

示す治具の回転中心から術具模擬体までの距離と回転半径の比較により行った。角度に関する精度評価は、図 56 の各回転角度に対して行った。

実験手順について説明する。まず、内視鏡レンズの歪みは、レンズ固有の歪みであるから事前実験により算出しておく。この歪みの情報は、仮想内視鏡画像を作成する際に用いる。次に、ある位置・方向で設定した術具模擬体の内視鏡画像を取得する。取得した画像に対して、平均相互情報量が大きくなるように仮想術具の位置・方向を変更していく。仮想術具の位置は 0.01mm、角度は 0.1deg の精度で変更することができる。取得した実内視鏡画像と仮想内視鏡画像の平均相互情報量が最大となったところで、仮想術具の位置・方向を記録する。このようにして検出した術具の位置・方向と、治具で設定した位置・角度とを比較し、それぞれの精度を評価した。ただし、この実験では、目視で実際の内視鏡画像と仮想内視鏡画像を比較して術具の凡その位置を合わせこみ、その後に平均相互情報量が最大となる位置・方向を手動で探索する手段を採った。

図 58-60 に実験結果の一例を示す。図 58 は術具模擬体を写した実内視鏡画像である。この画像との平均相互情報量が最大となった仮想内視鏡画像を図 59 に示す。図 34 は仮想術具をワイヤーフレームにして実内視鏡画像上に重畳した画像である。なお、仮想内視鏡画像を作成するのに要する処理時間は、一画像あたり 160msec 程度であった。

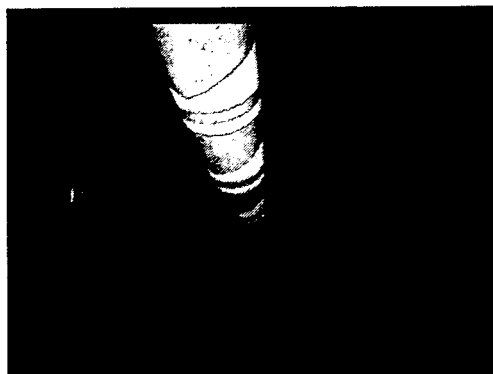


図 58 実内視鏡画像(距離 30mm)

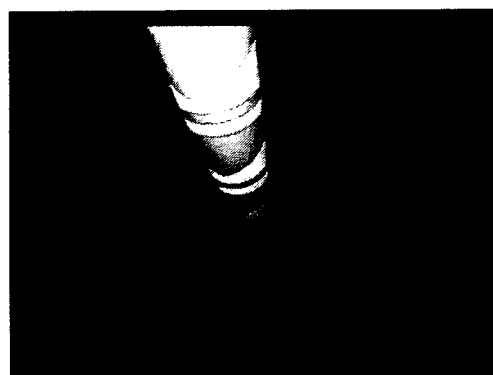


図 59 仮想内視鏡画像

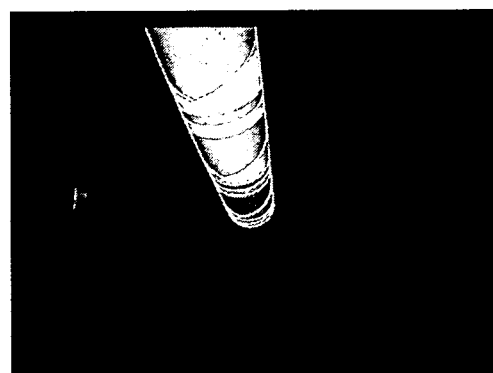


図 60 実内視鏡画像+ワイヤーフレーム

精度評価結果について説明する。内視鏡と術具模擬体の距離は 10mm, 30mm, 50mm、各回転軸の角度は 0deg, 30deg, 60deg に設定し、親内視鏡模擬体と治具が干渉する場合を除く 39 パターンの計測を行った。この各パターンにおけるそれぞれの設定値と検出値との差分値を計算した。

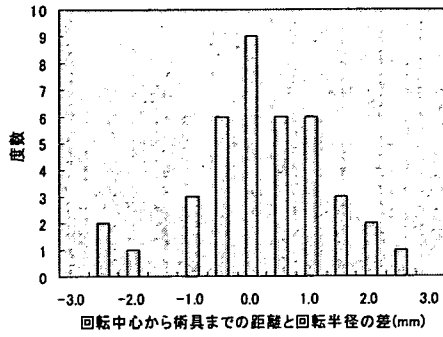


図 61 回転中心と術具の距離の差の分布

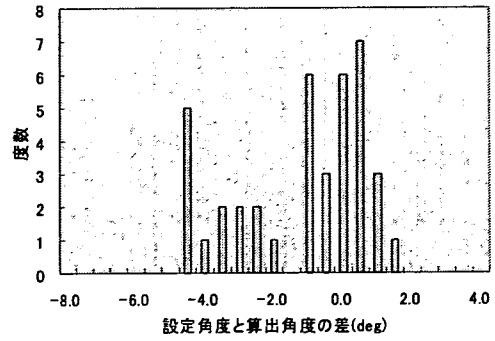


図 62 回転軸 1 の角度の差の分布

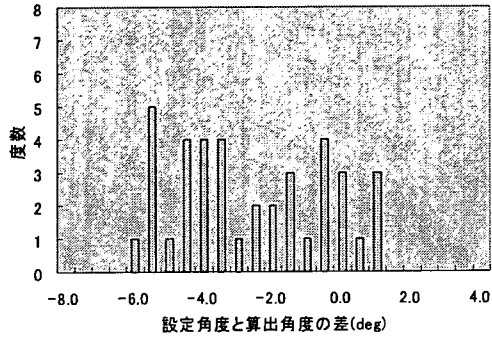


図 63 回転軸 2 の角度の差の分布

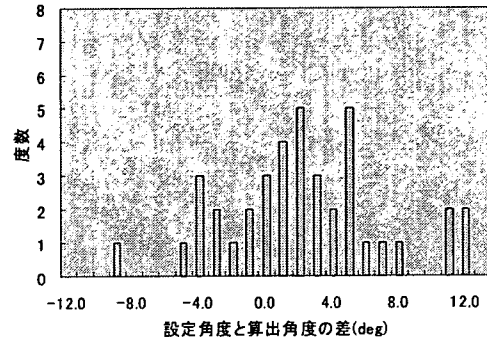


図 64 回転軸 3 の角度の差の分布

表 1 術具位置センシング手法の精度評価結果

	平均値	標準偏差	上限値 下限値
回転中心と術具の距離の差	-0.07mm	±1.15mm	+2.45mm -2.90mm
回転軸 1 の角度の差	-1.43deg	±1.92deg	+1.31deg -4.85deg
回転軸 2 の角度の差	-2.91deg	±2.15deg	+0.78deg -6.00deg
回転軸 3 の角度の差	-1.81deg	±4.78deg	+11.92deg -9.61deg

この差分値の分布を図 61-64 に表す。表 1 には、それぞれの差分値の平均値、標準偏差および上限値・下限値を示す。

次に、処理の高速化を図るために実施した、術具の位置・姿勢の探索自由度を限定化する処理について説明する。上述の平均相互情報量を用いた手法は原理的には優れた手法であるが、三次元空間上で術具の位置・姿勢を決定するためには六自由度のパラメータを決定する必要があり、処理時間の面で現実的ではない。仮想内視鏡画像の作成から平均相互情報量を算出するまでに要する時間は、一画像当たり 200msec 程度である。一自由度当たり 10 通りのサンプリングを行い、六自由度に関してすべての組み合わせを考えると、処理時間は $2 \times 105 \text{sec}$ すなわち 55.5h となる。したがって、如何に最大値探索処理を最適化するかが課題となる。

前述した、実術具側面の二つのエッジを用いた自由度を限定化する処理は、六自由度のうち四自由度を決定するので処理の高速化が見込める。しかし、内視鏡画像は広角画像であるため、直線検出などの処理を行うには適していない。したがって、レンズ歪み補正処理を施した後、直線を直線として検出する処理が可能になるように、遠近歪み補正を行う必要がある。

図 65-67 に内視鏡画像のレンズ歪み補正処理結果を示す。内視鏡レンズの歪みは前述した事前実験により算出した歪み補正マップを用いる。そのマップを基にレンズ歪み補正した画像を図 66 に示す。この補正画像に対して、遠近歪み補正を施した画像を図 67 に示す。図から明らかなように、術具の背景にある格子がほぼ直線上に並んでい

ることが確認できる。このように内視鏡画像を遠近歪み補正することにより、実術具側面の二つのエッジを直線として認識することが可能になり、二つのエッジの画像上での傾きから、術具の三次元空間上における配置を限定化することができる。

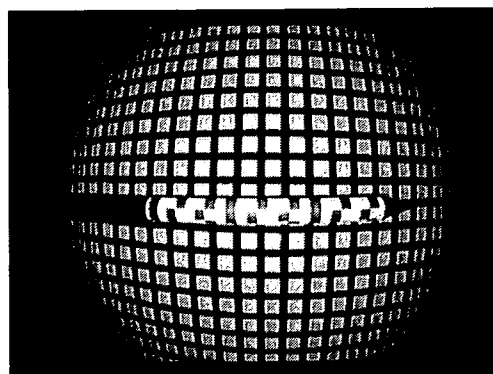


図 65 実内視鏡画像

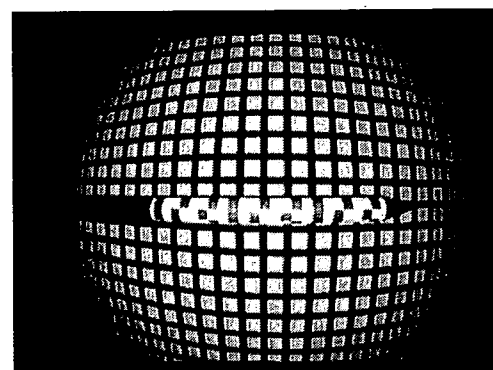


図 66 レンズ歪み補正画像

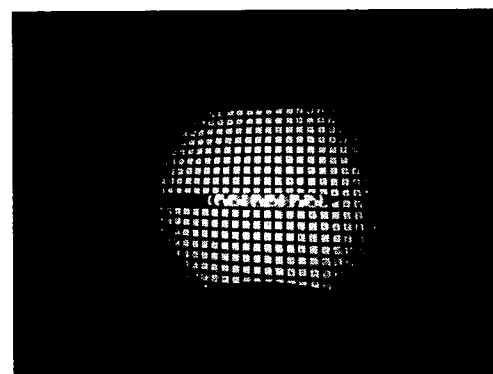


図 67 遠近歪み補正画像

上記の処理を行った後、平均相互情報量を用いた術具位置センシング手法で、術具の位置を検出した結果を図 68-70 に示す。

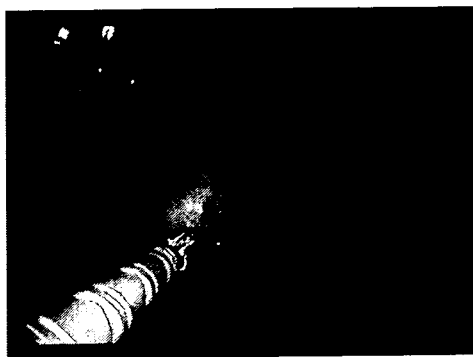


図 68 実内視鏡画像

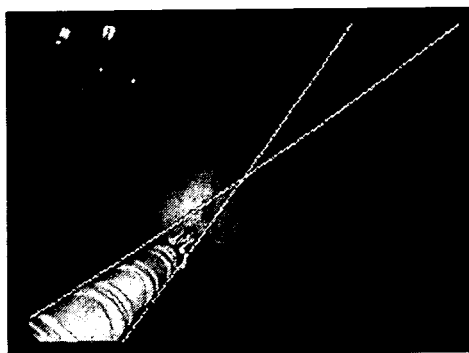


図 69 術具側面エッジ処理

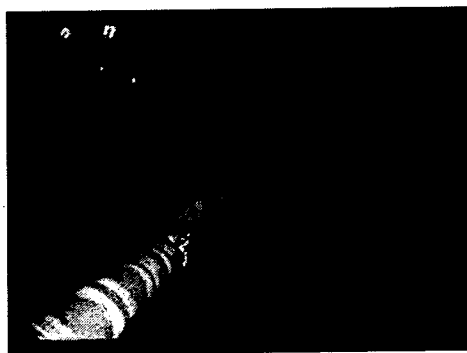


図 70 術具位置検出結果

図 68-70 は、実術具の二つの側面エッジの直線検出結果に対し、遠近歪み補正変換およびレンズ歪み補正変換の逆変換を施し、実内視鏡画像上に重畳した画像である。実

術具の側面エッジを確かに捉えていることが確認できる。処理時間はおよそ 1.5sec である。次に、算出した四自由度を固定し、平均相互情報量を用いた手法により、残りの二自由度を決定する。まず、術具が存在する凡その位置から開始し、平均相互情報量が少しでも大きくなるパラメータ方向を探索する。

どのパラメータ方向に行っても平均相互情報量が小さくなる場合には、最大値に達したとして二つのパラメータを決定する。ただし、実際には、既に決定している四自由度のパラメータについても当然誤差があると考えられるので、一度最大値に達した後に、六自由度のすべてのパラメータ空間に対して、平均相互情報量が最大となる方向を再び探索する。この処理において、平均相互情報量が最大値に達したところで、最終的な術具の位置・姿勢の検出結果として表示する。図 68 の実内視鏡画像に対して、すべての処理を行い、最終的に得られた結果を図 70 に示す。図 69 は、実内視鏡画像の上に、検出した位置・姿勢での仮想内視鏡画像を重畳している。すべての処理が終了するまでに要する時間は、画像によってかなりばらつきはあるが、平均的に 1.5min 程度である。

〔III〕 手術支援基盤システムの開発

〔領域抽出〕

手術ナビゲーションの処理機能を十全とするためには、臓器・腫瘍領域を手術ナビゲーションシステムのモニタ上に表示し、それに付随するさまざまな情報を提示する必要がある。そのため、術前・術中における臓器・腫瘍領域抽出機能開発は、手術ナ

ビゲーションシステムにとって欠くことのできない重要な項目である。

本研究では、腫瘍性病変の識別能向上を目的として、医用診断画像による病変部領域抽出手法を開発した。開発した手法は SNAKES と呼ばれる動的輪郭法 (Active Contour Model) を応用したものである。

SNAKES を表す変数として区間[0,1) 上で定義されるパラメータ s を導入する。輪郭線上の全ての座標は、このパラメータ s によって決定される。すなわち、輪郭線上の任意の点の座標 v は (式 2) で表される。

$$v = v(s) \quad (\text{式 2})$$

ここで、SNAKES の内部エネルギー ε_{inter} を (式 3) で定義する。

$$\varepsilon_{inter}(s) = \frac{\alpha}{2} \left| \frac{dv(s)}{ds} \right|^2 + \frac{\beta}{2} \left| \frac{d^2v(s)}{ds^2} \right|^2 \quad (\text{式 3})$$

ここで、 α 、 β は、正定値のフリーパラメータであり、経験則に則って決められる。右辺第 1 項は、SNAKES 上の隣接した 2 点間の距離に比例したエネルギーを表しており、SNAKES の収縮性に対応する。右辺第 2 項は、SNAKES 上の隣接した 2 点間の位置差分ベクトルの差分の二乗に比例したエネルギーであるので、その 3 点が直線からどれだけずれているかを表し、SNAKES の直線性に対応する。

次に、輪郭抽出のために画像の輝度勾配が大きいところでエネルギーが小さくなるように画像エネルギー $\varepsilon_{imag}(s)$ を、(式 4) で表わす。

$$\varepsilon_{imag}(s) = -\omega |\nabla I(v(s))|^2 \quad (\text{式 4})$$

ただし、 ∇ は v による微分であり、 I は座標 v における画像輝度である。ここで、 ω は、正定値のフリーパラメータであり、経験則に則って決められる。これより、画像エネルギーは最大値がゼロ値であり、輝度勾配が大きくなればなるほど値が小さくなるのが理解できる。

輪郭線に不自然な凹凸が発生すると内部エネルギーが大きな状態になりうるので、これを押さえるために、外部から与える拘束のエネルギーを $\varepsilon_{exter}^{(pressure)}(s)$ とし、(式 5) で与える。

$$\varepsilon_{exter}^{(pressure)}(s) = \frac{\eta}{2} |v(s) - v_{c.m.}|^2 \quad (\text{式 5})$$

ここで、 η は、圧力ポテンシャルのフリーパラメータであり、経験則に則って決められる。 $v_{c.m.}$ は収束先を示す座標であり、通常は輪郭線の重心が用いられる。つまり、輪郭線上の各点はその重心に向かって距離に比例した圧力を感じるようになる。これにより、全エネルギーの増加を抑え、凹凸のある対象も抽出可能となる。

さらに、輪郭線が形成する閉領域の面積が大きいほど輪郭線に内向きの力を発生させる面積力ポテンシャルエネルギーを定義する。このポテンシャルエネルギー導入の目的は、凹領域に輪郭線を浸入させることである。面積項 $\varepsilon_{exter}^{(area)}(s)$ は、(式 6) で表わされる。

$$\begin{aligned} \varepsilon_{exter}^{(area)}(s) &= \frac{\kappa}{2} \left\{ y(s) \frac{dx(s)}{ds} - x(s) \frac{dy(s)}{ds} \right\} \\ &= -\frac{\kappa}{2} \varepsilon_{jk} v_j(s) \frac{dv_k(s)}{ds} \end{aligned}$$

$$\begin{aligned} \text{where } \varepsilon_{12} &= -\varepsilon_{21} = 1 \\ \varepsilon_{11} &= \varepsilon_{22} = 0 \quad (\text{式 6}) \end{aligned}$$

ここで、 κ は、面積力ポテンシャルの大きさを決めるフリーパラメータであり、経験則に則って決められる。また、 ε_{jk} は、Levi-Civitaの完全反対称テンソルである。ただし、ここでは、輪郭線上の点を表わすパラメータ s は、時計回りの方向に定義する。

以上から、全エネルギー $\varepsilon_{total}(s)$ は、(式 7) で与えられる。

$$E_{total}(s) = E_{inter}(s) + E_{imag}(s) + E_{exter}(s) \quad (\text{式 7})$$

SNAKES では、(式 7) を小さくするように曲線を逐次変形する方法を用いる。系の全エネルギー E_{total} は(式 8) で表わされる。

$$\begin{aligned} E_{total} &= E_{total}[\mathbf{v}(s)] \\ &= \int ds \mathcal{E}_{total}(s) \\ &= \int ds \mathcal{E}_{total}(\mathbf{v}(s)) \quad (\text{式 8}) \end{aligned}$$

ここで、系の全エネルギー E_{total} は、位置 \mathbf{v} の汎関数であり、最小作用原理により、この場合の安定解は位置 \mathbf{v} による変分をとることによって導かれる。

最終的には、変分から(式 9)の非線形連立微分方程式が得られる。

$$\begin{cases} -\alpha \frac{d^2 x(s)}{ds^2} + \beta \frac{d^4 x(s)}{ds^4} \\ \quad + \eta(x(s) - x_{c.m.}) - \kappa \frac{dy(s)}{ds} = \omega F_x(\mathbf{v}(s)) \\ -\alpha \frac{d^2 y(s)}{ds^2} + \beta \frac{d^4 y(s)}{ds^4} \\ \quad + \eta(y(s) - y_{c.m.}) + \kappa \frac{dx(s)}{ds} = \omega F_y(\mathbf{v}(s)) \end{cases} \quad \text{where } F_i(\mathbf{v}(s)) = \nabla_i |\nabla J(\mathbf{v}(s))|^2 \quad (\text{式 9})$$

(式 9)において、左辺は輪郭線を内側に引き寄せる力として働き、右辺は画像のエッ

ジ部分から受ける外向きの力である。求めるべき静的安定解は、この2つの力が釣りあった場所に定まる。(式 8)をSNAKESが形作るすべての点について解けば、静的安定解が得られる。

しかし、現実には、以下に示すような問題点がある。第一に、輪郭線上のすべての点をトレースするので、ノイズに対して脆弱である。第二に閉曲線が包んだ領域の内部情報が反映されない。閉領域内に抽出対象が複数存在しても、閉領域が複数に分裂することはない。第三に、画像エネルギー以外は外向きの力が発生しないので、対象のエッジを越えて閉曲線が内部に入ると、元に戻れないということが挙げられる。

以上のような問題点を解決するために、下記のような対策を施した。第一の問題点に関しては、輪郭線にある程度幅を持たせることで、ノイズに対して弱いことを克服した。第二の点に関しては、閉曲線の内部の統計量を加味して、(式 9)を解くようにした。また、第三の点に関しては、輪郭線上の各点を局所の領域に分け、局所の情報からパラメータ制御により、局所に膨張したり、収縮したりするメカニズムを取り入れたので、ある一定方向に同時に収縮または膨張しかしないという欠点を改良した。

今回開発した領域抽出手法の処理結果の一例を示す。図 71 は、腹部に病変を持つ患者のMRI画像に手書きで初期の閉領域をマウスなどのポインティングデバイスで書き込んだ領域を赤色で示す。この初期領域を初期値として、領域抽出した結果が図 72 である。青色の領域が抽出された結果である。SNAKES が、病変部のエッジ領域を適切に抽出できていることが明確に理解できる。

しかし、動的輪郭モデルを用いた画像処理には、

- i) 抽出領域が大きくなるにしたがって処理速度が極端に増大する。
- ii) 領域抽出結果がユーザインターフェースから入力する初期輪郭に大きく依存しており、初期輪郭の与え方によっては処理結果が意図した領域とは著しく異なる場合がある。

という問題があった。

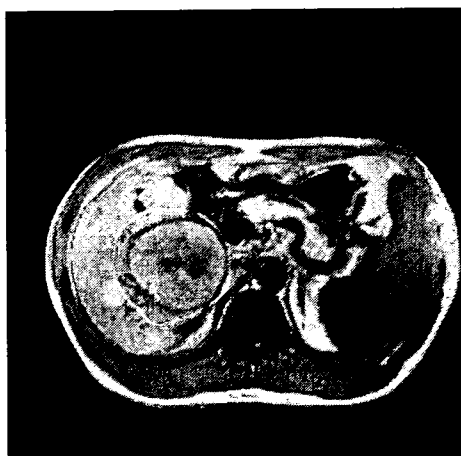


図 71 初期閉領域 (赤色)

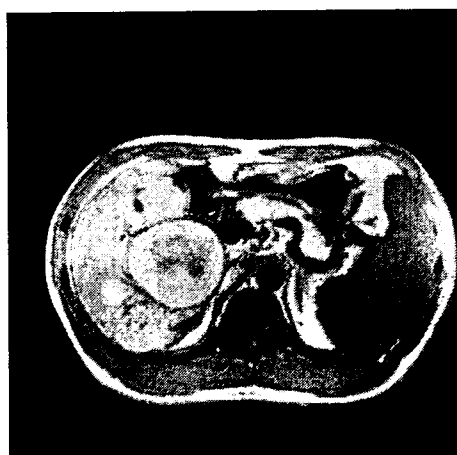


図 72 領域抽出判定結果 (青色)

動的輪郭モデルは優れた領域抽出技術であるので、長所を生かしながらこれらの課題

を克服する一手段として、エッジ抽出処理を事前に行った。この処理により抽出するエッジは連結性が保たれるわけではないが、抽出したエッジから動的輪郭モデルに対する初期輪郭を作成することができる。これにより、初期輪郭の与え方による領域抽出結果の差異が低減し、且つ、既に抽出したエッジ部分に関しては動的輪郭モデルの処理の対象としないことにより処理速度の向上を図ることが可能である。

エッジ抽出処理は次の 4Step で行うこととした。

1. 微分フィルタ処理
2. 非極大エッジ抑制処理
3. エッジ細線化処理
4. エッジ情報再構成処理

腹部 MRI 画像 (256×256) にこれらの処理を施して得た結果を示す (図 73-76)。図 73 は処理を行う前の元画像である。図 74 は上記のエッジ抽出処理によって得たエッジの画像である。図 75 は図 76 で示したエッジを元画像に重畳した画像である。図 76 は、ユーザインターフェースにてカーソルを合わせることによりエッジを選択し、その一本のエッジを強調表示している。選択したエッジは体表を囲むエッジとなっており、エッジの連結性を確認することができる。



図 73 元画像



図 76 エッジの選択



図 74 抽出エッジ



図 75 抽出エッジの重畳画像

[歪み補正]

本歪み補正方法は、大別すると 2 段階の補正処理ステップを踏んだ。1 段目は共線性の条件を用いた粗い歪み補正であり、2 段目はヒルベルト変換を用いた微細な歪み補正である。以下に歪み補正方法の手順を示す。

1) 共線性の原理による歪み補正方法

共線性の原理とは、複数の点が同一直線上にあることを示し、画像上から算出した点列を結ぶ複数の折れ線が共線となる条件で、最急降下法により点列を探索する方法である。点列 (x_a, y_a) が共線である必要十分条件は、すべての点列のモーメント行列、(式 10) の最小固有値が 0 となることである。但し、 f は任意の整数であり、今回は $f = 1$ を用いた。

$$M = \begin{bmatrix} \sum_{a=1}^N x_a^2 & \sum_{a=1}^N x_a y_a & \sum_{a=1}^N x_a f \\ \sum_{a=1}^N y_a x_a & \sum_{a=1}^N y_a^2 & \sum_{a=1}^N y_a f \\ \sum_{a=1}^N f x_a & \sum_{a=1}^N f y_a & \sum_{a=1}^N f^2 \end{bmatrix}$$

(式 10)

点列 (x_α, y_α) を算出し、ある初期値から開始した補正係数を積算した点列（補正点列）からモーメント行列を算出する。モーメント行列の固有値が最小になるまで、補正係数を更新し、新しい補正点列に対して同様に反復処理を行う。

a) モーメント行列の算出

次に、共線性の条件による歪み補正方法における補正点列のモーメント行列について記す。

画像上の点 (x, y) が変位 (u, v) によって点 (x', y') に補正されるとすると、補正公式は次の式で表すことができる。

$$x' = x + u(x, y), \quad y' = y + v(x, y) \quad (式 11)$$

ここで現実の歪み画像から、補正モデルとして(式 12)で与えられる奇関数系を考える。

$$\begin{aligned} x' &= x + Ax^3 + Bxy^2 \\ y' &= y + Cx^2y + Dy^3 \end{aligned} \quad (式 12)$$

今、画像上に M 本の直線の像があるとき、各直線像の点列をサンプルする。サンプルされた点列を

$$(x_\alpha^{(k)}, y_\alpha^{(k)})_1, \alpha = 1, \Lambda, N^{(k)}, k = 1, \Lambda, M$$

とすると、補正される点列は(式 13)で与えられる。

$$\begin{aligned} x_\alpha'^{(k)} &= x_\alpha^{(k)} + Ax_\alpha^{(k)3} + Bx_\alpha^{(k)}y_\alpha^{(k)2} \\ y_\alpha'^{(k)} &= y_\alpha^{(k)} + Cx_\alpha^{(k)2}y_\alpha^{(k)} + Dy_\alpha^{(k)3}, \\ \alpha &= 1, \Lambda, N^{(k)}, k = 1, \Lambda, M \end{aligned} \quad (式 13)$$

このとき、補正点列のモーメント行列は、(式 14)のように表わされる。

$$M^{(k)} = \begin{bmatrix} \sum_{\alpha=1}^{N^{(k)}} x_\alpha'^{(k)2} & \sum_{\alpha=1}^{N^{(k)}} x_\alpha'^{(k)}y_\alpha'^{(k)} & \sum_{\alpha=1}^{N^{(k)}} x_\alpha'^{(k)}f \\ \sum_{\alpha=1}^{N^{(k)}} y_\alpha'^{(k)}x_\alpha'^{(k)} & \sum_{\alpha=1}^{N^{(k)}} y_\alpha'^{(k)2} & \sum_{\alpha=1}^{N^{(k)}} y_\alpha'^{(k)}f \\ \sum_{\alpha=1}^{N^{(k)}} fx_\alpha'^{(k)} & \sum_{\alpha=1}^{N^{(k)}} fy_\alpha'^{(k)} & \sum_{\alpha=1}^{N^{(k)}} f^2 \end{bmatrix} \quad (式 14)$$

$\alpha = 1, \Lambda, N^{(k)}, k = 1, \Lambda, M$
モーメント行列の固有値 $\sigma^{(k)}$ や固有ベクトル $u^{(k)}$ はヤコビ法により求め、画像に歪みがない場合、補正点列からなる直線すべては共線であるため、補正点列からなる直線の固有値の合計は 0 となる。そこで、(式 15)の左辺を最小にするよう補正係数 A, B, C, D を決定する。

$$\sum_{k=1}^M \sigma^{(k)} \leq \text{threshold} \quad (式 15)$$

threshold は任意である。

b) 補正係数の決定

固有値の最小値を決定するため、その導関数の評価を行う。その評価方法は、以下に示す「固有値の摂動定理」にもとづくものとする。

「固有値の摂動定理」は、実対称行列 $M^{(k)}$ の、ある固有値を $\sigma^{(k)}$ 、対応する単位固有ベクトルを $u^{(k)}$ とすると、行列 $M^{(k)}$ の各要素がパラメータ $p^{(n)} = \{A, B, C, D\}, n = 1, \Lambda, 4$ の関数であるとき、(式 16)が成り立つ条件とする。

$$\begin{aligned}\nabla\sigma^{(n)}(p^{(n)}) &= \frac{d\sigma^{(k)}}{dp^{(n)}} = (u^{(k)}, \frac{dM^{(k)}}{dp^{(n)}}u^{(k)}) \\ &= \sum_{k=1}^M u^{(k)'} \left(\frac{\partial M^{(k)}}{\partial p^{(n)}} u^{(k)} \right)\end{aligned}$$

(式 16)

(式 14)より $M^{(k)}$ の A, B, C, D に関する導関数はそれぞれ、(式 17)~(式 11)で与えられる。

$$\frac{\partial M^{(k)}}{\partial A} = \begin{bmatrix} 2 \sum_{\alpha=1}^{N^{(k)}} x_{\alpha}^{(k)'} x_{\alpha}^{(k)3} & \sum_{\alpha=1}^{N^{(k)}} y_{\alpha}^{(k)'} x_{\alpha}^{(k)3} & \sum_{\alpha=1}^{N^{(k)}} f x_{\alpha}^{(k)3} \\ \sum_{\alpha=1}^{N^{(k)}} y_{\alpha}^{(k)'} x_{\alpha}^{(k)3} & 0 & 0 \\ \sum_{\alpha=1}^{N^{(k)}} f x_{\alpha}^{(k)3} & 0 & 0 \end{bmatrix}$$

(式 17)

$$\frac{\partial M^{(k)}}{\partial B} = \begin{bmatrix} 2 \sum_{\alpha=1}^{N^{(k)}} x_{\alpha}^{(k)'} x_{\alpha}^{(k)2} y_{\alpha}^{(k)2} & \sum_{\alpha=1}^{N^{(k)}} y_{\alpha}^{(k)'} x_{\alpha}^{(k)2} y_{\alpha}^{(k)2} & \sum_{\alpha=1}^{N^{(k)}} f x_{\alpha}^{(k)2} y_{\alpha}^{(k)2} \\ \sum_{\alpha=1}^{N^{(k)}} y_{\alpha}^{(k)'} x_{\alpha}^{(k)2} y_{\alpha}^{(k)2} & 0 & 0 \\ \sum_{\alpha=1}^{N^{(k)}} f x_{\alpha}^{(k)2} y_{\alpha}^{(k)2} & 0 & 0 \end{bmatrix}$$

(式 18)

$$\frac{\partial M^{(k)}}{\partial C} = \begin{bmatrix} 0 & \sum_{\alpha=1}^{N^{(k)}} x_{\alpha}^{(k)'} x_{\alpha}^{(k)2} y_{\alpha}^{(k)} & 0 \\ \sum_{\alpha=1}^{N^{(k)}} x_{\alpha}^{(k)'} x_{\alpha}^{(k)2} y_{\alpha}^{(k)} & 2 \sum_{\alpha=1}^{N^{(k)}} y_{\alpha}^{(k)'} x_{\alpha}^{(k)2} y_{\alpha}^{(k)} & \sum_{\alpha=1}^{N^{(k)}} f x_{\alpha}^{(k)2} y_{\alpha}^{(k)} \\ 0 & \sum_{\alpha=1}^{N^{(k)}} f x_{\alpha}^{(k)2} y_{\alpha}^{(k)} & 0 \end{bmatrix}$$

(式 19)

$$\frac{\partial M^{(k)}}{\partial D} = \begin{bmatrix} 0 & \sum_{\alpha=1}^{N^{(k)}} x_{\alpha}^{(k)'} y_{\alpha}^{(k)3} & 0 \\ \sum_{\alpha=1}^{N^{(k)}} x_{\alpha}^{(k)'} y_{\alpha}^{(k)3} & 2 \sum_{\alpha=1}^{N^{(k)}} y_{\alpha}^{(k)'} y_{\alpha}^{(k)3} & \sum_{\alpha=1}^{N^{(k)}} f y_{\alpha}^{(k)3} \\ 0 & \sum_{\alpha=1}^{N^{(k)}} f y_{\alpha}^{(k)3} & 0 \end{bmatrix}$$

(式 20)

補正係数は(式 16)を代入した (式 21)の最急降下法によって求める。

$$p'^{(n)} = p^{(n)} - \varepsilon \nabla \sigma^{(n)}(p^{(n)}) \quad (\text{式 21})$$

ステップ長 ε は任意である。この値により最小値に収束する速度が決定される。

共線性の条件による歪み補正は点列に対する補正であり、画像全体に対する補正ではないが、各々の補正点列 (x', y') の間を各々の変位 $u(x, y), v(x, y)$ で高次補間することにより画像全体の点 (座標) に対する変位を算出することができる。

2) ヒルベルト変換による歪み補正方法

ヒルベルト変換による歪み補正方法の概略図を図 77 に示す。

ヒルベルト変換による歪み補正では、図 77(a)のように等ピッチの格子データを撮影した画像を使用する。同図 (b) は同図 (a) をフーリエ変換して得たスペクトルを示し、中央部のピークが直流成分、その周辺のピークが基本波成分となる。同図 (b) のスペクトルに基本波成分を取り出すフィルタをかけることにより、基本波成分の第一高調波成分のみを抽出する。そのスペクトルに対し、逆フーリエ変換して得た正弦波形 (同図 (c))、すなわち、平面波画像の実信号をヒルベルト変換し、実信号の振幅変化と位相変化を独立に判別することで位相画像 (同図 (d)) を作成する。位相画像の任意のプロファイルを抽出すると、等ピッチの格子データの位相変化は直線で表されるのに対し、歪んだピッチの格子データの位相変化は曲線で表される。この位相変化の違いから、歪み量 (同図 (e)) を算出し、歪みをうけた内視鏡像を補正する (同図 (f))。

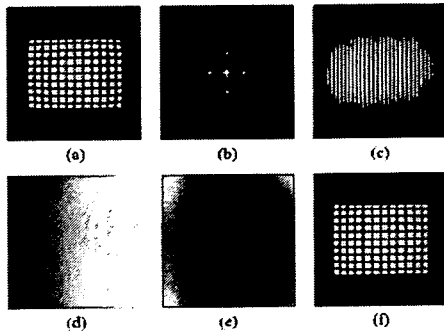


図 77 ヒルベルト変換による歪み補正方法の概略図

a) 格子画像から平面波画像作成までの処理過程

格子画像から平面波画像作成までの処理過程を図 78 に示す。信号を有限の範囲でそのまま直交変換等を行うと、繰り返しの接続部分に不連続性を伴う誤差が発生する。この誤差を軽減するために格子画像(256×256)を中央部に収め、周辺部をゼロ埋めした格子画像(512×512)を作成する。また、この誤差を軽減するために切り出し窓を乗じた処理を行う。

ゼロ埋めした格子画像を 2 次元フーリエ変換し、スペクトル画像を得る。このスペクトルに横方向の基本波成分を取り出すための 2 次元フィルタをかけ、横方向の第一高調波成分のみを抽出する。2 次元フィルタは第一高調波成分の最大値を中心とした円形フィルタである。第一高調波成分抽出後、2 次元逆フーリエ変換により、横方向に進行する平面波画像を得る。縦方向に進行する平面波画像は、上記スペクトルから縦方向の第一高調波成分を抽出することにより作成される。

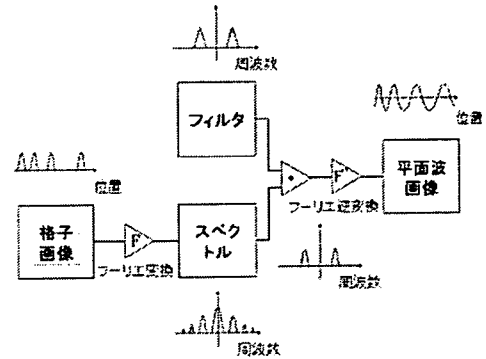


図 78 格子画像から平面波画像作成までの処理過程

b) 平面波画像から位相画像作成までの処理過程

平面波画像から位相画像作成までの処理過程を図 79 に示す。平面波画像の実信号を $S(x, y)$ とする。画像の中心を原点とし、横、縦方向を各々 x, y 方向とする。実信号が(式 22)のように極座標表現の場合、信号の変化が振幅の変化に起因するものか、位相の変化に起因するものかを判別できない。そこで、(式 23)のように実信号を、実数部と虚数部が直交関係にある複素信号に変換すれば(直交座標表現)、振幅変化と位相変化を独立して判別することができる。

$$S(x, y) = A(x, y)e^{j\theta(x, y)} \quad (\text{式 22})$$

$$S(x, y) = I(x, y) + jQ(x, y) \quad (\text{式 23})$$

ここで、 $A(x, y)$ は振幅、 $\theta(x, y)$ は位相であり、 $I(x, y)$ は同相成分、 $Q(x, y)$ は直交成分である。 $I(x, y)$ は、複素信号を実軸へ投影したものと考えることができる。実数成分と直交関係にある虚数成分 $Q(x, y)$ は複素信号を虚軸へ投影したものと考えることができる。 $I(x, y)$ と $Q(x, y)$ が得られれば、 $A(x, y)$ 、 $\theta(x, y)$ は(式 24)により独立

に求めることができる。

$$A(x, y) = \sqrt{I(x, y)^2 + Q(x, y)^2} \quad (式 24)$$

$$\theta(x, y) = \tan^{-1}\{Q(x, y)/I(x, y)\}$$

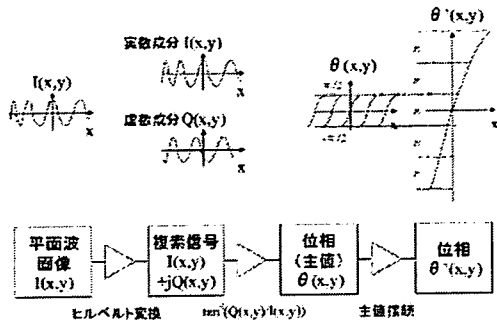


図 79 平面波画像から位相画像作成までの過程

直交座標表現への変換として、実信号をヒルベルト変換する。ヒルベルト変換を導出する方法を述べる。実数信号 $I(x, y)$ をフーリエ変換すると、周波数軸上において 0 を対称に正と負の 2 つの周波数成分が現れる。 $I(x, y)$ のフーリエ変換 $F(\omega)$ を

$$F(\omega) = F_-(\omega) + F_+(\omega) \quad (式 25)$$

とすると、正の周波数成分と負の周波数成分の合成であると考えられる。これは実数信号 $I(x, y)$ を

$$I(x, y) = 1/2\{(I(x, y) - jQ(x, y)) + (I(x, y) + jQ(x, y))\} \quad (式 26)$$

と変形することにより、互いに共役な複素信号の加算と捉えることができる。ここで、

$$F_-(\omega) \Leftrightarrow 1/2(I(x, y) - jQ(x, y)) \quad (式 27)$$

$$F_+(\omega) \Leftrightarrow 1/2(I(x, y) + jQ(x, y))$$

の関係である。 $S(x, y) = I(x, y) + jQ(x, y)$ を得るためには、信号 $I(x, y) = 1/2\{(I(x, y) - jQ(x, y)) + (I(x, y) + jQ(x, y))\}$ に対し、負の周波数成分を除去する以下のフィルタ処理を行えばよい。

$$u(\omega) = \begin{cases} 2 & \omega \geq 0 \\ 0 & \omega < 0 \end{cases} \quad (式 28)$$

まとめると、平面波画像の 2 次元フーリエ変換し、周波数成分に(式 28)のフィルタ処理をかけて 2 次元逆フーリエ変換することで、 $I(x, y)$ と $Q(x, y)$ を算出する。(式 23) から、位相 $\theta(x, y)$ を算出する。ところで、 \tan 関数は周期関数であり、

$$\theta' = \theta_m \pm n\pi \quad (式 29)$$

ここで、 θ_m : 主値 ($-\pi/2 < \theta \leq \pi/2$)、 n : 周期 (0 及び自然数) である。位相は主値 θ_m のみ意味を持ち、 $\pm n\pi$ の自由度を持つ。内視鏡像の歪みの最小点を位相 $\theta = 0$ とした主値接続を行い、位相画像 $\theta'(x, y)$ を算出する。尚、歪みの最小点を手動で設定する。

c) 位相画像から歪み補正ベクトル算出までの処理過程

平面波画像から位相画像作成までの過程で得られる位相画像の進行方向のプロファイル有位相曲線と呼び、補正前の位相曲線を $\phi(x)$ 、補正後の位相曲線を $\theta(x)$ とする。 $\theta(x)$ は直線となる。補正前後の位置 x_j 、

X_j は位相により次の関係がある。

$$\begin{aligned} \phi(x_j) &= \theta(X_j) \\ \phi(x_j) \leq \theta(X_j) \leq \phi(x_{i+1}) \end{aligned} \quad (式 30)$$

補正前の位置 x_j は線形補間により(式 31)で求めることができる。

$$x_j = \frac{x_i \cdot \{\phi(x_{i+1}) - \theta(X_j)\} + x_{i+1} \cdot \{\theta(X_j) - \phi(x_i)\}}{\phi(x_{i+1}) - \phi(x_i)} \quad (式 31)$$

上記により求められた補正前の座標 x_j と補正後の座標 X_j により歪み補正ベクトル V_j は(式 32)により与えられる。

$$V_j = x_j - X_j \quad (式 32)$$

今回開発した内視鏡像歪み補正方法は、すでに述べたように、2段階の補正処理ステップを持つ補正方法である。1段目に共線性の条件による歪み補正を実施し、2段目にヒルベルト変換による歪み補正を実施することで、より高精度な歪み補正を実現している。図 80 に示すように、強い歪みを持つ軟性内視鏡画像に対して歪みを補正することができた。

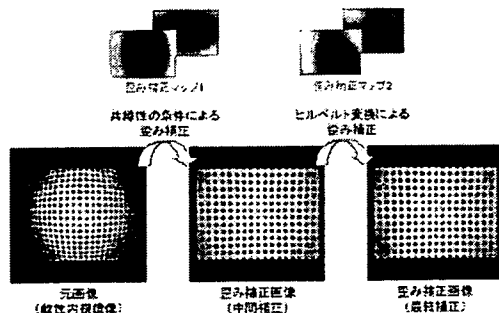


図 80 本内視鏡像歪み補正方法

[画像重畳]

歪み補正技術を用いて、各種医用診断画像と内視鏡画像を逆歪補正し、内視鏡に重畳する機能を開発した。管腔ファントムを X線CT装置で撮影したものを、仮想内視鏡表示し、軟性内視鏡のレンズ歪を逆補正して表示した結果を図 81 に示す。図 81 (a) は、管腔を模擬したファントムを内視鏡で観察した画像を示している。ファントムの管腔内に突起を設けてあるのが、観察できる。この突起が画像重畳の良否の指標である。同図(b)は、管腔ファントムを X線CT装置で撮影したものを、仮想内視鏡表示し、かつ、軟性内視鏡カメラの歪を逆補正して表示している。画像の輪郭が歪んで見えるのは、元の X線CT像を、カメラの収差で歪ませたせいである。この画像と実内視鏡像を重畳させた画像が同図(c)である。内視鏡で見える範囲内の画像だけが表示される。内視鏡で視認できない領域をも観察できるようにするために、内視鏡のカメラの歪関数を延長し、X線CT像を重畳したものが、同図(d)である。明らかに、実内視鏡の周辺部も、X線CT像で確認できることが分かる。内視鏡だけでは視認不可能な臓器の正確な位置関係を直感的に把握できることになり、視認性が高まった。

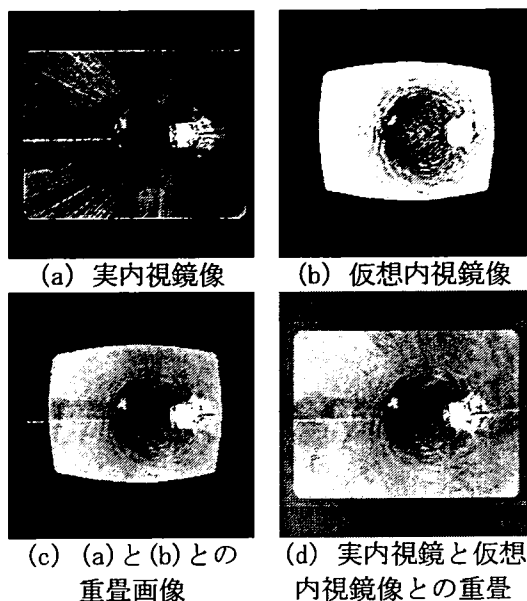


図 81 画像重畳例

[術具位置追従]

内視鏡画像による術具位置センシング要素技術を用いて、術具の位置を追従するアプリケーションを開発し、手術ナビゲーションシステムに実装した。技術的方法は、[II]で述べたので、ここでは割愛する。

軟性内視鏡画像による術具先端位置・姿勢の検出には、電子データ上に発生させた術具の仮想内視鏡画像と、術具が写った実内視鏡画像との間で、類似度の一種である平均相互情報量を算出し、平均相互情報量が最大となる仮想内視鏡画像での術具の位置・姿勢を、実際の術具の位置・姿勢とする方法を採用した。仮想内視鏡画像の作成では、内視鏡に写る被写体の歪み（内視鏡レンズによる歪みと遠近歪み）を考慮し、任意の位置にある被写体が内視鏡画像上でどのように写るかを電子データとして再現した。術具先端位置・姿勢が医用画像空間上でどの位置を指し示すかを、三断面 MPR 画像、及び 3D ガイド画像に表示した。

本件では、模擬実験（動物未使用）、及びブタを用いた動物実験により、その性能を評価した。

術具位置追従機能のブタへの適用にあたり、事前に模擬実験で動作検証を行った。

手術ナビゲーションシステムにおけるターゲットの設定や術具位置のガイドに CT データを使用した。なお、CT データと実データ（ブタ）との位置合わせは行わないこととした。内視鏡は、VNL1171K（ペンタックス社製）を使用した。模擬実験前に内視鏡レンズによる歪みを補正する歪補正マップを算出した。

電子データとして発生させた仮想の術具（以下、仮想術具）と、LIVE 映像の実際の術具（以下、実術具）とが重なり合う位置や方向を逐次検索し（マッチング処理）、実術具に最も近似する仮想術具から、実術具の三次元的な位置や方向を算出した。実術具に特徴的なマーカーを貼付することで、実術具とそれ以外とを区別し、マッチング処理の精度を高めた。また、実術具の長手方向の軸角もマーカーのパターンから算出した。

術具位置追従処理結果の一例を図 82 に示す。左図は実術具であり、処理過程で発生する術具方向に交線を付けた。右図は処理後の仮想術具であり、実術具に重畳表示された。これにより、実術具と仮想術具とのマッチングのずれを映像として確認することができた。また、図 83 のように三次元的な位置や方向を数値で示すこともできた。

図 84 にターゲット、及び親内視鏡位置設定画面を、図 85 に術具位置追従画面を示す。設定画面では、三面図 MPR 画像、及び 3D ガイド画像を用いて位置や方向、WW/WL

等を設定できた。術具追従画面では、処理結果に連動して位置や方向で切断された三面図 MPR 画像や 3D ガイド画像を正確に表示でき、カメラ座標系での三次元的位置や仰角・方位角・軸角を数値化できた。



図 82 実術具 (左): 処理過程で発生する術具方向を赤線で表示、仮想術具 (右): 実術具との重畳表示

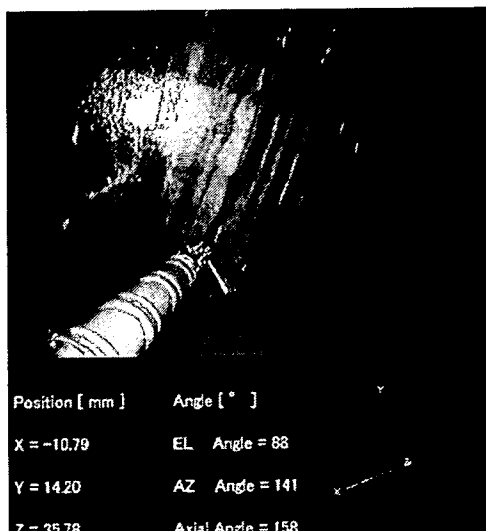


図 83 術具の三次元的位置や方向の表示

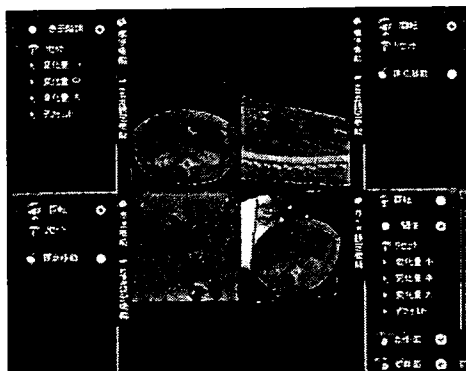


図 84 ターゲット、及び内視鏡位置設定画面

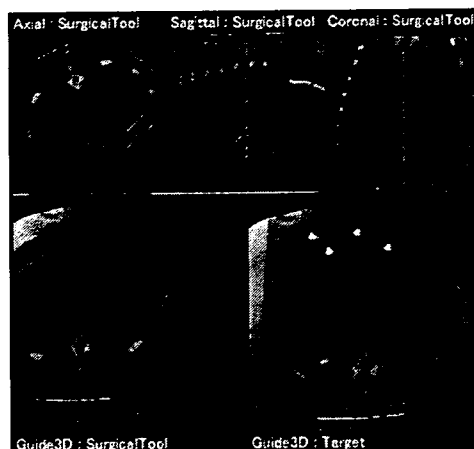


図 85 術具位置追従画面

ブタを用いた動物実験では、模擬実験で使用した内視鏡から、ペンタックス製の親子内視鏡に変更した。親子内視鏡の内径は術具とほぼ同じである。術具を用いず、マーカーを子内視鏡に貼付したものをを用いた。ターゲットや親内視鏡の位置を模擬実験と同様に設定した。安定して術具位置を追従できなかったが、処理が成功した例を図 86 に示す。

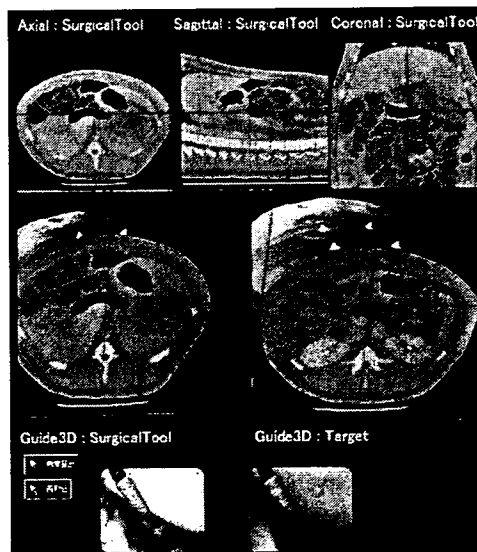


図 86 術具位置追従機能のブタ実験による動作検証結果

[手術ナビゲーション機能統合化ソフトウェア]

手術ナビゲーションシステムの要素技術開発の結果に関し、(a)スイッチド・ファブリック技術を使用した StarFabric データ通信制御ソフトウェアの開発、(b)時間同期した複数の動画像を管理する術中画像管理装置の開発、について記載する。

(a) スwitchド・ファブリック技術を使用した StarFabric データ通信制御ソフトウェアの開発

図 87 に本ソフトウェアのシステム構成図を示す。また、実装した各機能を以下に記載する。

- StarFabric 制御機能

StarFabric の初期化、コネクション作成、ホットプラグによるノード検出などの StarFabric 固有の処理を行うための機能を実装した。StarFabric に接続されるノードは、ファブリックの初期化やファブリック内で発生したイベントを検出するノード (Root) と、それ以外のノード (Leaf) に分類され、一般的に Root ノードが StarFabric のコネクションの作成やイベントの検出を一括で行った。本件では、ナビゲーションコントロール装置が常に存在するノードであることから、Root ノードとしての処理を行い、それ以外の装置は Leaf ノードとして処理を行った。また、各装置にノード固有の ID (ノード ID) を付加し、各ノードの初期化時にノード ID を通知することにより、各ノードの種別判定を行った。なお、ノード数は最大で 8 ノード (Root ノード + Leaf ノード 7 台) とした。

StarFabric でデータ通信を行うには、コネ

クションを作成する必要がある。本件では、制御用コネクションとデータ通信に使用するデータ通信用コネクション、及び複数のノードに同時にデータ転送するためのマルチキャストコネクションの 3 種類を作成した。制御コネクションは、各装置間で制御用データを通信するために使用するもので、各装置のノード情報とコネクション情報の通知に使用するものである。データ通信用コネクション、及びマルチキャストコネクションは、各装置間で行うデータ転送に使用するものである。

- メモリ管理機能

StarFabric で通信を行う際に必要となる物理メモリの確保や、物理アドレスのユーザ空間へのマッピングなどの、通信する上で最低限必要なメモリ管理を行うための機能を実装した。

- 通信制御機能

StarFabric の Write Message Event 機能を使用し、装置間で割り込みベースのメッセージ通信を行うため機能、及び DMA 対応 StarFabric ボードを使用する場合の DMA 通信制御を行う機能を実装した。

- エラー検出機能

これまで開発してきたものと同様に、StarFabric 上のエラーを検出し、エラー発生時にユーザアプリケーションに対して通知を行うための機能、各ノードの動作状態を定期的に監視し、動作停止検出時にもユーザアプリケーションに対して通知を行うための機能を実装した。

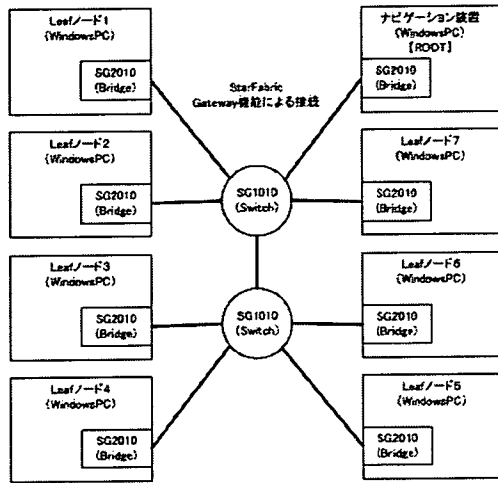


図 87 システム構成図

(b) 術中画像管理装置の開発

画像サーバとナビゲーションコントロール装置とで、動画・静止画や、以下の手術関連情報をデータ送受信するためのインターフェースを構築した。また、DICOMサーバ、DICOMIFサーバ、ナビゲーションコントロール装置との連携機能を以下のように開発した。

<インターフェース>

ナビゲーション装置やDICOMIFサーバのDICOM変換機能により発生したDICOM/非DICOM情報は、WebServiceとソケット通信でデータを送受信した。DICOM情報はソケット通信でデータ送受信を行った。非DICOM情報はXMLで保管されるものと、AVIやJPEGで保管されるものがあるが、前者はWebServiceで後者はソケット通信でデータ送受信を行った。

<DICOMサーバ連携機能>

- DICOM画像受信プログラム
DICOM通信におけるSOPの役割を担うサービスを提供する。DICOMサーバからの画像自動受信に使用される。
- DICOM変換プログラム

ビデオコンバータから出力されるMotionJPEG形式ファイルをDICOM形式に変換するサービスを提供する。デフォルトでは2分周期で定期的に行われる。変換と同時に元ファイルのデータベース登録やサムネイル作成が行える。

- DICOM画像送信プログラム
ローカルに保存されているDICOM画像をDICOMサーバへ転送するサービスを提供する。デフォルトでは2分周期で定期的に行われる。ビデオコンバータから出力された動画をDICOM変換後、DICOMサーバへ出力するために利用される。
- DICOM画像削除プログラム
ローカルに保存されているDICOM画像を自動で削除するサービスを提供する。デフォルトでは2分周期で定期的に行われる。DICOM画像転送後に一定期間でデータが削除される。削除対象は既にDICOMサーバへ転送されている画像のみとした。
- ナビゲーション装置用インターフェース
ナビゲーション装置とのソケット通信やWebServiceに関するサービスを提供する。
- DICOM/非DICOM情報管理機能
DICOMのTAG情報にtime stampを設定する機能や、定期的にサムネイルを作成する機能、動画を管理する機能(AVI管理機能)を提供する。

「手術関連情報」

- 患者情報...患者ID、患者名、患者性別、血液型
- 検査情報...検査日、検査時刻、モダリ

- ティ、検査記述、検査番号
 - ▶ シリーズ情報...シリーズ番号、シリーズ記述、検査部位、撮像体位
 - ▶ 画像情報...画像番号、フレームカウント、行、列
- 手術情報...手術日、手術時間、手術年齢、麻酔法、再発、輸血、診断、術式名称、診療科、スタッフ、装置、医薬品、手術体位、部位・臓器
 - ▶ スタッフ...執刀医、助手、麻酔医、臨床検査技師、放射線技師、看護師
 - ▶ 装置...モダリティ、映像機器、生体モニタリング、マーカー、ツール、術具、麻酔器具、その他
 - ▶ 医薬品...神経系、免疫系、炎症、感染症、悪性腫瘍、外科的手術、泌尿器系、その他
 - ▶ 位置...ツール、マーカー、レジストレーション
 - ▶ モジュール...画像処理、計測、領域抽出、音声
 - ▶ LIVE...カメラ、医用画像、センシング、イベントレコーダー、生体信号

[手術ナビゲーションソフトウェアアーキテクチャ]

手術ナビゲーションシステムの要素技術開発に関し、

- (1) ストーリー制御エンジンの開発
- (2) データ構造、及び制御マネージャのアーキテクチャ設計
- (3) 術具位置追従アプリケーションの実装
- (4) データ蓄積・検索機能、及びデータ通

信機能の改良

について、研究結果をそれぞれ記載する。

(1) ストーリー制御エンジンの開発

与えられたストーリー（ここでは、事例と記載する）を正確に表現するため、過去の経緯によって次の選択肢の確率が変化する非マルコフ過程を基礎として、個々の情報を繋ぎ合わせることで事例を表現した。その事例を繰り返し利用することで、その事例が持つ優先度や繋がりが変化するネットワーク構造の事例ベースを考案した。

本件における事例ベースのデータ構造を図 88 に示す。情報の最小単位をエレメント、エレメントを幾つか組み合わせる意味を持つ単位をセル、幾つかのセルと当該セルに含まれるエレメントとの組み合わせを繋ぎ合わせ、優先度を持ったネットワークとして表現したものをパスと定義した。このパスが最終的に表現された事例となる。

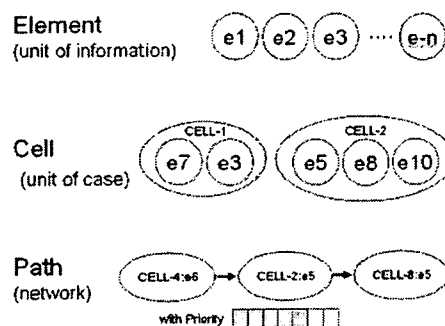


図 88 事例ベースのデータ構造

このデータ構造を用いて、順序の有無や優先度 (Positive、Negative) で学習・推論する機能を設けた。また、ストーリー制御エンジンとの通信インターフェースに Web Service を用いた。推論機構は与えられた問

題に類似した事例を検索し、各事例が持つ優先度で並び替えて提示した。

ストーリー制御エンジンによる手術関連情報設定支援や術中ナビゲーションストーリー支援を評価するために、図 89 に示すストーリー制御エンジン評価用イベントビューワを試作した。これには、以下の機能を備えた。

- テストケース設定
- 順序の有無設定
- 優先度 (Positive、Negative) 設定
- 学習・推論・抹消機能
- Web Service によるデータ通信機能

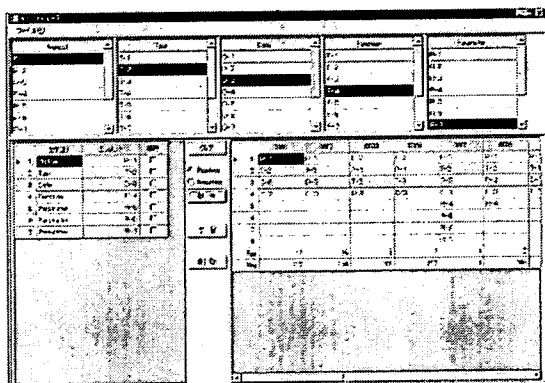


図 89 ストーリー制御エンジン評価用イベントビューワ

イベントビューワの操作手順には、学習過程、推論過程がある。学習過程では、プロトコル、タスク、制御パラメータ（データ、操作、情報、属性）を組み合わせたテストデータに対して優先度や順序の有無を設定し、ストーリー制御エンジンに学習要求を発行する。ストーリー制御エンジンは、それを受けてプロトコルの振る舞いを学習し、その結果を学習完了応答として転送する。

推論過程では、順序の有無を設定し、ストーリーやストーリーの構成要素であるプ

ロトコルを検索キーとして、ストーリー制御エンジンに推論要求を発行する。ストーリー制御エンジンは、類似したストーリーを検索し、ストーリーを優先度順に並び替えてイベントビューワに転送する。

(2) データ構造、及び制御マネージャのアーキテクチャ設計

ストーリーはプロトコルの処理フローの一単位、プロトコルはオペレータが操作するアプリケーションクラスの一単位、タスクはアプリケーションの機能クラスの一単位である。タスクは制御パラメータ（データ、操作）から構成される。また、それぞれのクラスは、データの実体を示す情報や操作の実体を示す属性をもつ。

本件のデータ構造は、上記構成要素をノードとして構成したプロトコルオブジェクトモデル (POM : Protocol Object Model と記述) ツリーで表現した。POM ツリー上のノードを全てオブジェクトとして捉えた。Element オブジェクトは、要素（ストーリー、プロトコル、タスク、操作、データ）を表し、情報オブジェクトや属性オブジェクトは要素の内容を表した。ただし、情報オブジェクトや属性オブジェクトは、どのエレメントオブジェクトのノードにも成りえるとした。POM ツリーのルートとして Story オブジェクトがストーリーの全体を表した。

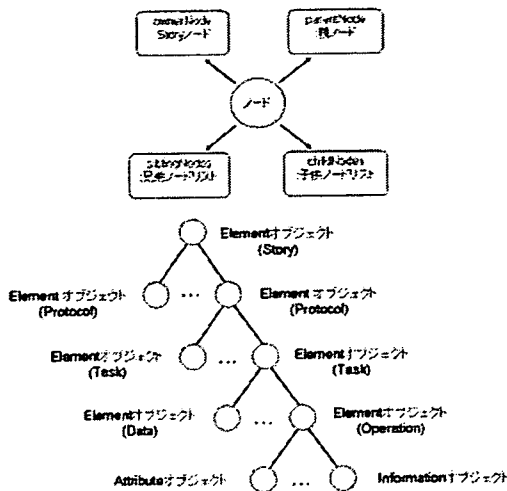


図 90 POM ノード構造

POM ノードツリーのクラス図を図 91 に示す。

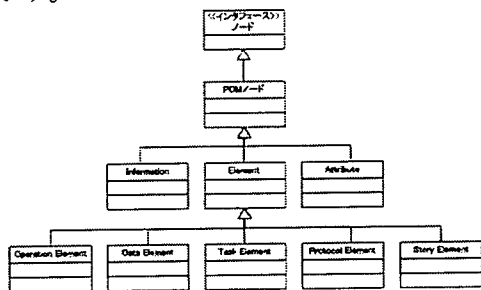


図 91 POM ノードクラス図

図 92、93 のようにプロトコル管理ツール、イベントビューワ、ナビゲーションビューワを管理できるようプロトコル制御マネージャを設計した。

- GUID 管理...プロトコル管理ツールからの GUID 発行要求により、プロトコルやタスクの GUID を発行する。
- ストーリー管理...プロトコル管理ツールでプロトコルやタスク、インターフェースをカスタマイズし、ストーリーを組立てて、ライブラリで管理する。また、ナビゲーションビューワで術中

に操作したプロトコルから、ストーリーを構成し、ライブラリで管理する。

- 通信管理...ストーリー制御エンジンと通信し、ストーリーの検索・学習・抹消を行う。
- プロトコル管理...イベントビューワからのアプリ起動要求により、プロトコルを生成し、アプリケーションを起動する。アプリ停止要求により、アプリケーションを停止する。また、ナビゲーションビューワからアプリケーションを強制的に起動する。

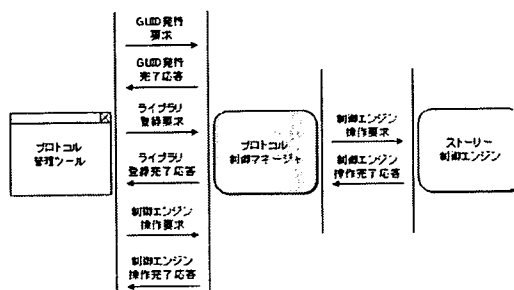


図 92 プロトコル管理ツールにおけるプロトコル制御マネージャの機能

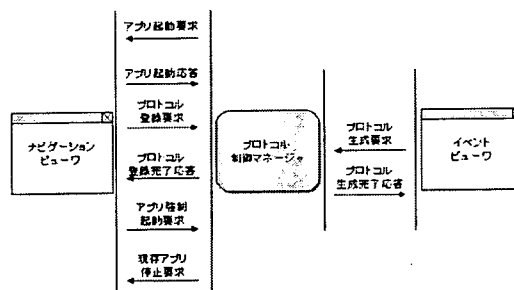


図 93 イベントビューワ、ナビゲーションビューワにおけるプロトコル制御マネージャの機能