

臓器手術ナビゲータやシミュレータにおけるメス誘導ナビゲーションアルゴリズムとその提示に関する研究

| | |
|--------|---|
| 著者 | 國居 貴浩 |
| 学位名 | 博士(情報学) |
| 学位授与機関 | 大阪電気通信大学 |
| 学位授与年度 | 2022 |
| 学位授与番号 | 34412乙第16号 |
| URL | http://doi.org/10.57427/00000319 |



博士学位論文

題 目

臓器手術ナビゲータやシミュレータにおけるメス誘導ナビゲーションアルゴリズム
とその提示に関する研究

担当指導教員名 登尾啓史

申請年月日 2023年1月24日

申請者専攻名 総合情報学専攻

学 生 番 号 _____

氏 名 國居貴浩

大阪電気通信大学大学院

| | |
|---------------------------------------|----|
| 第1章 はじめに | 1 |
| 1-1 研究の背景 | 1 |
| 1-2 研究の経緯 | 3 |
| 1-3 研究の概要 | 3 |
| 1-4 本論文の構成 | 7 |
| 第2章 医療事故防止システムの開発 | 8 |
| 2-1 仮想臓器を構成する 3D メッシュ群 | 8 |
| 2-2 メス先が危険領域に接近した場合に警告音を鳴らす過程 | 9 |
| 2-3 GPGPU で深度バッファを使った距離計測 | 12 |
| 2-3-1 深度バッファ | 13 |
| 2-3-2 深度バッファの深度値と画素の位置を使った 3D 空間座標の復元 | 15 |
| 2-3-3 GPGPU による深度バッファの評価 | 18 |
| 2-3-4 CPU と GPGPU の速度比較 | 19 |
| 2-4 まとめ | 20 |
| 第3章 切削経路探索アルゴリズム | 21 |
| 3-1 手術ナビゲーションアルゴリズム | 21 |
| 3-2 ポテンシャルベース法 | 21 |
| 3-2-1 障害物からの斥力関数 | 24 |
| 3-2-2 腫瘍への引力関数 | 25 |
| 3-2-3 腫瘍からの斥力関数 | 26 |
| 3-2-4 メス先に発生する力の大きさと方向 | 27 |
| 3-2-5 ポテンシャル関数パラメータ値の調整 | 29 |
| 3-2-6 ポテンシャル関数パラメータ値の考察 | 32 |
| 3-3 まとめ | 33 |
| 第4章 センサーベース法 | 34 |
| 4-1 ポテンシャルベース法の懸念点 | 34 |
| 4-2 センサーベース法の経路探索ステップ | 34 |
| 4-3 センサーイメージの生成法 | 38 |
| 4-4 センサーベースでの経路探索実装について | 40 |
| 4-5 ポテンシャルベース法とセンサーベース法の評価 | 43 |
| 4-6 ポテンシャルベース法とセンサーベース法の併用 | 47 |
| 4-7 まとめ | 47 |
| 第5章 ボクセルベース法 | 48 |
| 5-1 境界面切削への課題とその対応アルゴリズム | 48 |
| 5-2 ボクセル群の生成 | 48 |
| 5-3 ボクセル階層間の移動 | 49 |

| | |
|-------------------------|----|
| 5-4 ボクセルベース法の経路探索ステップ詳細 | 51 |
| 5-5 ボクセルの濃度測定方法 | 52 |
| 5-6 ボクセルベース法の評価 | 55 |
| 5-7 DICOM ボリューム | 58 |
| 5-8 DICOM ボリュームの利用と表示 | 58 |
| 5-9 レイマーチング法 | 59 |
| 5-10 疑似法線 | 60 |
| 5-11 まとめ | 64 |
| | |
| 第6章 DICOM ボリュームの複合表示 | 65 |
| 6-1 DICOM の自動セグメンテーション | 65 |
| 6-2 2つの DICOM の利用 | 66 |
| 6-3 2つの DICOM の位置合わせ | 67 |
| 6-4 レイマーチング法の対応 | 68 |
| 6-5 まとめ | 68 |
| | |
| 第7章 おわりに | 70 |
| 7-1 成果と考察 | 70 |
| 7-2 課題 | 70 |
| | |
| 謝辞 | 72 |
| | |
| 参考文献 | 73 |

第1章 はじめに

この章では、本研究の背景や研究の概要について、経緯を振り返りつつ説明した後、本論文の構成を説明する。

1-1 研究の背景

肝臓腫瘍は、周囲に炎症をおこしたり、黄疸を引き起こしたりと、緊急処置を必要とすることがある疾患である。対応処置としては、肝臓を切削して腫瘍を切除することになる。この手術が比較的難しいとされる理由は、肝臓という臓器の内部構造にある。流入血管として肝動脈、消化管からの静脈血としての門脈、流出血管の肝静脈、また胆汁を運ぶ肝内胆管というように、生命維持に重要な脈管が交錯するように走行している。肝臓の切削では、これらの微細な脈管構造を把握して、切り進む中で無数にでてくる脈管を正確に処理しながら進めていかなければならない。血管を傷つけることで大出血を引き起こしてはならないし、腫瘍等の切削すべき部位と、そうでない部位を的確に区別して切除する必要がある。

しかし、肝臓が内部に持つ複雑な脈管構造は、メスを使って切り開くまでは目視不能である。メス先を切り込む前に考慮すべき脈管構造が、メス先を切り込むまでわからない。この問題は肝臓の手術に限ったものではない。そのため、近年では、手術の際に腫瘍の位置や人体の内部構造をできるだけ正確に術者に伝えるために、手術用ナビゲータに関する研究が盛んに行われている。

手術用ナビゲータが外科医に伝える体内情報には、患者本人の体内を断層撮影した CT¹ や MRI² といった DICOM³ が利用される。例えば肝臓手術では、患者の体への負担の少なさ、術後の回復の早さから、腹腔鏡を使った手術がよく行われる。この手術では、患者の腹部に数か所、穴を開け、そこから複数の操作用直管を差し込み、管の先に付けたカメラのライブ画像を見ながら、別の管の先に付いた鉗子やハサミを操作して切削を行う。ナビゲータでは、このライブ画像に、術前に撮影した CT や MRI を合成したりする。

画像の合成方法は様々で、単に画面を分割して並べたりするものから、手術室に超音波診断装置を持ち込み、プローブから得られた超音波画像と、プローブの位置と角度に対応した断面像を、CT や MRI のボリュームデータからリアルタイムに再構成して表示を行うものもある [1]。CT や MRI のボリュームデータとは、複数の連続した断層画像を重ね、2次元の画像を3次元に拡張したものを指す。プローブの位置と角度の検出には、磁気位置検出ユニットを利

¹ CT (Computed Tomography) : 患者の周囲から X 線をあて、体の中の吸収率の違いをコンピュータで処理して断層画像にしたもの。放射線被曝による放射線障害を考慮する必要がある。検査の目的によっては、造影剤を使用する場合がある。

² MRI (Magnetic Resonance Imaging) : 核磁気共鳴現象を利用して物体を走査し、コンピュータで処理して断層画像にしたもの。生体組織別の画像のコントラストが CT よりも高い。生体が高磁場にさらされるため、ペースメーカーや人工内耳などは電磁誘導により発生する誘導電流によって故障してしまう。そのため、それらを埋め込んだ患者は撮影できない。身に着けた金属製品は取り外す必要がある。検査の目的によっては、造影剤を使用する場合がある。

³ DICOM (Digital Imaging and Communications in Medicine) : 医用画像システム、医療情報システムなどの間で、デジタル画像データや関連する診療データをやり取りするために定められた国際標準規格。

用している。[1]で合成された画像は、超音波画像に合わせた 2D 断面画像だが、CT や MRI のボリュームデータを元に、コンピュータ上で人体組織の立体像を再構成して表示する場合もある。例えば肝臓なら、肝臓の中を通る血管群といったものが立体物化され、PC モニタ上で、任意の角度や位置から眺められるようになっている。このようなコンピュータ上の人体組織の立体物を、本論文では仮想臓器と呼ぶが、先にあげた腹腔鏡手術でのナビゲーションで、この仮想臓器の表示をライブ映像に合成する研究もある[2]。また、ダビンチという腹腔鏡手術用ロボットでは、外科医は立体視可能なステレオモニタを覗きながら、ロボットのアームを操作し手術を行うが、このステレオモニタのライブ画像に仮想臓器を映し込ませるものもある[3]。

また、そのような画像表示を必要としないものもある。著者が加わった医療事故防止システムの開発[4]では、手術中の肝臓の危険領域にメスが入ったかどうかを音で知らせるようにしていた。この研究では、特定マーカの位置計測装置や深度センサを用いて、実物の肝臓やメスの位置や姿勢とコンピュータ内の仮想の肝臓やメス先の位置を比較して、危険領域判定を行っている。

以上のように、ほとんどが視覚や聴覚を使うナビゲーションにおいて、手術用ロボットアームが持つ触覚機能を利用し、対象部位との距離間に応じた触覚を指先に伝える研究[5]もあるが、外科医の指先を直接引っ張るようなナビゲーションの研究までは見あたらなかった。そもそも、実際の手術で、外科医の指を誘導するような装置自体が見当たらない。しかし、プログラム制御で、グリップ部に任意方向の力を発生させることができる触覚デバイス自体は存在する。そこで著者は、以前開発に加わった、上述の触覚デバイスが利用可能な肝臓切除手術シミュレータ[6]に、外科医の指を切削経路に誘導するナビゲーションシステムを組み込むことにした。この手術ナビゲーションシステムの中で特に独創的なのは、仮想臓器と実臓器の視察できる表面範囲を GPU の Z-buffer である仮想デプス画像とデプスカメラから撮像した現実デプス画像をマッチングさせ、仮想臓器と実臓器の位置・姿勢合わせをする点である[7]。これは、臓器全体を点群表示し、移動前と移動後の点群をマッチングさせる Iterative closest point (ICP)[8]でそれらの位置と姿勢を合わせるのと比較して、臓器の一部が見えればそこだけで高速にマッチングが取れる点で新規性や独創性が高い。

このようなナビゲーション付き手術シミュレータがあると、術前の手術計画作成時の検討や、外科医を目指す学生の独習にも利用できる。また、通信機能を組み込めば、教師が学生へ遠隔指導するといった利用も可能となる。そのため、肝臓手術ナビゲータのシステム開発から肝臓手術シミュレータも含めた仮想臓器でのナビゲーションアルゴリズムに興味に移り、現在は DICOM 臓器の提示を平面モニタから没入感体感 HMD へ移す研究を進めている。本論文は、このナビゲーションやシミュレーションで使う経路探索アルゴリズムの開発、およびその経路を DICOM 臓器で提示する方式の拡張が主題となっている。

1-2 研究の経緯

時間軸的には、1-1 で述べた肝臓の悪性腫瘍を切除する手術ナビゲータの開発から始まった。著者は、このシミュレータの開発において、GPGPU⁴で深度バッファ⁵を利用した仮想臓器の変形処理を担当した。そこで、この GPGPU と深度バッファについて、第2章で説明する。これに、1-1 で述べた医療事故防止システムの開発が続く。著者が担当したのは、仮想臓器の危険領域とメス先との距離測定で、こちらも GPGPU で深度バッファを利用している。

その後、切削方向ナビゲーション付き手術シミュレータの開発 [9]に進み、適切な切削方向を誘導するナビゲーションアルゴリズムを、上述の手術シミュレータに組み込んだ。この時、ナビゲーションを行うための切削経路探索アルゴリズムとして、ロボットの移動経路探索で使われる、ポテンシャルベース法[10]に基づいたアルゴリズムを提案している。そして、ポテンシャルベース法で懸念される問題点对応のために、同じくロボットの移動経路探索で使われるセンサーベース法[11]に基づくアルゴリズムを新たに提案し、2つの方式の評価を行い、切削経路探索アルゴリズムの安定性をはかった。ポテンシャルベース法については第3章で、センサーベース法については第4章で説明する。

その後も、ナビゲーション付き手術シミュレータの開発を続け、面切削を視野に入れた切削経路探索アルゴリズムとして、ボクセルベース法[12][13]を提案した。ボクセルベース法は3Dメッシュを必要とせず DICOM を直接利用する。それに合わせ、手術シミュレータでの仮想臓器表示も、DICOM を直接表示するものに変更している。3Dメッシュは、三角形平面で構成された多面体を指し、ボクセルベース法の研究を始めるまで、本研究の手術シミュレータで、仮想臓器の基本構造として使っていたもの。ボクセルベース法については第5章で説明する。

現在は、手術シミュレータを没入型装置に対応[14]させつつ、DICOM の直接表示の拡張を研究している。DICOM の直接表示の拡張については第6章で説明する。

1-3 研究の概要

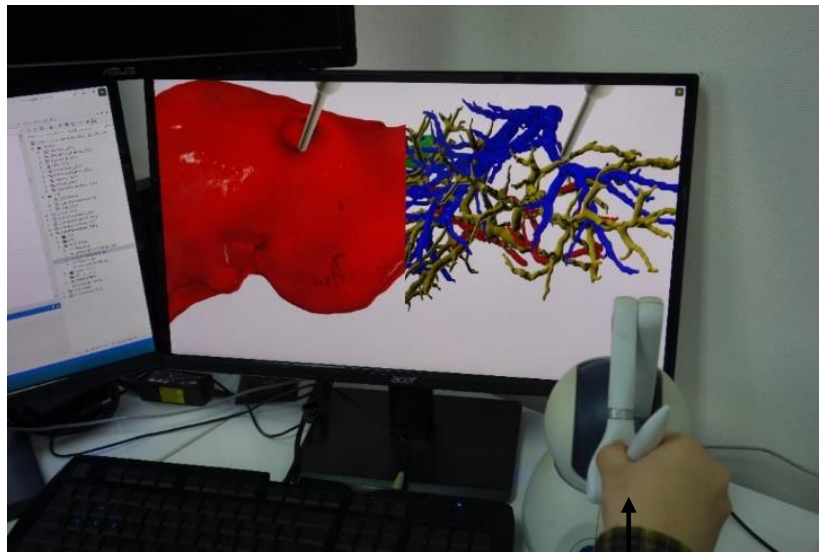
各研究の概要は以下の通り。

肝臓切除手術シミュレータの開発

この研究では、PC モニタに表示される 3D 仮想空間内に肝臓とメスを表示し、触覚デバイスを用いて、仮想空間内を自由に動くメス先で、肝臓を切削できる手術シミュレータ(図 1-1)を開発した。触覚デバイスとは、グリップの 3D 空間内の動きを PC に伝えることができ、フォースフィードバックと呼ばれる、グリップの部分に任意方向の力を返す機能を持つ装置である。

⁴ GPGPU (General Purpose computing on Graphics Processing Units): GPU による汎用計算。画像処理用に用意された GPU を、画像処理以外の目的に応用する技術。

⁵ 3D 多面体を 2D 画像として表示する際に、3D 多面体を構成する平面の前後関係を調整するために作られる情報バッファ。



触覚デバイスによるメス先位置の指定

図 1-1 肝臓切除手術シミュレータ.

仮想臓器である肝臓は、動脈、静脈、リンパ管、腫瘍、肝臓表面領域で分けられ、それぞれが 3D メッシュで構成されている。これらの 3D メッシュは、患者を撮影した DICOM からセグメント化し抽出される。専用アプリケーションを使い DICOM2 次元断層画像情報について、血管部、腫瘍部、リンパ管部というように領域分割することをセグメント化と呼ぶ。セグメント化された領域から抽出される 3D メッシュ群は、三角形平面で構成される STL⁶多面体として提供される。著者は、このシミュレータの開発において、3D メッシュ群で構成される仮想肝臓の「表面を押された時の弾性表現」といった、視覚的なリアリティを持たせるための変形を担当した。この時の変形には、GPGPU で深度バッファを利用している。肝臓 3D メッシュを 2D 画像として描画することで、深度バッファを生成し、深度バッファという単純な 2 次元配列で、形状変形や反発力計算などを行った。著者の担当部ではないが、その後マーチングキューブ法 [15]を用いて、深度バッファから肝臓 3D モデルを再構築している。これらの処理は GPGPU の並列処理により高速に実行され、操作へのリアルタイムな反応を保証している。

仮想臓器の変形や切削の詳細は、論文[6]を参照してもらおうこととし、本論文では割愛する。なお、GPGPU、深度バッファについては第2章で説明する。

医療事故防止システムの開発

この研究では、肝臓腫瘍手術における医療事故を、未然に防止するシステムの開発を行っている。手術中に、メス先が危険領域に接近した場合に、警告音を鳴らして外科医に注意するようにした。

⁶ STL (Standard Triangulated Language) : 三次元形状を表現するデータを保存するファイルフォーマットのひとつである。

術前に撮影した患者自身の DICOM から、仮想臓器として肝臓 3D メッシュを作成しておき、コンピュータの仮想空間で利用する。そのために、最初に手術台上の実空間を測定し、肝臓 3D メッシュを配置した仮想空間と対応付けるようにした。そして手術中は、逐次、実空間のメスや肝臓の位置・姿勢を計測し、仮想空間のメスと肝臓 3D メッシュを連動させることで、コンピュータの仮想空間で、メス先が肝臓の危険領域に入ったかどうかを判定している(図 1-2)。

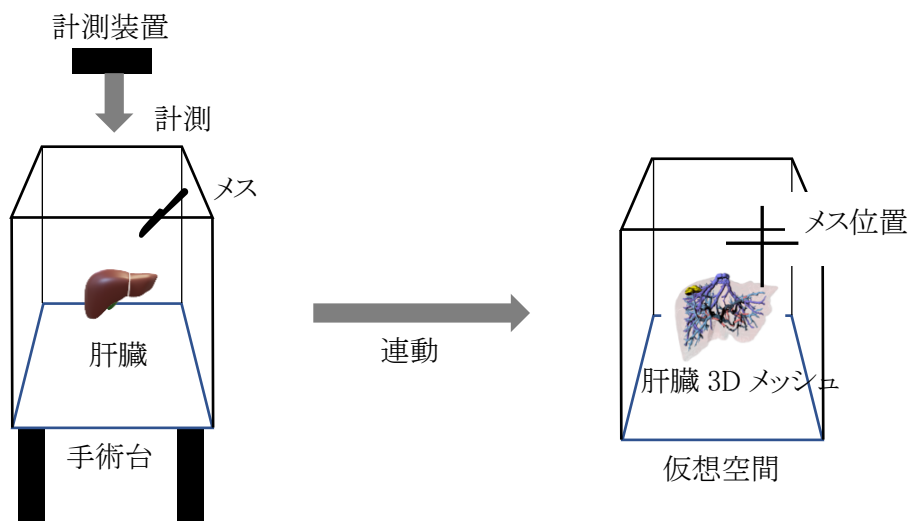


図 1-2 実空間と仮想空間の連動.

メス先位置、姿勢の推定には特定マーカの位置計測装置を利用した。肝臓の位置、姿勢の推定には深度センサを利用した。実際の肝臓の上に、人工的なマーカを配置すれば、肝臓の位置、姿勢、形状を正確に検知することはできる。しかし、手術中、肝臓は何らかの損傷を受けることになるので、そこにマーカを貼り付けることは望ましくない。また、位置、姿勢、形状を計算するためには、膨大な数のマーカ数を必要とし時間もかかる [16] [17] [18] [19] [20] [21] [22]。この点を解決するために、深度センサを用いて、外科医に操作される実肝臓の深度画像を撮影し、仮想肝臓側で用意した深度画像を比較し、位置合わせを行なっている。ここでも GPGPU を使い効率的に比較している。また、こちらの計測装置は光学カメラであり、[1]のように磁気位置検出ユニットを使わないので、磁場の影響を考慮する必要はない。

著者が担当したのは、仮想臓器の危険領域とメス先との距離測定で、GPGPU で深度バッファを利用して測定している。深度センサを用いた位置推定や実肝臓、仮想肝臓の連動については論文[4]を参照してもらうことにし、本論文では割愛する。著者が担当した 3D モデルとメス先との距離測定については第2章で説明する。

切削方向ナビゲーション付き手術シミュレータの開発

この研究[9]では、腫瘍切除シミュレータに、切削の際、適切な経路をナビゲーションする仕組みを追加した(図 1-3)。

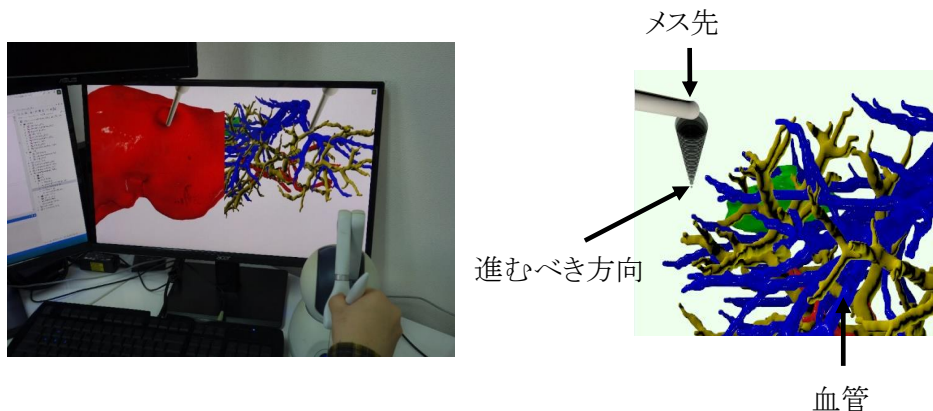


図 1-3 ナビゲーションシステムを組み込んだ手術シミュレータ.

手術対象患者の DICOM から作成した肝臓 3D メッシュを使い、切削のシミュレーションをおこなう。その際、触覚デバイスのフォースフィードバックを使い、切開口から腫瘍付近まで、血管などを傷つけず、メス先を安全に誘導するようにした。誘導するための経路探索アルゴリズムには、ポテンシャルベースのアルゴリズムを提案している。提案したポテンシャルベース法では、血管と腫瘍にポテンシャル関数を適用し、切削経路を案内する力を計算する。計算された、切削経路を案内する力は、触覚デバイスのフォースフィードバックを用いて、外科医の指先にかかる力として提示している。この誘導に従いメス先を動かせば、安全かつ確実に血管群を通過し、腫瘍の切除手術を行うことができる。画面に切削方向の表示も行なっている。

手術シミュレータに加えた、ナビゲーションシステムやポテンシャルベース法については第3章で説明する。

ポテンシャルベース法とセンサーベース法の併用

この研究では、ポテンシャルベース法で予測される、引力斥力の影響により、2点間の往復を繰り返すデッドロック状態を回避するためセンサーベースの切削経路探索アルゴリズムを提案した。センサーベースでは、血管等の障害物を、深度バッファを用いて感知し避けながら腫瘍に進むようにしている。そして2つの方法それぞれの特性を比較研究[23] [24]した。

センサーベース法と、2つの切削経路探索アルゴリズムの比較については第4章で説明する。

ボクセルベース法を使った手術シミュレータの開発

この研究では、ある程度、障害物領域を通過できる、ボクセルベース法を提案した。

前述の2つのアルゴリズムは、障害物領域を進入禁止領域として完全に回避する。ボクセルベース法では、生体組織の弾性を利用し、血管等を移動させて直進性を優先させるといったことも、実際の手術では有りうる点を考慮し、ある程度、障害物領域を通過させる。

ボクセルベース法では、メス先位置から、腫瘍までの直線を Z 軸とする 3D 空間を考え、これを格子状に分割した立方体空間の集合を利用する。分割された立方体空間をボクセルと呼ぶが、それぞれのボクセルに障害物の濃度を持たせ、濃度の薄いボクセル間を移動し腫瘍に進む。濃度による判定なので、ポテンシャルベース法、センサーベース法のように、障害物領

域を進入禁止領域として完全に回避することがない。また、ボクセルに濃度を持たせる過程には、直接 DICOM を使うことが可能なので、3D メッシュを必要としない。手術シミュレータの仮想臓器表示側でも、DICOM から直接表示できるようにし、3D メッシュを不要にした。これにより、DICOM をセグメント化し3D メッシュを作成する工程が省略できることになる。

DICOM からの直接表示には、レイマーチング法[25] [26]を使っている。ボクセルに濃度を持たせるところでも、レイマーチング法を使っている。画面側の表示では、仮想臓器の視認性を向上させるために疑似法線[27]を計算し利用している。

ボクセルベース法、レイマーチング法、疑似法線については第5章で説明する。

没入型装置型手術シミュレータの開発

この研究では、手術シミュレータの表示に没入型装置も使えるようにし、メス先といった 3D 空間位置を指定する際の優位性を検証した。DICOM からの直接表示にも改良を加え、CT、MRI といった種類の異なる DICOM を組み合わせる DICOM の複合表示を提案する。

3D 空間位置を指定する際の優位性検証については割愛する。DICOM の複合表示については第6章で説明する。

1 - 4 本論文の構成

本章に続き、2 章では、研究で利用する手術シミュレータや、DICOM、3D メッシュ、GPGPU について説明する。GPGPU については、深度バッファを使った距離測定についても説明する。3 章では手術シミュレータへのナビゲーションシステムの組み込みと、ナビゲーションで使う切削経路探索アルゴリズムのポテンシャル法について説明する。4 章ではセンサーベース法について説明し、ポテンシャル法との比較評価をおこなう。5 章ではボクセル法について説明する。6 章では DICOM の複合表示について説明し、7 章で結論と今後の課題を示す。

第2章 医療事故防止システムの開発

ここでは、最初に、ナビゲーションシステムで利用される仮想臓器について説明する。その後、医療事故防止システムの構成を説明し、どのように実空間の肝臓やメス先位置を推定するかを説明する。そして、著者が担当した、GPGPUによる、深度バッファを使った、仮想空間での臓器とメス先の距離測定について説明する。

2-1 仮想臓器を構成する3Dメッシュ群

本研究で開発したナビゲーションシステムは、メス先が危険領域に接近した場合に警告音を鳴らすようになっている。危険領域の判定には、術前に患者のCT、MRIから作成した肝臓の仮想臓器が利用されている。この仮想臓器は、腫瘍、動脈、静脈、リンパ管別に用意された3Dメッシュで構成される。ここでの3Dメッシュとは、三角形平面で構成された多面体を意味する(図2-1)。三角形平面以外で構成された3Dメッシュも存在するが、本論文では扱わない。

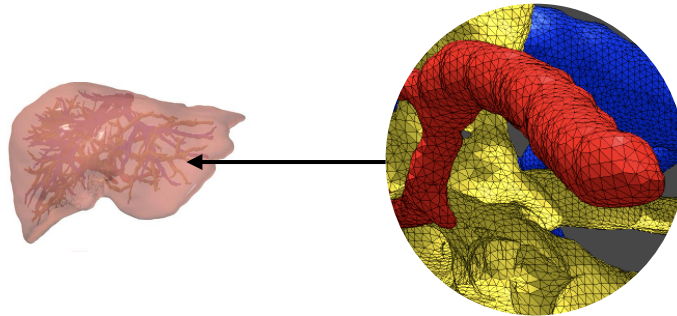


図2-1 3Dメッシュは腫瘍、肝臓の表面、血管、リンパ管別に用意され、これらを合わせて肝臓の仮想臓器とする。

腫瘍、動脈、静脈、リンパ管別に3Dメッシュを作る際は、患者自身のCTやMRIを参照することになる。いずれも複数枚の人体断層画像からなり、PCモニターには白黒の濃淡画像として表示したりする(図2-2)。これらはDICOMと呼ばれる国際標準規格に基づいたファイルで提供される。

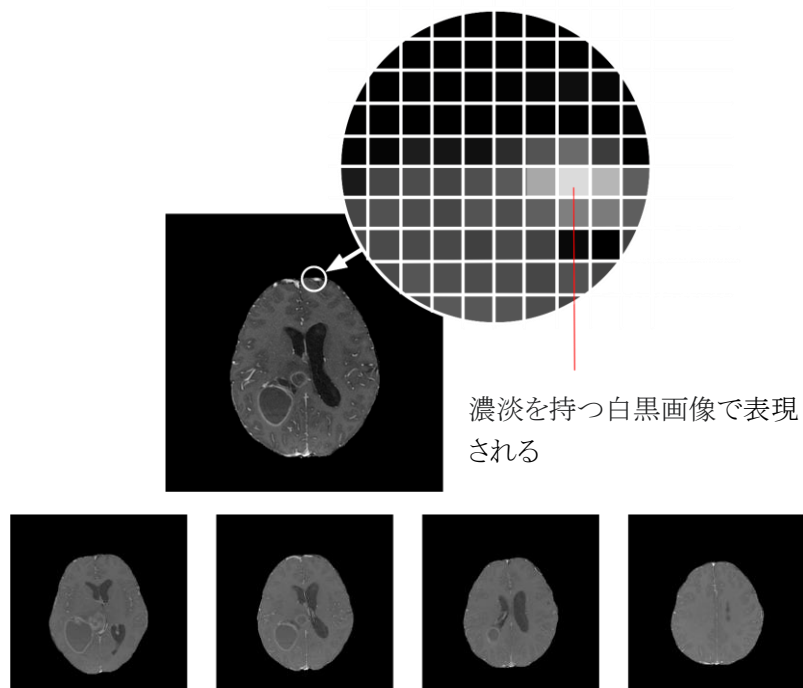


図 2-2 MRI 装置で撮影した頭部の断層画像.

DICOM 断層画像群からの、血管等の 3D メッシュ抽出は熟練者がおこなう。3D Slicer[28]や ITKSnap[29]といった専用アプリケーションを使い、PC モニタを見ながら、断層画像のどの部分が血管で、どの部分が肝臓で、どの部分が腫瘍かを決定していく。この作業は、セグメント化と呼ばれ、依頼された部位ごとに STL 形式のファイルが作成される(図 2-3)。



図 2-3 DICOM 断層画像群からの血管等 3D メッシュ抽出.

STL 形式のファイルは、三角形平面で構成された多面体情報なので、読み込んでそのまま 3D メッシュとして利用できる。3D メッシュにできるなら、STL 形式である必要はないが、医療機関から提供される 3D メッシュは STL 形式が主流となっている。

2-2 メス先が危険領域に接近した場合に警告音を鳴らす過程

提供された 3D メッシュ群は、コンピュータの仮想空間に配置され、同じ空間に配置される仮想のメス先との最近接距離が測定される。そして、その測定値を評価し、メス先が危険領域に

入ったと判定されれば警告音を鳴らすことになる。このシステムを実際の手術現場で利用するには、実空間の肝臓とメス先が、仮想空間の肝臓とメス先に連動する必要がある。

メス先については、末尾にマーカを装着したメスと、マーカを追跡するセンサを用意して、位置や向きを追跡するようにしている(図 2-4)。マーカ追跡センサは、複数のマーカをセンサで追跡し、その 3 次元位置をマーカ別に検出している。メスの末尾に付けた、複数のマーカとメス先の相対位置をあらかじめ測定しておくことで、検出されたマーカ群の位置から、メス先の位置や姿勢を推定している。



マーカを装着したメス



追跡センサ

図 2-4 実験で使用したマーカ装着済みのメス代替物(手術では同じマーカを実際のメスに付けておこなう)と追跡センサ。

肝臓については、手術台全域の深度を測定する深度センサを用意した(図 2-5)。

1-2 で述べたように、肝臓は手術中、何らかの損傷を受けることになるので、マーカを貼り付けることは望ましくない。位置、姿勢、形状を計算するために、膨大な数のマーカ数を必要とし時間もかかる。そのため深度センサが撮影した肝臓の深度画像と、仮想空間の肝臓 3D メッシュの深度画像を重畳させて、位置合わせを行う。肝臓 3D メッシュの深度画像は、コンピュータ内で計算によって作り出す。深度画像の作成法については 2-3 で説明する。

深度センサ マーカー追跡センサ



図 2-5 深度センサは手術台全域を撮影し、マーカー追跡センサはメスに付けたマーカーを追跡する(写真は研究室でおこなった実験時のもの).

実際にメス先が危険領域に接近した場合に、警告音を鳴らすまでの過程は以下のとおり。最初に1度、準備作業をおこない、以後、位置、姿勢推定、警告判定作業を繰り返す。

準備

手術台の上方に深度センサとマーカー追跡センサを配置する。

手術台を深度センサで撮影し、深度センサの深度情報と、コンピュータの仮想空間の座標系を関連づけておく。

深度センサに関連付けた、コンピュータの仮想空間座標系と、マーカー追跡センサの座標系も一致させておく。そのために、手術台に複数のマーカーを置き、深度センサとマーカー追跡センサ、それぞれが測定したマーカー位置情報をもとに、マーカー追跡センサ座標系からコンピュータの仮想空間座標系への変換行列を用意した。この変換行列を使うことで、マーカー追跡センサによって推定したメス先の位置が、深度センサに関連付けた仮想空間の座標系で使えるようになる。

位置、姿勢推定

手術台上の肝臓を深度センサで計測する。深度センサの深度画像と、仮想空間の肝臓3Dメッシュの深度画像を対応づけて肝臓3Dメッシュの位置、姿勢を実肝臓と連動させる。加えて、マーカー追跡センサが追跡しているマーカー位置から、メス先の位置・姿勢を推定する(図2-6)。

警告判定

血管やリンパ管3Dメッシュを危険領域とみなし、メス先との最近接距離を測定し、危険な距離と判断したら警告音を鳴らす。

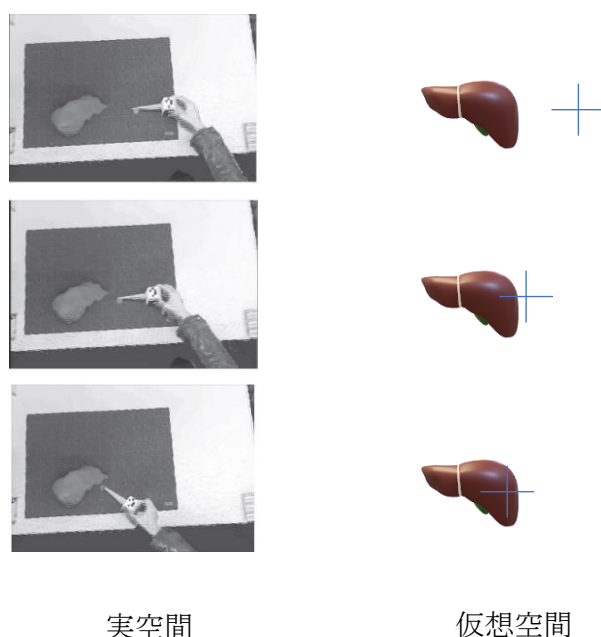


図2-6 実空間の肝臓、メス先と、仮想空間の肝臓、メス先を連動させる
(写真は研究室でおこなった実験時のもの)。

深度センサの深度画像と、仮想空間の肝臓3Dメッシュの深度画像を重畳させる方法、マーカーの位置から、メス先の位置、姿勢を推定する方法については、論文[4]を参照してもらうことにし、ここでは割愛する。

2-3 GPGPUで深度バッファを使った距離計測

本研究で危険領域とみなす“血管やリンパ管の3Dメッシュとメス先位置との最近接距離”は計算可能である。3Dメッシュを構成する三角形平面と、メス先位置との距離を、平面の各頂点群から数学的に計算すればよい。しかし、3Dメッシュを構成する三角形平面群は、今回使った肝臓の動脈で数十万枚となり、1つ1つメス先位置との距離を計算して、その中で最も近い距離を調べているのは、40ms近くの処理時間になってしまう。本研究の警告システムでは、警告判定の周期を16Hzとしたので、1回の判定に使えるのは60msであり、これでは使える時

間の 2/3 を距離測定だけで使ってしまう、メス先や肝臓の位置推定の処理時間が圧迫されてしまう。そのため距離測定処理を高速化する必要がある。空間分割等で、距離計算の対象面を絞り込む手法もあるが、本研究では、GPGPU で深度バッファを利用する手法を提案した。

この手法では、3D 描画用 API で 3D メッシュを 2D 画像に描画させ、その時に作成される深度バッファを利用する。1章から登場している深度画像とは、この深度バッファの事を指す。

本研究では、3D 描画用 API 群として DirectX を利用したが、OpenGL、Vulkan、Metal といった別の 3D 描画用 API 群でも、深度バッファは利用できる。

2-3-1 深度バッファ

3D メッシュは、そのままでは、3次元立体を表現する座標数値の集まりでしかない。これを人間に、目の前の光景として認識させるには、3D メッシュを眺めている位置、眺めている方向、眺めている視野角などを設定し、計算により 2D 画像として描き出す必要がある。

この描き出し作業は、現実世界での、カメラで目の前の光景を写し取る作業に例えることができるので、以後、仮想カメラでの撮影と表現する。また、上述の 3D メッシュを眺めている位置、眺めている方向、眺めている視野角等の指定は、仮想カメラへの設定として説明する。

この仮想カメラでの撮影では、GPU が 3D メッシュの三角形平面群を 2D 画像として描き出していく。三角形平面を順に書き出していく時、三角形の向きや位置、大きさによっては、他の三角形と重なる部分が出てくる。その場合、各平面の前後関係を、正しく 2D 画像に描き出すには、2D 画像の画素ごとに、どちらの平面がよりカメラに近いかが判定できなければならない。深度バッファはそのために用意され、結果、画素ごとの深度の集まりとなる。例えば、横 200 画素、縦 100 画素の 2D 画像は、列数 200、行数 100 で構成される画素情報の 2 次元配列と捉えることもできるわけだが、その場合、深度バッファも列数 200、行数 100 の深度の 2 次元配列として構成されることになる(図 2-7)。

この深度バッファの深度情報を画素情報と捉えた表現が、深度画像ということになる。

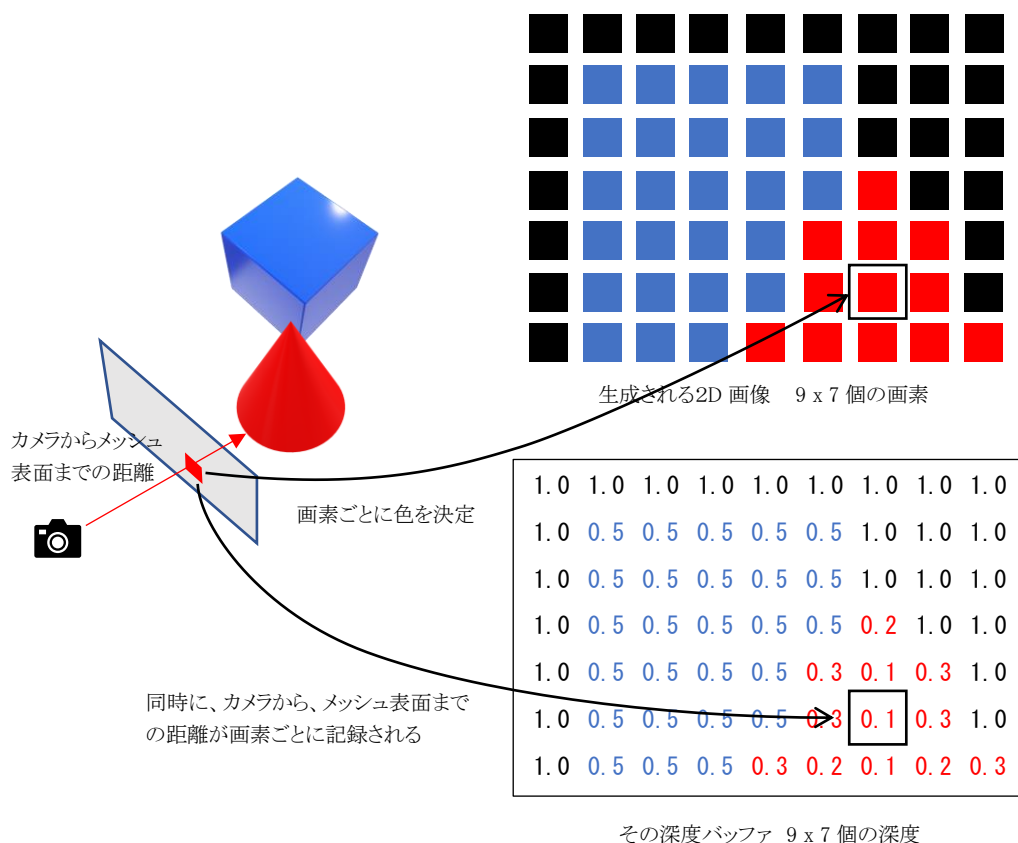


図 2-7 撮影された画像は画素ごとにカメラからの距離を深度情報として深度バッファに持つ。

この深度バッファは、カメラに映り込む、全 3D メッシュの全三角形平面が描画されるまで、絶えず参照、更新される。そして最終的には、2D 画像の画素ごとに、カメラから 3Dメッシュ表面の三角形平面までの距離を記憶したものとなる。

ここで、カメラ位置をメス先位置にして撮影すれば、深度バッファに記録された深度は、メス先からの距離ということになる。例えば、描き出す 2D 画像の画素数を横 100 個、縦 100 個とすれば、10,000 個の深度が出来上がる。この深度バッファから最も浅い深度を選び出せば、カメラの撮影方向限定ではあるが、手前にある 3D メッシュとの最近接距離が確定する。この処理は、数十万とある三角形平面からの距離を計算するより、はるかに負荷の少ない処理となる。

本研究では、メス先から周辺の障害物 3Dメッシュまでの距離を検出するために、メス先をカメラ位置にして、X 軸の±方向、Y 軸の±方向、Z 軸の±方向、合計 6 つの方向にカメラを向けて撮影をおこない、その 6 方向の深度バッファから、もっともメス先に近い距離を、最近接距離としている(図 2-8)。

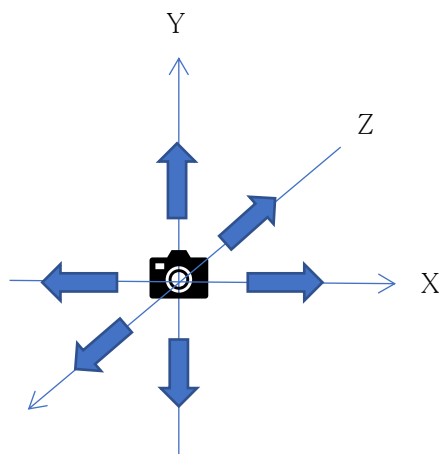


図 2-8 軸に沿った 6 方向にカメラを向けて撮影.

深度バッファに記録されたカメラからの距離も、GPU 側が三角形平面から計算したものなので、三角形平面からの距離計算処理自体が消えてなくなっているわけではない。しかし、計算を担当する GPU は、3D グラフィック表示を高速化するために用意された専用プロセッサなので、CPU を使うより数倍～数十倍速く計算ができる。

以上の説明からもわかるように、この深度バッファを利用する手法は、GPU の性能に依存した手法となっている。そういった制限はあるが、近年頭打ちである CPU の計算能力向上ペースに比べ、GPU の計算能力向上ペースは大きいことから、将来性があるものと考ええる。

2-3-2 深度バッファの深度値と画素の位置を使った 3D 空間座標の復元

深度バッファに記録される深度値は、カメラからの距離そのままではなく、0~1 の範囲に正規化されたものになっている。遠近を判断するだけならそれでよいが、元の 3D 空間での距離が欲しい場合、正規化前の 3D 空間座標に戻す必要がある。その場合、2D 画像の画素の縦横位置と、その画素の深度値を利用することになる。

この 2D 画像上の画素位置とその深度から、3D 空間座標への変換について説明するために、最初に、3D 空間座標から、2D 画像上の画素位置 (以後、2D 座標と呼ぶ) と深度に変換される過程を説明する。

3D 座標から 2D 座標と深度への変換には以下の 2 つの行列が使われる。

ビュー変換行列 V: 仮想空間での 3D 空間座標から、カメラを原点とし、カメラの向きを Z 軸とした 3D 空間座標に変換する行列。

射影変換行列 P: カメラに設定されたパラメータをもとに、3D 空間座標から 2D 座標へ変換する行列。

ビュー変換行列 V を作り上げるには、カメラの位置、カメラの向きが使われる。射影変換行列 P を作り上げるには、カメラに設定された視野角と、nearZ、farZ という深度値が使われる (図 2-9)。GPU は、カメラを原点に、XZ 平面、YZ 平面それぞれ、視野角の範囲で扇状に広がる領域に対し、Z 軸が nearZ から farZ までの範囲を 2D 画像として映しこむ。射影変換行列 P

は、この空間領域を、2D 座標と深度に変換する。与えられたパラメータから、どのようにビュー変換行列や射影変換行列を作るかについては、Direct X の解説ページ [30] など説明されているので、ここでは割愛する。

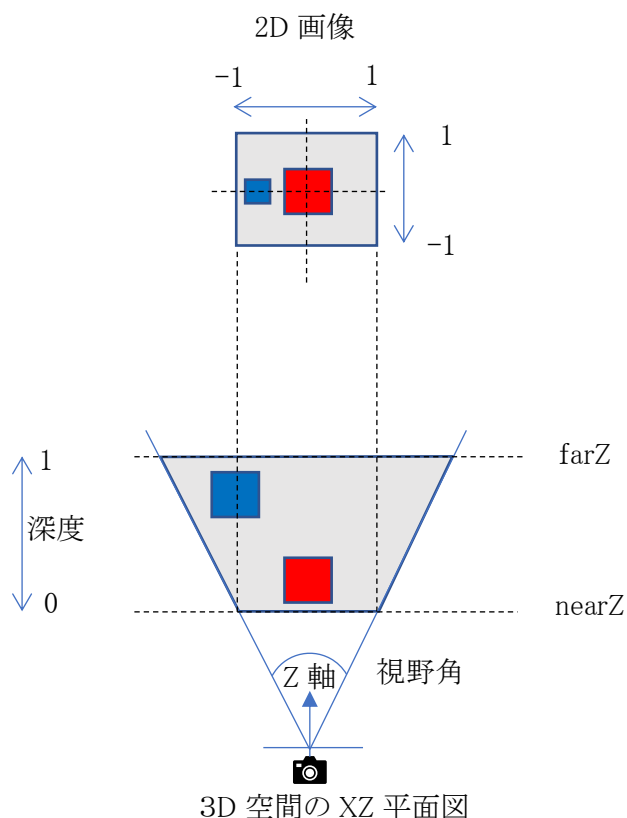


図 2-9 カメラに設定されるパラメータと、撮影される 2D 画像の関係。

行列V、Pはどちらも 4 行 4 列で提供され、仮想空間の 3D 座標 $[x\ y\ z]$ は、同次座標系 $[x\ y\ z\ 1]$ にされてから、式(1)、(2)の順で計算され $[u\ v\ d\ 1]$ となる。ここでは、 u が 2D 画像の X 軸の座標、 v が Y 軸の座標を示し、値は、 $-1\sim 1$ に正規化される(図 2-9)。 d は深度を示し、値は $0\sim 1$ に正規化される。

$$[x\ y\ z\ 1]VP = [x' y' z' w] \quad (1)$$

$$[x' y' z' w] \frac{1}{w} = [u\ v\ d\ 1] \quad (2)$$

2D 画像の 2D 座標 $[u, v]$ と深度 d は、このようにして決まる。そのため、逆に 2D 座標 $[u, v]$ と深度 d から仮想空間の 3D 座標 $[x\ y\ z]$ に戻すには、上記の計算式をもとにして $[u\ v\ d\ 1]$ から $[x\ y\ z\ 1]$ を求める方法を考えればよい。

まず VP の逆行列 $(VP)^{-1}$ を式(1)の両辺にかけると式(3)になる。そして、式(2)を変形して式(3)に当てはめることで式(4)ができる。

$$[x\ y\ z\ 1] = [\acute{x}\ \acute{y}\ \acute{z}\ w](VP)^{-1} \quad (3)$$

$$[x\ y\ z\ 1] = w[u\ v\ d\ 1](VP)^{-1} \quad (4)$$

ここで仮に式(5)とおくと、式(4)は式(6)と書け、式(7)がなりたち、式(6)は式(8)とおける。そして式(8)と式(5)から式(9)が導かれる。

$$[u\ v\ d\ 1](VP)^{-1} = [\acute{u}\ \acute{v}\ \acute{d}\ \acute{w}] \quad (5)$$

$$[x\ y\ z\ 1] = w[\acute{u}\ \acute{v}\ \acute{d}\ \acute{w}] \quad (6)$$

$$1 = w\acute{w} \quad (7)$$

$$[x\ y\ z\ 1] = \frac{1}{\acute{w}}[\acute{u}\ \acute{v}\ \acute{d}\ \acute{w}] \quad (8)$$

$$[x\ y\ z\ 1] = \frac{1}{\acute{w}}[u\ v\ d\ 1](VP)^{-1} \quad (9)$$

これで、VP の逆行列 $(VP)^{-1}$ を用意しておけば、式(9)を使い、正規化された 2D 座標 $[u\ v]$ と深度 d から、元の 3D 座標 $[x\ y\ z]$ が導き出せることになる。

ところで、ここで利用した射影変換行列 P は、カメラのパラメータに透視投影法を指定した場合の行列である。透視投影法は、実際のカメラのように、同じ大きさの物体でも、近くにある方が大きく写り、遠くにある方が小さく写りこむよう描かせたい時に指定する。通常、3Dメッシュを 2D 画像に描かせる場合は、この透視投影法を指定することで自然な画像となる。しかし、3D 描画用 API は、平行投影法という別の投影法も指定できるようになっている。平行投影法は、建築図などのように、遠近に関係なく、同じ大きさなら同じ大きさに描かせたい場合に指定する。その場合、視野角ではなく、写し込みたい X 軸、Y 軸の範囲をカメラに指定する(図 2-10)。

平行投影用の射影変換行列 P は、3D 座標 $[x\ y\ z]$ の x 、 y 座標を、それぞれ X 軸、Y 軸の指定範囲 $width$ 、 $height$ の範囲で $-1\sim 1$ に正規化して 2D 座標 $[u\ v]$ にする。また、平行投影用の射影変換行列 P では、[30]より、式(1)の w が 1 になることが保証されているので、深度は単純に式(10)で正規化されることになる。

$$\frac{z - nearZ}{farZ - nearZ} \quad (10)$$

行列計算が不要となり、計算が単純になるため、本研究ではカメラのパラメータに平行投影を指定するようにした。

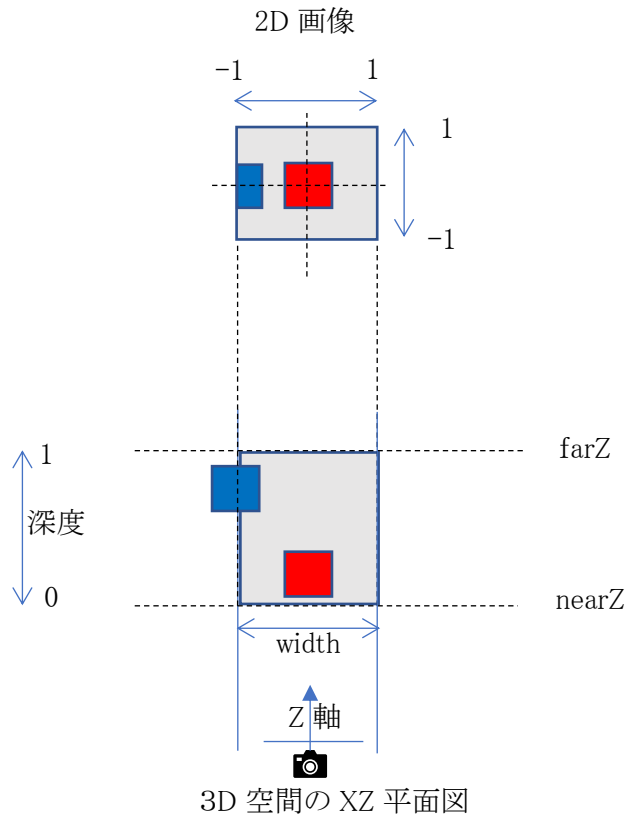


図 2-10 平行投影の場合、視野角ではなく X 軸の範囲 width と図には描かれていないが Y 軸の範囲 height を指定する。

2-3-3 GPGPU による深度バッファの評価

前述した計算で、深度バッファを使って、3D 空間位置を扱うことができるわけだが、この深度バッファが置かれているメモリは GPU 側にあり、一般的に CPU がアクセスできない場所にある。そのため CPU で深度バッファの値を使った計算をしたければ、GPU 上のメモリから CPU 上のメモリへの転送が必要になる。本研究では、深度バッファ自体の転送を避けるためと、GPU の並列処理を期待して、GPU で最近接距離値を計算するようにした。警告音をならずといった操作は CPU 側が行っているため、最近接距離値のメモリ間転送は避けられないが、深度バッファ自体の転送より、はるかに少ない転送量で抑えられる。

GPU は 2000 年頃に、3D メッシュの 2D 画像描画を、プログラマがカスタマイズできるよう専用の言語が提供された。この頃から GPU は、2D 画像生成だけでなく、数値計算に利用されはじめ、これを「GPU による汎用計算」という英文の頭文字から GPGPU (General Purpose Computing on Graphics Processing Units) と呼んでいる。DirectX でも、シェーダ言語と呼ばれる、C 言語に似た言語でプログラムを書けば、GPGPU が可能で、その場合、GPU 側のメモリに置かれた深度バッファに直接アクセスできる。

また、GPGPU では、GPU が持つ数百から数千の演算プロセッサを、並列で動作させることで、より高速な計算ができる。例えば 100×100 で 10,000 個の深度の比較を、単独のプロセッサでループ処理するのではなく、深度ごとにプロセッサを割り当てて並列で処理する方が速い。どれだけ演算プロセッサを割り当てられるかなどの情報は、GPU から API を通して受け取

れるので、それを元に決めることになる。例えば同時に 40 個のプロセッサが使えるなら、10,000 個の画素を 40 個ずつ並列で処理する。このようなスケジューリング機能も API で提供されている。

同じメモリ番地に対し、各プロセッサが同時にアクセスする場合には、注意しなければならない。本研究だと、最近接距離値を決定するために、それまでに見つかった最近接距離値を記憶するメモリは、それぞれのプロセッサが共有しなければならない。そのため、最近接距離値にアクセスする時は排他処理をおこなうようにした。DirectX が提供するシェーダ言語は、メモリ番地のアクセスを排他制御する機能を提供してくれるのでこれを利用した。

2-3-4 CPU と GPGPU の速度比較

最後に論文[31]で測定した CPU と GPGPU の計算速度を提示しておく。この手法は、物体と物体をどちらも高速に短冊群にデジタル化し、それらの短冊の干渉や距離を並列処理で計算するものである。この物体の短冊群への自動変換、および排他処理で並列処理するのは GPU とそのプログラミングで実施している。これらは従来のアルゴリズムを工夫するもの[32], [33], [34] に比べて格段の高速性能を持つところに新規性がある。

ここではその一例として、メス先から肝臓3D メッシュまでの最近接距離を求める計算を、CPU での三角形平面総当たり手法と、GPGPU での深度バッファを使う手法とで比較している。GPGPU では6面の撮影をおこなって最近接距離を求めている。

三角形平面の数をグラフの横軸にとり、肝臓 3D メッシュの三角形平面の数を 357,992 まで単調に増加させた。一方、CPU と GPGPU それぞれの計算時間(単位は ms)をグラフの縦軸に表示した(図 2-11)。

仮想カメラの撮影範囲 X:106mm×Y:106mm
深度バッファサイズ 2048×2048 画素

開発環境:

IDE Microsoft Visual Studio 2012
Direct X 11.0
OS Windows 8.1 Professional 64 ビット
Language C++ / HLSL
GPGPU Direct Compute

CPU スペック:

CPU Intel Core i5-2500K
クロック 3.30GHz
Memory 16GB DDR3

GPU スペック:

VGA GeForce GTX480 CUDA プロセッサコア数 480
グラフィッククロック 700MHz

プロセッサクロック 1401MHz
メモリ 1536MB GDDR5

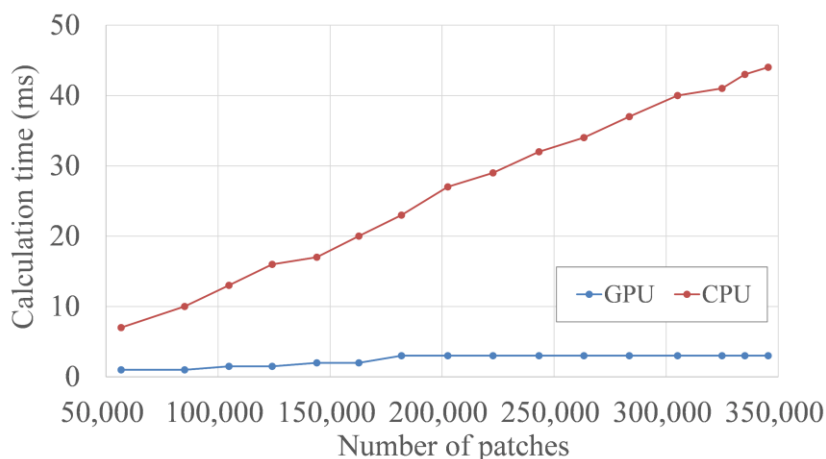


図 2-11 CPU と GPGPU の処理時間比較チャート

30 万枚の三角形平面の場合、CPU が 40ms に対し、GPGPU では 5ms を切っている。前述したように、本研究での危険領域判定周期は 16Hz であり、1 回の更新に避ける時間は 60ms 程度なので、CPU の 40ms では他の処理に影響を与えかねず、3D メッシュをもう少し単純な形状にする必要がある。GPGPU であればもう少し複雑な 3D メッシュも対応可能だろう。

このチャートで平面数が増えると、GPGPU 側も、わずかに処理時間が増加するのは、深度バッファを用意するまでの描画処理と考えていい。三角形平面数の増加に対して、緩やかな上昇で済んでいるのは、GPU 内部で効率的な描画処理が行われているものと判断できる。プロセッサあたりのクロック数の上昇が頭打ちの現在、並列処理でプロセッサを増やすことで高速化する GPU の利用は、大量の三角形平面で構成される仮想臓器を扱う本研究に適していると考えられる。

2-4 まとめ

ここでは、医療事故防止システムの構成を説明した。このシステムでは、深度センサ、マーカー追跡センサにより、肝臓とメス先位置や姿勢を推定する。推定した位置や姿勢で仮想空間に配置した肝臓 3D メッシュとメス先位置を連動させ、メス先と危険領域までの距離測定を行っている。仮想空間でメス先が危険領域に入った時には、実空間のメス先も危険領域に入ったとみなし警告音を鳴らすようにしている。

ここで著者は仮想空間での距離測定を担当し、GPGPU で深度バッファを使い、肝臓 3D メッシュの各三角形とメス先との距離を求める方法を開発した。開発した手法は CPU に比べ 8 倍程度の速度で距離を測定できた。

第3章 切削経路探索アルゴリズム

ここでは、手術シミュレータへのナビゲーションシステムの組み込みを説明し、そこで利用する切削経路探索アルゴリズムとして、ポテンシャルベース法を提案する。

3-1 手術ナビゲーションアルゴリズム

1章で紹介した手術シミュレータは、触覚デバイスを使い、メス先位置を3次元で入力できるようにしている。これに2章のような警告システムを組み込み、音を鳴らすだけでなく、切削経路をナビゲーションするようにすれば、術前に、手術の流れを準備・検討する際のツールとして役立つだろう。そのため、手術シミュレータにナビゲーションアルゴリズムを追加することにした。

本研究で、手術シミュレータに組み込むナビゲーションシステムは、腫瘍以外の血管、リンパ節等の重要器官を避けながら、腫瘍に到達する経路を、医師に提案することを目的としたものとした。そのために、切削経路を決定するためのアルゴリズムや、見つけた経路の効果的な提案方法について研究した。

ナビゲーションでは、まずメス先位置を取得し、次に、メス先位置周辺の患者の仮想臓器を参照し切削経路決定、最後に、決定した切削経路を提案するという作業を繰り返すことになる。2章の警告システムでは、危険域に入った時に音で知らせていたが、今回、手術シミュレータに組み込む切削経路ナビゲーションでは、触覚デバイスのグリップにフォースフィードバックで力を発生させ、見つけた切削経路の方向を提案するようにした。切削経路方向は、表示中の画面にも表示する。

3-2 ポテンシャルベース法

切削経路決定アルゴリズムとしては、ロボットの移動経路探索で使われる、ポテンシャルベース法の応用を提案する。名称はそのままポテンシャルベース法とする。

応用元の[10]では、ポテンシャル関数を複数の障害物に配置し、ロボットがいる地点で、周囲のポテンシャル関数が発生させる引力、斥力を合計し進行方向を決定する。本研究の場合、障害物は血管3Dメッシュやリンパ管3Dメッシュが相当する。枝分かれて複雑に交錯する血管に触れずに、目的地である腫瘍に到達させるには、3Dメッシュが持つ全頂点にポテンシャル関数を置くことになる。数十万になる頂点群にポテンシャル関数を配置し、それらを合計してはリアルタイムの反応は難しい。2章では16Hzの周期で、メス先が危険領域に入るかどうかを判定していたが、本研究でも最低限、その周期は維持したい。

事前に空間分割をするにしても、前処理にはそれなりに時間がかかる。臓器は切削され変形もするので、その都度空間分割をしては前処理とは言えない。空間分割をGPGPUで行うことも考えたが、まずは2章で用意した最近接点測定手法を応用することにした。

経路検索ステップは次のようになる。

まず、メス先への引力ベクトルとして、メス先から腫瘍に向かうベクトルを生成する。

次に、メス先への斥力ベクトルとして、メス先に最も近い障害物 3D メッシュの最近接点からメス先に向かうベクトルを生成する。そして、引力、斥力2つのベクトルを合成したものを、メス先の進むべき方向ベクトルとする(図 3-1)。

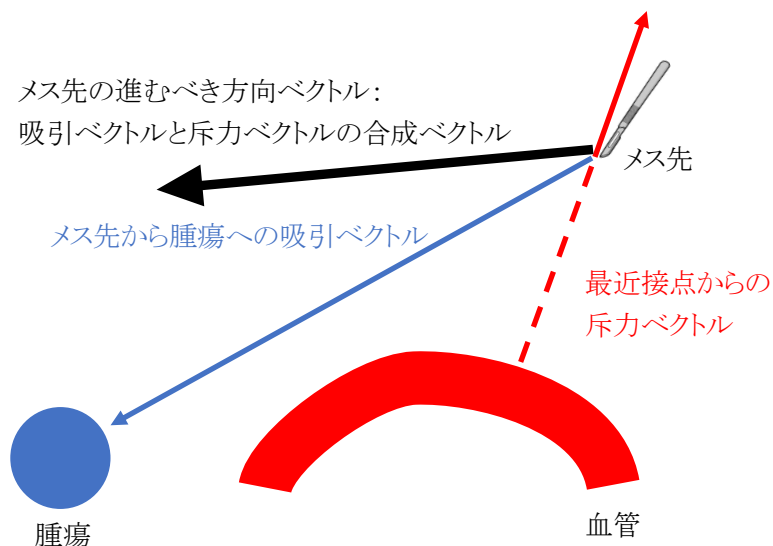


図 3-1 2つの方向ベクトルの生成

メス先の進むべき方向ベクトルが得られたら、微小距離として、0.1mm~2mm 程度(研究で作成した手術シミュレータでは 0.5mm を指定した)だけメス先を進める(図 3-2)。この時、進む先で障害物にぶつかるようなら、ぶつかる手前までに移動量を制限もする。

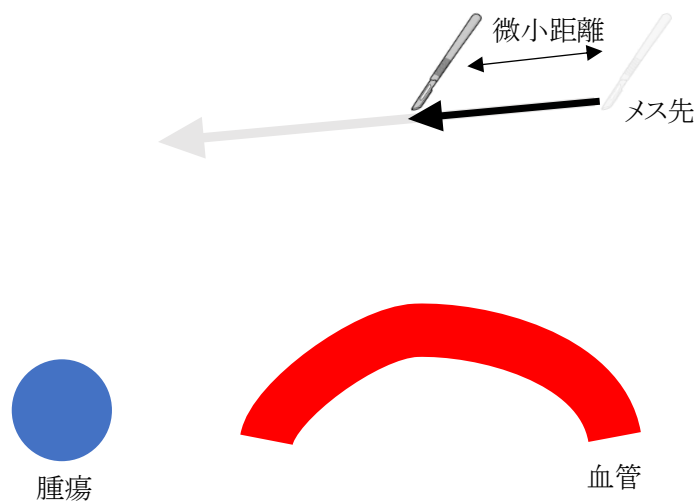


図 3-2 移動ベクトルに従い、微小距離だけ移動

そして新しい位置で同じ計算をおこない、また少しメス先を進める(図 3-3)。これを繰り返し最終的に腫瘍に到達させる。

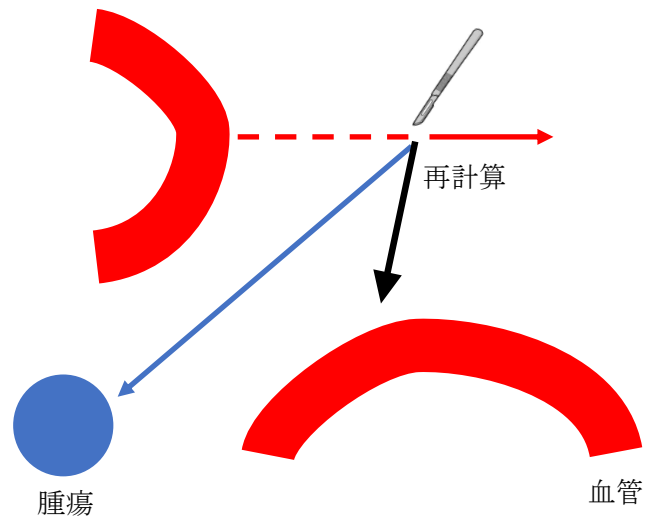


図 3-3 新しい場所で再計算

そのために吸引力、斥力を生み出すポテンシャル関数を設定した。
障害物からの斥力を計算する関数は F_{rv} と定義する。
腫瘍への吸引力を計算する関数は F_{at} と定義する。

3-2-1 障害物からの斥力関数

関数 F_{rv} の値は、障害物表面の、メス先に対する最近接点 P_{vessel} 座標と、メス先点 P 座標をパラメータとし式(11)、(12)を使い決定する。 D_{rv} 、 C_{rv} は定数。

$$n = \min (|P_{vessel} - P|, D_{rv}) / D_{rv} \quad (11)$$

$$F_{rv} = -C_{rv}(1 - n)^2 \quad (12)$$

\min は引数 2 つのうち小さい方を返す関数

n は常に[0.1]内で正規化される

関数 F_{rv} から出力される力は図 3-4 のようになる。

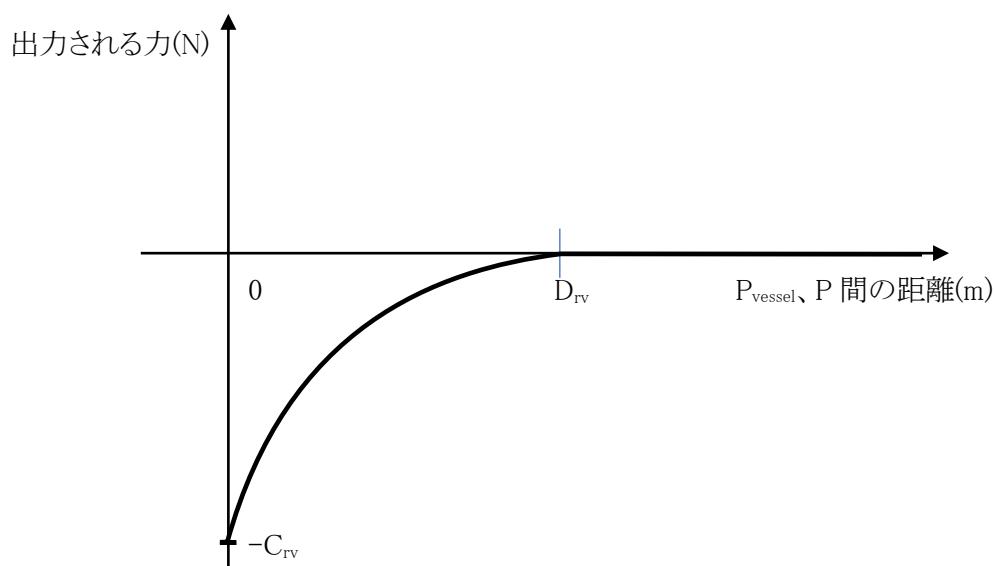


図 3-4 障害物からの斥力関数.

3-2-2 腫瘍への引力関数

関数 F_{at} の値は、腫瘍表面の、メス先に対する最近接点 P_{tumor} 座標と、メス先点 P 座標をパラメータとし式(13)、(14)を使い決定する。 D_{at} 、 C_{at} は定数。

$$n = \min(\max(|P_{tumor} - P| - margin, 0), D_{at}) / D_{at} \quad (13)$$

$$F_{at} = C_{at}(1 - n)^2 \quad (14)$$

関数 F_{at} から出力される力は図 3-5 のようになる。

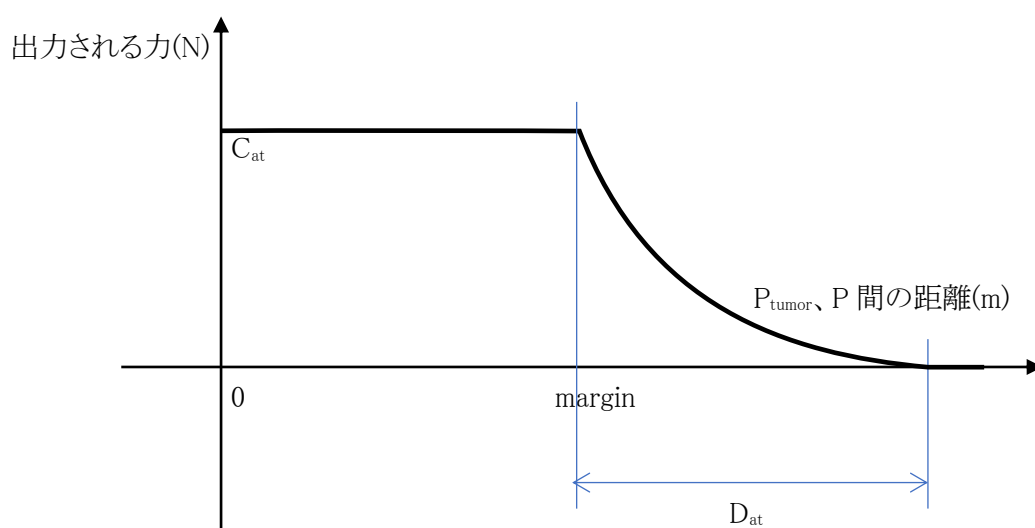


図 3-5 腫瘍への引力関数.

また、腫瘍除去という手術の目的から、基本の 2 関数に加えて、腫瘍に切り込めないよう腫瘍側にも斥力を生成する関数 F_{rt} を定義する。 F_{at} と F_{rt} が釣り合うところでは力が無くなることになるが、ここが腫瘍の切削点であり終着点となる。

3-2-3 腫瘍からの斥力関数

関数 F_{rt} の値は、腫瘍表面の、メス先に対する最近接点 P_{tumor} 座標と、メス先点 P 座標をパラメータとし式(15)、(16)を使い決定する。 D_{rt} 、 C_{rt} は定数。

$$n = \min(\max(\text{margin} - |P_{tumor} - P|, 0), D_{rt}) / D_{rt} \quad (15)$$

$$F_{rt} = -C_{rt}n^2 \quad (16)$$

\max は引数 2 つのうち大きい方を返す関数

関数 F_{rt} から出力される力は図 3-6 のようになる。

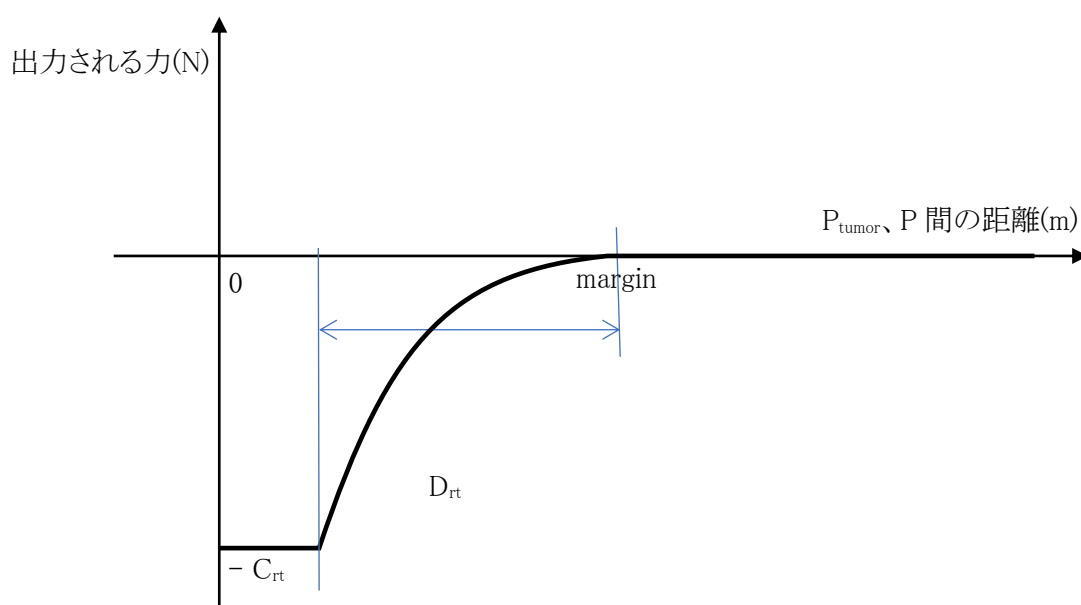


図 3-6 腫瘍からの斥力関数.

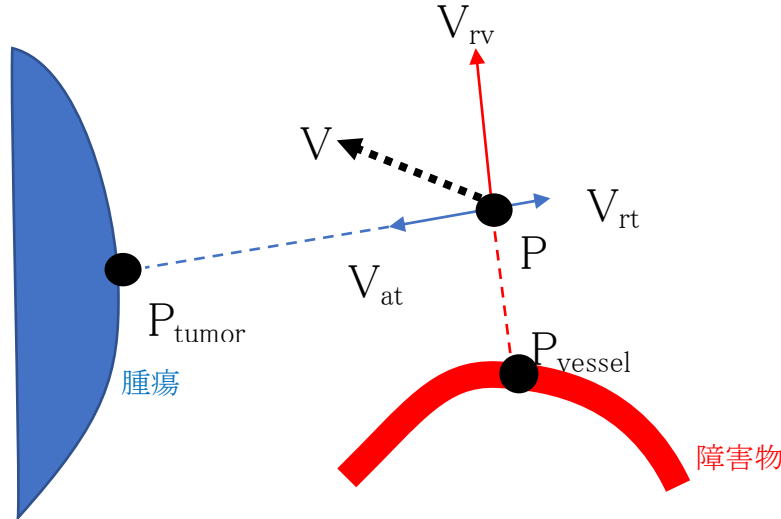
以上 3 つの関数に設定する定数を含め、ポテンシャルベース法は表 3-1 にあけるパラメータで、距離に応じた引力や斥力カーブを調整する。

表 3-1 ポテンシャルベース法のパラメータ

| パラメータ名 | 説明 |
|----------|---------------------------|
| C_{at} | 腫瘍への引力を決める係数 |
| D_{at} | 腫瘍への引力を変化させる距離 |
| C_{rv} | 障害物からの斥力を決める係数 |
| D_{rv} | 障害物からの斥力を変化させる距離 |
| C_{rt} | 腫瘍からの斥力を決める係数 |
| D_{rt} | 腫瘍からの斥力を変化させる距離 |
| margin | 腫瘍表面からの距離で指定する腫瘍と共に取り除く領域 |

3-2-4 メス先に発生する力の大きさと方向

ここでは、**図 3-7** を用いて、メス先に力のベクトル V を発生させる過程を説明する。この力は触覚デバイスにフォースフィードバックするためと、切削経路探索で次の位置に移動するための移動ベクトルに使うために発生させている。発生した V の単位を 1N とし、1/60 秒と掛け合わせ力積として、触覚デバイスの API に渡すことで、デバイスのグリップに、指定した力が発生するようになっている。切削経路探索で、次の位置に移動するための移動ベクトルに使う際は、単位ベクトルにしてから 1 回のステップで移動させる距離を掛け、移動ベクトルとして使っている。



- P メス先の 3D 座標
- P_{vessel} 障害物表面のメス先への最近接点の 3D 座標
- P_{tumor} 腫瘍表面のメス先への最近接点の 3D 座標
- V_{rv} 点 P_{vessel} から点 P への斥力ベクトル
- V_{at} 点 P_{tumor} から点 P へ引力ベクトル
- V_{rt} 点 P_{tumor} から点 P へ斥力ベクトル
- V V_{rv} 、 V_{at} 、 V_{rt} から合成されたベクトル

図 3-7 ポテンシャル関数で作られるベクトル V_{rv} 、 V_{rt} 、 V_{at} とその合成ベクトル V 。

P_{vessel} はメス先 P と障害物の最近接点、 P_{tumor} はメス先 P と腫瘍の最近接点として、P から逐次計算される。

この時も2章で説明したように GPGPU による距離測定を利用している。本研究では、最近接距離ではなく、最近接点の 3D 空間座標を戻すようにした。そのようにして求めた P_{vessel} 、 P_{tumor} 、P をパラメータとし、式(17)、(18)、(19)、(20)を使い V を決定する。

$$\mathbf{V}_{rv} = \text{normalize}(\mathbf{P}_{vessel} - \mathbf{P})\mathbf{F}_{rv} \quad (17)$$

$$\mathbf{V}_{at} = \text{normalize}(\mathbf{P}_{tumor} - \mathbf{P})\mathbf{F}_{at} \quad (18)$$

$$\mathbf{V}_{rt} = \text{normalize}(\mathbf{P}_{tumor} - \mathbf{P})\mathbf{F}_{rt} \quad (19)$$

$$\mathbf{V} = \mathbf{V}_{rv} + \mathbf{V}_{at} + \mathbf{V}_{rt} \quad (20)$$

normalize はベクトル正規化関数

このようにポテンシャルベース法では、引力、斥力の影響を受けメス先移動方向を決めるベクトル場を用意し、その中でメス先を移動させる(図 3-8)。

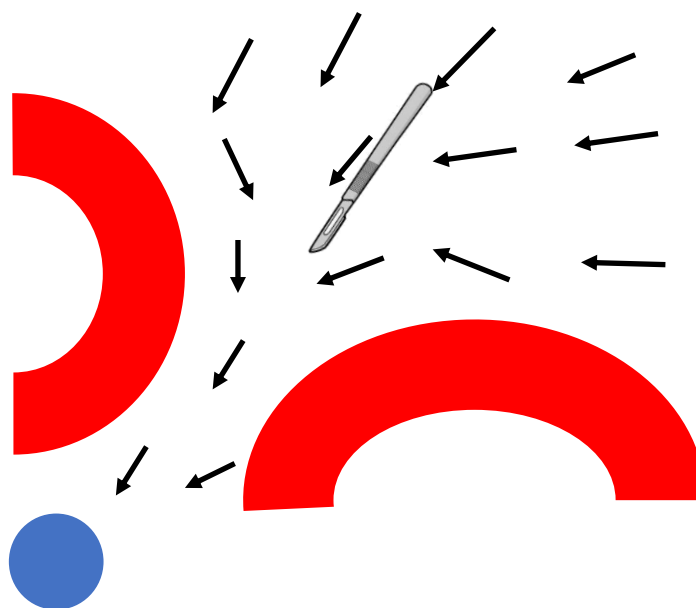


図 3-8 ベクトル場を移動するメス先.

3-2-5 ポテンシャル関数パラメータ値の調整

実用的なパラメータ値を見つけるため、ポテンシャルベース法を手術シミュレータに組み込み、実際にナビゲーションを行い、使用しやすいように徐々にパラメータ値を変更する実験をおこなった。手術シミュレータでは、触覚デバイスでのフォースフィードバックだけでなく、画面上でもフォースフィードバックする方向等、以下の情報を提示し、被験者に誤切除の危険性を知らせるようにした。以後、フォースフィードバックする方向および力のことを手術ナビゲーションベクトルと呼ぶこととする。

- メス先と血管等の障害物の距離に応じ、背景色を緑(安全)→水色→黄色→濃いオレンジ→赤(危険)と変化させる
- メス先端に働いている手術ナビゲーションベクトルを、メス先端に表示する半透明物体の形状で示す(図 3-9)

また、PC のコンソールにメス先と腫瘍の距離等の数値を出力、ログファイルを残すようにして評価した。

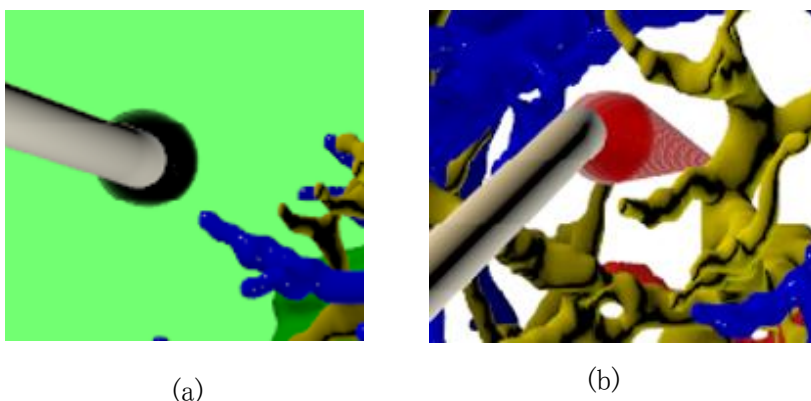


図 3-9 メス先端に働いている手術ナビゲーションベクトルの画面表現(a)球体:力が働いていない状態、(b)円錐:力が働いている状態、尖塔部が力の方向、長さが力の強さ.

この時、調整に使用した代表的なパラメータ値のセットを表 3-2 に示す。表先頭列のセット名は以下の意味を持つ。

- A 初期値(実験を始めるにあたり、妥当と思える値を設定した)
- B 肝臓の手術自体が不可能になる値
- C メス先が腫瘍に近づくことが優先され、血管が傷つくような値
- D 腫瘍の斥力が強すぎる値
- E 最終調整値

表 3-2 実験で使った代表的なパラメータセット

| セット名 | margin [mm] | C_{at} | D_{at} [mm] | C_{rt} | D_{rt} [mm] | C_{rv} | D_{rv} [mm] |
|------|----------------|----------|------------------|----------|------------------|----------|------------------|
| A | 10 | 1.0 | 5 | 2.0 | 5 | 1.0 | 5 |
| B | 12 | 1.5 | 20 | 4.5 | 5 | 2.0 | 51 |
| C | 12 | 1.5 | 20 | 1.0 | 5 | 2.0 | 11 |
| D | 15 | 1.5 | 20 | 6.0 | 5 | 2.0 | 11 |
| E | 12 | 1.5 | 20 | 4.0 | 5 | 2.0 | 11 |

注意)この表では、読みやすさを考え、長さの単位は m ではなく mm とした。実際のパラメータでは m 単位の値を与えている。

実験は次のようなステップで行われた。

ステップ1 血管の反応距離と反発力を調整する

実験初期値である A セットで、手術ナビゲーションを行ったところ、障害物である血管等からの反発はほとんど感じられなかった(図 3-10(a))。このような状態では誤って血管を切ってしまう可能性があるため、最初に障害物との反応距離を調整することにし、 D_{rv} を大きくすることにした。この時 C_{rv} をそのままにしておく、 D_{rv} の増加により、距離に対応して発生する斥力が小さくなってしまふ。これを避けるため、同時に C_{rv} も大きくするようにしている。 D_{rv} は、大きすぎると、どの位置にメス先があっても血管群からの反発が起こり、肝臓の手術自体が不可能になる。例えば、B セットのように D_{rv} を 51mm に設定すると、肝臓に触れていない点からの反発が発生し、肝臓へのアプローチすら不可能になった。

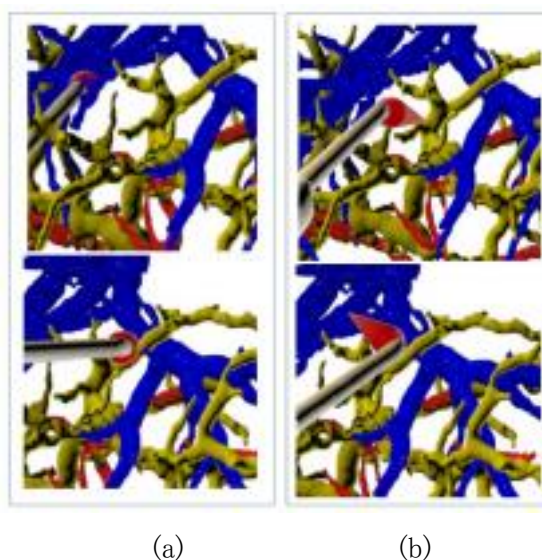


図 3-10 (a) 血管からの反発が弱い場合の状態、(b) 血管からの反発が適切な場合の状態。

D_{rv} の値は、Cセットの11mm程度が適切なようで、血管からの距離としては小さいように見えるが、この値以上の血管からの距離で反発が発生すると、メスの先端が肝臓自体から遠ざかるように手術ナビゲーションベクトルが指示されることになってしまう。そのため、 D_{rv} は11mmが適切であると判断した。この値に合わせて C_{rv} は2とした。

ステップ2 腫瘍への引力の調整

次に、腫瘍への引力を調整した。初期状態のAセットでは、メス先がかなり腫瘍に近づかないと吸引力が発生せず、吸引力が発生する範囲も小さかった。そのため D_{at} を増加させた。この増加により、より遠くの距離から引力が発生する。こちらも C_{at} の値が1のままだと、 D_{at} の増加により、距離に対応して発生する引力が小さくなる。このため C_{at} も合わせて増加させている。Cセットのように C_{at} が C_{rt} よりも大きくなってしまうと、メス先が腫瘍に近づくことが優先され、腫瘍に切り込んでしまうことがある(図3-11(b))。そのため C_{rt} も同時に増加させた。

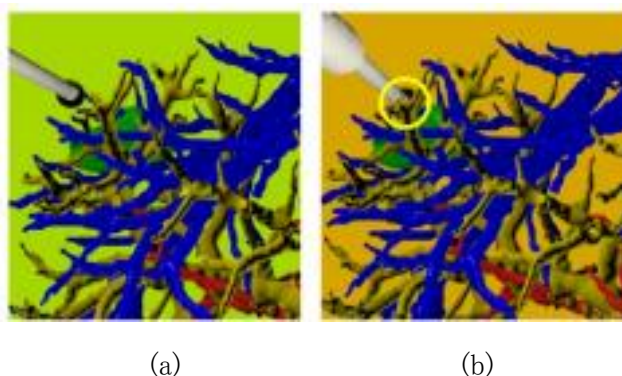


図3-11 C_{at} が C_{rt} より大きい場合の(a)侵入開始から(b)腫瘍到達まで(黄色い丸で囲んだ部分で腫瘍に切り込んでいる)。

ステップ3 腫瘍からの反発の調整

最後に、切除範囲を腫瘍外縁10mmとするためにmarginを調整した。この値は、肝臓外科医の意見として、腫瘍の周囲10mmの範囲を切除すべきとされていることに基づくものである。転移の状態など、患部の状態により異なるが、一般に手術で腫瘍を切除する際、腫瘍外縁の正常な組織は、わずかしこ切除しないことが望ましいとされている。腫瘍外縁10mmで手術ナビゲーションベクトルが無くなるようにすることで、切除最適位置を知らせるようにしたい。そのためには腫瘍の引力と斥力が均衡し、手術ナビゲーションベクトルが無くなる地点を腫瘍外縁10mmに定めればよい。腫瘍の引力と斥力が均衡する位置をZとすると、以下の恒等式(21)で各パラメータの関係を記述できる。

$$C_{at} - ((\text{margin} - Z)/D_{rt})^2 C_{rt} = 0 \quad (21)$$

例えば、 $Z=10\text{mm}$ としてCセットのパラメータの C_{rt} のみ再設定するなら

$$C_{at} = 1.5$$

$$D_{rt} = 5$$

$$\text{margin} = 12$$

$$Z = 10$$

より

$$1.5 - ((12 - 10)/5)^2 C_{rt} = 0$$

$$C_{rt} = 9.375$$

となる。ただ、この C_{rt} の値だと、少し近づいただけで手術ナビゲーションベクトルが極端な反応となったので、 C_{rt} を調整することとした。また位置 Z を 10mm として、そこで力が無くなるようにしていても、人間はそれより遠くから力を感じなくなる。margin も調整し位置 Z もいくつか試すこととした。Dセットだと $C_{rt}=6$ なので、計算される位置 Z は 12.5mm となっている。今回は C_{rt} を 4 とし、位置 Z は 9mm となるパラメータ値を最終調整値とした。

3-2-6 ポテンシャル関数パラメータ値の考察

この実験では、表 3-2 E セットのパラメータ値が最適だということに落ち着いた。この設定で腫瘍位置を変える、血管配置を変えるなど、いくつかのシナリオを試した結果、腫瘍からの吸引力を感じ始める距離も適切であることもわかった。しかし、位置 Z よりも少しでも切り込むと手術ナビゲーションベクトルが極端に反発してしまう点については、満足な解決値を見つけることができなかった。この改善にはパラメータ値ではなくポテンシャル関数の作りこみをおこなうべきだろう。以下に実験で得られた知見をまとめておく。

- 出血を考慮して D_{rv} を大きくしすぎると、肝臓を切開できず、移動が不可能になる
- C_{rv} 、 C_{rt} が大きいと、手術ナビゲーションベクトルの変化が大きすぎるため、使い勝手が悪くなる
- 腫瘍切除開始位置の正確な位置合わせを行うためには、 C_{at} と C_{rt} の比率に着目する必要がある

表 3-2 A セットのパラメータ値で手術シミュレーションをした時の手術ナビゲーションベクトルの変化を図 3-12 に、表 3-2 E セットでの手術ナビゲーションベクトルの変化を図 3-13 に示す。A セットの、絶えず力が派生して振動しているような状態が、最適設定では、力がかかっている時には A セットより大きく反応し、力がかかっていない時はより小さく振動する様子が確認できる。

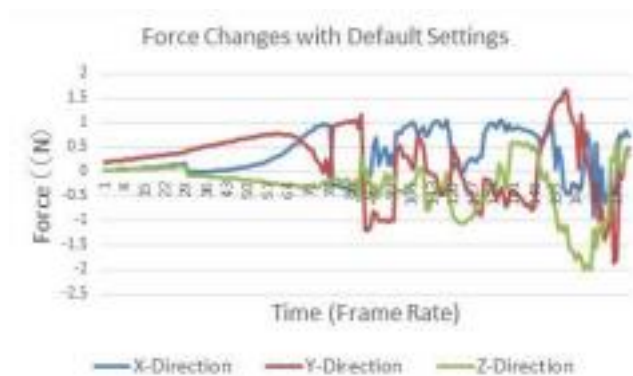


図 3-12 (表 3-2 A セット)のパラメータを用いた手術ナビゲーションベクトルの変化.



図 3-13 (表 3-2 E セット)のパラメータを用いた手術ナビゲーションベクトルの変化.

3-3 まとめ

ここでは、手術シミュレータへのナビゲーションシステムの組み込みを説明し、そこで利用する切削経路探索アルゴリズムとしてポテンシャルベース法を提案した。

提案したポテンシャルベース法では、ポテンシャル関数を設置する場所を2つに限定したことで実用的な 60Hz でナビゲーションの更新を行えている。パラメータの調整により、触覚デバイスのフォースフィードバックが調整されることも確認できた。

いくつかの試験により、ナビゲーションを指に体感できるパラメータ値も見つかり、計画したように腫瘍周辺で力のなくなることも確認できた。

しかし、障害物に少しでも切り込むと手術ナビゲーションベクトルが極端に反発してしまう点については、満足な解決値を見つけることができなかった。この改善にはパラメータ値ではなくポテンシャル関数の作りこみをおこなうべきだろう。

第4章 センサーベース法

ここでは、ポテンシャルベース法で予測される、移動不能状態を回避するために、別の切削経路探索アルゴリズムとしてセンサーベース法を提案する。これは、基本的にはメスを腫瘍方向に直進させ、深度バッファを用いて障害物に見立てた血管等を感知してそれを回避し、回避し始めたところよりも腫瘍に近く、かつそれに直進できるところから血管を離れ、腫瘍に向かって直進するものである。ここでは、ポテンシャルベース法とセンサーベース法の特性を比較・評価する。

4-1 ポテンシャルベース法の懸念点

ポテンシャルベース法は、停留(デッドロック)問題を理論的に内包するのみならず、2点間で往復を継続する振動現象をも生み出すことがある(図4-1)。これにより、メスが多くの組織を傷つけ、大出血を引き起こし術後の治りも悪くする。このような問題を回避するため、本研究ではセンサーベース法を提案した。

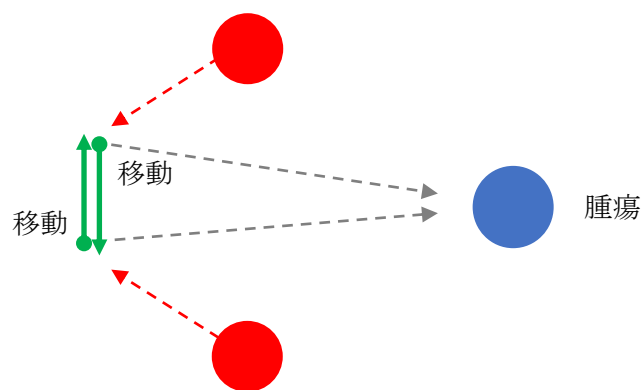


図 4-1 デッドロックの発生.

センサーベース法も、移動ロボットの障害物回避の研究[35]で提案された各種のアルゴリズムを利用したもので、この利用時はパラメータで種類を選択できるようにしている。ここでは、その中の1つについて簡単に説明する。

4-2 センサーベース法の経路探索ステップ

センサーベース法では、メス先を腫瘍に向かい直進させ、もし、そのままメス先が腫瘍に届いたら、探索は終了する(図4-2)。そうでなければ、直進を妨げる血管が感知されるので、メス先は血管の手前で直進を止め、そこから腫瘍までの距離を半径としたエリアを設定する(図4-3)。



図 4-2 障害物がない場合.

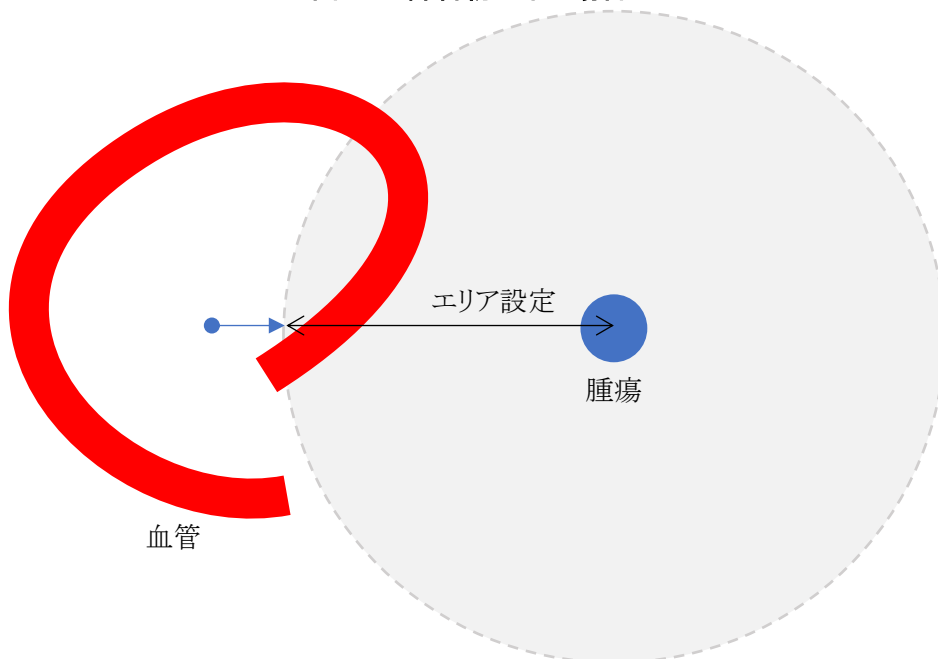


図 4-3 障害物を感知、エリアを設定.

そして、血管を常に進行方向の右側に置くよう、血管の周りを移動する(図 4-4)。ここで、移動の単位距離は、大きすぎると血管を切開したりして血管の周囲から離れてしまう。しかし、小さすぎると経路探索に時間がかかる。このような観点を考慮して、経験的な試行錯誤で予め設定する。

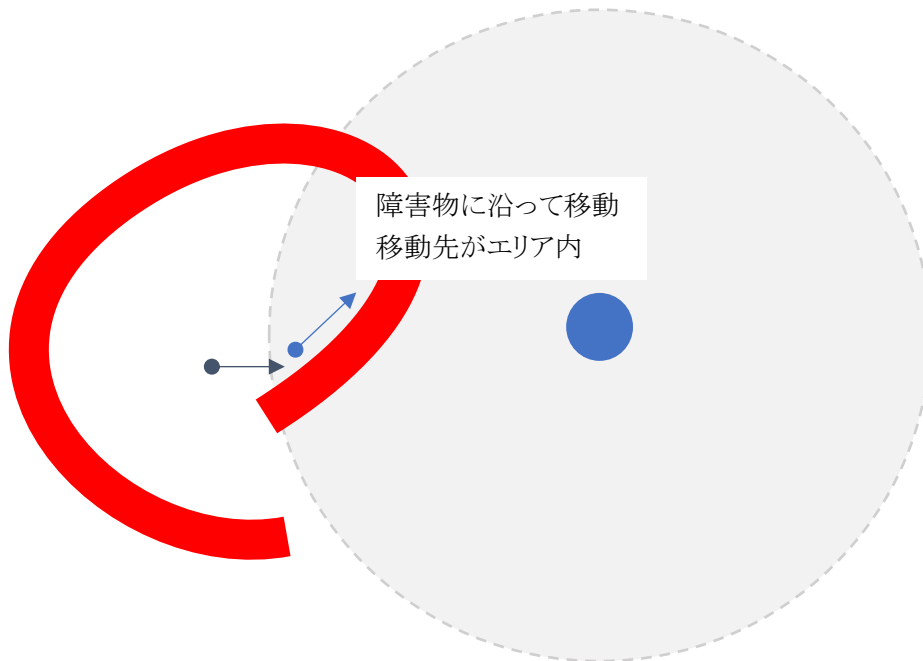


図 4-4 障害物に沿って移動.

移動先がエリア内なら、最初にやったように腫瘍に向け直進をこころみ、障害物がなければそのまま腫瘍まで進み終了する。障害物があれば、そこから腫瘍までの距離を半径としたエリアを更新する。そしてメス先は、障害物にそって移動する(図 4-5)。

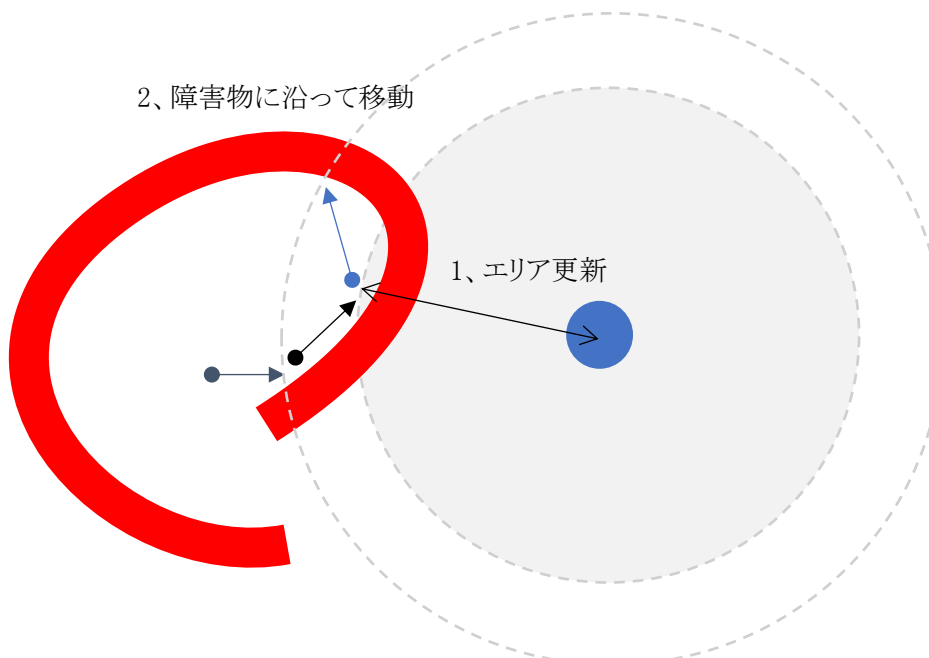


図 4-5 移動先がエリア内で障害物があった場合.

エリア外では、現在位置から腫瘍までの間に障害物が無くならない限り、現状の障害物にそって移動を繰り返す(図 4-6)。

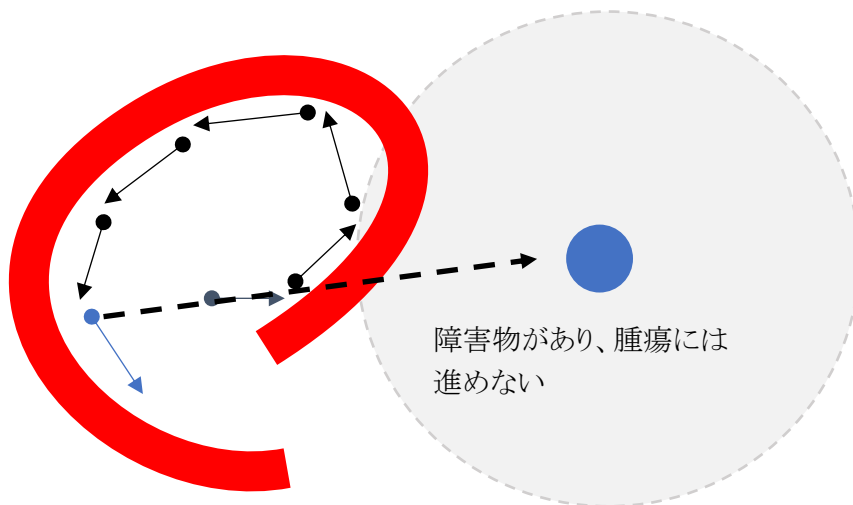


図 4-6 移動先がエリア外の場合.

もしエリア外で、現在位置から腫瘍までの間に、障害物が無くなった場合は、エリア内でやったように腫瘍に向け直進をこころみ、障害物がなければそのまま腫瘍まで進み終了する(図 4-7)。

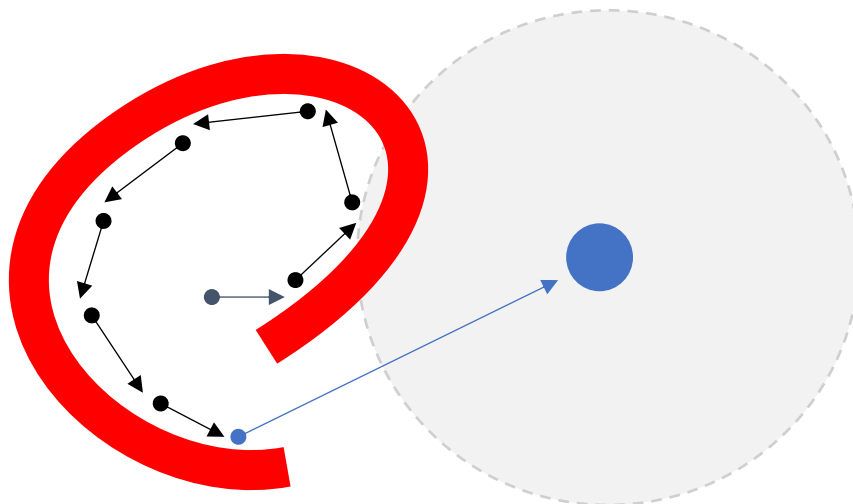


図 4-7 移動先がエリア外で腫瘍まで障害物がない.

そのため、最悪のケースではかなりの大回りになる(図 4-8)。

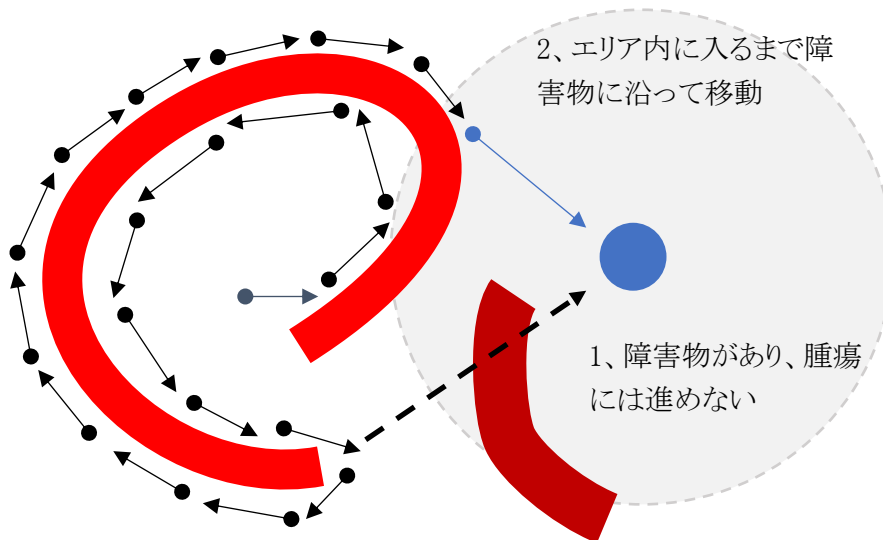


図 4-8 エリア内に入るまで大回りになるケース.

このように、センサーベース法ではメス先が血管の周囲をなぞるように動く(図 4-9(b)).

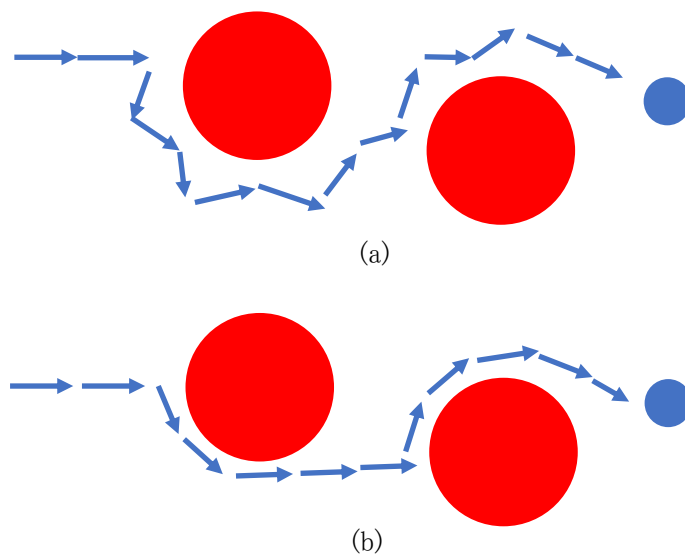


図 4-9 ポテンシャルベース法とセンサーベース法の移動経路の違い。(a)ポテンシャルベース法ではメスが血管を避けるように動き、(b)センサーベース法ではメスが血管の周囲をなぞるように動く。

4-3 センサーイメージの生成法

このように腫瘍への距離を縮めながら、障害物に沿って移動していくのがセンサーベース法だが、移動ロボットでは“障害物に向かって反時計回りに移動”というように、2D 平面上での移動が前提となっている。そのため、今回の3D 経路探索で利用するにあたっては、メス先から

腫瘍に向けた方向を移動用平面の Y 軸とした。移動用平面の X 軸は、移動用平面の Y 軸と、グローバル空間の Y 軸との外積から決定する(図 4-10)。グローバル空間の Y 軸が移動用平面の Y 軸と重なるときは、グローバル空間の X 軸を代用する。

3次元の血管を回避する方向、すなわち上記の平面の選択肢は無限にあることから、上記のような取り決めより本来存在する経路を見逃す可能性を排除できないが、無限にある位置から無限に存在する平面を選択するのは無限であり未解決である。

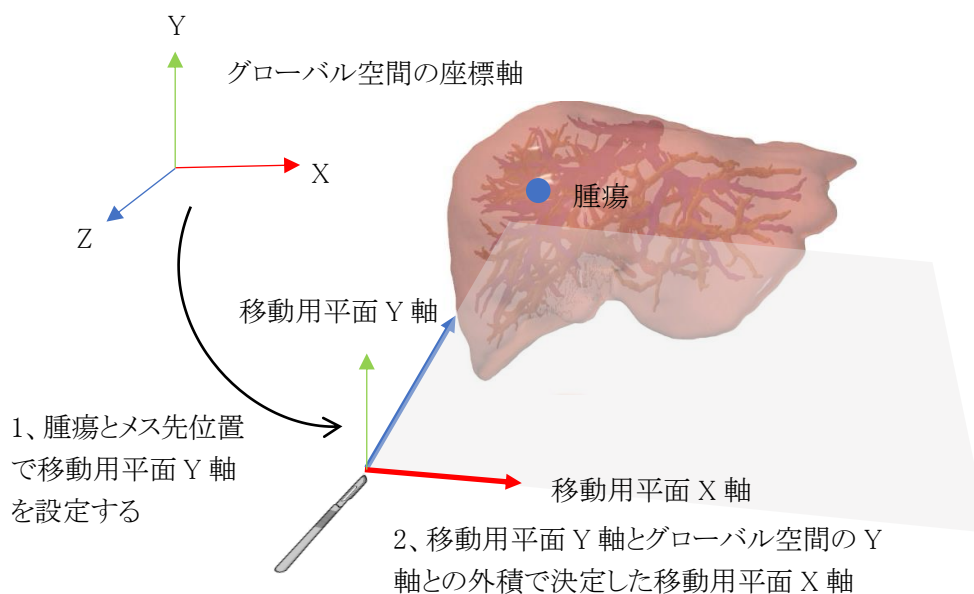


図 4-10 移動用 2D 平面の決定.

メス先のセンサは、メス先位置をカメラ位置とし、腫瘍に向けて撮影した深度バッファを利用した。

センサーベース法は表 4-1 のパラメータで制御する。

表 4-1 センサーベース法のパラメータ

| 名称 | 説明 | 設定値 |
|-----------------|--|------|
| step | 1 回に行うメス先の移動距離(mm) | 0.02 |
| radius | メス先から障害物までの許容される近接距離 (mm) | 1.0 |
| sensorMargin | メス先と障害物を衝突とみなす許容誤差 (mm) | 5.0 |
| sensorStepCount | 移動不可とみなす上限回数 | 1500 |
| sensorAlgorithm | アルゴリズムの選択(クラス 1 ~クラス 5) | 1 |
| sensorDpmm | 1mm あたりに割り当てるセンサの画素数 | 10 |
| sensorWidth | センサの幅 (mm) | 5.0 |
| sensorHeight | センサの高さ (mm) | 5.0 |
| sensorType | 投影法(平行投影・透視投影)の選択 | 平行投影 |
| area | 深度バッファの深度を、そのまま使うなら 1、周辺の深度と平均するなら、その範囲をピクセル数で指定 | 1 |
| fovDegree | センサの視野角(度) | 90.0 |

注意) sensorWidth、sensorHeight、sensorMargin は、5mm 程度の周辺を調べながら進まそうという程度の設定。step も 1mm 動かす間に 50 回くらい周囲を確認させながら動かそうといった程度のものである。

4-4 センサーベースでの経路探索実装について

メス先の現在位置を開始位置、腫瘍をゴールとして、センサーベースでの移動処理を具体的に説明する。前述したように、センサーベース法では、エリア内で、ゴール方向の障害物を調べながら進む移動と、エリア外で、障害物に沿って移動するフェーズが存在する。ここでは前者をフェーズ1、後者をフェーズ2として説明する。

フェーズ1

初期状態はフェーズ1となる。

フェーズ1では、現在位置からゴールに向けて進む移動ベクトルを設定する(図 4-11)。

移動ベクトルを設定した後は、表 4-1 step の距離だけ移動をおこなうことになるが、移動する前に障害物の有無をセンサで確認する。

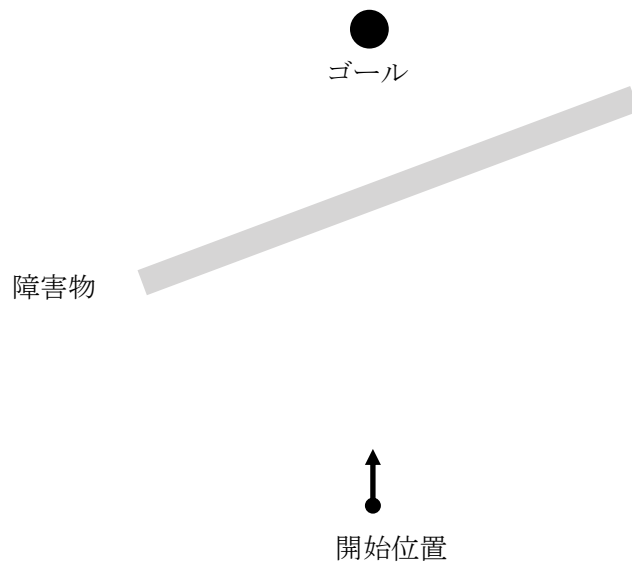


図 4-11 初期状態.

センサによる障害物の確認では、調べたい方向にカメラを向けて撮影、深度バッファを作成し、障害物までの距離を測定する(図 4-12)。

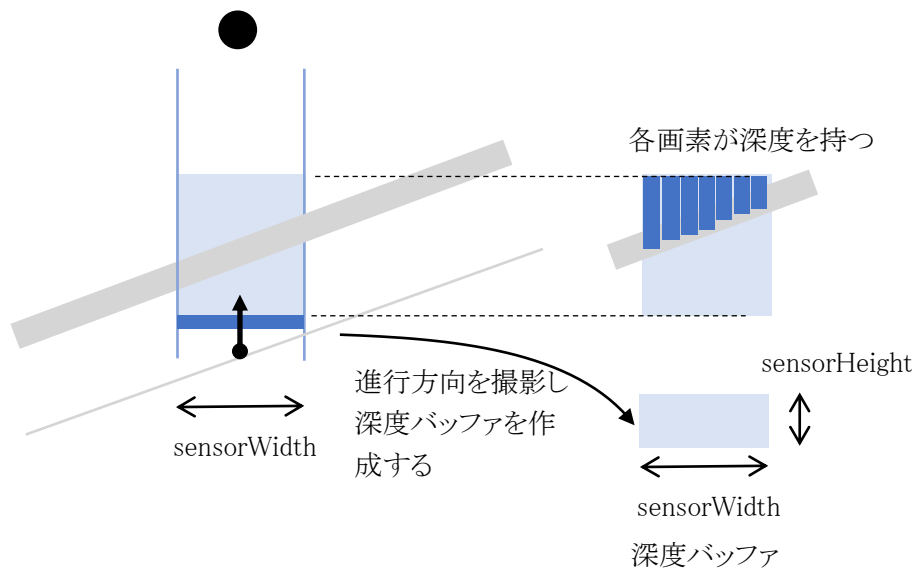


図 4-12 センサによる障害物確認.

この測定で障害物が見つかった場合も、step での移動後の障害物までの距離が $sensorMargin$ より大きいなら、そのまま移動し、次のフェーズ1の処理に移行する。そうでなければ $sensorMargin$ の距離を保つよう移動し、フェーズ 2 に進む準備をする(図 4-13)。

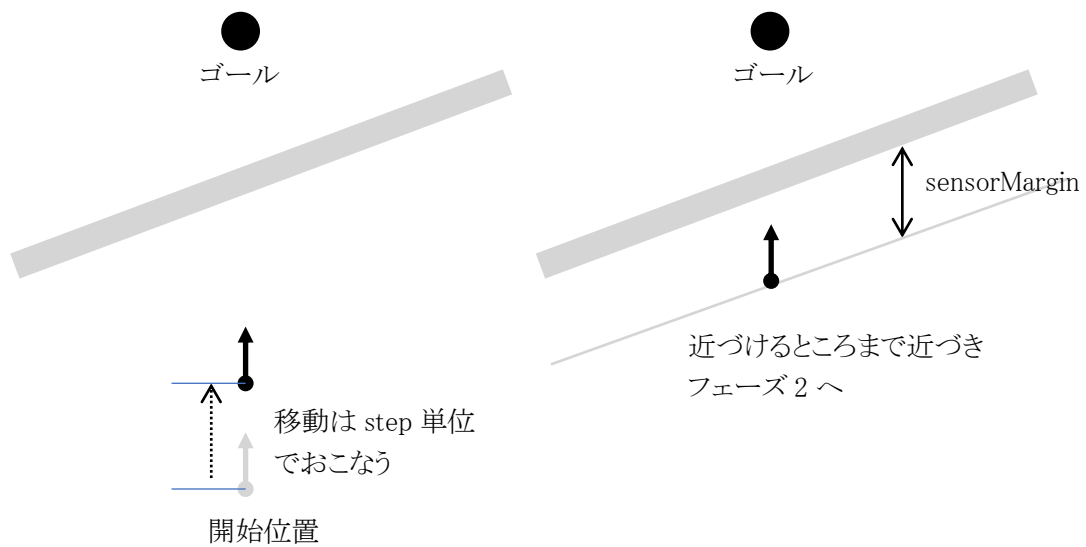


図 4-13 障害物までの距離によるフェーズ遷移判断.

フェーズ 2 に進む準備では、深度バッファの左半分から最短距離の画素を探し、現在位置から最短距離位置へのベクトル N を生成する。そして生成したベクトル N と移動平面の Z 軸との外積により、新しい移動ベクトルを決定する(図 4-14)。

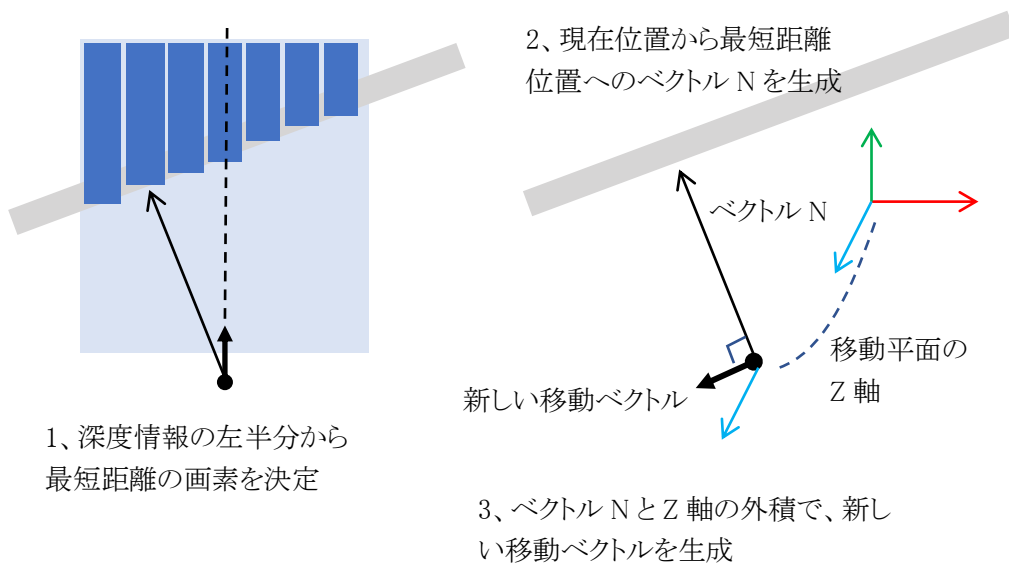


図 4-14 フェーズ 2 に進む準備.

これで壁を沿って進むフェーズ 2 の準備が完了し、次の処理ではフェーズ 2 としての処理を実行する。

フェーズ 2

フェーズ 2 では、最初に現在位置とゴールまでに障害物の存在を確認し、存在しないなら、そのままゴールまで直進する。障害物が存在するなら、次は、現在位置とゴールまでの距離を

確認し、4-2 で説明したエリア内ならフェーズ1に戻り、そうでないなら、そのままフェーズ 2 の処理を続ける。

フェーズ 2 の処理を続ける場合は、進行方向からみて右側垂直方向を撮影し、最短距離 (こちらは深度バッファの全幅を調べる) にある障害物の位置を見つけ、現在位置から、見つけた位置へのベクトル N を作成する。そしてベクトル N の長さが sensorMargin を保つように現在位置を調整しつつ、移動平面の Z 軸との外積を使い進行方向を補正する。この処理により、障害物の角に来ていた時などには、障害物に沿った新しい移動ベクトルが作り出されるようになっている(図 4-15)。

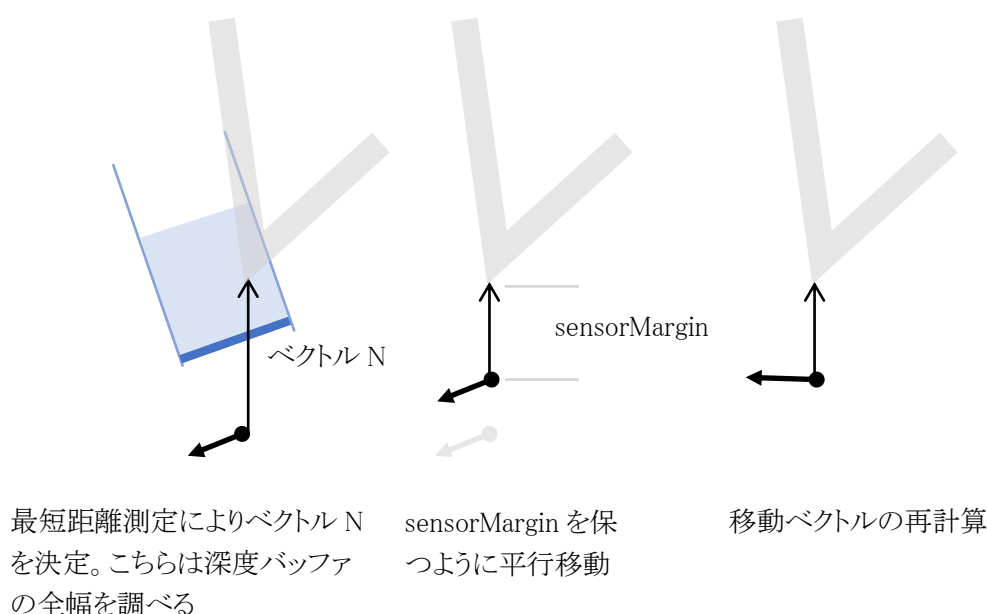


図 4-15 障害物に沿った新しい方向ベクトルが作り出される過程.

その後、進行方向に障害物があるかを、深度バッファを作成し確認する。障害物がある場合、障害物までの距離が sensorMargin を下回るなら移動は許さず、その場でフェーズ 2 に進む準備でおこなった同じ方法で、深度バッファから新しい移動ベクトルを決定する。この新しい移動ベクトルの進行先も同じように移動不可なら、また同じ方法で新しい移動ベクトルを決定する。1回転してしまったら移動不可能とし、経路探索を終了する。

移動可能なら、step だけ移動したのち、次のフェーズ2の処理に移行する。

4-5 ポテンシャルベース法とセンサーベース法の評価

提案したポテンシャルベースとセンサーベースの切削経路探索アルゴリズムが、メス先端から腫瘍までの経路を確実に選択できるかどうかを、患者の肝臓モデルを変形させて様々な肝臓モデルを作り出して検証した。検証1の結果を図 4-16、検証 2 の結果を図 4-17 に示す。

各図の(a)が、アルゴリズムを適用した肝臓血管モデルで、ピンク色のチューブがポテンシャルベース法の経路、緑色のチューブがセンサーベース法の経路を示す。(b)、(c)は同じ経路を赤線で抽出して表示している。表 4-2 は検証1の総移動量と移動ステップごとの向きの変化の

標準偏差となる。**表 4-3** は検証 2 の総移動量と移動ステップごとの向きの変化の標準偏差となる。

この結果を見ると、一般に、ポテンシャルベース法は、メスから腫瘍までの距離が比較的直線的で短い経路を選択する。しかし、切開の方向を決定する各移動ステップでは何度も試行錯誤が発生している。この切り込み方向を試行錯誤することは、臓器組織にダメージを与え、出血量を増加させるので好ましくない。逆に、センサーベース手術ナビゲーションでは、メスの先端から腫瘍まで比較的長い経路を選択する。しかし、各移動ステップでの試行錯誤は少なく、ステップでの切り込み方向の大きな変化はないため、出血のリスクを軽減することができる。血管を太くしたり、追加したり、腫瘍の位置を変えたりして検証したが、どの場合でも同じような特性が得られた。

ポテンシャルベース法の各移動ステップでの試行錯誤は、2 点のポテンシャル関数のみで移動経路を決定しているのが原因である。これより、試行錯誤の結果を整理するなど今後改善が期待される。

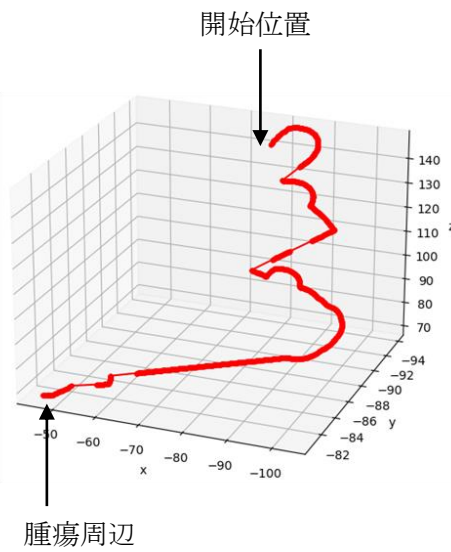
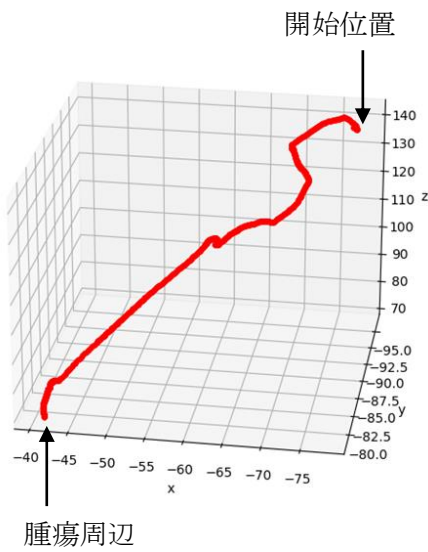
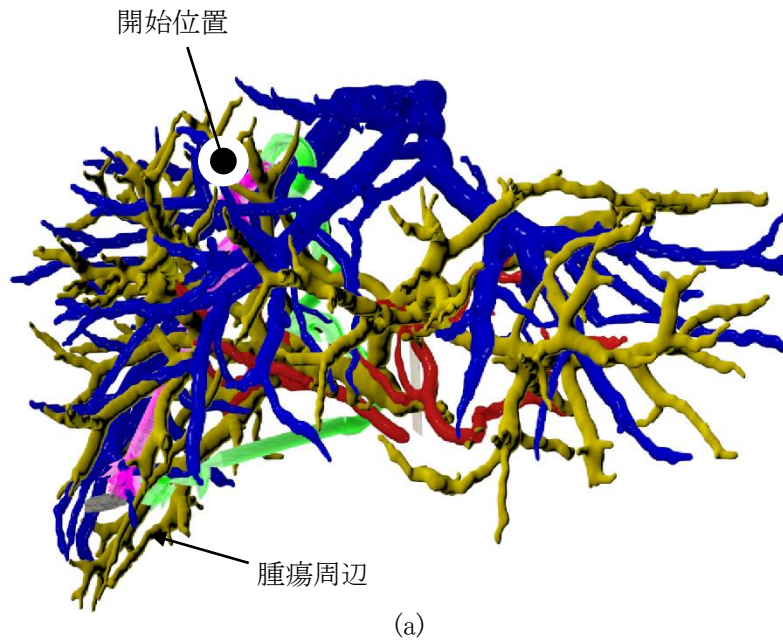


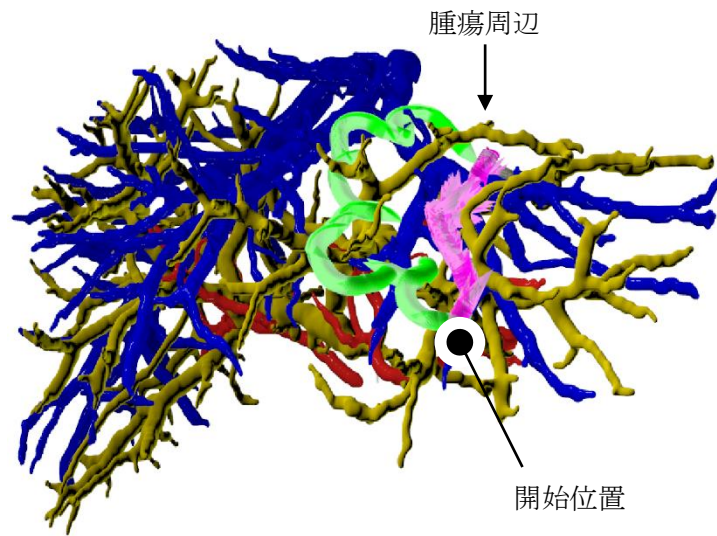
図 4-16 (a) 検証 1 のキャプチャ画像

(b)ポテンシャル法の 3D 経路【開始位置(-77,-99,135) →腫瘍周辺(-42,-82,69)】

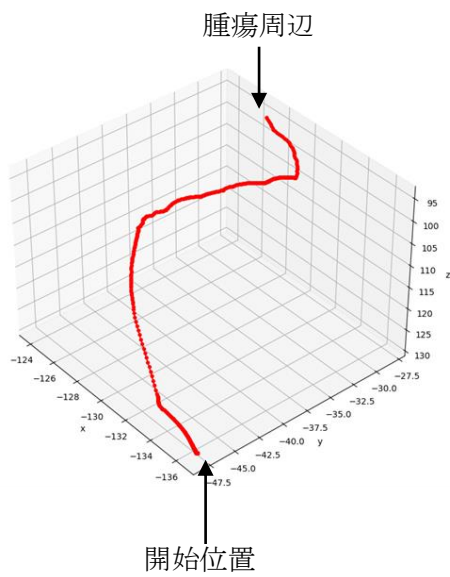
(c)センサーベース法の 3D 経路【開始位置(-77,-98,137)→腫瘍周辺(-42,-83,69)】
の 3D 経路表示。スケールは 3D 経路全体が入るように自動的に調整している。

表 4-2 検証 1 の総移動量と、移動ステップごとの向きの変化の標準偏差.

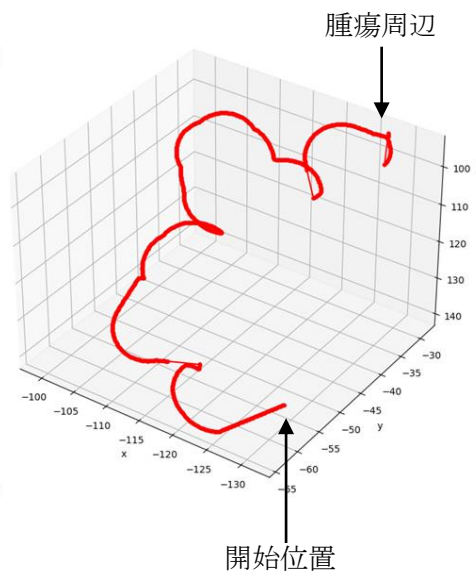
| | ポテンシャル法 | センサーベース法 |
|----------|---------|----------|
| 総移動量(mm) | 99.35 | 171.36 |
| 標準偏差 | 0.38 | 0.17 |



(a)



(b)



(c)

図 4-17 (a) 検証 2 のキャプチャ画像

(b)ポテンシャル法の 3D 経路【開始位置(-126,-26,98) →腫瘍周辺(-136,-47.5,132)】

(c)センサーベース法の 3D 経路【開始位置(-126,-26,98)→腫瘍周辺(-127.5,-52.5,142)】

の 3D 経路表示。スケールは 3D 経路全体が図に入るように自動的に調整している。

表 4-3 検証 2 の総移動量と移動ステップごとの向きの変化の標準偏差。

| | ポテンシャル法 | センサーベース法 |
|----------|---------|----------|
| 総移動量(mm) | 52.81 | 166.79 |
| 標準偏差 | 1.16 | 0.18 |

4-6 ポテンシャルベース法とセンサーベース法の併用

先の検証では、血管の直径を大きくし、結果を新たに追加するなどして、血管網の複雑さを意図的に増加させ、血管間の距離を減少させたが、どちらのアルゴリズムも確実に経路を選択することができ、予想されたポテンシャルベース法でのデッドロックを確認することができなかった。

細かい移動ステップではなく、全体的な経路を示す場合、ポテンシャルベース法を使い、直線的な経路を求めるほうが良い。加えて、デッドロックに陥るようならセンサーベース法に切り替える仕組みを組み込めば、より安定性を増すことができる。センサーベース法は、障害物に沿って移動する関係で、ポテンシャルベース法に比べて遠回りの経路を見つけがちのため、ある程度の距離を移動したら、ポテンシャルベース法に切り替えて探索を続けるようにしている。併用することで、安定した経路探索が可能となる。

4-7 まとめ

ここでは、ポテンシャルベース法、およびセンサーベース法の切削経路探索アルゴリズムを提案した。また、それらを使い分ける方法も提案した。ポテンシャルベース法は悪性腫瘍に向かって直線的な経路を見つけ出しやすく、一方、センサーベース法は複雑な血管群を大回りして回避する経路を見つけがちである。両者の特性をうまく組み合わせれば、出血量を管理したより安定的な経路探索ができると考えている。

第5章 ボクセルベース法

ここでは、ナビゲーションアルゴリズムとして、パラメータによって直進性などがコントロールできるボクセルベース法を提案する。最初に直進性をコントロールする目的を説明し、そのアルゴリズムを説明する。その後、パラメータを変化させ直進性などがコントロールできるかを評価する。また、このアルゴリズムの特性から、3D メッシュではなくDICOMを直接扱えるので、その対応もおこなった。シミュレータ側のDICOM対応と合わせて説明する。

5-1 境界面切削への課題とその対応アルゴリズム

腫瘍切除手術は、腫瘍周辺領域を切り取る手術である。腫瘍が発生した臓器と、切除領域の境界面を切削していくことになる。境界面を、切削経路の積み重なりと考えると、外科医が指定するメス位置から腫瘍まで、ある程度直線的な切削経路をナビゲーションしたい。そうすれば、外科医が臓器表面の切り込みたい位置をメス先でなぞることで、境界面の候補を見つけられることになる。そういった手術シミュレータがあれば、術前の手術計画作成の支援になる。

このような切削経路検索に、前出のアルゴリズムをそのまま利用することは難しい。ポテンシャルベース法が、ある程度直進性を持つとはいえ、両方式とも障害物領域を通過不可とし、完全に回避しようとするアルゴリズムに変わりはない。数センチ程度の局所的なナビゲーションでは有用であるが、もう少し長い距離では、生体組織の弾性を利用し、血管等を移動させて直進性を優先させるといったこともあり得るので、この点からすると、どちらの方法も複雑な経路になりがちと言える。そのため、障害物領域を通過不可とせず、ある程度通過を許容できるアルゴリズムを新しく提案する。

このアルゴリズムでは、3D 空間を格子状に分割し、小さな立方体空間の集合(図 5-1)として扱い、各空間に占める障害物の濃度をもとに経路探索を行う。プログラムの世界では、この立方体空間のことをボクセルと呼ぶことが多いので、このアルゴリズムをボクセルベース法と呼ぶことにする。

5-2 ボクセル群の生成

ボクセルベース法では、メス先と腫瘍を結ぶ直線を Z 軸とした空間を設定し、これを任意幅で格子状に分割したボクセル群を使う(図 5-1)。

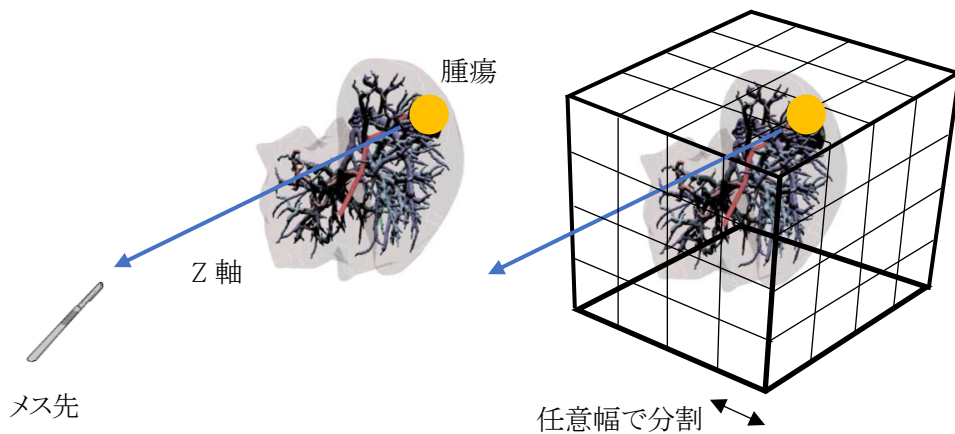
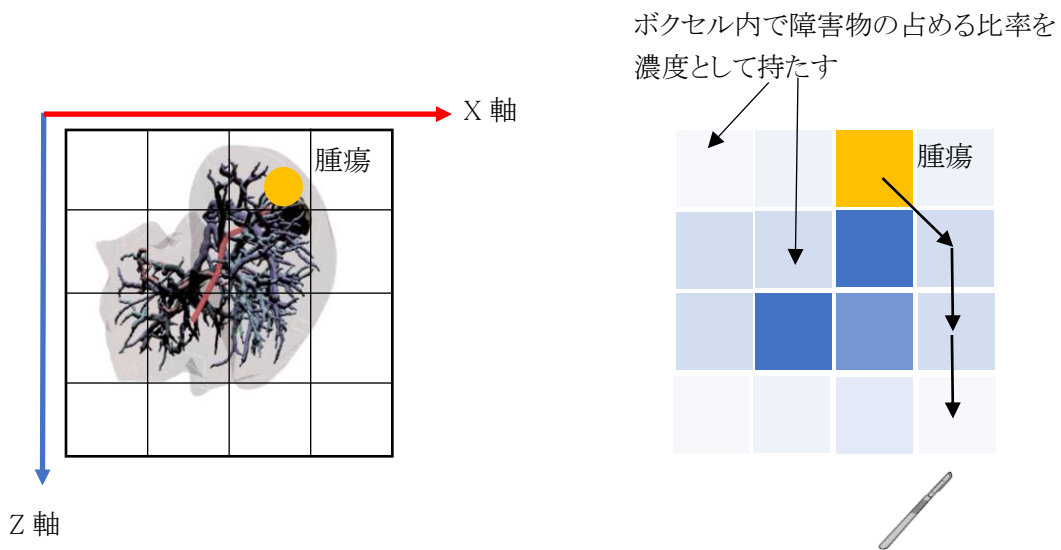


図 5-1 ボクセルベース法では、空間を格子状に分割しボクセル群として扱う。

各ボクセルには、ボクセル内で障害物の占める比率を濃度として持たせる。そして腫瘍を含むボクセルから初めて、メス先方向に、できるだけ濃度の薄いボクセルを通過するよう移動させ、見つかった経路の逆方向を切削経路とする(図 5-2)。



分割されたボクセル群を XZ 平面から見た状態

メス先方向に、できるだけ濃度の薄いボクセルを通過するよう移動

図 5-2 できるだけ濃度の薄いボクセルを通過するよう移動させる。

5-3 ボクセル階層間の移動

ボクセルベース法での切削経路探索では、まず、腫瘍が存在するボクセルから、Z 軸に沿って 1 つボクセルの階層を進める。進んだ先のボクセルを V1 とし、V1 と周囲、合わせて 9 つのボクセルを移動先候補とする(図 5-3)。

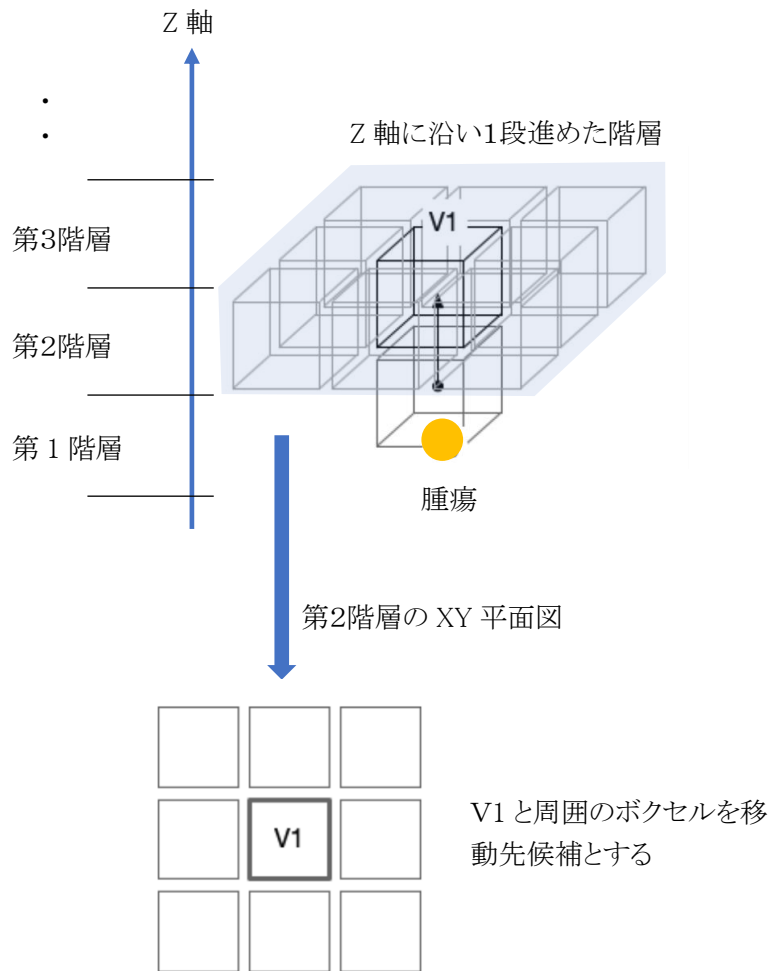


図 5-3 1つ先の階層の移動先候補.

この9つの移動先候補の濃度値から、移動するボクセルを決定するのだが、そのまま使うと、パラメータでのコントロールができないので、V1を基準にした位置関係でボクセルの濃度を加工する関数を適用し、判定に使う濃度にはこの関数値を使うようにした。本研究では重みづけ関数として、式(22)を用意し、周辺の濃度がより大きくなるようにしている。

$$D + C(\sqrt{dx^2 + dy^2})^3 \quad (22)$$

D 調整対象のボクセルが持つ元々の密度

C 濃度調整用係数

dx V1から調整対象までのX軸オフセット値(-1, 0, 1)

dy V1から調整対象までのY軸オフセット値(-1, 0, 1)

Cの値を調整することで、図5-4のように中心のV1ボクセルに比べ、周辺ボクセルの濃度が増え、より通過しにくくなり、直進性をコントロールできる。

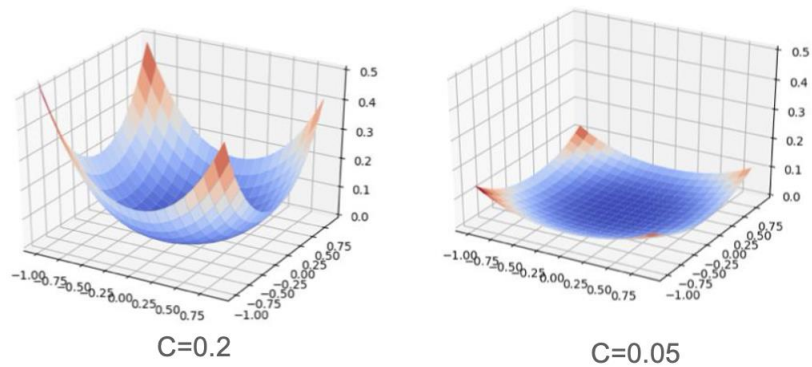


図 5-4 C 値の変化に対応した重みづけ分布.

通過可能かどうかは、補正された濃度に対して、パラメータとして用意した閾値で判断し、9つの候補から、通過可能で最も濃度の低いボクセルを、次の階層の通過ボクセルとする。そして次の階層でも、先と同じように Z 軸に沿って、新しい V1 を決め同じ操作を繰り返す。このようにして見つけたボクセルの中心を、線で結び切削経路とする。

5-4 ボクセルベース法の経路探索ステップ詳細

Z 軸に沿って階層を1つ進めた先でも、上述した方法で、次の階層のボクセル候補を選ぶのだが、次の階層の9つの候補すべてが通過不可の場合もある。この場合は、1つ前のボクセル階層に戻る(図 5-5)。

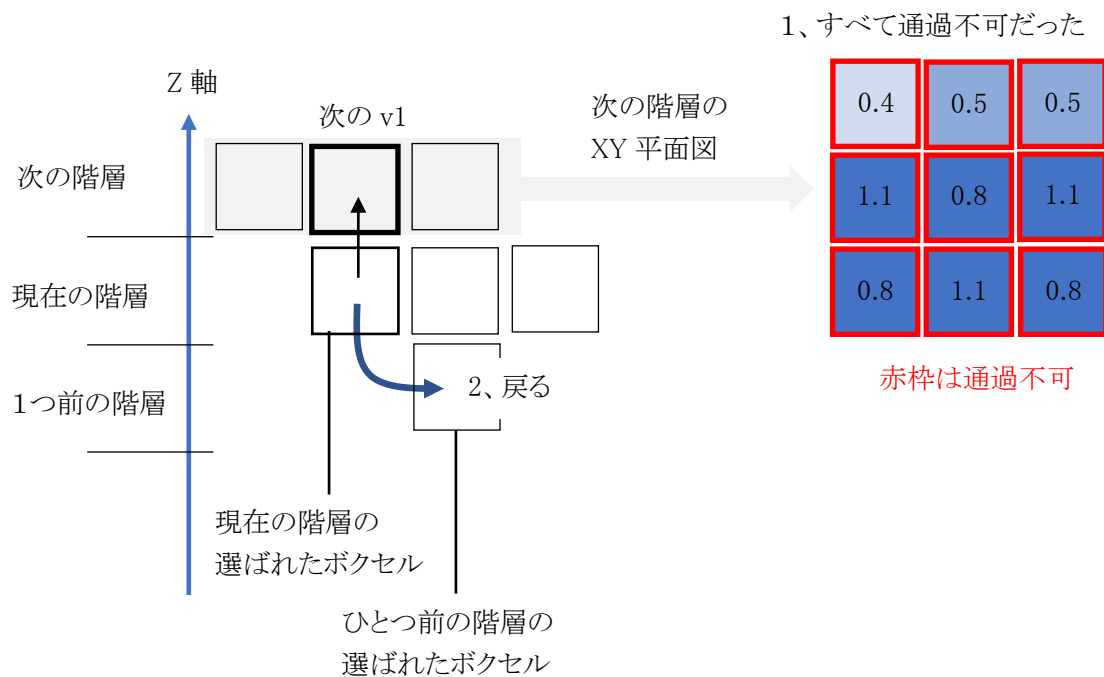


図 5-5 次の階層のボクセル候補がすべて通過不可なら、1つ前の階層に戻る.

1 段戻った階層に、選んだボクセル以外に通過可能なボクセルがあるなら、そのうちの一番濃度の薄いボクセルを新しい通過ボクセルとして探索をやり直す(図 5-6)。

階層を戻りきって、最初の腫瘍を含むボクセルまで戻ってしまったなら、切削経路決定不可とする。

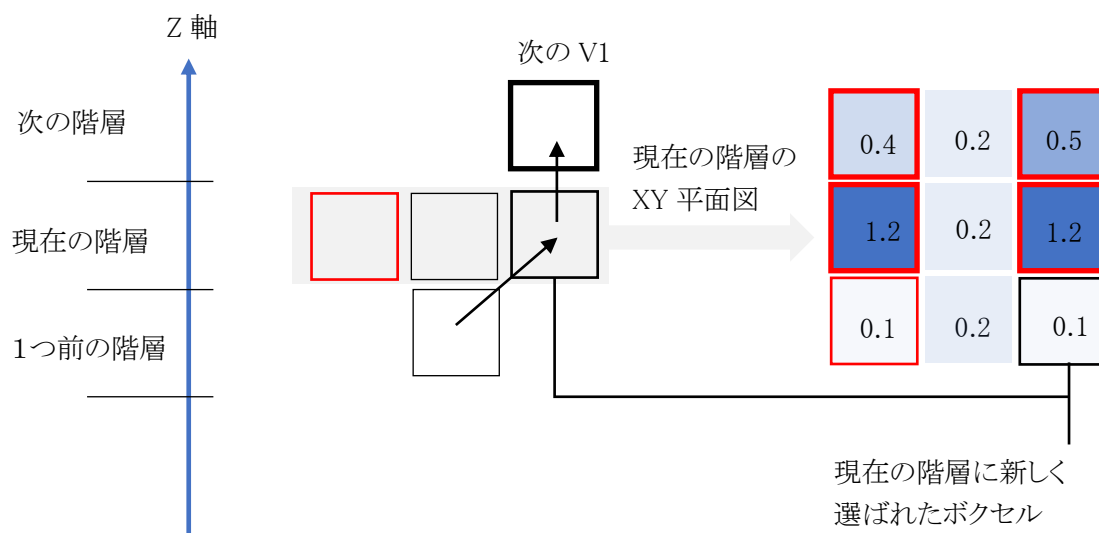
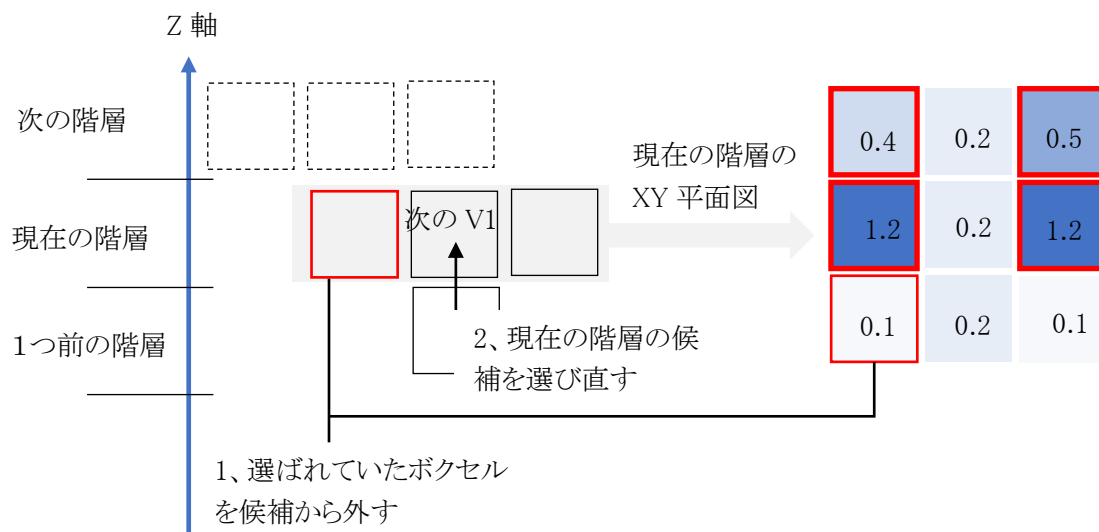


図 5-6 1つ前の階層で別の候補を選び直してやり直す。

このようにして見つけた通過可能ボクセルの中心を結び切削経路とする。

5-5 ボクセルの濃度測定方法

各ボクセルの濃度測定は、Z 軸のボクセル階層ごとにおこなう。

カメラ方向を Z 軸にとり、カメラから見た、測定する階層の手前側 Z 座標を nearZ に、後ろ側 Z 座標を farZ に設定し、平行投影で撮影する(図 5-7)。撮影する 2D 画像のサイズは、臓器全体を映し込める大きさとして、あらかじめ決定しておく。

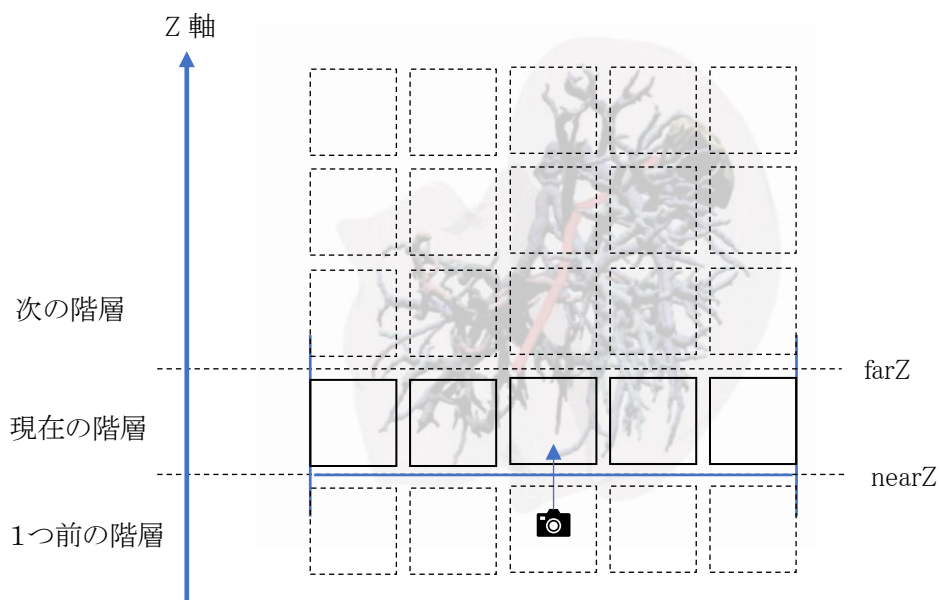


図 5-7 濃度測定では1階層ごとに撮影する.

この指定で生成される 2D 画像および深度バッファは、nearZ から farZ の範囲で輪切りにされた 3D メッシュの画像と、深度バッファとなる。このうち、濃度測定には深度バッファを使用する。深度バッファは、ボクセル群の 1 階層分なので、縦横をボクセル幅ごとに分割し、深度のある部分を 1、ない部分を 0 として集計し、画素数で割った値を濃度とする(図 5-8)。

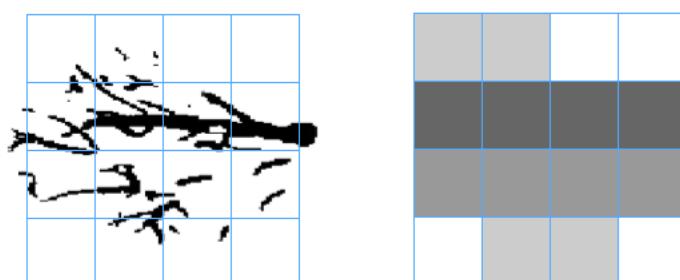
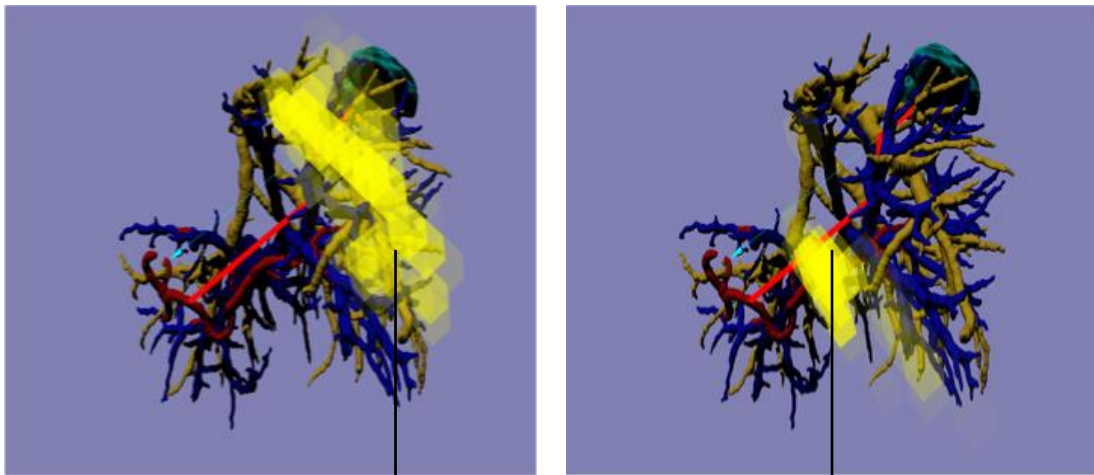


図 5-8 深度バッファからの濃度決定.

この作業を、Z 軸の腫瘍を含むボクセルの階層からメス先を含むボクセルの階層まで繰り返す。図 5-9 は階層ごとに生成されたボクセルと、その元になった深度バッファを表示させたもの。深度バッファは障害物が存在する画素を黒、存在しない画素を白として 2 値で表示させている。



表示されているボクセル階層の元になった深度バッファ

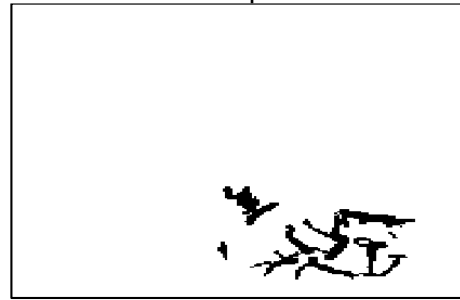
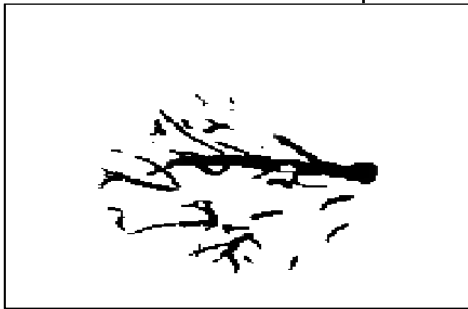


図 5-9 深度バッファから生成される 1 階層分のボクセル群。

輪切りの画像を撮影する際の注意点として、面の法線による描画の省略がある。3D メッシュは三角形平面で構成されているが、これらの平面には裏面と表面を判別するための、法線が設定されている。通常、多面体は裏面が見えるようには構成されないため、3D 描画用 API は描画高速化のために、裏面を描画しない設定になっている。今回のような輪切りの画像を撮影する際は、多面体が切り取られ、裏面が露出する可能性があるため、必ず裏面も描画する設定にしておく。そうしないと裏面分が消えて、正しく濃度を測定できない(図 5-10)。

また、この処理を行っても、血管の内部の空間を障害物から除外してしまうわけだが、ボクセルベース法では人体組織を軟体としてある程度移動可能と考えるので、除外されても良いとした。

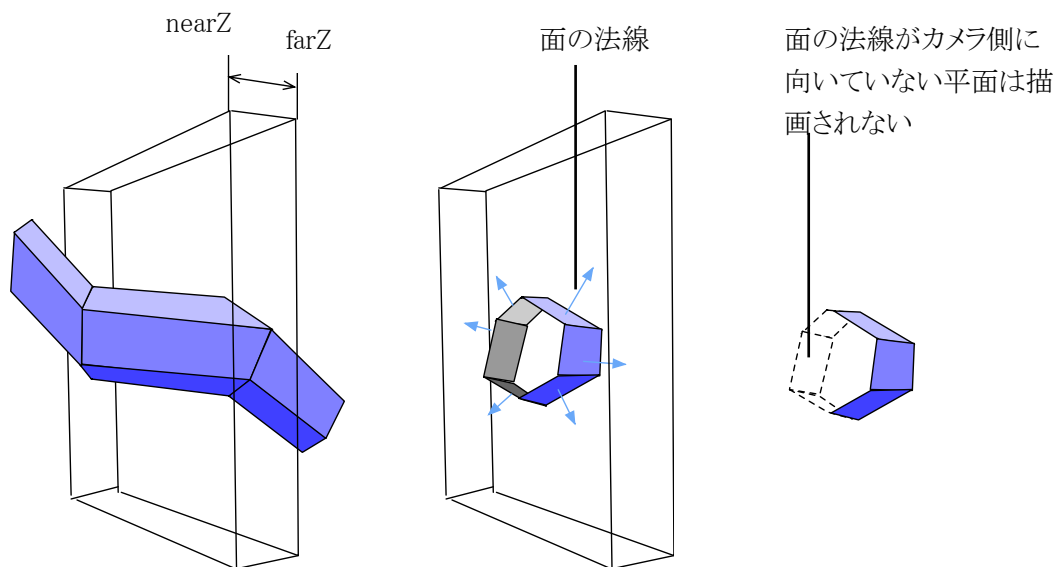


図 5-10 裏面を描画する設定にしないと3D メッシュの一部が欠損する.

5-6 ボクセルベース法の評価

ボクセル幅(5、7、10mm)、閾値(0.6、0.4)、C 値(0.05、0.2、0.4)を変化させて、本アルゴリズムの実際の効果の評価した。

図 5-11(a) に評価で使った肝臓モデルの形状を示す。このうち、腫瘍と見つかった経路、腫瘍からメス先へのベクトルだけにしたものが図 5-11(b) となる。黄色い物体は腫瘍を、緑の矢印は腫瘍からメス先へのベクトルを、赤い線は見つかった経路を表している。

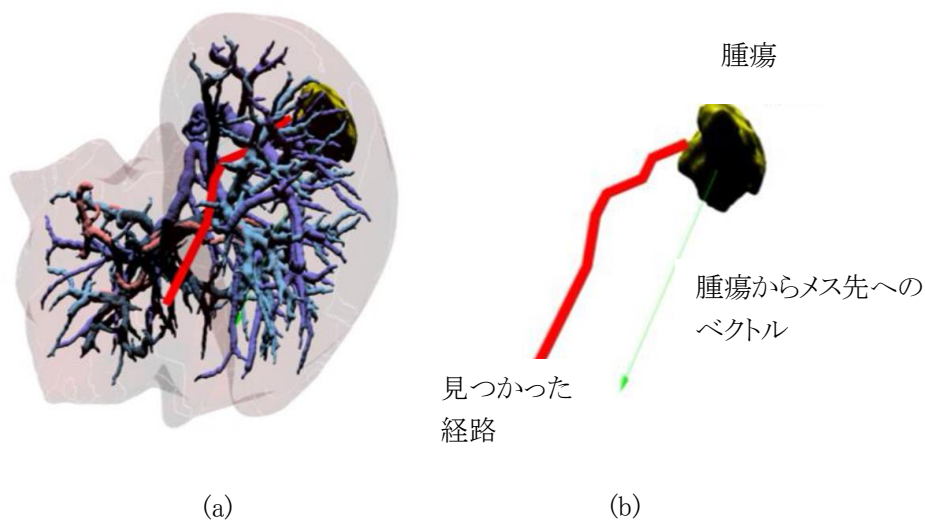


図 5-11 評価に使った肝臓モデル.

各ボクセル幅(5、7、10mm)で作成するボクセル群は**図 5-12**のようになる。半透明の黄色い立方体が1つのボクセルである。密度の値が低いほど透明になる。見やすくするため、Z軸に沿った1階層のボクセル群のみを表示している。

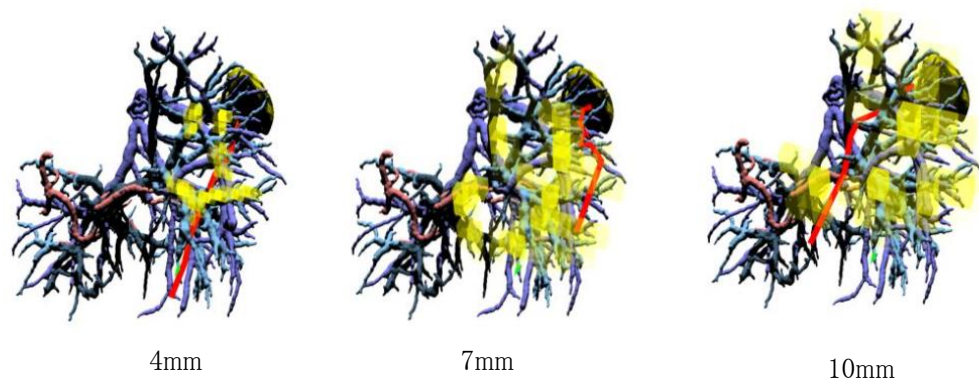


図 5-12 各ボクセル幅(5、7、10mm)で作成するボクセル群

肝臓のモデル、腫瘍からメス先へのベクトルを変えずに、ボクセル幅、閾値、C 値を変えて実行した結果は**表 5-1**～**表 5-3**のようになる。

表 5-1 ボクセル幅 5mm

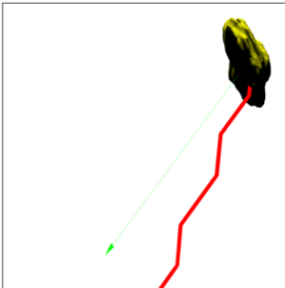
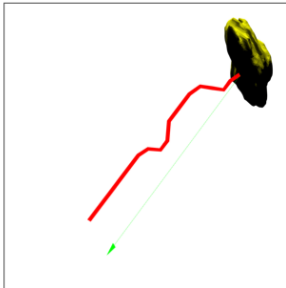
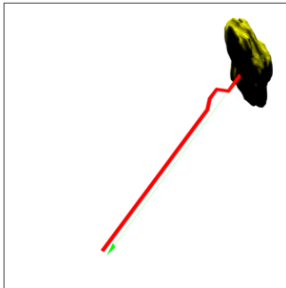
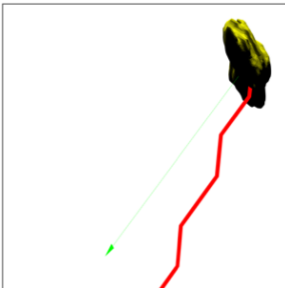
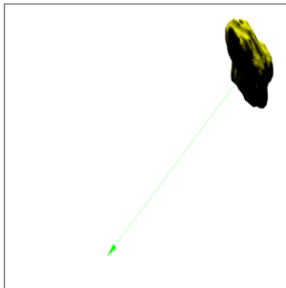
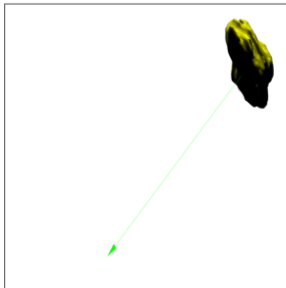
| 閾値 | C 値:左から 0.05, 0.2, 0.4 | | |
|-----|--|---|--|
| 0.6 |  C:0.05 completion |  C:0.2 completion |  C:0.4 completion |
| 0.4 |  C:0.05 completion |  C:0.2 Failed |  C:0.4 Failed |

表 5-2 ボクセル幅 7mm

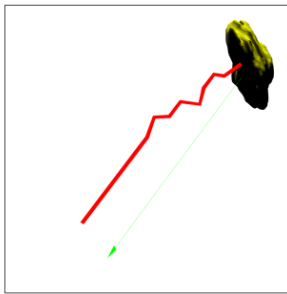
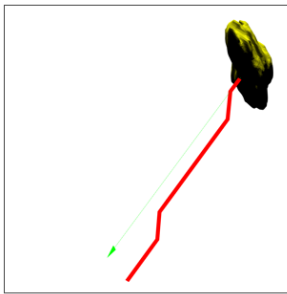
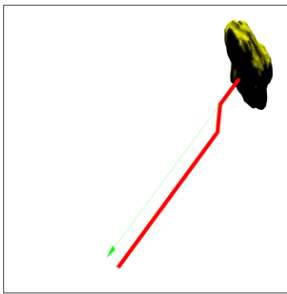
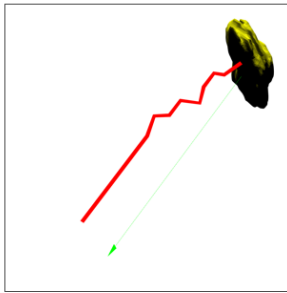
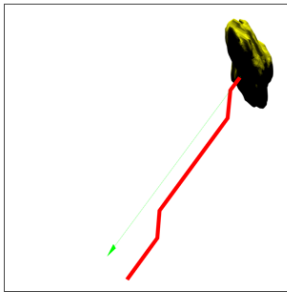
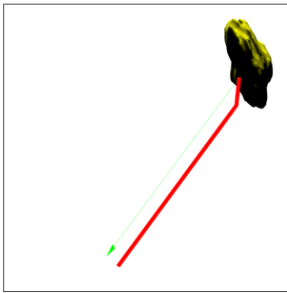

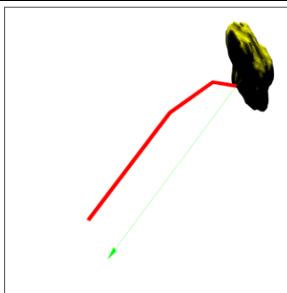
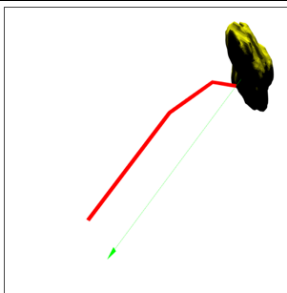
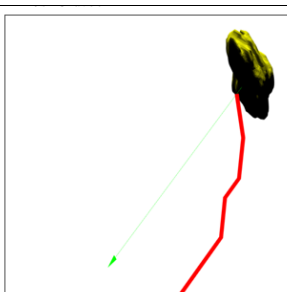
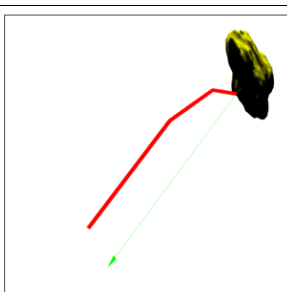
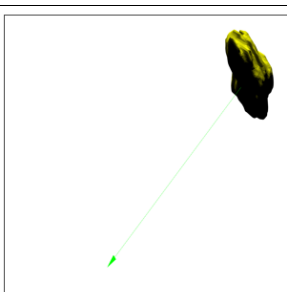
| 閾値 | C 値:左から 0.05, 0.2, 0.4 | | |
|-----|--|---|--|
| 0.6 |  C:0.05 completion |  C:0.2 completion |  C:0.4 completion |
| 0.4 |  C:0.05 completion |  C:0.2 completion |  C:0.4 completion |

表 5-3 ボクセル幅 10mm

| 閾値 | C 値:左から 0.05, 0.2, 0.4 | | |
|-----|--|---|--|
| 0.6 |  C:0.05 completion |  C:0.2 completion |  C:0.4 completion |
| 0.4 |  C:0.05 completion |  C:0.2 completion |  C:0.4 Failed |

ボクセルベース法の場合、ボクセル幅は切削経路の解像度を意味するので、幅が大きいほど検出した経路の折れ線具合は粗くなるのがわかる。C の値で直進性を制御することができ、値を大きくすると直進性が高くなった。ただし、真直度を優先しすぎると、検出ができなくなる。

閾値は切削経路の形状よりも、切削経路を検出できるかどうかに関与することがわかる。また、この表の画像では確認できないが、経路が血管に触れることも発生する。障害物領域の、ある程度の通過を許容できる方法であることが確認できた。

5-7 DICOM ボリューム

このようにして提案したボクセルベース法だが、濃度測定に、3D メッシュは必須ではない。いくつかの取り決めで、DICOM から直接測定することが可能である。2章で説明した DICOM 断層画像群から 3D メッシュを作成するセグメンテーション処理が省けることは、手術前シミュレーション準備の簡略化に貢献できる。そのため、ボクセルベース法では 3D メッシュではなく DICOM 断層画像群から自動生成できる DICOM ボリュームを利用するようにした。

DICOM ボリュームは DICOM 断層画像ごとの各画素値(測定値)を 2 次元配列とし、これを連続する画像分で 3 次元配列にしたものを指す(図 5-13)。

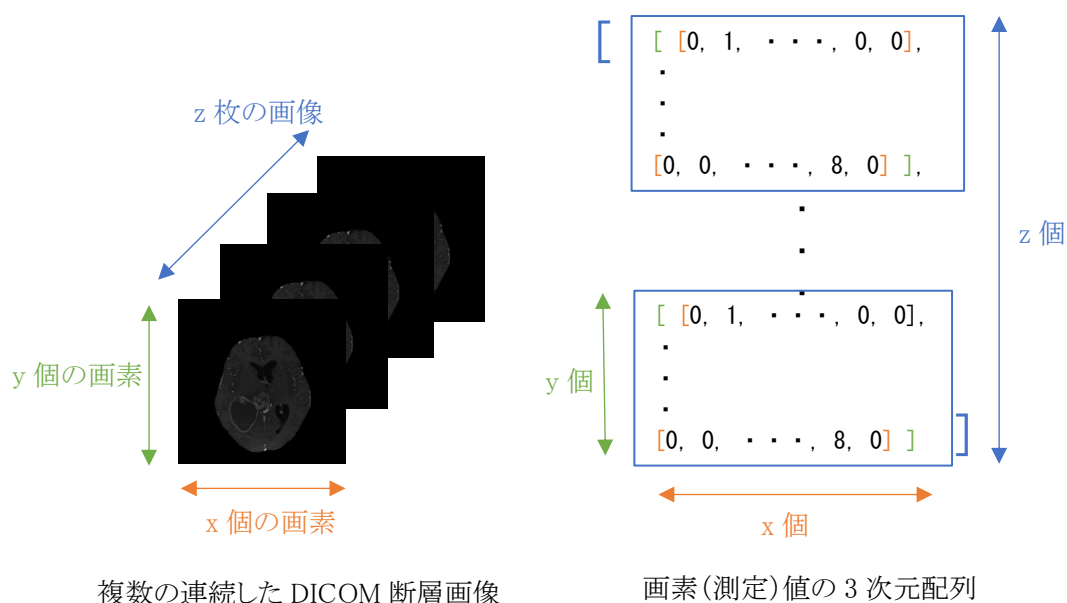


図 5-13 DICOM ボリューム。

5-8 DICOM ボリュームの利用と表示

この DICOM ボリュームから、どのようにメス先から腫瘍へのボクセル群を生成するかが、まず課題となるが、3D メッシュなら手術シミュレータにそのまま表示できていた血管やリンパ管などを、DICOM ボリュームからどのように表示するかも次の課題となる。

例えば、DICOM ボリュームが持つ測定値を、3D 空間に測定値の粒子として配置して表示するという方法(図 5-14)もある。しかし、これでは実際の体組織のつながりが表現できない。



DICOM の測定値を、3D 空間に配置して
測定値の粒子として表示する

図 5-14 粒子をそのまま表示.

5-9 レイマーチング法

そこで、配置した測定値粒子をそのまま表示するのではなく、測定値粒子内を視線にそって少しずつ進みながら、視線周辺の測定値粒子間の補正で、視線上の測定値を計算するというレイマーチング法を使用した(図 5-15)。

本研究の手術シミュレータでは、DICOM ボリュームを囲む、仮想の六面体を用意し、その六面体の表面にレイマーチングの結果を表示するようにしている。

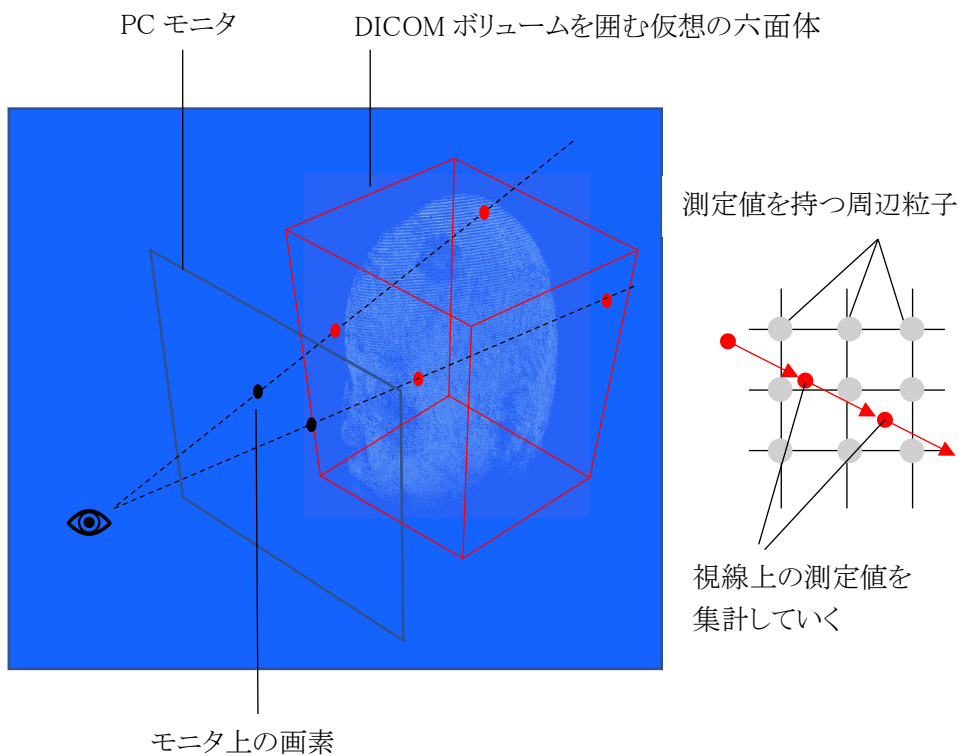


図 5-15 レイマーチング法.

レイマーチング法で表示した結果が**図 5-16**となる。本研究では、ボクセルの障害物濃度計算にも、このレイマーチング法を使うことにした。



レイマーチング法を使った DICOM ボリュームの表示

図 5-16 レイマーチング法での表示.

ボクセルを作成する過程で、カメラのパラメータ設定は 3D メッシュの時と変わらない。3D メッシュと違い、裏面を描くという処理はなく、nearZ 位置から farZ 位置に向けてレイマーチングを行うことになる。

5-10 疑似法線

濃度を計算するには、上述の処理で十分だが、シミュレータ画面としては視認性がよくない。この点を解決するために、粒子ごとに疑似法線を用意した。

X、Y、Z 軸にそって隣り合う粒子の測定値の勾配(**図 5-17**)から、疑似法線の X、Y、Z 成分それぞれの値を決定する(**図 5-18**)。

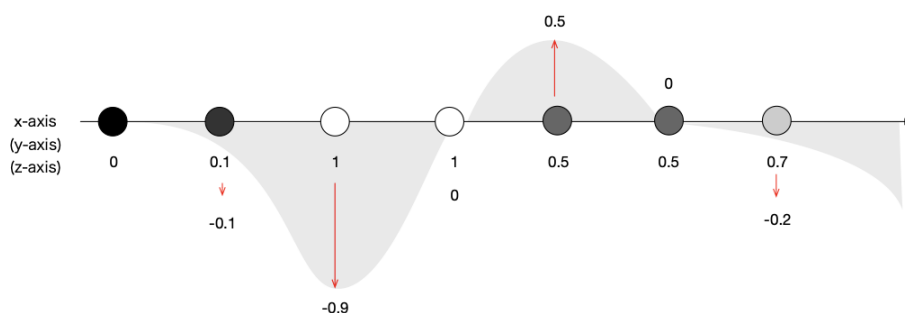


図 5-17 隣り合う粒子の測定値の勾配.

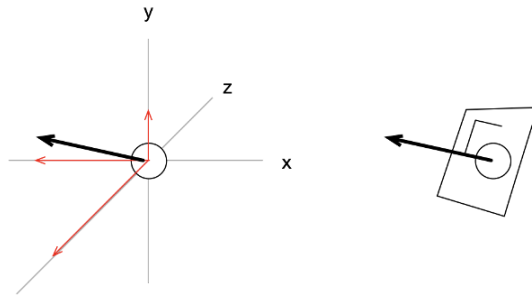
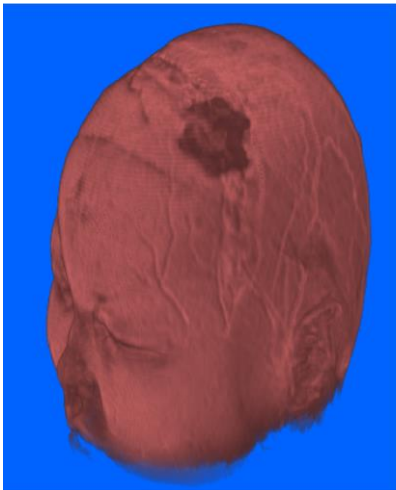


図 5-18 勾配値を X、Y、Z 成分にして法線生成.

図 5-19 に、シミュレータ画面表示のレイマーチング処理に、疑似法線を適用したものを示す。これでシミュレータでの視認性があがる。この疑似法線を適用するのはシミュレータの画面のみであり、ボクセルの濃度計算には使わない。



疑似法線適用版レイマーチング法を使った
DICOM ボリュームの表示

図 5-19 疑似法線を適用した表示.

レイマーチング法での表示は、DICOM の測定値を元に行っているため、閾値をもうけた表示で、生体組織の特定部分のみの表示を行うことになる。図 5-20 は閾値を 0.3、0.4、0.5 と変化させたもの。すべて閾値を中心に ± 0.15 の範囲を表示させている。DICOM の測定値は 0-1 に正規化している。閾値 0.5 で血管網を極端に表示した。測定値に色情報は無いので、ここでは、肌色を設定して濃淡を表示している。

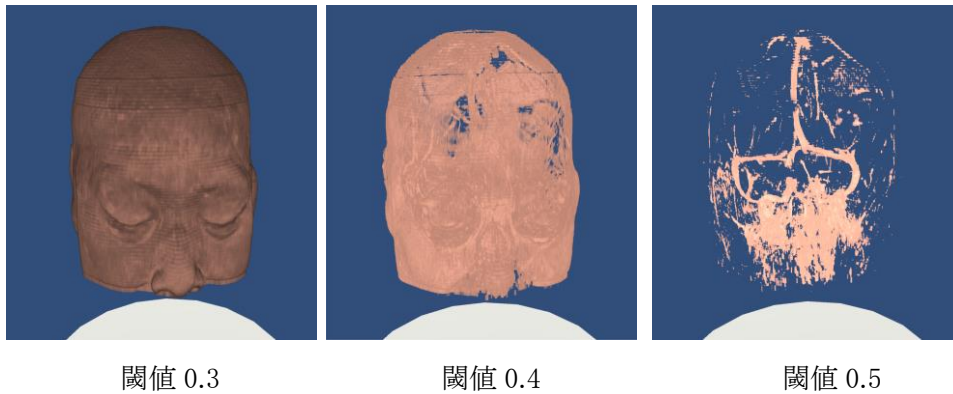


図 5-20 閾値による変更.

図 5-21 にレイマーチング法を使い作成したボクセル群を示す。作成時の閾値は、画面上の表示の閾値に合わせている。1つの立方体が、1つのボクセルを表す。ボクセルが持つ濃度は黒赤色のキューブの透明度で表現した。ボクセルのサイズに 10mmを指定したものは、領域の中に障害物が少ないので濃度が薄くなる。

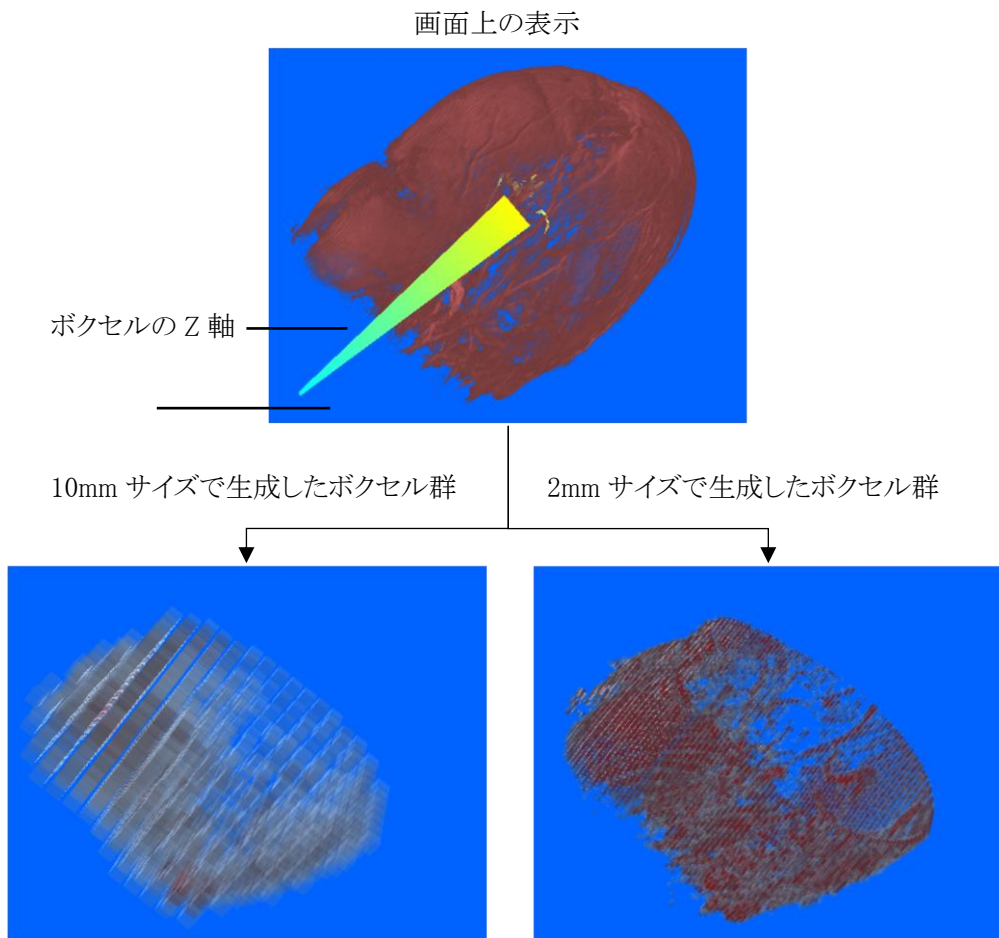


図 5-21 レイマーチング法で作成されたボクセル群.

また、レイマーチング中に、そのまま DICOM の測定値を使うのではなく、任意の距離関数を用意し、関数値が0以下の場合は、測定値に関わらず閾値外とすれば、距離関数の定義に従って DICOM の領域を切り取ることも可能となる。図 5-22 は、空間に置いた球体で DICOM の一部を削り取った状態を表示している。距離関数には式(23)を使った。

$$\text{length}(C_{sp} - R_{ray}) - R_d \quad (23)$$

length は 3D ベクトルの大きさを返す

C_{sp} 球体の中心位置 3D 座標ベクトル

R_{ray} レイマーチング中の視線の現在位置 3D 座標ベクトル

R_d 球体の半径

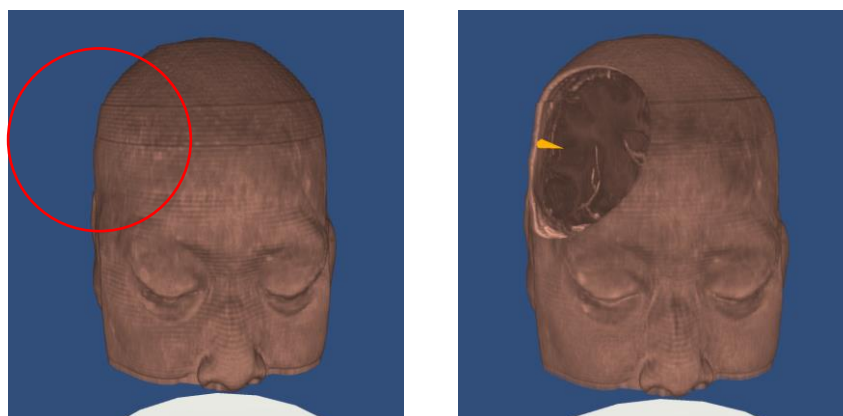


図 5-22 空間に置いた球体で一部を削り取った状態.

最後にボクセルベース法で検出した切削経路を、画面表示に合成したものを図 5-23 に示す。赤線がボクセルベース法で検出した経路となり、腫瘍周辺を黄色くしている。現在、腫瘍位置は手動で設定している。DICOM からどのように腫瘍を識別するかの研究には、まだ取り掛かっていない。

切削経路検索時の閾値は 0.4 ± 0.15 にした。画面上の頭部も同じ閾値で表示している。画面上の頭部表示に使う閾値と、経路探索に使う閾値は独立させており、連動させることも、別々に指定することも可能としている。

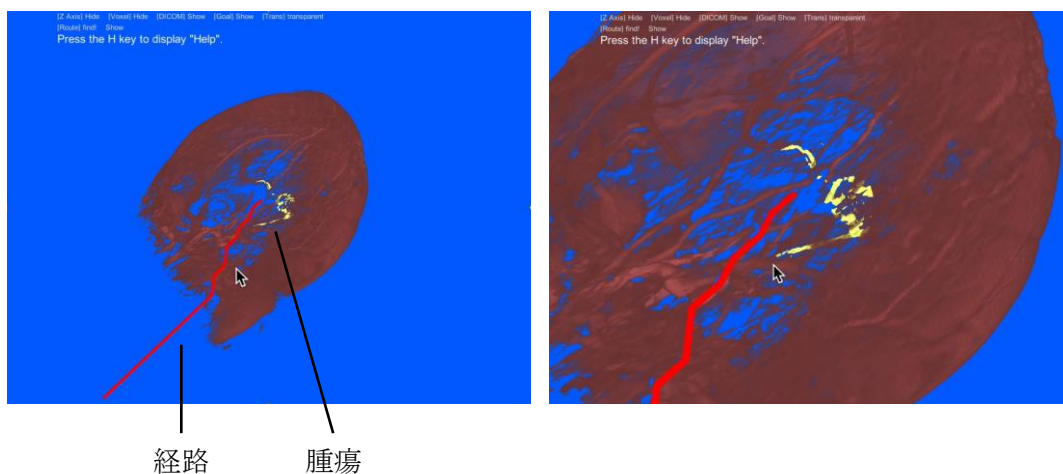


図 5-23 ボクセルベース法で検出した経路を合成した表示.

5-11 まとめ

ここでは、腫瘍周辺領域を面として切り取る切削を前提に、外科医が指定するメス位置から腫瘍まで、ある程度まっすぐな切削経路を示すための切削経路探索アルゴリズムとしてボクセルベース法を提案した。ボクセルベース法では、パラメータを変化させ直進性などがコントロールできることを確認した。また、その特性からDICOMを直接扱えるので、その対応をおこなった。DICOMの直接利用は、3D メッシュを作る手間を省き、患者を指定した手術シミュレーションを行うための前処理を省略できる。シミュレータでは、レイマーチング法を導入し、DICOM ボリュームから直接 2D 画面表示をおこなっている。この表示を利用して、ボクセルベース法のボクセル濃度測定もおこなうようにし、3D メッシュが不要となった。

第6章 DICOM ボリュームの複合表示

ここでは、手術シミュレータでの DICOM ボリューム表示の改良を説明する。CT と MRI で撮影した2種類の DICOM を融合させて表示するためのレイマーチング法の改良点を説明する。

6-1 DICOM の自動セグメンテーション

現在開発中の手術シミュレータは、肝臓に限らず脳外科などにも対応していこうと調査中である。その中で、頭部の手術シミュレータの在り方として、脳神経外科医の希望を伺ったところ、表皮を切削し、頭蓋骨が見え、これを開けると脳が見えるようなものを製作してほしいとの希望が出された。

以前の、3D メッシュを使った手術シミュレータであれば、それほど難しくはないが、現在の DICOM ボリュームを直接表示するシミュレータでは難しい。そもそも DICOM の情報に、肌、骨、内臓といった判別情報は無い。特に、脳外科でよく使われる MRI 撮影の DICOM は、数値からだけでは、骨と他の組織の識別ができない(図 6-1)。



図 6-1 MRI では、骨を識別する閾値で頭蓋内部も表示されてしまう。

このため、人間が測定値に基づいた画面表示を見ながら領域を指定していき、より分ける必要がある。2章で述べたセグメント化のことで、3D メッシュはこの工程を経た表示といえる。セグメント化作業を自動化しようと、機械学習を使ったクラス分けなどの研究[36]もおこなわれている。いずれは、このような自動化されたセグメント化作業により、MRI 撮影の DICOM を肌、骨、内臓というように色分けしたいと考えるが、現時点での実現は難しい。

ここでは別のやり方として、レイマーチング法で CT 撮影の DICOM を参照しながら MRI 撮影の DICOM の表示を行う方法を提案する。

6-2 2つの DICOM の利用

まず、骨のセグメンテーションは、MRI 撮影の DICOM で閾値を使うだけでは曖昧になるが、CT 撮影の DICOM だと閾値だけで明確に行える(図 6-2)。

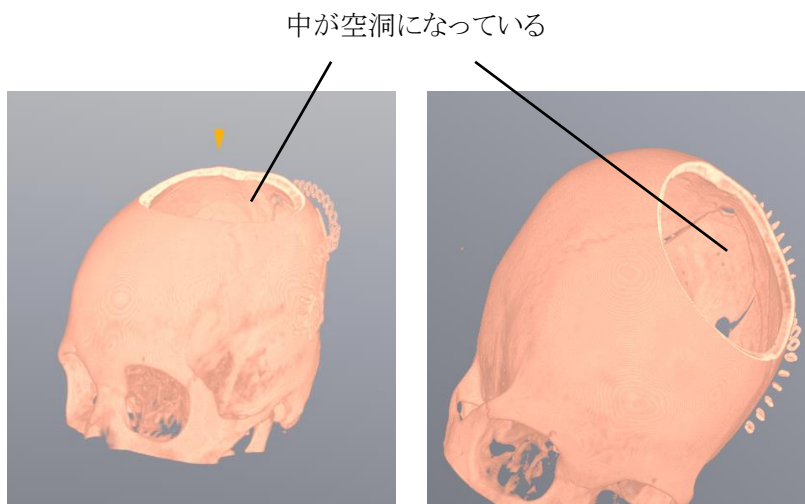


図 6-2 CT では、骨を識別する閾値で頭蓋内部は空洞になる。

このことから、レイマーチング中に MRI と CT を同時に参照し、CT 側で閾値範囲内の値がある場合、CT 側の値を優先し、骨の色で表示するようにした。また、骨を通過した後は、MRI で割り当てる色を変更し、表皮や内部の脳の色の変化にも対応した(図 6-3)。

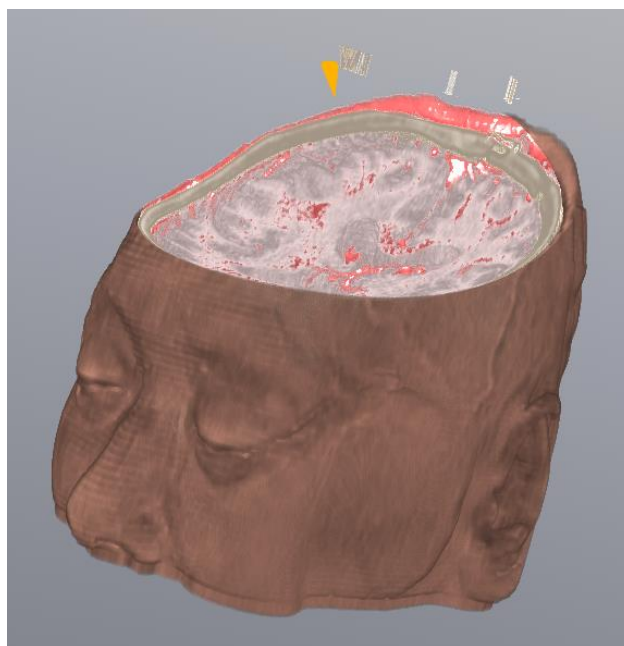


図 6-3 MRI と CT の融合表示。

このように、CT の領域を参考にしながら、MRI の表示色を変えることで、単純に2つを重ね合わせる以上の効果が期待でき、従来、手動で指定しなければならない領域分けが、閾値の調整程度で可能となった。

5章で実施した距離関数を使った切り取りの際にも、これを応用することが可能となる。骨を通過した領域かどうかを判定し、通過した領域であれば、切り取りをキャンセルする。このようにすると、**図 6-4** のように骨の内部にある脳だけを残し、外部を削り取るという表示が可能になる。

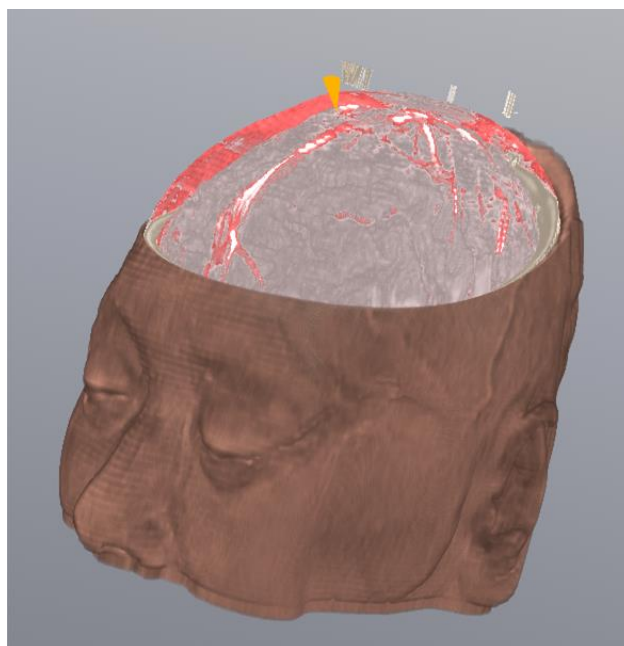
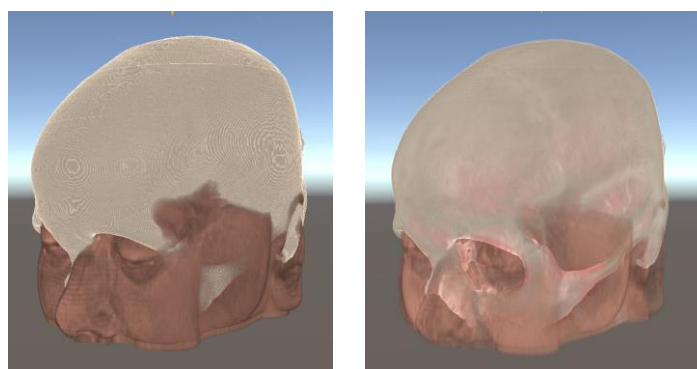


図 6-4 脳だけを残し外部を削り取る.

6-3 2つの DICOM の位置合わせ

同時に撮影するわけではないので、CT と MRI で各々撮影した DICOM の初期位置・姿勢はずれている。実際、本研究で使った同一患者の CT 撮影の DICOM と MRI 撮影の DICOM をそのまま読み込ませると、**図 6-5** のような位置関係となり、位置や姿勢、および大きさも微妙にずれている。



不透明表示

半透明表示(参考用)

図 6-5 初期状態では CT と MRI の位置は一致しない.

この点は、MRI 側の DICOM を基準として、CT 側の DICOM を合わせるように手動で調整した。皮膚表面の形状を利用し、2章で使われた深度バッファを使えば、自動位置合わせも可能と考えるが、本研究では今後の課題とした。

6-4 レイマーチング法の対応

レイマーチングで本研究のような表示を行うためには、レイマーチング中 MRI の DICOM の内部位置から、CT 側の DICOM が対応する内部位置を計算し、参照できるようにする必要がある。これには MRI と CT で撮影したそれぞれの DICOM ボリュームに仮想空間での傾き、位置、スケール情報を持たせることで対応する。そのために各 DICOM ボリュームにはアフィン行列を持たせることにした。2つの DICOM ボリュームが属する仮想空間座標をワールド座標とし、それぞれがどのくらい傾き、原点からずれ、スケールされたかをアフィン行列に持たせている。

例えば、MRI 用アフィン行列を A_{mri} 、CT 用を A_{ct} とするなら、式(24)で合成したアフィン行列をレイマーチングの位置座標ベクトルにかけることで、MRI 側の位置から CT 側の対応する位置へ変換することができる。

$$A_{mri}^{-1} A_{ct} \quad (24)$$

6-5 まとめ

ここでは、レイマーチング法で、MRI と CT で撮影した2種類の DICOM ボリュームを同時に参照することにより、画面表示の拡張をおこなった。頭部表示において、CT を使った骨領域認識で、MRI 側の表皮と頭蓋内部とで、測定値・色変換処理を変更することが可能になった。このため、手動によるセグメント化無しに、肌、骨、内臓といった色表現が、手術シミュレータで可能となる。CT 撮影の DICOM の代わりに、造影剤を変更した MRI 撮影の DICOM という組み合わせも考えられる。切削経路探索でもこの複合表示は利用価値があると考えられる。

しかし、本研究は、頭部だからできる対応であり、どの部位にも適用できるものではない。複数の DICOM ボリュームを同時に参照することでどのようなことができるかは、将来の研究課題

としたい。現行、手動で行っている、MRIとCTで撮影した2種類のDICOMの位置・姿勢合わせの自動化も今後の課題とする。

第7章 おわりに

最後に本研究で得られた成果と考察、今後の課題について述べる。

7-1 成果と考察

本研究では次のような成果と考察が得られた。

1. 3D メッシュ臓器においてメスが血管や腫瘍に食い込みことを防ぐ、深度バッファを利用した距離計算による医療事故防止システムの製作。
2. 3D メッシュ臓器における、ポテンシャルベース法による手術ナビゲーション(メス先の血管群を回避した腫瘍までの誘導)。ここでは、多種のパラメータを試行錯誤で調整し、触覚デバイスがグリップに発生させる力によって、外科医の指先を誘導できた。ただ、危険部位に切り込んだ時の極端な反発反応はパラメータ調整では対応できなかった。この点において、アルゴリズムのさらなる改良が必要である。
3. 3D メッシュ臓器におけるセンサーベース法による手術ナビゲーションアルゴリズムの製作。
4. 3D メッシュ臓器におけるポテンシャルベース法とセンサーベース法の手術ナビゲーションの比較。センサーベース法は入り組んだ切削経路を見つけがちで、利用には注意が必要と考える。全体の切削方向は外科医が決め、数センチといった局所的な領域でナビゲーションを使うのが、これらのアルゴリズムの適切な使い方なのではとも考えている。これらは脳神経外科医との擦り合わせが必要である。
5. DICOM ボリューム臓器におけるボクセルの濃度値による手術ナビゲーションアルゴリズムの製作。
6. DICOM ボリューム臓器の複合表示をおこなう手術シミュレータ。CT、MRI、それぞれの特性を生かすことで、DICOM ボリュームの正確な表示に効果があることを確認した。今後も組み合わせ方を考えていきたい。

7-2 課題

今後の課題としては次のようなものがある。

1. 現在の線による移動を示す切削経路ナビゲーションから、臓器の切り取るべき面を示す平面切削ナビゲーションへの拡張。
2. 例えば脳なら、前頭葉、側頭葉といった、MRI や CT の数値では差異が見られない領域に対し DICOM ボリュームのセグメント分けが可能かを探る。この場合、一般の医療用モデルとして、前頭葉、側頭葉に分けられ用意されている、それぞれの 3D メッシュの、姿勢、位置、スケールを調整し、表示中の DICOM ボリュームに合わせたうえで参照するなどが考えられる。
3. 複数種 (CT、MRI) の DICOM 測定値を融合させた DICOM ボリュームの作成。

4. 各 DICOM ボリューム間の、表面形状を使った自動位置合わせ。例えば、各々の DICOM 脳の表面点群が自動抽出できれば、前述の Iterative closest point (ICP)[8]でそれらを正確に重ね合わせることができる。
5. 手術中の実物との合成表示。没入型デバイスを使ったパススルー機能による、カメラを通した実空間と仮想空間の合成表示。

謝辞

本論文を結ぶにあたり、長年にわたる多大なご指導、ご鞭撻を賜りました大阪電気通信大学大学院総合情報学研究科総合情報学専攻コンピュータサイエンスコース 登尾啓史教授に深く感謝申し上げます。また、長年にわたる共同研究において、何度も手術現場を臨場体感させて頂き、ゼミナール等で数多くのご意見やコメントを頂きました京都大学医学研究科泌尿器科教室の小川修名誉教授、関西医科大学外科学講座の海堀昌樹教授、松井康輔講師、関西医科大学脳神経外科講座の埜中正博教授、岡山県立大学の小枝正直准教授、株式会社 Embedded Wings の水篠公範様、大阪国際大学短期大学部ライフデザイン学科の朝野美穂講師、奈良学園大学看護学科の溝口みちる助教、大阪電気通信大学情報学科の大西克彦教授、渡辺郁教授、大阪電気通信大学医療福祉工学科(現医療科学科)の橘克典准教授に深く感謝申し上げます。

本論文の作成にあたり、主査を引き受けて頂きました大阪電気通信大学大学院総合情報学研究科総合情報学専攻コンピュータサイエンスコース 登尾啓史教授には多くの貴重なご意見を頂き、深くお礼申し上げます。

本論文の作成にあたり、副査を引き受けて頂きました大阪電気通信大学大学院総合情報学研究科総合情報学専攻コンピュータサイエンスコース 上善恒雄教授、升谷保博教授には多くの貴重なご意見を頂き、心よりお礼申し上げます。

参考文献

- [1] Satou, S., Aoki, T., Kaneko, J., Sakamoto, Y., Hasegawa, K., Sugawara, Y., Arai, O., Mitake, T., Miura, K., Kokudo, N., “Initial experience of intraoperative three-dimensional navigation for liver resection using real-time virtual sonography,” *Journal of Surgery*, Vol.155, No.2, pp.255–262, Feb. 2014.
- [2] Nicolas, C. B., Francesco, V., François, P., Christian, T., Matteo, F., Kate, G., Pietro, E. M., Matthias, P., Stefan, W., Philippe, M., “Augmented environments for the targeting of hepatic lesions during image guided robotic liver surgery,” *J. Surg. Res.*, Vol.184, No.2, pp.825–831, Oct. 2013.
- [3] Pessaux, P., Diana, M., Soler, L., Piardi, T., Mutter, D., Marescaux, J., “Towards cybernetic surgery: robotic and augmented reality-assisted liver segmentectomy,” *Langenbecks Arch. Surg.*, Vol.400, No.3, 381–385, April 2015.
- [4] Koeda, M. et al., “Depth Camera Calibration and Knife Tip Position Estimation for Liver Surgery Support System,” In: Stephanidis, C. (eds) *HCI International 2015 – Posters’ Extended Abstracts*. HCI 2015. Communications in Computer and Information Science, vol 528. pp.496-502, Springer, Cham. 2015.
- [5] Kobayashi, M., Sato, I., and Nakamura, R., “Surgical navigation with distance sensation using force feedback for robotic surgery,” *J. Medical Imaging and Health Informatics*, Vol. 3, No. 1, pp. 120-124, March 2013.
- [6] Onishi, K. et al., “Virtual Liver Surgical Simulator by Using Z-Buffer for Object Deformation,” In: Antona, M., Stephanidis, C. (eds) *Universal Access in Human-Computer Interaction. Access to Learning, Health and Well-Being. UAHCI 2015. Lecture Notes in Computer Science*, vol 9177., pp.345-351, Springer, Cham., 2015.
- [7] Zhang, Zhengyou, “Iterative point matching for registration of free-form curves and surfaces,” *International Journal of Computer Vision*. Vol.13, No.12, pp.119-152, 1994.
- [8] Noborio, H., Onishi, K., Koeda, M., Watanabe, K., and Asano, M., “Depth-Depth Matching of Virtual and Real Images for a Surgical Navigation System,” *International Journal of Pharma Medicine and Biological Sciences*, doi: 10.18178/ijpmb.10.2.40-48, Vol. 10, No. 2, pp. 40-48, April 2021.
- [9] Noborio, H., Aoki, K., Kunii, T. and Mizushino K, “A Potential Function-Based Scalpel Navigation Method that Avoids Blood Vessel Groups during Excision of Cancerous Tissue,” *Proceedings of the 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC’16)*, pp.6106-6112, Orlando Florida USA, 2016.
- [10] Khatib, O., “The Potential Field Approach And Operational Space Formulation In Robot Control,” In: Narendra, K.S. (eds) *Adaptive and Learning Systems*. Springer, Boston, MA., pp.367-377, 1986.
- [11] Lumelsky, V., “Algorithmic issues of sensor-based robot motion planning,” *Proc. of the 26th IEEE Conference on Decision and Control*, pp. 1796-1801, 1987.

- [12] Kunii, T., Asano, M., Noborio, H., "Voxel-Based Route-Search Algorithm for Tumor Navigation and Blood Vessel Avoidance," In: Kurosu, M. (eds) Human-Computer Interaction. Interaction Techniques and Novel Applications. HCII 2021. Lecture Notes in Computer Science, vol 12763. Springer, Cham. pp.566-581, 2021.
- [13] Kunii, T., Asano, M., Noborio, H., "DICOM-Based Voxel-Supported Blood-Vessel-Avoiding Scalpel Navigation," In: Kurosu, M. (eds) Human-Computer Interaction. Technological Innovation. HCII 2022. Lecture Notes in Computer Science, vol 13303. pp.251-263, Springer, Cham, 2022.
- [14] Kunii, T., Watanabe, K., Mizoguchi, M., Noborio, H., "System to check organs, malignant tumors, blood vessel groups, and scalpel paths in DICOM with a 3D stereo immersive sensory HMD," HCII 2023, (to appear).
- [15] マーチングキューブ法:Wikipedia. <https://ja.wikipedia.org/wiki/マーチングキューブ法>. (オンライン).
- [16] Besl, P. J., McKay, N. D., "A method for registration of 3-D shapes," IEEE Trans. Pattern Anal. Mach. Intell., Vol.14, No.2, pp.239-256, 1992.
- [17] Zhang Z., "Iterative point matching for registration of free-form surfaces," Int. Journal of Computer Vision, Vol.13, No.2, pp.119-152, 1994.
- [18] Granger, S., Pennec, X., "Multi-scale EM-ICP: A fast and robust approach for surface registration," Proc. of the 7th European Conference on Computer Vision, Vol.4, pp.69-73, 2002.
- [19] Liu, Y., "Automatic registration of overlapping 3D point clouds using closest points," Image and Vision Computing, Vol.24, Issue 7, pp.762-781, 2006.
- [20] Salvi, J., Matabosch, C., Fofi, D., Forest, J., "A Review of Recent Range Image Registration Methods with Accuracy Evaluation," Image and Vision Computing, Vol.25, Issue 5, pp.578-596, 2007.
- [21] Rusu, R. B., Cousins, S., "3D is here: point cloud library (PCL)," Proc. of the IEEE Int. Conf. Robotics and Automation, pp.1-4., 2011.
- [22] Wu, Y. F., Wang, W., Lu, K. Q., Wei, Y. D., Chen, Z. C., "A new method for registration of 3D point sets with low overlapping ratios," Proc. of the 13th CIRP conference on Computer Aided Tolerancing, Vol.27, pp.202-206, 2015.
- [23] Kunii, T., Asano, M., Fujita, K., Tachibana, K., Noborio, H., "Comparative Study of Potential-Based and Sensor-Based Surgical Navigation in Several Liver Environments," In: Kurosu, M. (eds) Human-Computer Interaction. Interaction Techniques and Novel Applications. HCII 2021. Lecture Notes in Computer Science, vol 12763. pp.551-565, Springer, Cham., 2021.
- [24] Noborio, H., Kawai, K., Watanabe, K., Tachibana, K., Kunii, T., Mizushino, K., "Deadlock-Free and Collision-Free Liver Surgical Navigation by Switching Potential-Based and Sensor-Based Functions," In: Kurosu, M. (eds) Human-Computer Interaction. Human Values and Quality of Life. HCII 2020. Lecture Notes in Computer Science, vol 12183. pp. 604-622, Springer, Cham. 2020.

- [25] Bentoumi,H., Gautron,P., and Bouatouch,K., "GPU-Based Volume Rendering for Medical Imagery," International Journal of Electrical, Computer, and Systems Engineering, Vol.4. No.1, pp.36-42, 2010.
- [26] RASMUSSEN, N., NGUYEN, D. Q., GEIGER, W., AND FEDKIW, R., "Smoke simulation for large-scale phenomena. " Proc. of SIGGRAPH 2003, pp. 703-707. 2003.
- [27] Graphics Runner: Volume Rendering 102: Transfer Functions : graphicsrunner.blogspot.com. <http://graphicsrunner.blogspot.com/2009/01/volume-rendering-102-transfer-functions.html>. (オンライン) .
- [28] 3D Slicer (3D Slicer image computing platform): <https://www.slicer.org>. (オンライン)
- [29] itk-SNAP home. itk-SNAP. <http://www.itksnap.org/pmwiki/pmwiki.php>. (オンライン)
- [30]<https://learn.microsoft.com/ja-jp/archive/msdn-magazine/2014/june/directx-factor-the-canvas-and-the-camera>
- [31] Noborio H., Kunii T., Mizushino K., "Comparison of GPU-based and CPU-based Algorithms for Determining the Minimum Distance between a CUSA Scalper and Blood Vessels." Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2016) – Volume 3: BIOINFORMATICS, Rome, Italy, pp.128-136, February 21-23, 2016.
- [32] Lin, M., and Canny, J., "A fast algorithm for incremental distance calculation," Proceedings of the Robotics and Automation, Vol.2, pp.1008-1014, 1991.
- [33] Gilbert, E., Johnson, D., and Keerthi, S., "A fast procedure for computing the distance between complex objects in three-dimensional space, " IEEE J. of Robotics and Automation, Vol. 4, No.2, pp.193-203, 1988.
- [34] Noborio,H., Fukuda,S., and Arimoto,S., "Fast Interference Check Method Using Octree Representation", Journal of Advanced Robotics, Vol.3, No.3, pp.193-212, Sept. 1989.
- [35] 吉岡 孝, 登尾 啓史, 富永 昌治, "移動ロボットのセンサベースナビゲーションについて," 日本ロボット学会誌, 19 卷 8 号, pp.991-1002. 2001..
- [36] Sergio P., Adriano P.,Victor A.,Carlos S. "Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images," IEEE Transactions on Medical Imaging 35(5):1-1, March 2016.