

3次元 Laws マスクによるポリリュームデータの解析に関する研究

鈴木 一 史 独立行政法人メディア教育開発センター研究開発部准教授

1 はじめに

近年のコンピューター技術の発達によって、3次元ポリリュームデータが様々な分野で活用されるようになってきた。コンピューターグラフィックスハードウェアの高性能化によって、一般的なパーソナルコンピューターでも高速に3次元グラフィックスデータを表示することが可能になってきており、3次元ポリリュームデータの利用は増加の傾向にある。大量の3次元ポリリュームデータのデータベースがある時、データベースシステムは、高速かつ効果的にデータの検索・分類などができることが重要である。ポリリュームデータの検索・分類を行うためには、データの形状特徴解析が必要である。2次元画像の検索・分類においては、コンポリューション演算、フーリエ変換、ウェーブレット、フラクタル次元、自己相関関数等を応用した様々な研究がなされてきた。しかし、3次元ポリリュームデータの検索・分類に関する研究は、十分に行われているとはいえない。3次元ポリリュームデータに関する手法としては、3次元ポリリュームデータを2次元画像に変換し、各2次元画像に対して画像解析手法を応用し、その後、3次元情報を再構築するというものが多かった。しかし、近年、3次元医療データを対象に2次元画像解析で用いられてきた Co-Occurrence Matrices の3次元拡張[Kurani2004]やRun-Length Encoding の3次元拡張[Xu2004]などの方法が注目されてきている。これらの手法では、3次元データを連続する2次元画像へ変換してから特徴抽出を行うのではなく、直接、3次元ポリリュームデータから形状特徴を抽出する。そのため、2次元画像へのスライス化による情報の損失を防ぐことができる。そして、形状特徴を直接抽出することによって、精度の高いデータのセグメンテーションや分類を実現している。本研究では、Kurani や Xu らの研究と同様に、3次元ポリリュームデータから、直接、形状特徴を抽出した。具体的には、2次元画像解析（テクスチャー解析）に有効な手法と知られている Laws マスク[Laws1980]の3次元化を行った。そして、シミュレーションプログラムによって生成された3次元 Laws マスクを用いて、3次元ポリリュームデータの検索・分類を実現した。

2 3次元ポリリュームデータ

2-1 Laws' Texture Energy Measures

1979年、K. I. Laws は Texture Energy Measures 法を提案した。この手法は2次元テクスチャー画像の解析に有効であり、様々なソフトウェアに 응용されている。この手法では、Gaussian、Edge-Detector、Laplacian の画像フィルターを用いて、解析の対象となる2次元画像に”畳み込み演算”(Convolution)を行い、テクスチャーエネルギーが高くなる2次元画像内のピクセルを強調することができる。この画像フィルターは Laws のマスクとも呼ばれ、以下の3つの 1×3 の基本マスクから構成される。

$$L3 = \{1 \ 2 \ 1\}$$

$$E3 = \{-1 \ 0 \ 1\}$$

$$S3 = \{-1 \ 2 \ -1\}$$

それぞれのマスクのラベルは、Lはレベル(Level または Local Average)、Eはエッジ検出(Edge Detection)、Sはスポット検出(Spot Detection)を意味しており、ラベル末尾の数字はマスクの長さである。Laws はこれらの3つの 1×3 の基本マスクと、それを拡張した 1×5 の基本マスクを用いて2次元画像解析を行った。

1×5のマスクは、3つの1×3基本マスクそれぞれに対して”畳み込み演算”を行うことで作成することができる。この計算によって9個の1×5の基本マスクが作成できるが、同値となるマスクを削除すると以下の5つとなる。

$$\begin{aligned} L5 &= \{1 \ 4 \ 6 \ 4 \ 1\} \\ E5 &= \{-1 \ -2 \ 0 \ 2 \ 1\} \\ S5 &= \{-1 \ 0 \ 2 \ 0 \ -1\} \\ R5 &= \{1 \ -4 \ 6 \ -4 \ 1\} \\ W5 &= \{-1 \ 2 \ 0 \ -2 \ 1\} \end{aligned}$$

1×5の基本マスクでは、2つの特性を持つマスクが追加される。Rは波紋検出(Ripple Detection)、Wは波検出(Wave Detection)である。1×3のマスクおよび1×5のマスクでは、Lマスクを除いてマスク内の数値の和は0となる(Zero-sum masks)。マスク内の数値の和が0であると、”畳み込み演算”を行う際に、演算結果が各2次元ピクセルで独立となり、画像処理の計算をする際に都合がよい。マスク内の数値の和が0とされないLマスクは、一般的に画像のコントラスト特徴を抽出するのに利用されることが多い。アプリケーションによっては特徴として利用しない場合もある。

マスク長は、”畳み込み演算”を繰り返すことによって拡大することができる。例えば、1×7のマスクは1×5マスクと1×3マスクの”畳み込み演算”を行うことで算出できる。同値となるマスクを削除すると以下に示す7つの1×7のマスクを得ることができる。

$$\begin{aligned} Xa7 &= \{1, 6, 15, 20, 15, 6, 1\} = L5L3 \\ Xb7 &= \{1, 4, 5, 0, -5, -4, -1\} = L5E3 = -E5L3 \\ Xc7 &= \{-1, -2, 1, 4, 1, -2, -1\} = L5S3 = E5E3 = S5L3 \\ Xd7 &= \{1, 0, -3, 0, 3, 0, -1\} = E5S3 = -S5E3 = -W5L3 \\ Xe7 &= \{1, -2, -1, 4, -1, -2, 1\} = S5S3 = R5L3 = W5E3 \\ Xf7 &= \{1, -4, 5, 0, -5, 4, -1\} = R5E3 = W5S3 \\ Xg7 &= \{-1, 6, -15, 20, -15, 6, -1\} = R5S3 \end{aligned}$$

さらに、1×9のマスクは1×7のマスクと1×3のマスクの”畳み込み演算”で以下のようになる。

$$\begin{aligned} Ya9 &= \{1, 8, 28, 56, 70, 56, 28, 8, 1\} \\ Yb9 &= \{1, 6, 14, 14, 0, -14, -14, -6, -1\} \\ Yc9 &= \{-1, -4, -4, 4, 10, 4, -4, -4, -1\} \\ Yd9 &= \{1, 0, -4, 0, 6, 0, -4, 0, 1\} \\ Ye9 &= \{1, 2, -2, -6, 0, 6, 2, -2, -1\} \\ Yf9 &= \{-1, 2, 2, -6, 0, 6, -2, -2, 1\} \\ Yg9 &= \{-1, 4, -4, -4, 10, -4, -4, 4, -1\} \\ Yh9 &= \{1, -8, 28, -56, 70, -56, 28, -8, 1\} \\ Yi9 &= \{1, -6, 14, -14, 0, 14, -14, 6, -1\} \end{aligned}$$

様々な長さのマスクを利用することにより、解像度の異なる多種多様な特徴を2次元画像から抽出することが可能になる。しかし、多くのマスクを利用すると、”畳み込み演算”の計算量が増加することから、一般的なアプリケーションでは、1×3と1×5のマスクが用いられるケースが多い。

Lawsは、これらの1次元の基本マスクをそれぞれ掛け合わせることで、2次元のマスクを作成した。例えば、L3とE3を掛け合わせることで2次元のマスクL3E3が作成できる。同様の計算を1×3の基本マスクに行うと、以下の9個(3個×3個)の2次元マスクがある。

E3E3 E3L3 E3S3
 L3E3 L3L3 L3S3
 S3E3 S3L3 S3S3

1 × 5 の基本マスクでは、以下の 25 個（5 個 × 5 個）のマスクを得ることができる。

L5L5 L5E5 L5S5 L5R5 L5W5
 E5L5 E5E5 E5S5 E5R5 E5W5
 S5L5 S5E5 S5S5 S5R5 S5W5
 R5L5 R5E5 R5S5 R5R5 R5W5
 W5L5 W5E5 W5S5 W5R5 W5W5

2-2 3次元 Laws マスク

3次元ボリュームデータを Laws マスクで解析するためには、ボリュームデータを連続する2次元の画像に分割し、分割した2次元画像に対して前節で述べた2次元マスクを用いて各画像から形状特徴を計算することが可能である。しかし、このような方法では、3次元ボリュームデータを2次元画像に分割した際、一部の情報が失われてしまう恐れがある。そこで、本研究では、3次元ボリュームデータを2次元画像へ変換せずに、3次元のマスクを用いることで、3次元ボリュームデータから直接、形状特徴を計算する手法を提案する。2次元のマスクは1次元の基本マスクを掛け合わせることで作成することができる。3次元マスクの場合には、さらに、2次元マスクに1次元の基本マスクを掛け合わせることで3次元のマスクを作成した。以下は1 × 3の基本マスクを掛け合わせて作成した3次元マスクであり、組み合わせの数は27個（3 × 3 × 3）である。

E3E3E3, E3E3L3, E3E3S3, E3L3E3, E3L3L3, E3L3S3, E3S3E3, E3S3L3, E3S3S3,
 L3E3E3, L3E3L3, L3E3S3, L3L3E3, L3L3L3, L3L3S3, L3S3E3, L3S3L3, L3S3S3,
 S3E3E3, S3E3L3, S3E3S3, S3L3E3, S3L3L3, S3L3S3, S3S3E3, S3S3L3, S3S3S3

この3次元マスク（3 × 3 × 3）内の数値を正規化して画像化すると図1を得ることができる。図1では、3次元のマスクを、3 × 3の2次元画像として3枚並べている。ピクセル色が白くなるほど数値が大きく、ピクセル色が黒くなるほど数値が小さくなっている。

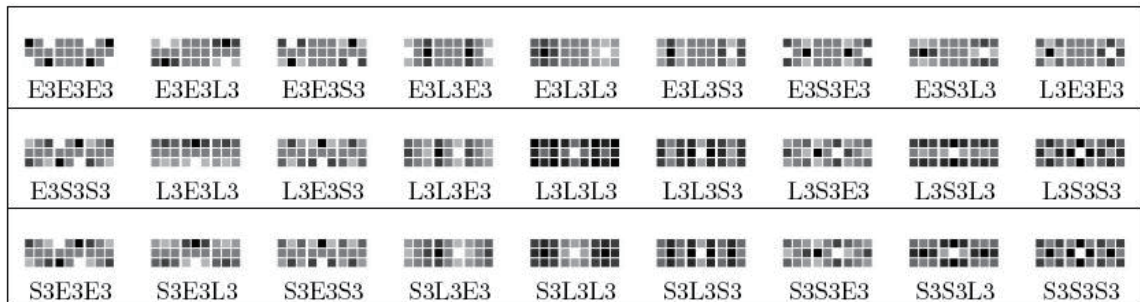


図1：3 × 3 × 3 の Laws マスク。マスクの係数を濃淡値として2次元画像化した。

以下は1×5の基本マスクを掛け合わせることで作成した3次元マスクであり、組み合わせの数は125個(5×5×5)である。

L5L5L5, L5L5E5, L5L5S5, L5L5R5, L5L5W5,
L5E5L5, L5E5E5, L5E5S5, L5E5R5, L5E5W5,
L5S5L5, L5S5E5, L5S5S5, L5S5R5, L5S5W5,
L5R5L5, L5R5E5, L5R5S5, L5R5R5, L5R5W5,
L5W5L5, L5W5E5, L5W5S5, L5W5R5, L5W5W5,
E5L5L5, E5L5E5, E5L5S5, E5L5R5, E5L5W5,
E5E5L5, E5E5E5, E5E5S5, E5E5R5, E5E5W5,
E5S5L5, E5S5E5, E5S5S5, E5S5R5, E5S5W5,
E5R5L5, E5R5E5, E5R5S5, E5R5R5, E5R5W5,
E5W5L5, E5W5E5, E5W5S5, E5W5R5, E5W5W5,
S5L5L5, S5L5E5, S5L5S5, S5L5R5, S5L5W5,
S5E5L5, S5E5E5, S5E5S5, S5E5R5, S5E5W5,
S5S5L5, S5S5E5, S5S5S5, S5S5R5, S5S5W5,
S5R5L5, S5R5E5, S5R5S5, S5R5R5, S5R5W5,
S5W5L5, S5W5E5, S5W5S5, S5W5R5, S5W5W5,
R5L5L5, R5L5E5, R5L5S5, R5L5R5, R5L5W5,
R5E5L5, R5E5E5, R5E5S5, R5E5R5, R5E5W5,
R5S5L5, R5S5E5, R5S5S5, R5S5R5, R5S5W5,
R5R5L5, R5R5E5, R5R5S5, R5R5R5, R5R5W5,
R5W5L5, R5W5E5, R5W5S5, R5W5R5, R5W5W5,
W5L5L5, W5L5E5, W5L5S5, W5L5R5, W5L5W5,
W5E5L5, W5E5E5, W5E5S5, W5E5R5, W5E5W5,
W5S5L5, W5S5E5, W5S5S5, W5S5R5, W5S5W5,
W5R5L5, W5R5E5, W5R5S5, W5R5R5, W5R5W5,
W5W5L5, W5W5E5, W5W5S5, W5W5R5, W5W5W5

この3次元マスク(5×5×5)内の数値を正規化して画像化すると図2を得ることができる。図2では、3次元のマスクを、5×5の2次元画像として5枚並べている。ピクセル色が白くなるほど数値が大きく、ピクセル色が黒くなるほど数値が小さくなっている。

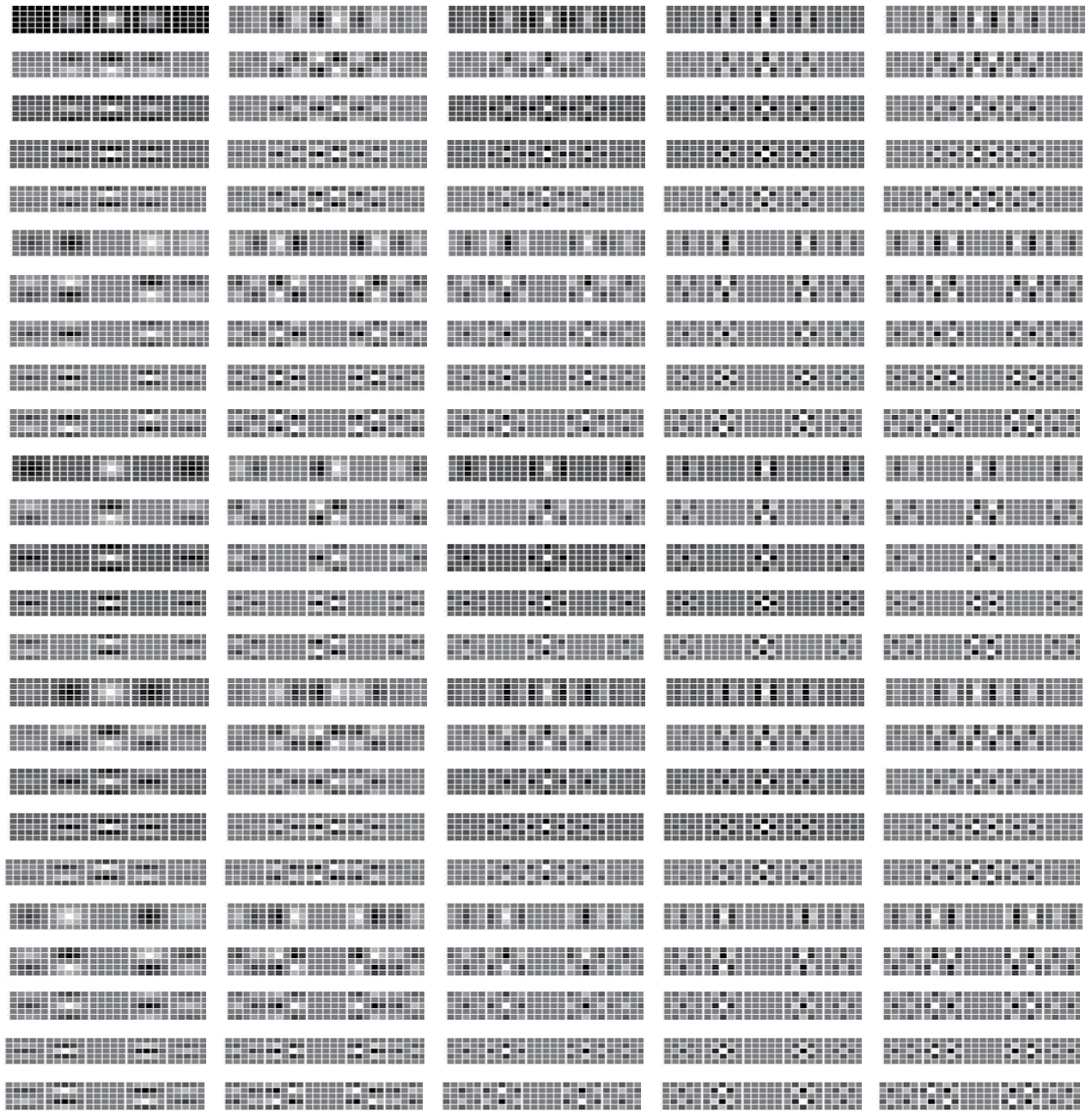


図 2 : $5 \times 5 \times 5$ の Laws マスク。マスクの係数を濃淡値として 2 次元画像化した。

2-3 3次元 LAWS マスクを用いた”畳み込み演算”

作成した3次元マスク（ $3 \times 3 \times 3$ ）を用いて、”畳み込み演算”（Convolution）を行った結果の例を図3に示す。”畳み込み演算”にもちいた入力3次元ボリュームデータには、2種類のテクスチャーが含まれている。ボリュームデータのサイズは $64 \times 64 \times 64$ である。図3の画像は、3次元ボリュームデータの一部をスライスした2次元画像（ 64×64 ）として切り出したものである。演算の結果、マスク L3E3L3 では、2種類のテクスチャーの境界を分離する白色の線ができる。（図3では線になっているが、実際には3次元ボリュームデータであり、平面で分離されている。）マスク L3S3L3 では、黒色の線によって、2種類のテクスチャーが分離されている。2次元のLawsマスクと同様に3次元のLawsマスクはボリュームデータのセグメンテーションに利用することができる。どのマスクがセグメンテーションに最適となるのかは、入力となる3次元ボリュームデータに依存している。学習用のサンプルデータから抽出されるパターンの特徴を、統計的手法などで解析することによって、どのマスクがセグメンテーションに利用できるのかを計算することができる。

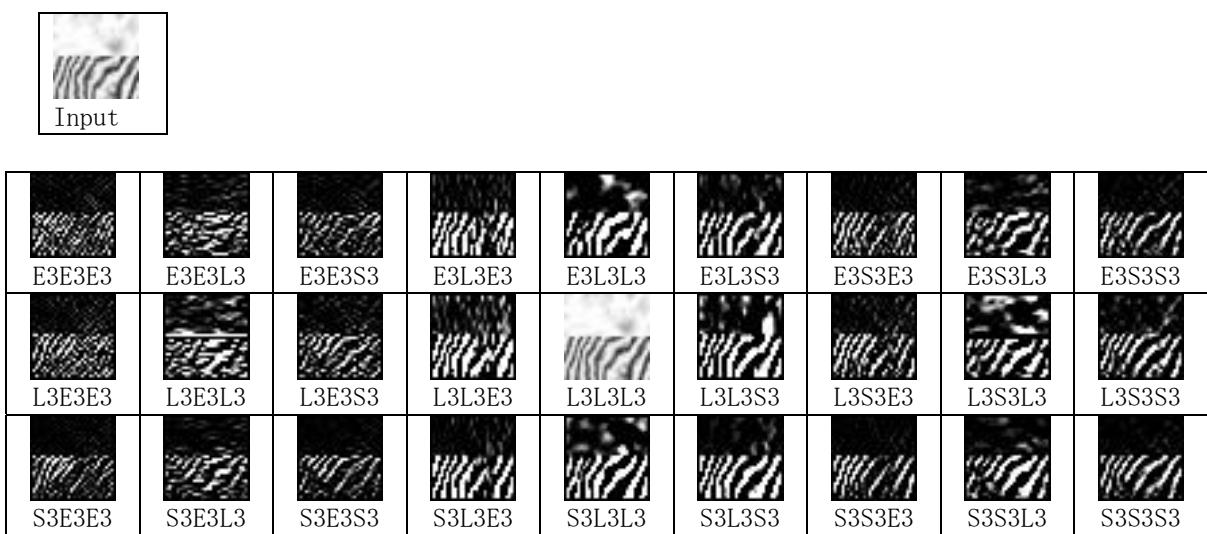


図3：3次元マスク（ $3 \times 3 \times 3$ ）を用いて、”畳み込み演算”を行った結果の例

図4に3次元マスク（ $5 \times 5 \times 5$ ）を用いて”畳み込み演算”を行った結果を示す。マスク L5S5L5 やマスク L5E5L5 など黒色の分離線（実際には平面によって分離されている）、マスク L5W5L5 で白色の分離線が得られている。マスクサイズが $3 \times 3 \times 3$ から $5 \times 5 \times 5$ へ大きくなったため、”畳み込み演算”の結果として抽出されるパターン画像は粗くなっている。また、”畳み込み演算”ができない画像の外周部分（黒色の縁）も増加している。マスク長が長くなることによって、抽出できる”畳み込み演算”後の画像の数は増加するが、図4からも推測できるように、冗長な演算結果（パターン特徴）も多くなると思われる。大量の冗長な形状特徴を計算に用いると検索・分類の結果に悪影響が出る場合がある。マスク長が長いマスクを用いてセグメンテーション、分類、認識等を行うためには、類似したパターン特徴の統合や、適切な重み付けを行うなどの冗長な形状特徴を排除する処理が必要になると考えられる。

L5L5L5	L5L5E5	L5L5S5	L5L5R5	L5L5W5	L5E5L5	L5E5E5	L5E5S5	L5E5R5	L5E5W5
L5S5L5	L5S5E5	L5S5S5	L5S5R5	L5S5W5	L5R5L5	L5R5E5	L5R5S5	L5R5R5	L5R5W5
L5W5L5	L5W5E5	L5W5S5	L5W5R5	L5W5W5	E5L5L5	E5L5E5	E5L5S5	E5L5R5	E5L5W5
E5E5L5	E5E5E5	E5E5S5	E5E5R5	E5E5W5	E5S5L5	E5S5E5	E5S5S5	E5S5R5	E5S5W5
E5R5L5	E5R5E5	E5R5S5	E5R5R5	E5R5W5	E5W5L5	E5W5E5	E5W5S5	E5W5R5	E5W5W5
S5L5L5	S5L5E5	S5L5S5	S5L5R5	S5L5W5	S5E5L5	S5E5E5	S5E5S5	S5E5R5	S5E5W5
S5S5L5	S5S5E5	S5S5S5	S5S5R5	S5S5W5	S5R5L5	S5R5E5	S5R5S5	S5R5R5	S5R5W5
S5W5L5	S5W5E5	S5W5S5	S5W5R5	S5W5W5	R5L5L5	R5L5E5	R5L5S5	R5L5R5	R5L5W5
R5E5L5	R5E5E5	R5E5S5	R5E5R5	R5E5W5	R5S5L5	R5S5E5	R5S5S5	R5S5R5	R5S5W5
R5R5L5	R5R5E5	R5R5S5	R5R5R5	R5R5W5	R5W5L5	R5W5E5	R5W5S5	R5W5R5	R5W5W5
W5L5L5	W5L5E5	W5L5S5	W5L5R5	W5L5W5	W5E5L5	W5E5E5	W5E5S5	W5E5R5	W5E5W5
W5S5L5	W5S5E5	W5S5S5	W5S5R5	W5S5W5	W5R5L5	W5R5E5	W5R5S5	W5R5R5	W5R5W5
W5W5L5	W5W5E5	W5W5S5	W5W5R5	W5W5W5					

図4：3次元マスク（ $5 \times 5 \times 5$ ）を用いて、“畳み込み演算”を行った結果の例

2-4 3次元 LAWS マスクの結合によるパターン特徴次元の削減

3次元の LAWS マスクの場合、1次元の基本マスクの組み合わせが多いため、2次元 LAWS マスクと比較して、マスクの数は非常に多くなる。その結果、算出される形状特徴の数も増加してしまう。形状特徴の数があまり多くなりすぎると、統計的な手法を用いたパターン認識等の計算では、より多くの認識用の学習用サンプルデータが必要になる。また、高次元の特徴量計算が必要になるなど、認識結果に悪影響があるケースがあることが知られている。そこで、本研究の実験では、冗長なマスクを統合することによって全体のマスク数（形状特徴の数）を削減する。3×3×3のマスクでは、X軸、Y軸、Z軸の各軸回りの90度の回転によって同値となるマスクが存在する。実験では、シミュレーションプログラムを作成した。そして、各マスクに90度および-90度の回転をX軸、Y軸、Z軸の各軸回り行い、少なくとも2つの軸方向で同値となるマスクを探索し、同値となったマスクを結合して1つのマスクとして扱うことにした。以下は、シミュレーションプログラムによって得られた、冗長なマスクを結合した3×3×3のマスクの組み合わせである。マスクの後尾にRがついたマスクが同値となったマスクで、90度の回転（Rotation）で同値となることを示している。例を上げると、マスク L3L3E3 とマスク E3L3L3 は回転によって同値となるので、L3L3E3_R という1つのマスクに統合できる。なお、特徴の正規化を行うため、L3L3E3_R の特徴は $(L3L3E3 + E3L3L3) \times 0.5$ という計算を行う。

3×3×3のマスクは全部で27個であったが、冗長なマスクを結合することによって、18個に削減することができる。

```
L3L3L3  L3L3E3_R  L3L3S3_R  L3E3L3  L3E3E3_R  L3E3S3_R  L3S3L3  L3S3E3_R  L3S3S3_R  E3L3E3
E3L3S3_R  E3E3E3  E3E3S3_R  E3S3E3  E3S3S3_R  S3L3S3  S3E3S3  S3S3S3
```

5×5×5のマスクの場合では、以下のような結果となった。125個あったマスクは80個に削減することができた。

```
L5L5L5  L5L5E5_R  L5L5S5_R  L5L5R5_R  L5L5W5_R  L5E5L5  L5E5E5_R  L5E5S5_R  L5E5R5_R  L5E5W5_R  L5S5L5
L5S5E5_R  L5S5S5_R  L5S5R5_R  L5S5W5_R  L5R5L5  L5R5E5_R  L5R5S5_R  L5R5R5_R  L5R5W5_R  L5W5L5  L5W5E5_R
L5W5S5_R  L5W5R5_R  L5W5W5_R  E5L5E5  E5L5S5_R  E5L5R5_R  E5L5W5  E5E5E5  E5E5S5_R  E5E5R5_R  E5E5W5
E5S5E5  E5S5S5_R  E5S5R5_R  E5S5W5  E5R5E5  E5R5S5_R  E5R5R5_R  E5R5W5  E5W5E5  E5W5S5_R  E5W5R5_R
E5W5W5  S5L5S5  S5L5R5_R  S5L5W5_R  S5E5S5  S5E5R5_R  S5E5W5_R  S5S5S5  S5S5R5_R  S5S5W5_R  S5R5S5
S5R5R5_R  S5R5W5_R  S5W5S5  S5W5R5_R  S5W5W5_R  R5L5R5  R5L5W5_R  R5E5R5  R5E5W5_R  R5S5R5  R5S5W5_R
R5R5R5  R5R5W5_R  R5W5R5  R5W5W5_R  W5L5E5  W5L5W5  W5E5E5  W5E5W5  W5S5E5  W5S5W5  W5R5E5
W5R5W5  W5W5E5  W5W5W5
```

3 3次元テクスチャーの特徴抽出実験

3-1 3次元ソリッドテクスチャーの生成

ノイズ関数により人工的に作成したソリッドテクスチャーを実験に用いた。以下は作成したソリッドテクスチャーの例である。実験では、図5に示すソリッドテクスチャーを Perlin ノイズ関数によって生成した。

ソリッドテクスチャーは、Marble、Turbulence、Cloud、Wood の4種類である。ソリッドテクスチャーのサイズは64×64×64である。

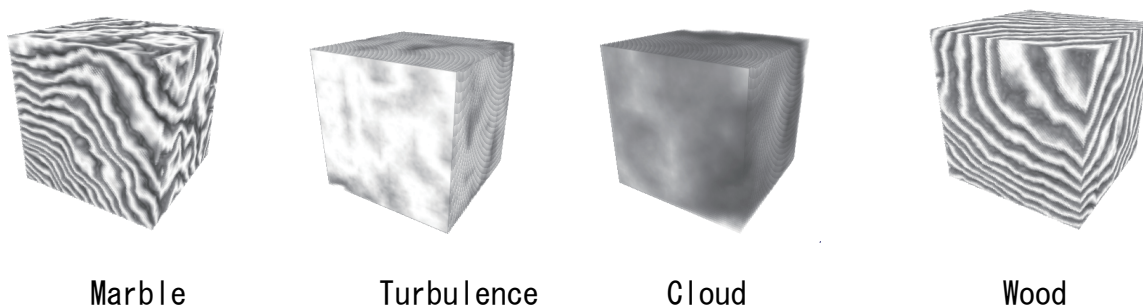


図5：64×64×64の3次元ソリッドテクスチャー

3-3 3次元ソリッドテクスチャーの類似検索

3次元ソリッドテクスチャーのデータベースを作成し、類似検索の実験を行った。データベースは4種類のソリッドテクスチャーを各400個生成し、合計1600個のデータをもつデータベースを作成した。そして、各ソリッドテクスチャーに対して3次元 Laws マスクを用いた”畳み込み演算”を行い、各ソリッドテクスチャーの形状特徴を抽出した。得られた形状特徴を多次元空間内の座標として扱い、検索キーとなるソリッドテクスチャーの座標とその他のソリッドテクスチャーの座標の距離を計算した。そして、計算された距離をソーティングして、その距離に近い順に表示することによって、類似検索を実現した。

図6に3次元ソリッドテクスチャーの類似検索の結果の例を示す。一番左のソリッドテクスチャーが検索キーである。図中の数字は検索キーに対してシステムが類似していると判断した順番を示している。検索システムは、図6に示すように類似したパターンを持つソリッドテクスチャーを検索することができた。検索結果は、形状特徴を索引としてデータベース内にあらかじめ計算しておけば、一般的な PC を用いて、約2秒以内で表示することができた。Marble と Wood のソリッドテクスチャーを誤検索するケースが幾つかあったが、その他のソリッドテクスチャーに関しては、ほぼ正しい類似検索結果を得ることができた。

検索キー	1	2	3	4	5

図6：類似検索結果の例。

3-3 3次元ソリッドテキストチャーの線形判別分析による分類

3次元ソリッドテキストチャーのデータベースを作成し、線形判別分析を用いたデータベースの分類に関する実験を行った。実験には4種類のソリッドテキストチャーを各50個生成し、合計200個のデータをもつデータベースを作成した。各ソリッドテキストチャーから3次元 Laws マスク ($3 \times 3 \times 3$ と $5 \times 5 \times 5$) による”畳み込み演算”を行い、各ソリッドテキストチャーから形状特徴を計算した。そして、200個のデータを半分に分割し、100個を学習用のデータ、残りの100個を線形判別分析のテスト用のデータとした。まず、学習用のデータを用いて、形状特徴を説明変数、ソリッドテキストチャーの種類(4種類)を目的変数として線形判別分析を行った。実験は、 $3 \times 3 \times 3$ の Laws マスクを利用した形状特徴(18個の形状特徴)と $5 \times 5 \times 5$ の Laws マスクを利用した形状特徴(80個の形状特徴)の場合の2つを行った。4つのソリッドテキストチャーを判別することから、3つの線形判別関数(LD1、LD2、LD3)が得られた。それぞれの判別関数がどれくらい全体のグループの分散を説明しているかを示すグループ間の比率は以下の通りである。

	LD1	LD2	LD3
$3 \times 3 \times 3$	0.9633	0.0314	0.0053
$5 \times 5 \times 5$	0.5403	0.3145	0.1452

比率を見ると、 $3 \times 3 \times 3$ の Laws マスクを用いた場合は1つの関数(LD1)、 $5 \times 5 \times 5$ の Laws マスクを用いた場合は2つの関数(LD1 と LD2) でほぼ判別ができるように判別関数が計算されている。

学習データの判別関数値の散布を図7と図8に示す。図7は $3 \times 3 \times 3$ の Laws マスクを用いた場合、図8は $5 \times 5 \times 5$ の Laws マスクを用いた場合である。それぞれ、3つの判別関数(LD1、LD2、LD3)があるので、合計6つの対となる散布図がある。図7では、テキストチャー“M”と“W”で重なり合う部分があり、この2つのテキストチャーの判別が難しくなっていることが推測できる。

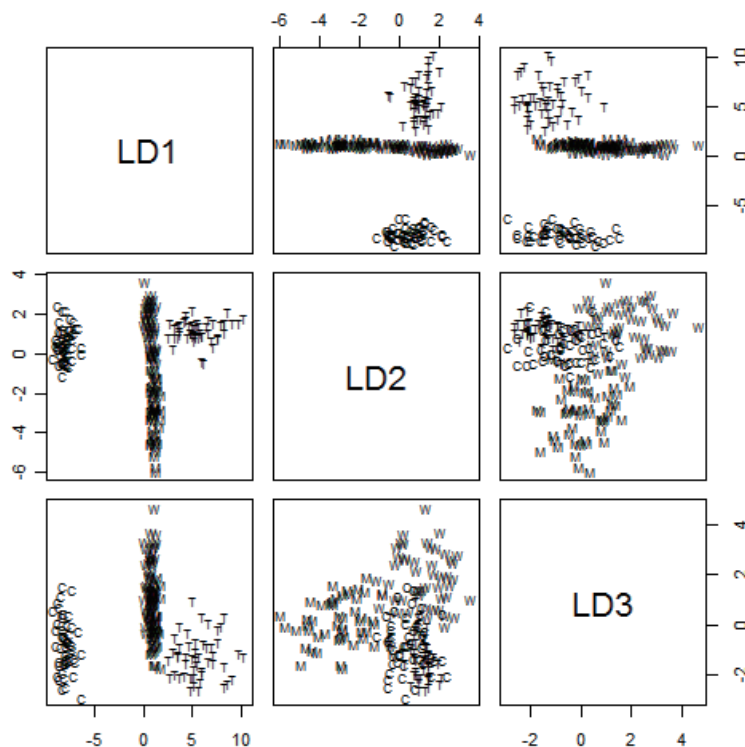


図7：学習データの判別関数値の散布図 ($3 \times 3 \times 3$)

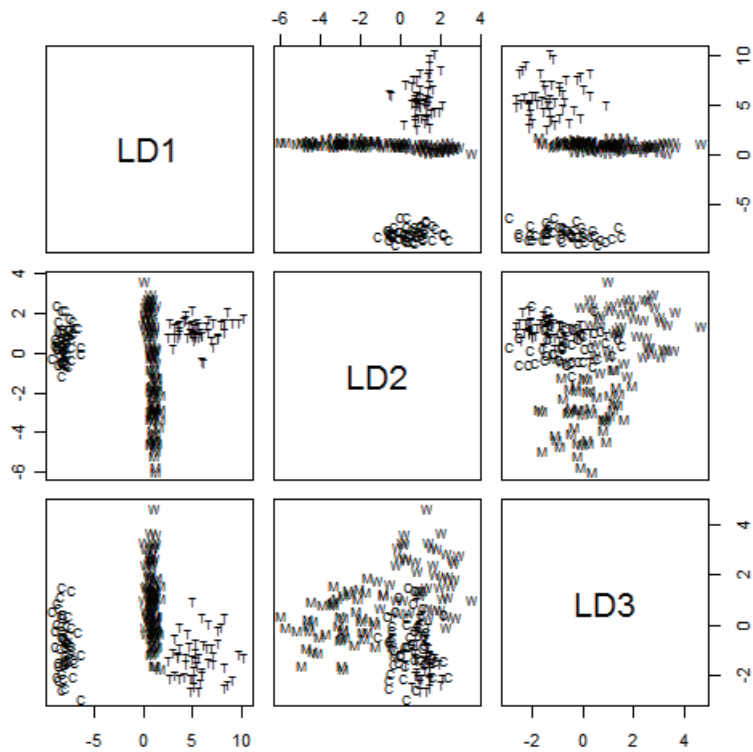


図 8 : 学習データの判別関数値の散布図 (5 × 5 × 5)

学習データを用いて計算した判別関数によって、テスト用のデータ 100 個が、どのように判別されたのかを示すクロス表を、図 9 と図 10 に示す。図 9 の結果のように、3 × 3 × 3 の Laws マスクを用いた場合には、“C” はすべて正しく判別されている。“M” は 9 個が “W” に誤判別、“T” は 1 個が “M” に誤判別、“W” は 9 個が “M” に誤判別されている。テストデータ全体 100 個の内 19 個 (9 + 1 + 9) が誤判別されており、正判別率は 81% である。特に “M” と “W” を誤判別するケースが多くなっている。それに対して、図 10 の結果のように 5 × 5 × 5 の Laws マスクを用いた場合には、すべてのデータが正しく判別され、この実験の場合、正判別率は 100% である。

	C	M	T	W
C	25	0	0	0
M	0	16	0	9
T	0	1	24	0
W	0	9	0	16

図 9 : クロス表 (3 × 3 × 3)

	C	M	T	W
C	25	0	0	0
M	0	25	0	0
T	0	0	25	0
W	0	0	0	25

図 10 : クロス表 (5 × 5 × 5)

4 まとめ

本研究では、3次元ボリュームデータから形状特徴を抽出するため、2次元画像解析に有効な手法と知られている Laws マスクを3次元に拡張した。そして、シミュレーションプログラムによって生成した3次元 Laws マスクを応用し、3次元ボリュームデータからパターン特徴を抽出した。抽出したパターン特徴を基にして、類似検索、判別分析を用いた統計的な分類を実現した。

【参考文献】

1. [Kurani2004] A. Kurani, D. H. Xu, J. D. Furst, and D. S. Raicu, "Co-occurrence matrices for volumetric data", The 7th IASTED International Conference on Computer Graphics and Imaging, CGIM 2004, USA, 2004.
2. [Xu2004] D. H. Xu, A. Kurani, J. D. Furst, and D. S. Raicu, "Run-length encoding for volumetric texture", The 4th IASTED International Conference on Visualization, Imaging, and Image Processing, VIIP 2004, 452-131, Marbella, 2004.
3. [Laws1980] K. Laws. Rapid texture identification. In SPIE Vol. 238 Image Processing for Missile Guidance, pages 376-380, 1980.
4. [Laws1979] K. I. Laws. Texture energy measures. In DARPA Image Understanding Workshop, pages 47--51. DARPA, Los Altos, CA, 1979.

1. 〈発表資料〉

題名	掲載誌・学会名等	発表年月
A Volume Data Classification Method Based on 3D Spherical Filters	The 2007 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2007), pp.1095--1100, ISBN:1-4244-0991-8, Montreal	10/2007.
A Solid Texture Analysis Based on Three Dimensional Convolution Kernels	IS&T/SPIE(The International Society for Optical Engineering) Electronic Imaging 2007, Videometrics IX, (EI-2007), Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6491, 64910W pp.1-8, 0277-786X/07/ San Jose, USA,	01/2007.