上の筋線維走向を得る。計算にかかる時間は数秒であり、ユーザは満足の行く結果が得られるまでストロークを追加および削除できる。

4. アルゴリズム

4.1 概要

3 次元モデルがロードされると、システムはまずその内部に含まれるボクセルを、各座標軸方向のスキャンに基づく手法で求める。このとき、モデル内外の境界にあたるボクセルには、その近傍のポリゴンの情報が記憶される。これは 2 段階の補間の際、メッシュ上のベクトル場をボクセル上のベクトル場に制約として与えるのに利用される。

表面のベクトル場は各メッシュ頂点に、内部のベクトル場は各ボクセルに割り当てられる。ユーザが描いたストロークを入力として受け取ると、システムは以下のように 2 段階の補間を行い内部のベクトル場を得る(図 4)。まず表面のストローク付近の頂点にそのストロークの向きと同じ向きのベクトルを設定する(図 5a)。次にそれらを制約としてメッシュ上で Laplacian smoothing を行い、表面のベクトル場を得る(図 4a)。次に、境界上にあるボクセルにその付近のメッシュ頂点のベクトルを混ぜ合わせて(図 5b) 制約として割り当てる(図 4b)。断面上に描かれたストロークもその付近のボクセルのベクトルに割り当てられる(図 5c)。最後にボクセル全体に対し Laplacian smoothing を行い、内部のベクトル場を得る(図 4c)。

我々の目的は方向を決めることなので、ベクトルの大きさは考慮しない。そのため、制約を与える際および補間の計算後には全てのベクトルは単位ベクトル化される。また、表面上で補間されたベクトルは必ずし

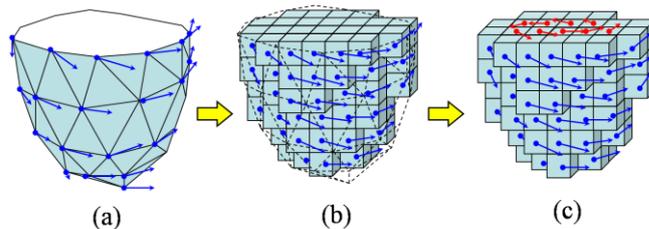


図 4: 2 段階の補間 (a)表面で補間 (b)境界上のボクセルに表面のベクトルを割り当てる (c)内部で補間

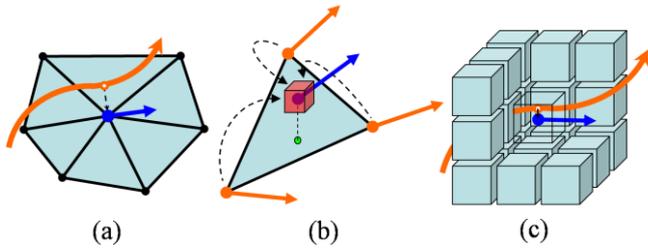


図 5: 制約の与え方 (a)頂点に表面ストロークの方向を設定 (b)境界のボクセルに付近の頂点の混合を設定 (c)ボクセルに内部ストロークの方向を設定

も表面に沿っているとは限らないので、補間の後に各メッシュ頂点についてその法線方向の成分を取り除き、表面上に沿わせる補正を行っている。

4.2 Laplacian Smoothing

ここでは本手法において Laplacian smoothing がどのように用いられたかを簡単に説明する(アルゴリズムに関する詳細は[7]を参照)。

x_1, \dots, x_n をメッシュ頂点またはボクセルに割り当てられた方向ベクトルとする。 x_i の Laplacian δ_i は次のように定義される:

$$\delta_i = x_i - \sum_{j \in N_i} w_j^i x_j$$

ここで N_i は x_i の近傍(メッシュ頂点の場合は頂点 i に隣接する頂点, ボクセルの場合はボクセル i に隣接するボクセル)である。重み w_j^i については単に $w_j^i = \frac{1}{|N_i|}$ とした。つまり δ_i は x_i とその近傍の平均との差となる。

我々の目的は以下の制約

$$x_{k_i} = b_i \quad (i = 1, \dots, m)$$

を満たしつつこれらの Laplacian の二乗を最小化することである。ここで k_i は i 番目の制約となる変数のインデックスであり, m は制約の総数である。メッシュ頂点の場合, 制約は表面上に描かれた入力ストロークによって与えられる(図 5a)。ボクセルの場合, 制約は表面に作られたベクトル場(図 5b)および断面上に描かれたストローク(図 5c)により与えられる。

この問題はベクトルと行列の表現を用いて以下のように書き換えられる:

$$\text{minimize} \left\| \begin{pmatrix} L \\ C \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix} \right\|^2 \quad (1)$$

ここで $\mathbf{x} = (x_1, \dots, x_n)^T$ および $\mathbf{b} = (b_1, \dots, b_m)^T$ はベクトルである。 $L = (l_{ij})$ および $C = (c_{ij})$ はそれぞれ $n \times n$ および $m \times n$ の行列であり, 以下で定義される:

$$l_{ij} = \begin{cases} -1 & (i = j) \\ w_j^i & (j \in N_i) \\ 0 & \text{otherwise} \end{cases}$$

$$c_{ij} = \begin{cases} 1 & (j = k_i) \\ 0 & \text{otherwise} \end{cases}$$

(1)は以下の線型方程式を解く問題に帰着される:

$$A^T A \mathbf{x} = A^T \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix} \quad (2)$$

ただし $A = \begin{pmatrix} L \\ C \end{pmatrix}$ である。行列 $A^T A$ およびベクトル $A^T \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}$ は, さらに次の単純な形に書き直せる:

$$\begin{aligned} A^T A &= (L^T \ C^T) \begin{pmatrix} L \\ C \end{pmatrix} \\ &= L^T L + C^T C \\ A^T \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix} &= (L^T \ C^T) \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix} \\ &= C^T \mathbf{b} \end{aligned}$$

なお, $C^T C$ の k_i 番目の対角要素は 1 で, $C^T \mathbf{b}$ の k_i 番目の要素は b_i であり, 残りの要素は全て 0 である。

システムはユーザがストロークを描く度に(2)を解く。メッシュ頂点および内部ボクセルの Laplacian 行列 L は与えられた 3 次元形状モデルに対して一定なので, $L^T L$ を事前に計算することが可能であり, 解く際に制約に対応する対角成分に 1 を加えれば良い。ボクセルに対する Laplacian smoothing では, $A^T A$ は表面上にストロークを描いている間は不変であり, 断面上にストロークを描いた時に初めて変化する。これらの行列は十分に疎であるため, 疎行列に最適化されたアルゴリズムを適用することができる。我々は LU 分解に基づく高速疎行列ライブラリを用いた[2]。

5. 結果およびユーザスタディ

現在のプロトタイプシステムは C++ で実装されており, Windows マシン上で動作する。図 6 は本システムを用いてモデリングした心筋線維走向の例である。用いた心臓モデルの頂点数および内部ボクセル数はそれぞれ 1992 および 7646 であり, 計算に要した時間は 2.1GHz の CPU および 2.0GB のメモリを搭載したマシンを使って約 3 秒であった。

提案手法についてフィードバックを得るため, 心臓電気生理シミュレーションの研究に取り組む一人の医師に本システムを試用してもらい, 実際に心筋線維走向をモデリングしてもらった。テスト環境には標準的なノート PC およびマウスを用い, モデリングの対象とする心臓モデルには市販の 3 次元ポリゴン心臓モデルを加工したものを使用した。まず初めにシステムの概要について簡単に説明し, 10 分程度のチュートリアルでシステムの操作法に慣れてもらった。その後実際にモデリングをしてもらったところ, 8 分程度で図 7a に示す心筋線維走向モデルが完成した。図 7b はそれを

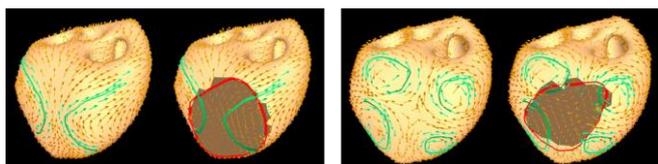


図 6: プロトタイプシステムでモデリングした心筋線維走向の例

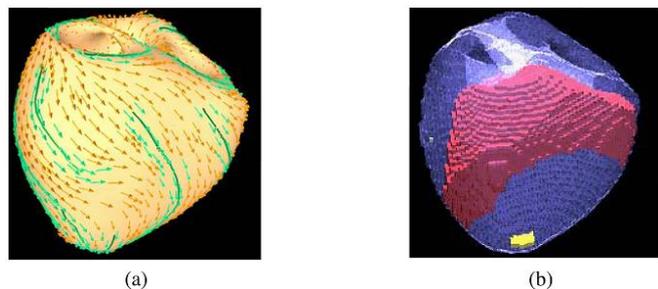


図 7: (a)医師によるモデリング結果 (b)簡易シミュレーション結果

簡易シミュレータ[4]に入力した結果の様子である。

その後、医師にインタビューを行い以下のようなフィードバックを得た。まず提案手法について、3次元的心筋線維走向を直感的にモデリングできるという点で、心臓シミュレーションの分野にとって重要な意味を持つものであるとの評価を得た。また作成されたモデルにおいて、心室壁中間層で線維走向がねじれるという心筋特有の構造を表現できている点も優れていると評価された。時間に関しては、仮に頂点やボクセルの数が増えて計算に時間がかかったとしても、一つ一つ手で設定するよりは間違いなく速いというコメントを得た。

また提案システムのもう一つの重要な点として、ベクトル場がポリゴンやストローク等の幾何形状に基づいて計算されていることが挙げられた。(本システムではまだサポートされていないが)それらの幾何形状を変形することによって一つのモデルから多数のバリエーションを簡単に作り出すことができれば、モデル作成が非常に容易になりシミュレーションのボトルネックを解消できるという可能性を指摘された。

また今後の課題として、より現実的なモデリングを支援するために MRI データの情報を提示できると良いということが挙げられた。これはノイズの多い医療データを人間の手でフィルタリングするという意味でも重要である。

最後に、心筋線維走向の可視化手法としては、断面表示という方法はあまり適さないことが指摘された。解剖学的には、心筋線維が束になって層を成し多数の層が折り重なって心臓を構成しているので、研究者は一般に心筋線維を理解するのに、断面ではなく層をイ

メージすることが多い。これに対する一つの改善案として、層を一枚ずつ剥いていき内側の線維走向をブラウズできるようなインタフェースが良いのではという提案を受けた。またそのように可視化手法を改善することにより、モデリングもさらに容易になるだろうとのコメントも得た。

6. まとめ

3次元心臓モデル内のボリューム上の心筋線維走向データを、スケッチによりモデリングするシステムを提案した。ユーザが描いたストロークを制約として表面と内部の2段階で Laplacian smoothing を適用することで、ボリューム上の連続的なベクトル場を得ることができる。また一人の医師にプロトタイプを試用してもらい、その有用性および課題点を確認した。今後も引き続きこの研究に取り組んでいく予定である。

謝辞

ユーザスタディにご協力いただいた滋賀医科大学循環器内科医の芦原貴司博士に深く感謝する。本研究は文部科学省科学研究費(基礎研究(B)18300157)およびリーディングプロジェクト「細胞・生体機能シミュレーションプロジェクト」の助成を受けたものである。

文 献

- [1] T. Ashihara et al., Breakthrough waves during ventricular fibrillation depend on the degree of rotational anisotropy and the boundary conditions: A simulation study. *Journal of Cardiovascular Electrophysiology*, vol.12, no.3, pp.312-322, 2001.
- [2] T. A. Davis, A column pre-ordering strategy for the unsymmetric-pattern multifrontal method, *ACM Trans. Math. Softw.*, vol.30, no.2, pp.165-195, 2004.
- [3] M. Fisher et al., Design of tangent vector fields, *Proc. SIGGRAPH'07*, 2007.
- [4] R. Haraguchi et al., Electrophysiological heart simulator equipped with sketchy 3-d modeling, in *Complex Medical Engineering*, ed. J. L. Wu et al., pp.107-116, Springer, 2007.
- [5] H. Hauser et al., State-of-the-art report 2002 in flow visualization, TR-VRVis-2002-003, VRVis, 2002.
- [6] K. Nakazawa et al., Computational analysis and visualization of spiral wave reentry in a virtual heart model, in *Clinical Application of Computational Mechanics to the Cardiovascular System*, ed. T. Yamaguchi, pp.217-241, Springer Japan, 2000.
- [7] O. Sorkine et al., Laplacian surface editing. *Proc. Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp.175-184, 2004.