コンピュータビジョン　58－3
（1989. 1. 26）

形状処理の集合的方法と代数的方法

Morphological Approach and Algebraic Approach for Image Analysis

中村　敏浩　　井宮　淳
Toshihiro NAKAMURA　　Atsushi IMIYA

金沢大学　工学部　電気・情報工学科
Department of Electrical and Computer Engineering,Faculty of Technology,Kanazawa University

あらまし　数理形態学（Mathematical Morphology）は，画像を画像平面上の部分集合として扱うため，そのデータ構造が画像の形状情報とうまく対応する．そのため，形状処理アルゴリズムを構成するのには有効な手法である．ところが，数理形態学の演算は計算機には不向きである．そこで本論文では，まず2値画像処理アルゴリズムの記述法としてブール束上の画像多項式を新しく定義する．次いで，数理形態学の演算とブール束上の画像多項式の演算との関係を明らかにし，変換する方法を示す．そして，数理形態学の演算をブール束上の画像多項式の演算に変換して実行することを提案する．これによって，数理形態学で記述された形状処理アルゴリズムが計算機でモジュール化して効率よく実行できることが明らかになる．また，ブール束上の画像多項式によってアルゴリズムの複雑を定量的に評価できることを明らかにする．

Abstract　Mathematical morphology is more useful than conventional image processing method, to describe image analysis algorithms, because the morphological operations, which are set transformations to subsets of Euclidean space, relate directly to shape. Set transformations are, however, not suitable for calculation by digital computers. So we suggest an efficient calculation method of morphological operations. The method treats the image value of the points as elements of binary Boolean algebra. Then we clarify that the operation of Minkowsky addition can be converted into Cauchy product of Boolean elements, and all operations of Mathematical morphology can be executed on computers by five modules of the Boolean algebra operation. Finaly we propose a measure of computational complexity of image analysis algorithms.

1　まえがき

物体の形状処理は，入力された画像の中の物体の個数の計数や，注目する部分の取り出し，形の特徴の抽出などの処理に用いられる．また，文字など線図形の認識にも形状処理の手法が用いられる．形状処理は，物体の形の情報に注目するため本質的に2値画像処理である．このように2値画像処理は画像処理の中で重要な位置をしめる．2値画像処理は，従来から盛んに研究されていて，文献［1］では最大値・最小値フィルタを中心とした2値画像処理が述べられている．最近，2値画像処理の新しい手法として数理形態学（Mathematical Morphology）が注目されている．数理形態学は従来の2値画像処理に対して多くの利点を持っている．従来の2値画像処理の手法では，データ構造が画像の形状情報とうまく対応していないため，画像処理の手順を視覚化しにくく，アルゴリズム設計の見とおしが悪いため，系統的にアルゴリズムを構成することが難しい．また，記述されたアルゴリズムの意味が分かりずらいなどの問題点がある．ところで，数理形態学は，2値画像を画像の図形要素に属する画像平面上の点または画素の集合として表し，集合として表された2値画像に対して種々の集合演算を行うことによって2値画像処理を行う手法である．すなわち，数理形態学では2値画像を2次元平面上の部分集合と考えて集合演算を行うことになる．集合演算は，集合を平面上の図形と直接的に対応させて行うことができる．したがって，数理形態学によって形状処理アルゴリズムを視覚的に構成することができ，効率的なアルゴリズム設計が可能になる．また，記述されたアルゴリズムの意味も分かりやすい．

数理形態学によって効率的に2値画像処理アルゴリズムを構成することができるが，構成されたアルゴリズムを計算機で実行するとき，数理形態学の集合演算を計算機でどのように扱うかという問題が生じる．そこで本論文では，著者らが画像修復のために提案した画像多項式によって2値画像の形状処理アルゴリズムを実行する手法を提案する．すなわち，係数をブール束上の値とする画像多項式，ブール束上の画像多項式を定義する．そして，数理形態学の集合演算を“ブール束上の画像多項式”の演算に変換する方法を

示し，ブール束上の画像多項式によって，数理形態学で記述された形状処理アルゴリズムをモジュール化して計算機で実行することを提案する．

ブール束上の画像多項式の演算の中で乗算が最も複雑である．そこで，数理形態学で記述されたアルゴリズムの複雑さを評価するときに，ブール束上の画像多項式に変換したときにアルゴリズムに含まれる乗算の回数によって評価することができる．多項式の乗算は係数のコーシー積を求めることと同じであるから，ブール束の元のコーシー積によってアルゴリズムの複雑さを定量的に評価できることになるのであるのである．また，数理形態学と最大値・最小値フィルタとの関係についても述べる．

本論文では，離散2値画像を取り扱うことにする．そして，離散2値画像を単に2値画像と呼ぶことにする．最後に，2値画像の形状処理アルゴリズムの例として距離変換，スケルトン，細線化について述べる．そして，これらをブール束上の画像多項式の演算に変換して計算機実験した結果を示す．

1 2値画像の表現

画面全体を通じて2種類の濃度値しか持たない画像を2値画像という．通常，この2種類の濃度値に対して0と1を割り当てる．本論文では，0は背景を表し，1は図形要素を表すとする．また，図形要素の連結成分を図形と呼び，濃度値が1である画素を1－画素と呼ぶことにする．

本章では，2値画像の種々の表現についてまとめる．画像平面上の画素または点 (m,n) の濃度値を f_{mn} とすると，2値画像は式(1)のように各要素が0または1をとる配列として表すことができる．

$$\mathbf{F}_B = \{f_{mn}\} \qquad f_{mn} = 0 \text{ または } 1 \tag{1}$$

式(1)で表される2値画像全体の集合を B とする．そして，式(1)を2値画像の配列表現と呼ぶことにする．配列表現は画像をそのまま行列として表したものである．

数理形態学では2値画像を1－画素の集合，すなわち図形に属する画素の集合として表す．そして，このとき画素は画素を代表する画像平面の座標で表すことにする．式(1)として配列表現された2値画像を数理形態学で表現すると

$$\mathbf{F}_M = \{(m,n) : f_{mn} = 1\} \tag{2}$$

となる．式(2)で表される2値画像全体の集合を M とする．そして，式(2)を2値画像の数理形態学表現と呼ぶことにする．以降，式(2)の集合の要素 (m,n) をベクトル x として書くこともある．

画素 (m,n) の濃度値が f_{mn} であるとき画像を

$$\mathbf{F}_P = \sum_{m,n} f_{mn} x^m y^n \tag{3}$$

のように濃度値を係数に持つ形式的べき級数として表したものを画像多項式という．ただし，式(3)の和分は平面上の全ての点 (m,n) にわたるものとする．さらに，2値画像の場合，式(3)の多項式の係数は，1または0の値をとることにし，これを2値画像の画像多項式表現と呼ぶことにする．また，式(3)で表される2値画像の画像多項式全体の集合を P とする．

2 数理形態学

2.1 集合演算

数理形態学は2値画像を集合として表し，集合として表された画像に対して種々の集合演算を行うことによって2値画像処理を行うものである．本章では，数理形態学の演算の基本となる集合演算についてまとめる．すなわち，数理形態学の全ての演算は以下の集合演算を基本として組み立てられる．

集合 $\mathbf{A}$，$\mathbf{B}$ を集合 M の要素とする．すなわち，$\mathbf{A}$，$\mathbf{B}$ は平面上の点の座標値を要素とする集合であるとする．

集合 $\mathbf{A}$ と対応する平面上の部分集合をベクトル x だけ平行移動した集合を $\mathbf{A}_x$ と書くことにする．すなわち，$x+y$ をベクトル和とすれば，

$$\mathbf{A}_x = \{x + y : y \in \mathbf{A}\} \tag{4}$$

となる．$\mathbf{A}_x$ を集合 $\mathbf{A}$ の平行移動と呼ぶことにする．

集合 $\mathbf{A}$ に対応する平面上の部分集合を原点に関して180度回転した集合を $\check{\mathbf{A}}$ と書くことにする．すなわち

$$\check{\mathbf{A}} = \{-x : x \in \mathbf{A}\} \tag{5}$$

とする．$\check{\mathbf{A}}$ を集合 $\mathbf{A}$ の回転と呼ぶことにする．

集合 $\mathbf{A}$ に対応する平面上の部分集合の，平面全体に対する補集合をとったものを $\overline{\mathbf{A}}$ と書くことにする．すなわち，

$$\overline{\mathbf{A}} = \{x : x \notin \mathbf{A}\} \tag{6}$$

とする．$\overline{\mathbf{A}}$ は2値画像の白地と黒地を反転したものに対応する．そこで，$\overline{\mathbf{A}}$ を集合 $\mathbf{A}$ の反転と呼ぶことにする．

集合 $\mathbf{A}$ と集合 $\mathbf{B}$ に対応する平面上の部分集合に対して和集合，差集合をとった集合をそれぞれ $\mathbf{A} \bigcup \mathbf{B}$，$\mathbf{A} \setminus \mathbf{B}$ と書くことにする．すなわち，

$$\mathbf{A} \bigcup \mathbf{B} = \{x : x \in \mathbf{A} \text{ または } x \in \mathbf{B}\} \tag{7}$$

$$\mathbf{A} \setminus \mathbf{B} = \{x : x \in \mathbf{A} \text{ かつ } x \notin \mathbf{B}\} \tag{8}$$

とする．$\mathbf{A} \bigcup \mathbf{B}$，$\mathbf{A} \setminus \mathbf{B}$ をそれぞれ，集合 $\mathbf{A}$ と $\mathbf{B}$ の和と差と呼ぶことにする．

2.2 数理形態学の演算

数理形態学の演算は2.1で述べた集合演算を基本として構成される．本節では，数理形態学の演算についてまとめる．

2つの集合 $\mathbf{A}, \mathbf{B} \in M$ に対して，演算

$$\begin{aligned} \mathbf{A} \oplus \mathbf{B} &\equiv \{x + y : x \in \mathbf{A}, y \in \mathbf{B}\} \\ &= \bigcup_{y \in \mathbf{B}} \mathbf{A}_y \end{aligned} \tag{9}$$

を $\mathbf{A}$ と $\mathbf{B}$ とのミンコウスキー和という．さらに，ミンコウスキー和から定義される演算

$$\begin{aligned}\mathbf{A}\ominus\mathbf{B} &\equiv \overline{(\overline{\mathbf{A}}\oplus\mathbf{B})}\\ &= \bigcap_{y\in\mathbf{B}}\mathbf{A}_y \end{aligned} \tag{10}$$

を $\mathbf{A}$ と $\mathbf{B}$ とのミンコウスキー差という．ミンコウスキー和，ミンコウスキー差に関して次の性質が成り立つ．

$$\mathbf{A}\oplus\mathbf{B}=\mathbf{B}\oplus\mathbf{A} \tag{11}$$

$$(\mathbf{A}\oplus\mathbf{B})\oplus\mathbf{C}=\mathbf{A}\oplus(\mathbf{B}\oplus\mathbf{C}) \tag{12}$$

$$(\mathbf{A}\ominus\mathbf{B})\ominus\mathbf{C}=\mathbf{A}\ominus(\mathbf{B}\oplus\mathbf{C}) \tag{13}$$

ミンコウスキー和とミンコウスキー差は数理形態学の基本演算となる．

ミンコウスキー和から集合 $\mathbf{A}$ の $\mathbf{B}$ による拡張（dilation）$D(\mathbf{A},\mathbf{B})$ が次のように定義される．

$$\begin{aligned}D(\mathbf{A},\mathbf{B}) &\equiv \mathbf{A}\oplus\check{\mathbf{B}}\\ &= \{x:\mathbf{A}\bigcap\mathbf{B}_x\neq\phi\}\end{aligned} \tag{14}$$

拡張は $\mathbf{A}$ をその境界に沿って $\mathbf{B}$ の大きさだけ太める働きをする演算である．

また，ミンコウスキー差から集合 $\mathbf{A}$ の $\mathbf{B}$ による侵食（erosion）$E(\mathbf{A},\mathbf{B})$ が次のように定義される．

$$\begin{aligned}E(\mathbf{A},\mathbf{B}) &\equiv \mathbf{A}\ominus\check{\mathbf{B}}\\ &= \{x:\mathbf{B}_x\subset\mathbf{A}\}\end{aligned} \tag{15}$$

侵食は拡張とは逆に $\mathbf{A}$ をその境界に沿って $\mathbf{B}$ の大きさだけ細める働きをする演算である．

さらに，拡張と侵食によって集合 $\mathbf{A}$ の $\mathbf{B}$ による開放（opening）$O(\mathbf{A},\mathbf{B})$ と閉鎖（closing）$C(\mathbf{A},\mathbf{B})$ が次のように定義される．

$$O(\mathbf{A},\mathbf{B})\equiv E(\mathbf{A},\mathbf{B})\oplus\mathbf{B} \tag{16}$$

$$C(\mathbf{A},\mathbf{B})\equiv D(\mathbf{A},\mathbf{B})\ominus\mathbf{B} \tag{17}$$

開放，閉鎖とも元の図形を平滑化する演算である．開放は $\mathbf{A}$ から $\mathbf{B}$ より小さい図形成分を除去し，$\mathbf{B}$ より細い部分を切断する演算である．また，閉鎖は $\mathbf{A}$ の中で $\mathbf{B}$ より小さい距離で隣接する図形どうしを接続する演算である．

集合 $\mathbf{A}$ が集合 $\mathbf{B}$ と $\mathbf{C}$ の直和であるとは，集合 $\mathbf{B}$ と $\mathbf{C}$ に関して

$$\mathbf{A}=\mathbf{B}\bigcup\mathbf{C} \tag{18}$$

$$\mathbf{B}\bigcap\mathbf{C}\neq\emptyset \tag{19}$$

が成り立つことである．以降，集合 $\mathbf{A}$ に関して式(18)，(19)が成り立つことを $(\mathbf{B},\mathbf{C})$ と略記する．

集合 $\mathbf{B}$ が

$$\mathbf{B}=(\mathbf{B}_1,\mathbf{B}_2) \tag{20}$$

であるとき，集合 $\mathbf{A}$ の $\mathbf{B}$ による適合変換（Hit or Miss Transformation）は次のような集合となる

$$HM(\mathbf{A},\mathbf{B})=\{x:\mathbf{B}_{1x}\subset\mathbf{A},\mathbf{B}_{2x}\subset\overline{\mathbf{A}}\} \tag{21}$$

適合変換は，その近傍の図形成分が $\mathbf{B}_{1x}$ に適合して，かつ背景が $\mathbf{B}_{2x}$ に適合するような状態にある点 x を $\mathbf{A}$ から抽出するマスク処理になっている．したっがて，さまざまな $\mathbf{B}$ を用いることにより広汎なマスク処理を行うことができるのである．

式(21)は拡張と侵食によって

$$\begin{aligned}HM(\mathbf{A},\mathbf{B}) &= E(\mathbf{A},\mathbf{B}_1)\bigcap E(\mathbf{A},\mathbf{B}_2)\\ &= E(\mathbf{A},\mathbf{B}_1)\setminus D(\mathbf{A},\mathbf{B}_2)\end{aligned} \tag{22}$$

と書くことができる．式(22)より，マスク処理が数理形態学の基本的な集合演算の組合せによって行うことができることが分かる．

数理形態学の演算は，全て $OP(\mathbf{A},\mathbf{B})$ のように2項演算になっている．$\mathbf{A}$ は入力2値画像に対応する集合である．一方，$\mathbf{B}$ は演算を制御する集合である．数理形態学では $\mathbf{B}$ のことを構成要素(structuring element)と呼ぶ．

集合はベン図の例での分かるように平面上の図形と直接対応させて操作することができるので，数理形態学の演算によって画像処理の種々の演算を視覚的に記述できる．したがって，数理形態学の演算は2値画像の形状処理アルゴリズムを記述，構成するための強力な道具となるのである．

3 集合演算から代数演算への変換

3.1 集合演算から代数演算への変換

数理形態学の演算は視覚的なため2値画像の形状処理アルゴリズムを構成するには非常に有効な方法である．数理形態学の演算で記述されたアルゴリズムを計算機に実行させるとき，集合演算を効率よく計算機に実行させなければならない．しかし，集合演算を直接実行するには計算機は不向きである．そこで，集合演算を計算機向きの代数演算で実行することにする．

零元を0，単位元を1としたときの2元からなるブール束の演算規則は以下の式で与えられる［3］:

$$1\cdot 1=,1\cdot 0=0\cdot 1=0,0\cdot 0=0 \tag{23}$$

$$1+1=1,1+0=0+1=1,0+0=0 \tag{24}$$

$$\overline{1}=0,\overline{0}=1 \tag{25}$$

ブール代数は集合演算を代数演算で実行することを可能にする．また，計算機の内部は2値ブール代数を基礎にした論理回路によって動作している．したがって，式(23)～(25)のブール束の演算によって集合演算を計算機で効率よく実行することができるのである．そこで，2値画像の画像多項式の係数をブール束の元とみなして，式(23)～(25)のブール束の演算を，多項式の係数の計算に適用することにする．

3.2 ブール束上の画像多項式

画像多項式による画像処理は，多項式の係数に置かれた濃度値をある体 K の要素と考えて，多項式の加算や乗算を行うことによって画像処理を行う方法である［4］．

2値画像の場合，式(3)の画像多項式の係数は1と0である．そこで，多項式の係数どうしの演算を行うときに，係数を式(23)，(24)を満たすブール束の元として計算することにする．すなわち，係数どうしの乗算には式(23)を使い，加算には式(24)を使って計算することにする．そして，このブール束上の元を係数とする画像多項式をブール束上の画像多項式と呼ぶことにし，2値画像のブール束上の画像多項式表現全体の集合をP_Bと書くことにする．

$\mathbf{F}$，$\mathbf{G}$，$\mathbf{H} \in P_B$を

$$\mathbf{F} = \sum_{m,n} f_{mn} x^m y^n \tag{26}$$

$$\mathbf{G} = \sum_{m,n} g_{mn} x^m y^n \tag{27}$$

$$\mathbf{H} = \sum_{m,n} h_{mn} x^m y^n \tag{28}$$

とする．そこで，ブール束上の画像多項式の和を

$$\mathbf{H} = \mathbf{F} + \mathbf{G} \tag{29}$$

と書くことにする．このとき，f_{mn}，g_{mn}，h_{mn}に関して

$$h_{mn} = f_{mn} + g_{mn} \tag{30}$$

が成り立つ．ここで，＋は式(24)を満たす演算である．ブール束上の画像多項式の積を

$$\mathbf{H} = \mathbf{F} * \mathbf{G} \tag{31}$$

と書くことにする．このとき，h_{mn}は

$$h_{mn} = \sum_{p,q} f_{m-p\,n-q} \cdot g_{pq} \tag{32}$$

によって与えられる．ここで，$\cdot$は式(23)で定義される演算である．そして，式(24)の演算に従って和分を計算することにする．

上のブール束の演算からは多項式の差は直接には導けない．そこで，$-$を論理差として係数どうしの計算をする．すなわち，

$$h_{mn} = f_{mn} - g_{mn} = f_{mn} \cdot \overline{g}_{mn} \tag{33}$$

によって係数の計算をする．そして，式(33)を通してブール束上の画像多項式の差を定義する．また，ブール束上の画像多項式の差を

$$\mathbf{H} = \mathbf{F} - \mathbf{G} \tag{34}$$

と書くことにする．

2値画像をブール束上の画像多項式で表して和，差をとることは，それぞれ数理形態学で和集合，差集合をとることと等価である．

3.3 数理形態学とブール束上の画像多項式

同じ2値画像を表すことから，P_BからMへの全単写が存在する．これを，Ψとする．このとき，次の定理が成り立つ．

[定理1] Ψは，$\langle P_B, * \rangle$から$\langle M, \oplus \rangle$への同型写像である．

（証明） $\mathbf{F}_P, \mathbf{G}_P \in P_B$をそれぞれ，

$$\mathbf{F}_P = \sum_{m,n} f_{mn} x^m y^n \tag{35}$$

$$\mathbf{G}_P = \sum_{m,n} g_{mn} x^m y^n \tag{36}$$

とする．また，$\mathbf{F}_M, \mathbf{G}_M \in M$をそれぞれ，

$$\begin{aligned} \mathbf{F}_M &= \Psi(\mathbf{F}_P) \\ &= \{z : z = (m,n), f_{mn} = 1\} \end{aligned} \tag{37}$$

$$\begin{aligned} \mathbf{G}_M &= \Psi(\mathbf{G}_P) \\ &= \{z : z = (m,n), g_{mn} = 1\} \end{aligned} \tag{38}$$

とする．このとき，

$$\begin{aligned} &\Psi(\mathbf{F}_P * \mathbf{G}_P) \\ &= \{z : z = (m,n), \sum_{p,q} f_{m-p\,n-q} \cdot g_{pq} = 1\} \\ &= \{z : z = (m,n), \exists (p,q), f_{m-p\,n-q} \cdot g_{pq} = 1\} \\ &= \{z : z = (m,n), (m-p, n-q) \in \mathbf{F}_M, (p,q) \in \mathbf{G}_M\} \\ &= \{z : z = x + y, x \in \mathbf{F}_M, y \in \mathbf{G}_M\} \\ &= \Psi(\mathbf{F}_P) \oplus \Psi(\mathbf{G}_P) \end{aligned} \tag{39}$$

が成立する． （証明終わり）

定理1より，ブール束上の画像多項式の積によって与えられる2値画像は，数理形態学のミンコウスキー和で与えられる2値画像と同じになることが分かる．

3.4 集合演算のモジュール化

$\mathbf{F}$を2値画像のブール束上の画像多項式表現とする．$\mathbf{F}$の表す2値画像の数理形態学表現を回転したものはブール束上の画像多項式では，$\mathbf{F}$のxをx^{-1}に，yをy^{-1}に入れかえることによって得ることができる．これを$\check{\mathbf{F}}$と書くことにする：

$$\check{\mathbf{F}} = \sum_{m,n} f_{mn} x^{-m} y^{-n} \tag{40}$$

$\check{\mathbf{F}}$をブール束上の画像多項式の回転と呼ぶことにする．

また，数理形態学の反転に相当するものは$\mathbf{F}$の係数の1と0を入れかえることによって得ることができる．これを$\overline{\mathbf{F}}$と書くことにする：

$$\overline{\mathbf{F}} = \sum_{m,n} \overline{f}_{mn} x^m y^n \tag{41}$$

$\overline{\mathbf{F}}$をブール束上の画像多項式の反転と呼ぶことにする．

次に，Ψの逆写像をΨ^{-1}とし，$\mathbf{F}_M, \mathbf{G}_M \in M$に対して，$\mathbf{F}_P$，$\mathbf{G}_P \in P_B$をそれぞれ，

$$\mathbf{F}_P = \Psi^{-1}(\mathbf{F}_M) \tag{42}$$

$$\mathbf{G}_P = \Psi^{-1}(\mathbf{G}_M) \tag{43}$$

とする．このとき，数理形態学の演算とブール束上の画像多項式の演算との間に次の定理が成り立つ．

[定理2] 集合演算とブール束上の画像多項式の演算との間に次の関係が成立する．

$$\Psi^{-1}(\mathbf{F}_M \cup \mathbf{G}_M) = \mathbf{F}_P + \mathbf{G}_P \tag{44}$$

$$\Psi^{-1}(\mathbf{F}_M \setminus \mathbf{G}_M) = \mathbf{F}_P - \mathbf{G}_P \tag{45}$$

$$\Psi^{-1}(\mathbf{F}_M \oplus \mathbf{G}_M) = \mathbf{F}_P * \mathbf{G}_P \tag{46}$$

$$\Psi^{-1}(\check{\mathbf{F}}_M) = \check{\mathbf{F}}_P \tag{47}$$

$$\Psi^{-1}(\overline{\mathbf{F}}_M) = \overline{\mathbf{F}}_P \tag{48}$$

2.1で数理形態学の演算の基本となる5つの集合演算を述べた．しかし，式(9)から分かるように平行移動をミンコウスキー和によって行うことができる：

$$\mathbf{A}_x = \mathbf{A} \oplus \{x\} \tag{49}$$

したがって，平行移動とミンコウスキー和を入れかえて5つの基本集合演算とすることができる．よって，定理2より，計算機上にブール束上の画像多項式の

- 乗算モジュール
- 加算モジュール
- 減算モジュール
- 回転モジュール
- 反転モジュール

を用意すれば，数理形態学の全ての演算をブール束上の画像多項式の演算で実行することができる．すなわち，数理形態学で構成した2値画像の形状処理アルゴリズムを，ブール束の代数演算で効率よく計算機で実行できるのである．そして，定理2は数理形態学の演算で記述されたアルゴリズムをブール束上の画像多項式の演算に変換する具体的な方法を与えているのである．

4 数理形態学と最大値・最小値フィルタ

従来から，最大値・最小値フィルタによる2値画像の形状処理が盛んに研究されている［1］．そこで，本章では数理形態学と最大値・最小値フィルタとの関係についてまとめる．

点 (m,n) の近傍 $\mathcal{N}((m,n))$ を

$$\mathcal{N}((m,n)) = \{(p,q) : (p-m, q-n) \in \mathbf{S}\} \tag{50}$$

によって定義する．ただし，$\mathbf{S}$ は $(0,0)$ を含む画像平面上の任意の有限部分集合である．入力画像 $\mathbf{F} = \{f_{mn}\} \in B$に対して，画像 $\mathbf{G} = \{g_{mn}\} \in B$を得る次の二つの演算

$$g_{mn} = \max_{(p,q)}\{f_{pq} : (p,q) \in \mathcal{N}((m,n))\} \tag{51}$$

$$g_{mn} = \min_{(p,q)}\{f_{pq} : (p,q) \in \mathcal{N}((m,n))\} \tag{52}$$

をそれぞれ（近傍 $\mathcal{N}((m,n))$ を用いた）最大値フィルタ，最小値フィルタという［1］．以降，最大値フィルタをμ，最小値フィルタをϕと書くことにする．

BからMへの全単写をΩとすると，次の定理が成り立つ．

[定理3] $\mathbf{S}$ が $\mathcal{N}((0,0))$ の数理形態学表現とする．そして，$\mathbf{F}_B \in B$，$\mathbf{F}_M = \Omega(\mathbf{F}_B) \in M$とすると

$$\Omega(\mu(\mathbf{F}_B)) = D(\mathbf{F}_M, \mathbf{S}) \tag{53}$$

$$\Omega(\phi(\mathbf{F}_B)) = E(\mathbf{F}_M, \mathbf{S}) \tag{54}$$

が成り立つ．

（証明）

$$\begin{aligned}
&(\mu(\mathbf{F}_B))_{mn} \\
&= \max_{(p,q)}\{f_{pq} : (p,q) \in \mathcal{N}((m,n))\} \\
&= \max_{(p,q)}\{f_{pq} : (p-m, q-n) \in \mathcal{N}((0,0))\} \\
&= \max_{(p,q)}\{f_{pq} : (m-p, n-q) \in \overline{\mathbf{S}}\} \\
&= \max_{(p,q)}\{f_{m-p\,n-q} : (p,q) \in \overline{\mathbf{S}}\}
\end{aligned} \tag{55}$$

となる．したがって，

$$(\mu(\mathbf{F}_B))_{mn} = \begin{cases} 1 & \exists (p,q) \in \overline{\mathbf{S}}, f_{m-p\,n-q} = 1 \\ 0 & \text{その他} \end{cases} \tag{56}$$

を得る．すなわち，

$$\begin{aligned}
&\Omega(\mu(\mathbf{F}_B)) \\
&= \{(m,n) : \exists (p,q) \in \overline{\mathbf{S}}, (m-p, n-q) \in \mathbf{F}_M\} \\
&= \mathbf{F}_M \oplus \overline{\mathbf{S}} \\
&= D(\mathbf{F}_M, \mathbf{S})
\end{aligned} \tag{57}$$

となる．式(54)も同様にして示すことができる．

（証明終わり）

定理3によって，2値画像に対する最大値・最小値フィルタの作用は数理形態学の拡張・侵食による作用と等価であることが分かる．

次に，最大値・最小値フィルタとブール束上の画像多項式の演算との関係について調べる．ΘをΨ^{-1}とΩの合成写像とする：

$$\Theta = \Psi^{-1} \circ \Omega \tag{58}$$

このとき，定理2，定理3より次の定理が成り立つ．

[定理4] $\mathbf{S}$ を $\mathcal{N}((0,0))$ の画像多項式表現とする．そして，$\mathbf{F}_B \in B$，$\mathbf{F}_P = \Theta(\mathbf{F}_B) \in P_B$であるとする．このとき，

$$\Theta(\mu(\mathbf{F}_B)) = \mathbf{F}_P * \check{\mathbf{S}} \tag{59}$$

$$\Theta(\phi(\mathbf{F}_B)) = \overline{\mathbf{F}_P * \check{\mathbf{S}}} \tag{60}$$

が成り立つ．

最大値・最小値フィルタは画像に対する近傍$\mathcal{N}$による位置不変な局所処理である．一方，ブール束上の画像多項式の乗算はブール束の元のコーシー積によって求めることができる．コーシー積は画像全体に対する大域処理である．したがって，定理4は，位置不変な局所処理が大域処理によって記述でき，実行できることを示している．さらに，処理の具体的な算法も与えているのである．

5 形状処理アルゴリズムの構成

5.1 距離変換

距離変換とは，図形成分の各点に対して背景からその点への距離を対応させる処理である．数理形態学によって距離変換アルゴリズムが次のように構成される．

集合 $\mathbf{A}$ を図1－(a)に示すような図形とする．そして，$\mathbf{A}$ を原点に中心がある半径1の円で侵食した結果 $\mathrm{E}(\mathbf{A},\mathbf{B})$ は図1－(b)に示すような図形となる．これは，$\mathbf{A}$ の中で背景からの距離が1以上の部分である．したがって，図1－(c)に示す $\mathbf{A}$ から $\mathrm{E}(\mathbf{A},\mathbf{B})$ を引いた部分は，$\mathbf{A}$ の中で距離変換の値が1である部分を表している．この操作を反復適用すると，$\mathbf{A}$ の中で距離変換の値が n である部分 $\mathbf{D}(n)$ を

$$\mathbf{D}(n) = E^{n-1}(\mathbf{A},\mathbf{B}) \setminus E^{n}(\mathbf{A},\mathbf{B}) \tag{61}$$

によって求めることができる．ただし $E^n(\mathbf{A},\mathbf{B})$ は $\mathbf{A}$ を $\mathbf{B}$ で n 回侵食した集合である．このアルゴリズムをブール束上の画像多項式による表現に変換すると

$$\begin{aligned} \mathbf{D}_P(n) &= \overline{\overline{\mathbf{A}}_P * \check{\mathbf{B}}_P^{n-1}} - \overline{\overline{\mathbf{A}}_P * \check{\mathbf{B}}_P^{n}} \\ \mathbf{B}_P^n &= \mathbf{B}_P^{n-1} * \mathbf{B}_P \end{aligned} \tag{62}$$

となる．これを計算機実験した結果を図2に示す．

5.2 細線化

細線化とは，図形の境界にある1－画素の消去可能性を判断しながら一層ずつ消去して細めていき，線幅1の図形を得る処理である．この処理はマスク処理で行うことができる．したがって，数理形態学の適合変換によって細線化を行うことができる．

細線化のアルゴリズムは適合変換の反復適用によって

$$\mathbf{X}_{n+1} = \mathbf{X}_n \setminus HM(\mathbf{X}_n,\mathbf{M}) \tag{63}$$

となる．ここで，構成要素 $\mathbf{M}$ は消去可能性を判定するマスクであり，$\mathbf{X}_n$ は入力画像 $\mathbf{X}$ に対する反復 n 回目の出力画像である．そして，式(63)の停止条件は $HM(\mathbf{X}_n,\mathbf{M}) = \emptyset$ である．

文献［2］では，4連結細線化のマスクとして図3に示す $\mathbf{H}(k)$，$\mathbf{D}(k)\,(k=1,2,3,4)$：

$$\mathbf{H}(k) = (\mathbf{H}_1(k),\mathbf{H}_2(k)) \tag{64}$$

$$\mathbf{D}(k) = (\mathbf{D}_1(k),\mathbf{D}_2(k)) \tag{65}$$

が用いられている．そして，細線化は $\mathbf{H}(k)$，$\mathbf{D}(k)$ を，$\mathbf{H}(1)$，$\mathbf{D}(1)$，$\mathbf{H}(2)$，$\mathbf{D}(2)$ の順に k を1から4まで変化させて，式(63)の $\mathbf{M}$ に代入することによって行われる．すなわち，式(63)は，1回の反復に8つのサブサイクルを持つことになる．

$\mathbf{H}(k)$，$\mathbf{D}(k)$ は横井の連結数［5］が1であり，かつ芯線を短縮しないような画素を図形の境界から抽出するマスクである．そして，$\mathbf{H}(k)$ は水平垂直方向から，$\mathbf{D}(k)$ は対角方向からこのような画素を抽出する．抽出された画素は式(63)より，元の画像から削除されるのである．式(63)の1回の反復によって入力画像の境界の一層が削られることになる．式(63)をブール束上の画像多項式による表現に変換すると，

$$\mathbf{X}_{Pn+1} = \mathbf{X}_{Pn} - (\overline{\overline{\mathbf{X}}_{Pn} * \check{\mathbf{M}}_P} - \mathbf{X}_{Pn} * \check{\mathbf{M}}_P) \tag{66}$$

となる．これを計算機実験した結果を図4に示す．

図4の結果では，芯線に余分な枝が生じている．そこで，余分な枝の発生を抑えるために，$\mathbf{H}(k)$，$\mathbf{D}(k)$ を適用する前に，図5に示すマスク $\mathbf{B}(k)\,(k=1,2,3,4)$：

$$\mathbf{B}(k) = (\mathbf{B}_1(k),\mathbf{B}_2(k)) \tag{67}$$

を式(63)適用することにする．$\mathbf{B}(k)$ によって余分な枝を発生させる画素を事前に除去するのである．結局，式(63)は，1回の反復に12のサブサイクルを持つことになる．これをブール束上の画像多項式に変換して計算機実験すると図6に示す結果を得る．結果より芯線の性質が改善されていることが分かる．

8連結細線化の場合は $\mathbf{D}(k)$ を図7に示すようにすればよい．

5.3 スケルトン

入力画像 $\mathbf{F}$ のスケルトンを求めるアルゴリズムは最大値・最小値フィルタを用いて次のように与えられる［1］．

$$\mathbf{SK}(n) = \phi^{n-1}(\mathbf{F}) - \mu(\phi^n(\mathbf{F})) \tag{68}$$

ただし，$\mathbf{SK}(n)$ は $\mathbf{F}$ の中で距離値がnのスケルトンである．そして，ϕ^n は最小値フィルタをn回適用することを表している．これを定理3により数理形態学の演算に変換すると

$$\mathbf{SK}(n) = E^{n-1}(\mathbf{F},\mathbf{S}) \setminus D(E^n(\mathbf{F},\mathbf{S}),\check{\mathbf{S}}) \tag{69}$$

となる．ただし，$\mathbf{S}$ は $\mathcal{N}((0,0))$ を最小値フィルタを適用するときの近傍として，$\mathbf{S} = \Omega(\mathcal{N}((0,0)))$ である．これをブール束上の画像多項式による表現に変換すると

$$\begin{aligned} \mathbf{SK}_P(n) &= \overline{\overline{\mathbf{F}}_P * \check{\mathbf{S}}_P^{n-1}} - \overline{\overline{\mathbf{F}}_P * \check{\mathbf{S}}_P^{n}} * \mathbf{S}_P \\ \mathbf{S}^n &= \mathbf{S}^{n-1} * \mathbf{S} \end{aligned} \tag{70}$$

となる．これを計算機実験した結果を図8に示す．

6 アルゴリズムの複雑さの基本単位

3.4で示したように数理形態学の演算で記述されたアルゴリズムは，5つのブール束上の画像多項式の演算モジュールを組み合わせて実行することができる．これらの中で最も複雑なのは多項式の乗算モジュールである．したがって，アルゴリズムの複雑さを左右するのは，画像多項式に変換されたアルゴリズムの中のアルゴリズムが停止するまでの多項式の乗算の回数であると考えられる．そこで，多項式の乗算を，アルゴリズムの複雑さを評価する基本単位とすることにする．多項式の乗算は，係数のコーシー積を求めることと同じである．したがって，ブール束のコーシー積

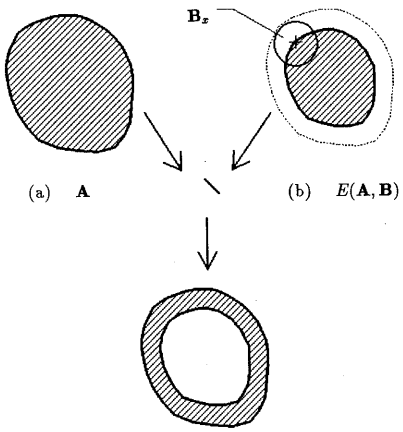
によってアルゴリズムの複雑さを定量的に評価できることになる．

多項式の乗算の回数によって，5 で示した距離変換，細線化，スケルトンのアルゴリズムの複雑さを評価すると表1のようになる．ここで，画面の大きさは $2N \times 2N$ であり，複雑さはアルゴリズムが停止するまでに必要な多項式の乗算の最大回数で評価する．

この結果から距離変換よりもスケルトン，スケルトンよりも細線化の方が複雑な処理であるといえる．

	乗算の数
距離変換	N
スケルトン	2N
細線化	24N

表1　複雑さの評価

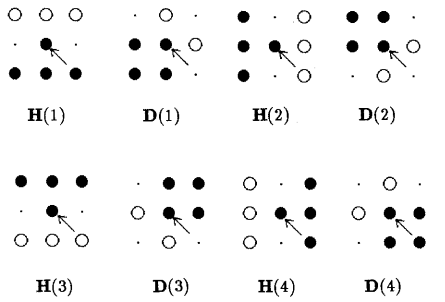


(c)　$D(1) = \mathbf{A} \setminus E(\mathbf{A},\mathbf{B})$

図1　距離変換アルゴリズムの構成

```
....................
....................
....................
.....1111111........
....12222222111.....
....123333332221....
.....12222233321....
......1111123321....
...........12321....
............1221....
............112211...
..............1221...
.............12321...
.............12321...
.............12321...
............12321....
..........112321.....
..........12221......
...........111.......
....................
....................
```

図2　距離変換の実験結果（4連結）



• points of structuring element which must belong to $\mathbf{X}$
o points of structuring element which must belong to $\overline{\mathbf{X}}$
· points of structuring element with no condition
↖ location of the origin associated with the structuring element

図3　4連結細線化のマスク $\mathbf{H}(k), \mathbf{D}(k)$

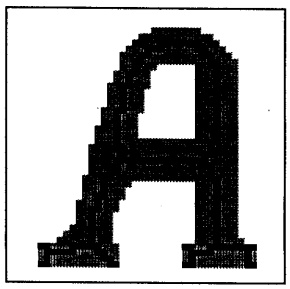


図4　細線化の実験結果（4連結）

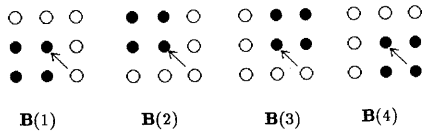


図5　マスク $\mathbf{B}(k)$

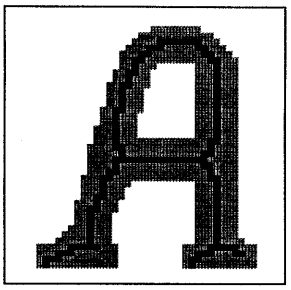


図6　$\mathbf{B}(k)$ を用いた細線化の実験結果

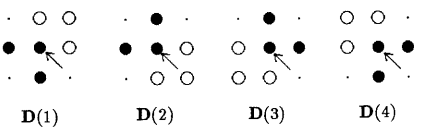


図7　8連結の場合のマスク $\mathbf{D}(k)$

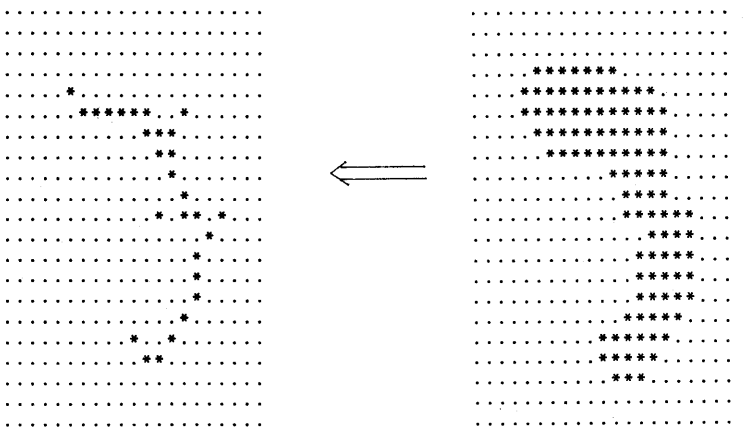


図8　スケルトン（4連結）

原画像

7　むすび

本論文では，はじめに数理形態学の演算が2値画像の形状処理アルゴリズムを構成するのに有効であることを見た．次いで，ブール束上の画像多項式を定義した．そして，数理形態学の演算をブール束上の画像多項式の演算に変換して，計算機上でモジュール化して効率よく実行できることを示した．

さらに，最大値・最小値フィルタと数理形態学，ブール束上の画像多項式の演算との関係を明らかにした．ここで，最大値・最小値フィルタの位置不変な局所処理が，画素値をブール束の元と考えることによって，画素値のコーシー積によって記述できることを示した．すなわち，形状処理アルゴリズムの多くは，位置不変なマスク処理，つまり局所処理で記述されるが，代数系をうまく選ぶことによって，位置不変な大域処理で表すことができることが分かる．

最後に，ブール束上の画像多項式の乗算をアルゴリズムの複雑さを，定量的に評価する基本単位とすることを提案した．

数理形態学および画像多項式による画像処理の記述法をそのまま記述できるプログラミング言語を開発することが今後の課題である．

本研究の一部は，文部省からの科学研究費補助金によるものである．

参考文献

[1] 鳥脇純一郎，“画像理解のためのディジタル画像処理[I],[II]”，昭晃堂，(1988)

[2] J.Serra,“IMAGE ANALYSIS AND MATHEMATICAL MORPHOLOGY”, ACADEMIC PRESS,(1982)

[3] 尾崎，樹下，“ディジタル代数学”，共立出版，(1966)

[4] 井宮，中村，“量子化された離散画像の記述法とその画像多項式への応用”，信学技報，PRU87-62,(1987)

[5] 横井，鳥脇，福村，“標本化された二値画像のトポロジカルな性質について”，信学論，Vol.56-D,No.11,pp662-669,(1973-11)