

6D-5

## 心電図同期撮像されたMRI-CT画像からの 3次元アニメーション合成

山本 強  
北海道大学大型計算機センター

### 1 まえがき

医療診断用映像化技術としてMRI-CTが注目されている。現在、主に断層写真として診断の用いられているが、ボリュームレンダリングを用いることにより臓器の3次元形状を再構築することが可能である。しかし心臓のように時間とともに形状が変化する部位に関しては原データは $x, y, z$ 軸に加え時間 $t$ が加わった4次元データとなる。本研究では人体胸部を心電図同期撮像して得られた4次元データから心臓の動特性を再構築する試みについて報告する。

### 2 4次元ボリュームデータの構造

最近のX線CT, MRI-CTには心電図同期撮像モードが備えられている機種があり、1心拍を10~25フレームに分解して撮像することが可能になって来ている。このモードではCT値を $(x, y, z, t)$ の4次元空間の体積データとして計測できる。計測された4次元データはある $z$ 値について、 $xy$ 平面の断層画像(基本断層画像)を $t$ を変化させて計測した10~25枚の画像群を1ブロックとし、さらに $z$ を変更して $n$ ブロック測定したものである。今回はMRI-CTによって計測された胸部画像を用いたが、心臓のように時相による形状変化が大きい部位では時間サンプリング幅を小さくするため、4次元体積セルが小さくなる。その結果、MRI信号のエネルギーが低下して画像のSN比を低下させることになる。SN比を確保するためには空間方向のサンプリング間隔を大きく取る必要がある。今回実験に使用した4次元データでは基本断層画像を人体の輪切りとしたため $x, y$ 方向についてはサンプリング間隔がほぼ1.5mmに固定されるため、 $z$ 方向のサンプリング間隔を1.5cm程度としている。時間変化を含む4次元データではデータ量が莫大となるのでMRI-CT装置の内部容量の限界からも、空間方向の解像度は犠牲にしてデータ量の削減を行う必要もある。実験に使用した4次元データは $256(x) \times 256(y) \times 5(z) \times 23(t) \times 16\text{bit}$ の構成であり、全体で約15MBであった。

### 3 補間アルゴリズムとレンダリング処理

4次元ボリュームからある時相について3次元ボリュームセットを抽出し3次元ボリュームレイトレース法で等密度面(isosurface)を画像化する。3次元ボリュームの抽出は単純に同時相の断面画像を抽出するのではなく隣接する時相の画像を重み加算してフレーム間の連続性を向上させている。3次元ボリュームは $x, y$ 方向が1.5mm間隔、 $z$ 方向が15mmでサンプリングされているため何らかの補間処理を行わなければ自然な画像を生成できない。補間処理をレンダリング前に行い、補間されたボリュームデータを作成するのが自然であるが、それによってレンダラが使用するデータサイズが増加しプロセスサイズが巨大化してしまう。補間処理が高速ならば光線追跡の段階で逐次計算することができ、プロセスサイズが小さくなり小型のワークステーションでレンダリングが実行可能となる。ここで使用した補間アルゴリズムは1次補間による再サンプリングであり、任意の座標 $(x, y, z)$ のCT値を3回の整数化(floor), 14回の実数乗算, 10回の加減算で求めることができる。

レンダリング計算は光線追跡アルゴリズムに基づいて行う。CT値が指定された値 $th$ の面(isosurface)を探索し、その面の法線ベクトルを用いてシェーディング計算することにより画素輝度を求める。光線追跡は視点からの距離 $t$ をパラメータとする探索直線に沿って行う。視点( $t=0$ )から一定間隔 $dt$ で3次元ボリュームを再サンプリングしていき、

$$ct1 = CT(x(t_1), y(t_1), z(t_1)) \quad (1)$$

$$ct2 = CT(x(t_1 + dt), y(t_1 + dt), z(t_1 + dt)) \quad (2)$$

$$ct1 < th \cap ct2 > th \quad (3)$$

である $t_1$ を探す。 $t_1, t_2, dt, ct1, ct2$ から交点の推定値 $t$ を次式から求める。

$$t = t_1 + \frac{dt(th - ct1)}{ct2 - ct1} \quad (4)$$

法線ベクトルは次式によりCT値傾斜ベクトル $n_x, n_y, n_z$ を求めそれを正規化して代用する。

$$n_x = CT(x_t + 0.5, y_t, z_t) - CT(x_t - 0.5, y_t, z_t) \quad (5)$$

$$n_y = CT(x_t, y_t + 0.5, z_t) - CT(x_t, y_t - 0.5, z_t) \quad (6)$$

$$n_z = CT(x_t, y_t, z_t + 0.5) - CT(x_t, y_t, z_t - 0.5) \quad (7)$$

ここで  $x_t = x(t)$ ,  $y_t = y(t)$ ,  $z_t = z(t)$  である。Isosurface の CT 値  $th$  は複数設定でき、設定値毎に不透明率 ( $\alpha$ )、表面色、鏡面反射強度を独立に指定できる。特に  $\alpha$  を 1 以下に設定することにより半透明表示が可能となり、多層構造が表示できる。動画像の生成では半透明表示により層間の動的な関係が把握しやすくなる。

## 4 動画像生成

4次元ボリュームから時間軸断面を変化させて3次元ボリュームを抜き出しボリュームレンダリングによりアニメーションフレームを作成する。心臓の動きは心電図と同期した周期運動であるので反復表示によって長時間アニメーションとする事が可能である。

今回使用した4次元ボリュームデータは心臓を含む断面のある時相の断面画像群である。断面画像は  $256 \times 256$  構成、1画素 12bit で量子化されている。4次元データの全体はこの断面画像 115 枚から構成されている。

図1は標準的な表面属性を定義した isosurface から再構成されたある時間断面画像の例である。時間方向のサンプリングのために空間方向、特に  $z$  軸 (体軸方向) のサンプリング間隔が断面方向と比べて 10 倍程粗いが補間処理の結果極めて自然な画像が得られることがわかる。

診断的な意味としては単純に臓器表面を画像化するよりも内部情報を含めて表示する事によって有効な情報が得られる場合がある。レンダラーでは複数の isosurface を指定でき、それらに独立の表面属性を指定できるようになっている。属性には不透明率 ( $\alpha$ ) が含まれ、それを 1 以下にすることで半透明面を表示できる。これによって臓器表面と内部を同時に表示できるようになる。このような複雑な表示は静止画では読み取りを困難にする場合が多いが、動画像では有効である。図2は筋肉や脂肪などの軟組織に対応する CT 値を青の半透明表面とし、血液に相当する CT 値を赤の不透明表面として画像化した例である。図2では  $t=0, t=11$  の異なる時間の画像再構成例を表示してある。図2の例で 23 フレームの画像生成に要した CPU 時間は全体で約 80 分 (IRIS-4D/30TG, R3000/30MHz) であった。

## 5 むすび

心電図同期撮像された4次元のMRI-CT画像からの3次元アニメーション合成実験の結果、MRI-CTの撮像時間と見合った時間でアニメーション合成が可能であることが示された。今回は4次元空間の処理は時間方向のスムージング処理だけであるが、4次元画像の特徴

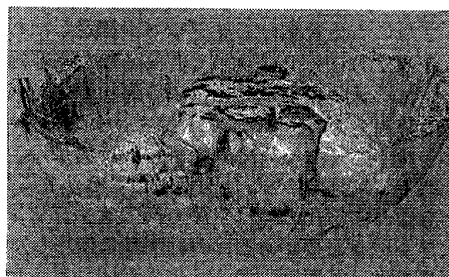
を生かした画像処理によって再生画像の質を向上させることも可能となると思われる。



図1 Isosurface 表示例



a. Reconstruction for  $t=0$



b. Reconstruction for  $t=11$

図2 半透明表示を用いた表示例 ( $t=0, t=11$ )

## 参考文献

- [1] 山本. "MRI-CT で計測された 3 次元ボリュームデータの可視化", 北海道大学大型計算機センターニュース Vol.24, No.3, pp.8-16, 1992