

病理画像における癌細胞特徴量の  
抽出アルゴリズムの検討

同志社大学工学部インテリジェント情報工学科  
卒業論文

2011年2月

学籍番号 070095

知的システムデザイン研究室

山口 浩明

# 目次

<b>1</b>	<b>画像処理</b>	<b>1</b>
1.1	デジタル画像 . . . . .	1
1.2	2値画像処理 . . . . .	1
1.3	雑音除去 . . . . .	7
1.4	輪郭抽出 . . . . .	9
<b>2</b>	<b>遺伝的プログラミング</b>	<b>12</b>
2.1	概要 . . . . .	12
2.2	木構造 . . . . .	12
2.3	アルゴリズム . . . . .	13
2.4	選択手法 . . . . .	15
2.5	ブロート . . . . .	16
<b>3</b>	<b>進化的画像処理</b>	<b>17</b>
3.1	概要 . . . . .	17
3.2	処理手順 . . . . .	17
3.3	GPによる画像処理フィルタの自動構築 . . . . .	18
3.4	評価関数 . . . . .	18
<b>4</b>	<b>進化的画像処理手法を用いた画像処理フィルタ自動構築の検証</b>	<b>20</b>
4.1	概要 . . . . .	20
4.2	手書きメモを除去する画像処理フィルタ構築の検証 . . . . .	20

# 1 画像処理

画像処理とは、処理する前の元の画像である原画像に対して何らかの変換処理を行い、目的に合った画像を得る処理であると定義されている<sup>1)</sup>。コンピュータ上で行う、デジタル画像を用いた画像処理はデジタル画像とも呼ばれ、コンピュータの発達と共に画像処理技術が考案されてきた。本章では、デジタル画像を用いた代表的な画像処理技術について述べる。

## 1.1 デジタル画像

デジタル画像とは、画面を Fig. 1.1 に示すような画素 (pixel) と呼ばれる小さな離散的な点に分割し、各画素における濃淡の値も階調値と呼ばれる整数値で表現したものである<sup>2)</sup>。コンピュータ上では、デジタル画像における各画素の階調値を基に画像処理が行われる。

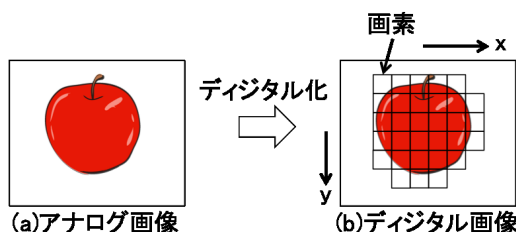


Fig. 1.1 デジタル画像

デジタル画像の静止画は、「2値画像」、「濃淡 (モノクロ) 画像」、「カラー画像」の3つに分類される。カラー画像には、原色として、赤 (R)、緑 (G)、青 (B) を利用し色を表現する「RGBのカラー画像表現」とシアン (C)、マゼンタ (M)、黄 (Y) に黒 (K) を加えた「CMYKのカラー画像表現」がある。濃淡画像とは、白黒の濃淡で表現される画像である。画像の階調値が  $n$  ビットで表現されている場合、その画像は  $2^n$  通りの階調値をもつ。実用では、階調値が8ビットで表現されているものがほとんどであり、階調値は0から255 (0が最も暗く、255が最も明るい) の値をとる。2値画像とは、濃淡がなく、画素の値が0, 1の2つの値しかとらない白と黒の色のみで表現された画像である。本章では、説明する画像処理は、すべて濃淡画像を対象として扱う。

## 1.2 2値画像処理

2値画像処理は、コンピュータ画像処理の中でも特異な位置にある。濃淡画像、カラー画像を2値画像へ変換することで、領域の数や面積、形状などを解析・認識が可能となり、濃淡画像処理に比べて理論や手法の体系化が進んでいるといえる。

### 1.2.1 画像の2値化

画像の2値化は、画像の特徴を解析するために、画像から対象物を切り出し、対象領域と背景領域を分離するために2値化処理を用いることが多い。通常、8ビット濃淡画像の各画素は、0～255の階調値を持っており、この階調値に対して基準とする閾値を設定し、2値化が行われる。画素  $(x, y)$  の明るさ  $f(x, y)$  に対する閾値  $t$  での閾値処理を式 1.1, Fig. 1.2 に示す。また、ある濃淡画像に対し

て閾値処理を行った結果を Fig. 1.3 に示す.

$$fitness = \begin{cases} 0 \text{ (黒色)} & (f(x, y)) > T \\ 1 \text{ (白色)} & (f(x, y)) \leq T \end{cases} \quad (1.1)$$

7	20	4	2値化	→	0	0	0
0	15	46			0	0	0
55	80	99			1	1	1

Fig. 1.2 閾値処理の例 (閾値 T=50)

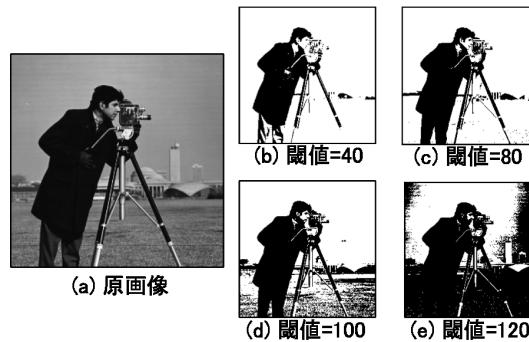


Fig. 1.3 2値化

Fig. 1.3(a) の原画像閾値処理によって抽出対象領域 (人物, カメラ) と背景領域を切り離すことを目標とした. Fig. 1.3 から明らかなように, Fig. 1.3(d) のような大きすぎる閾値は, 余分に背景領域も抽出してしまい, 一方, Fig. 1.3(a) のような小さすぎる閾値は, 抽出対象領域 (人物, カメラ) が欠けてしまう. この結果から 2 値化の処理において, 閾値の設定が重要な要素であることがわかる. しかし, 適切な閾値の選択は, 2 値化を行う前に画像の持つ情報から事前に調べる必要がある, また対象画像によって依存してしまう. そこで, 統計学を用いた閾値を自動的に選定する手法が 1979 年に大津<sup>3)</sup>によって提案されている. この手法は, 画像の濃度値ヒストグラムを 2 つのクラスに分類し, それぞれのクラスの分離度 (式 1.2) が最大となる閾値を自動で決定する.

$$\eta(T) = \frac{\sigma_B^2(T)}{\sigma_W^2(T)} \quad (1.2)$$

ここで,  $\sigma_W^2$  および  $\sigma_B^2$  はそれぞれクラス内分散, クラス間分散と呼ばれる. クラス内分散とは, 2 つのクラスの分散の和, クラス間分散とは, 2 つのクラスの平均値の分散のことである. 与えられた画像が, Fig. 1.4 のようなパラメータ, 統計量を持っているとするとクラス内分散, クラス間分散はそれぞれ式 1.3, 1.4 で与えられる.

$$\sigma_W^2 = \frac{\omega_1 \sigma_1^2 + \omega_2 \sigma_2^2}{\omega_1 + \omega_2} \quad (1.3)$$

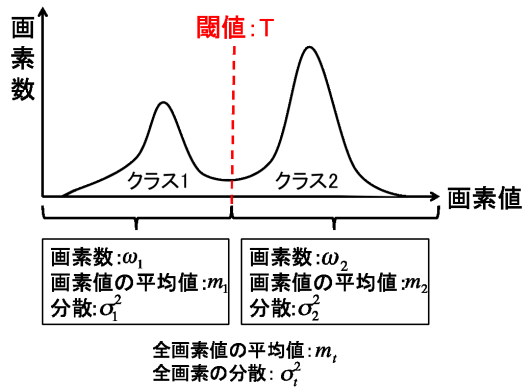


Fig. 1.4 判別分析によるクラス分け

$$\sigma_B^2 = \frac{\omega_1(m_1 - m_T)^2 + \omega_2(m_2 - m_T)^2}{\omega_1 + \omega_2} = \frac{\omega_1\omega_2(m_1 - m_2)^2}{(\omega_1 + \omega_2)^2} \quad (1.4)$$

さらに、全画素の分散は  $\sigma_T^2$  なので、式 1.5 の関係が成り立つ。

$$\sigma_T^2 = \sigma_W^2 + \sigma_B^2 \quad (1.5)$$

この関係より、式 1.2 は式 1.6 となる。

$$\eta(T) = \frac{\sigma_B^2(T)}{\sigma_W^2(T)} = \frac{\sigma_B^2}{\sigma_T^2 - \sigma_B^2} \quad (1.6)$$

全分散  $\sigma_T^2$  は閾値とは無関係な定数であるので、分離度を最大にするには  $\sigma_B^2$  を最大にすればよい。すなわち、 $\sigma_B^2$  が最大になるときの閾値  $T$  を求めればよい。

Fig. 1.3(a) の画像において判別分析法による 2 値化処理を行った結果を Fig. 1.5 に示す。



Fig. 1.5 判別分析法による 2 値化

Fig. 1.5 の結果では、背景領域を消し、人物とカメラを抽出している良好な結果となった。このように判別分析法を用いることで、2 値化の閾値を自動で決定することができる。

### 1.2.2 ラベリング

ラベリング処理とは、同じ連結成分に属するすべての画素に同じラベル（番号）を割り当て、異なった連結成分には異なったラベルを割り当てる処理のことである（Fig. 1.8）。2 値画像では、画像

処理上で、連結性を利用したラベリング（連結領域抽出）処理が可能となる。2値画像において、ある注目画素  $(x, y)$  と隣接する画素が同じ階調値である場合、それらはつながった（連続している）領域であると判断できる。ここで、注目画素  $(x, y)$  に隣接する画素のことを近傍と呼ぶ。この連結性には、「4近傍」、「8近傍」の2つの考え方がある。「4近傍」では、Fig. 1.6 に示すように注目画素の上下左右の位置の画素において、注目画素と同じ階調値をもつものが存在する場合に、それらはつながっていると考える。「8近傍」では、注目画素の上下左右に斜め方向の4つの画素を加えた考え方である（Fig. 1.7）。

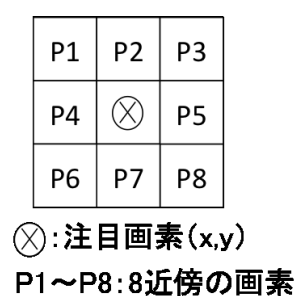
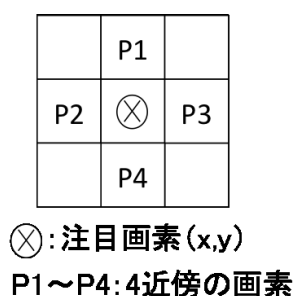


Fig. 1.6 画素  $(x, y)$  の4近傍

Fig. 1.7 画素  $(x, y)$  の8近傍

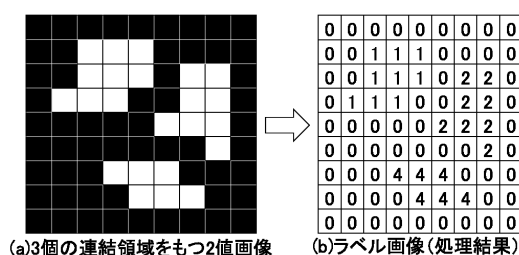


Fig. 1.8 ラベリング処理の例

画像内に複数の連結領域が存在する場合、それらの幾何学（形状）的特徴をコンピュータ上で個々に解析したい時に、この処理が利用される。以下に8近傍におけるラベリング処理の手順を示す。なお、ラベリングを行う前に、処理対象の2値画像と同サイズの「ラベル画像（初期値はすべて0）」を別に用意し、ラベル番号を1から順番に数え上げていくものし、背景を黒、解析を行う領域を白とする。

**Step1.** 原画像（2値画像）に対してラスタ走査の順で注目画素を移動させる。未処理（ラベル付けがされておらず、白色）の画素にたどり着いた場合、Step2の処理を行う。ラスタ走査とは、画像の左上の画素から水平に右方向へ移動し、その走査を順に垂直方向にずらしていくことで、全画素を探索する走査方法である（Fig. 1.9）。

**Step2.** 「ラベル画像」の注目点の近傍において、すでにラベル付けされた画素があるか調べる。調べる近傍画素は、8近傍なら、左上、上、右上、左、4近傍なら上、左の画素である。ラベル付け画素のラベルの種類の数により、行われる処理が変わる。

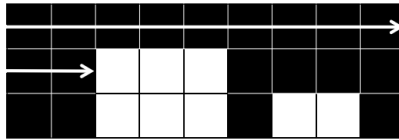


Fig. 1.9 ラスタ走査

(1) ラベル付けされた画素がない場合

新しいラベル（現在のラベル番号に 1 を加える）を注目画素に割り当てる（Fig. 1.10）.  
ラベル番号の初期値は 0 とする.

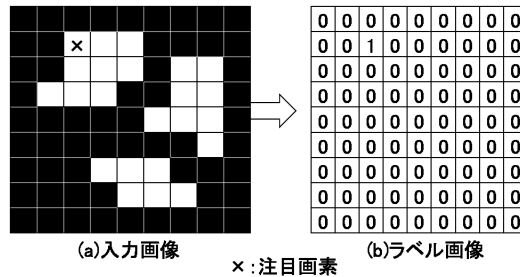


Fig. 1.10 ラベル付け 1

(2) ラベルが 1 種類の場合

見つかったラベルを注目画素に割り当てる（Fig. 1.11）.

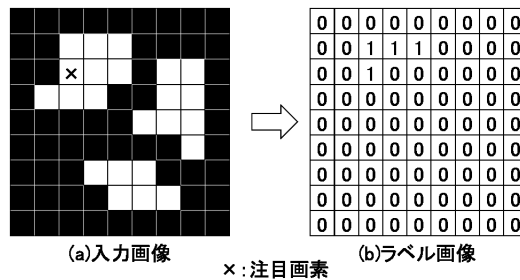


Fig. 1.11 ラベル付け 2

(3) ラベルが 2 種類以上の場合

最も小さいラベルを注目画素に割り当てる（Fig. 1.12）. このとき、用いなかったラベル番号（Fig. 1.12 の例では 3）は、どのラベル同士が同一とみなせるか記憶するためのテーブルに書き込む.

Step1, Step2 の処理を全画素に適用させた後, Step3 の処理を行う.

**Step3.** ラベルテーブルから, Step2 の (3) で, 同一となったラベル同士を整理し, その対応しているラベル（最も小さいラベル番号）に置き換える（Fig. 1.13）,

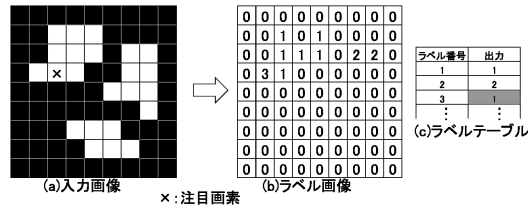


Fig. 1.12 ラベル付け 3

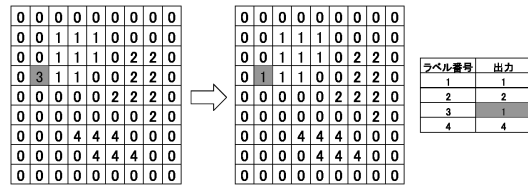


Fig. 1.13 ラベル付け 4

### 1.2.3 膨張・収縮

画像処理において、膨張とは、与えられた連結成分の境界画素をすべて取り除いて1画素分大きくする処理である。一方、収縮とは、逆に1画素分小さくする処理である (Fig. 1.14, Fig. 1.15)。膨張、収縮の処理による出力画像  $g(x, y)$  の階調値は式 1.7, 1.8 によって決定される。

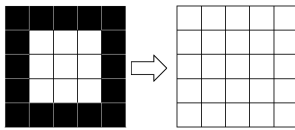


Fig. 1.14 膨張

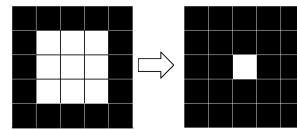


Fig. 1.15 収縮

$$\text{膨張} : g(x, y) = \begin{cases} 1 : \text{入力画像の注目画素 } (x, y) \text{ あるいはその近傍の階調値に } 1 \text{ が含まれる場合} \\ 0 : \text{入力画像の注目画素 } (x, y) \text{ とその近傍の階調値が全て } 0 \text{ の場合} \end{cases} \quad (1.7)$$

$$\text{収縮} : g(x, y) = \begin{cases} 0 : \text{入力画像の注目画素 } (x, y) \text{ あるいはその近傍の階調値に } 0 \text{ が含まれる場合} \\ 1 : \text{入力画像の注目画素 } (x, y) \text{ とその近傍の階調値が全て } 1 \text{ の場合} \end{cases} \quad (1.8)$$

膨張、収縮の処理を組み合わせることにより、2値画像の雑音除去が行うことができる。収縮、膨張の順に画像を処理することで、雑音によって連結してしまった領域を切り離すことができる (Fig. 1.16)。また膨張、収縮の順に処理を行うと、領域の切れ込みやごま塩雑音を除去することができる (Fig. 1.17)。これらの処理のことを「オープニング」、「クロージング」という。



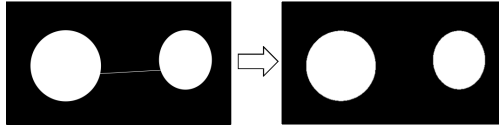


Fig. 1.16 オープニング

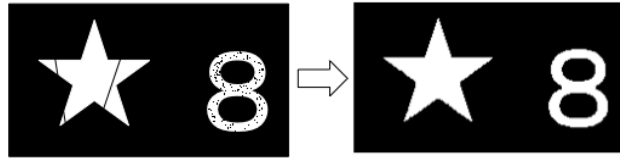


Fig. 1.17 クロージング

### 1.3 雑音除去

画像の生成や伝送の過程では、数々の要因で物理的に雑音が画像に含まれることがある。Fig. 1.18の画像には白と黒の点がランダムに含まれており、これらの点はごま塩雑音と呼ばれる。本節では、ごま塩雑音を除去する代表的な平滑化手法とされる、移動平均法、メディアンフィルタについて述べる。



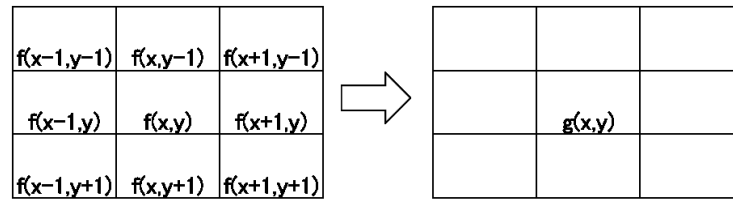
Fig. 1.18 ごま塩雑音を含む画像

#### 1.3.1 移動平均法

移動平均法は、入力画像の注目画素  $f(x, y)$  とその8近傍画素の階調値の平均値を計算し、それを出力画像の  $(x, y)$  点の階調値として決定する手法 (Fig. 1.19) である。

移動平均法を用いると、近傍画素との濃度のばらつきが抑えられるため、雑音との濃度差を低下させることができる。Fig. 1.18に移動平均法の処理を行った結果をFig. 1.20に示す。

移動平均法の処理によって、画素値が平均化されるため、Fig. 1.20(a)では雑音が抑えられた結果となった。しかし、処理回数を増やしたFig. 1.20(b)では、画像がぼけてしまっている。移動平均法では、平均化の処理によって近傍画素との濃度差を低下させるため、雑音以外、例えば人物と背景との境目などの輪郭付近の濃度差も抑えられ、画像がぼけるといった結果になる。



$$g(x,y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i,y+j)$$

Fig. 1.19 移動平均法

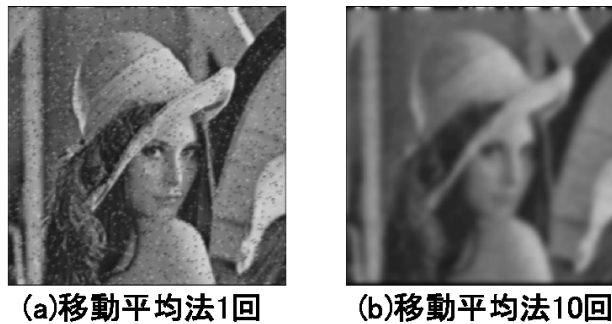


Fig. 1.20 移動平均法 (処理結果)

### 1.3.2 メディアンフィルタ

移動平均法では、注目画素とその8近傍における濃度の平均値を出力画像の濃度とした。メディアンフィルタでは、注目画素とその8近傍の中央値（メディアン）を出力画像の濃度として与える。メディアンフィルタによる処理の例を Fig. 1.21 に示す。

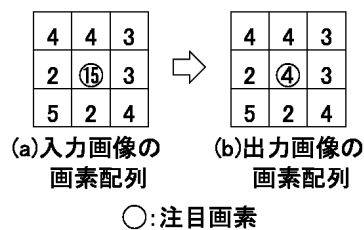


Fig. 1.21 メディアンフィルタ

Fig. 1.21(a) の画素値を昇順に並べると Fig. 1.22 となる。

要素数が9であるので、この要素内の中央値は左から5番目にある「4」となり、これが新しい濃度となる。周りと比べて極端に濃度の違うものは、大きさの順に並べたとき、左端か右端に集まり、中央値として選択されないため雑音を除去することができる。メディアンフィルタによる処理を行った結果を Fig. 1.23 に示す。

移動平均法では、処理回数が増加するにつれ、輪郭付近の濃度差が抑えられるため画像がぼけてし

Fig. 1.22 メディアン (中央値)



Fig. 1.23 メディアンフィルタ (処理結果)

まった。一方、メディアンフィルタでは、処理回数が増加しても輪郭が保存され、画像がぼけない特徴がある。

## 1.4 輪郭抽出

濃淡画像において輪郭とは、物体の外縁をあらわす線、または画像を特徴づける線要素であり、一般的に濃度の急激な変化と定義される<sup>4)</sup>。画像中の濃度の変化、すなわち輪郭を検出する最も基本的な方法として、関数の変化分を取り出す微分演算が輪郭抽出に利用できる。デジタル画像ではデータが離散的に並んでいるため、本当の意味での微分演算はできない。このため隣接画素同士の差をとる演算で微分を近似し、これを差分と呼ぶ。本節では、差分を利用した輪郭抽出手法である、ソーベルフィルタ、ラプラシアンフィルタについて述べる。

### 1.4.1 ソーベルフィルタ

輪郭抽出を行う画像処理は、微分演算を利用することで行われ、1次の微分演算は差分で近似できるので、 $x$ 方向の微分、 $y$ 方向の微分は式1.9、1.10のように考えることができる。

$$\frac{\partial}{\partial x} f(x, y) \approx f(x, y) - f(x - 1, y) \quad (1.9)$$

$$\frac{\partial}{\partial y} f(x, y) \approx f(x, y) - f(x, y - 1) \quad (1.10)$$

上記の式により、 $x$ 方向、 $y$ 方向の微分値を計算することができる。それぞれの微分値から、輪郭の強さが式1.11により求められる。

$$\sqrt{f_x^2 + f_y^2} \quad (\text{または、}|f_x| + |f_y|) \quad (1.11)$$

輪郭の強さとは、輪郭を表す濃度差の大きさを指しており、この値が大きいほど、輪郭が強調されやすい。上記の処理を $3 \times 3$ のマスクで表すと、Fig. 1.24のようになる。

0	0	0
-1	1	0
0	0	0

(a)水平輪郭マスク

0	-1	0
0	1	0
0	0	0

(b)垂直輪郭マスク

Fig. 1.24 通常の差分

マスクで示された値を，注目画素とその近傍画素に掛けることで，特定の部分を抽出することができる．Fig. 1.24(a)のマスクでは，注目画素とそれと隣接する左の画素値（ $x$ 方向），Fig. 1.24(b)では，注目画素とそれと隣接する上の画素値（ $y$ 方向）との差分を求めることができる．

Fig. 1.24のマスクでは， $x$ 方向， $y$ 方向の輪郭は抽出できるが，斜め方向の輪郭を抽出することは難しい．そのため，輪郭抽出の手法として，ソーベルフィルタ（Fig. 1.25）が使われることが多い．また，通常の差分，ソーベルフィルタを用いた結果をFig. 1.26に示す．

-1	0	1
-2	0	2
-1	0	2

(a)ソーベル水平輪郭マスク

-1	-2	-1
0	0	0
1	2	2

(b)ソーベル垂直輪郭マスク

Fig. 1.25 ソーベルフィルタ



Fig. 1.26 輪郭抽出結果

Fig. 1.26(a)の原画像において，通常の差分よりもソーベルフィルタを用いた方がよりはっきりと輪郭を抽出することができた．

#### 1.4.2 ラプラシアンフィルタ

ラプラシアンフィルタは2階微分値の差分近似を計算し（式1.12），輪郭部分を抽出する．

$$\begin{aligned}
 \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y) & \\
 \approx f(x-1, y) - 2f(x, y) + f(x+1, y) + f(x, y-1) - 2f(x, y) + f(x, y+1) & \\
 = f(x-1) + f(x+1, y) + f(x, y-1) + f(x, y+1) - 4f(x, y) & \quad (1.12)
 \end{aligned}$$

これをマスクで表すと，Fig. 1.27のようになる．

0	1	0
1	-4	1
0	1	0

Fig. 1.27 ラプラシアンフィルタ

ラプラシアンフィルタは、注目画素とその4近傍画素に対して平均化を行っているため、雑音が混じった画像や自然画像の輪郭抽出に向いていると考えられる。ラプラシアンフィルタを用いた処理結果を Fig. 1.28 に示す。



Fig. 1.28 輪郭抽出結果（ラプラシアン）

## 2 遺伝的プログラミング

### 2.1 概要

遺伝的プログラミング (Genetic Programming:GP) は, 1992 年に Stanford 大学の John Koza らにより提案された進化的計算手法である<sup>5)</sup>. GP は, 生物の進化を模倣した遺伝的アルゴリズム (Genetic Algorithms:GA) の遺伝子型 (Fig. 2.1) を構造的な表現 (木構造, グラフ構造) (Fig. 2.2) で扱えるように拡張したものである. 遺伝子型で表現することにより知識表現の獲得や関数・プログラムの自動生成などの階層的な表現能力を要する問題を直接的に扱うことができる.

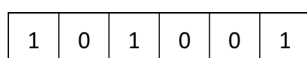


Fig. 2.1 GA の遺伝子例

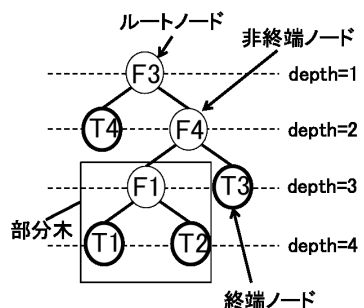


Fig. 2.2 GP の遺伝子例

GP では, Fig. 2.2 のような個体が集まった母集団に対して遺伝的操作 (選択, 交叉, 突然変異など) を行い, 世代交代を繰り返すことで, 問題に適した良好な個体 (解) を生成する.

### 2.2 木構造

#### 2.2.1 構成

Fig. 2.2 の木構造は, ノード (節点, 頂点) と, ノード間を結ぶエッジ (辺, 枝) で構成されている. ノードは, 木構造の底辺にある終端ノード (T1,T2,...) とそれ以外の非終端ノード (F1,F2,...) に区別される. さらに, 木構造の頂点にあるノード (F3) を根 (ルート) ノードと呼ぶ. また, ノードは, 子ノードをもつ非終端ノード (F1,F2,...) と, 子ノードを持たない終端ノード (T1,T2,...) に区別される.

非終端ノードである F4 に着目すると, F4 とつながっており, 1つ下の階層にある F1, T3 を F4 の子ノードと呼ぶ. 1つ上の階層にある F3 は F4 の親ノードと呼ぶ.

木構造の深さは, ルートノードから最底辺に至るまでのエッジの数に 1 を加えた数で表される. Fig. 2.2 は深さ 4 の木構造である. また, Fig. 2.2 の四角で囲まれた部分に着目すると, それ自身も木構造を為しており, これを部分木という.

#### 2.2.2 走査

木構造のノードを一定の規則に従い, 全てのノードを調査する処理を走査という. 走査法には前順走査 (pre-order), 間順走査 (in-order), 後順走査 (post-order) があり, それぞれの走査処理を行った例を Fig. 2.3 に示す.

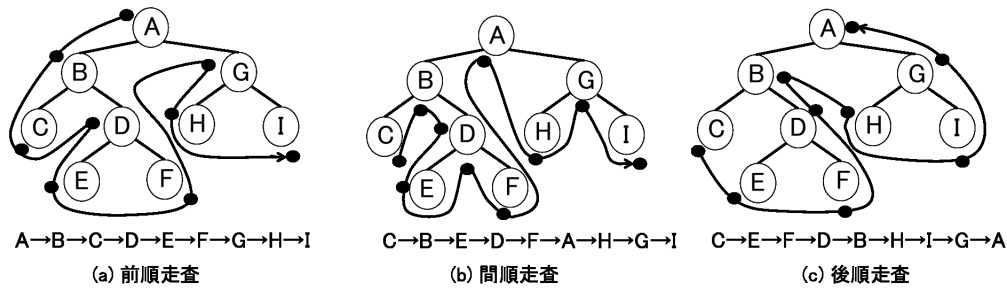


Fig. 2.3 走査

前順走査では、節、左部分木、右部分木の順に再帰的に走査を行い、全てのノードを調査する。間順走査では、左部分木、節、右部分木、後順走査では、左部分木、右部分木、節の順に再帰的に走査を行う。

### 2.2.3 基本要素

GP では、以下の5つの基本要素を設計することで、さまざまな応用例題への適用が可能になる。

- (1) 非終端記号  
非終端ノードで使用する記号、関数、演算子など。
- (2) 終端記号  
終端ノードで使用する記号、実値など。
- (3) 評価関数
- (4) パラメータ  
母集団サイズ、交叉、突然変異の起こる確率など。
- (5) 終了条件  
最大世代数、目標評価値など。

## 2.3 アルゴリズム

GP のアルゴリズムを Fig. 2.4 に示す。

- (1) 初期集団の木構造の生成 (Individual)  
あらかじめ設定した木の深さを超えない複数の個体をランダムに生成し、初期個体群 (母集団) とする (Fig. 2.5)。各個体は終端記号と非終端記号を組み合わせた木構造で表す。
- (2) 評価 (Evaluation)  
対象問題の評価関数によって、各個体の評価値を求める。対象とする問題の解として適する個体ほど高い評価値になるように問題に応じた評価式を作成する必要がある。

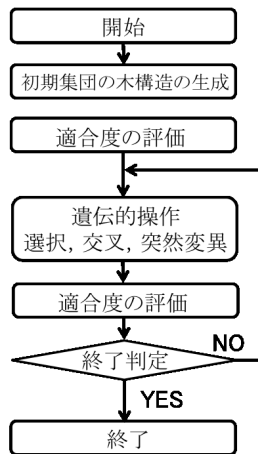


Fig. 2.4 GP のアルゴリズム

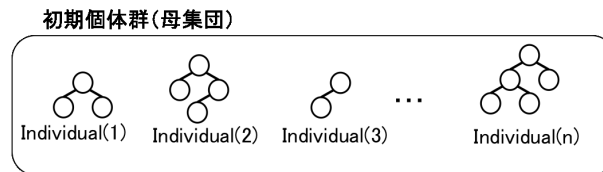


Fig. 2.5 初期集団

(3) 選択 (Selection)

各個体の評価値を判断基準にして、次世代に残す個体を母集団の数だけ選択する。選択方法には、トーナメント選択、ランキング選択、ルーレット選択、エリート選択などがあり、2.4節で述べる。

(4) 交叉 (Crossover)

個体の構造に変化を加える中心的な操作である。母集団中の個体に対して、交叉率に応じて交叉を行う個体のペア (親) をランダムに選出する。親 1 と親 2 の交叉点をランダムに選び、それぞれ交叉点に応じた部分木同士で交叉させることで、新しい子個体を生成する (Fig. 2.6)。

(5) 突然変異 (Mutation)

母集団中の個体に対して、突然変異率に応じて突然変異を行う個体をランダムに選出する。そしてランダムに突然変異点を選び、その点に応じたランダムに生成した部分木と突然変異木を入れ替える。(Fig. 2.7) ここで、部分木の作成には、(1) の初期個体の生成と同様の処理を行う。ただしルートノードは突然変異の対象外である。

(6) 終了条件 終了条件に達するまで、(2) ~ (5) の遺伝的操作を繰り返す。終了条件には、最大世代数や、目標評価値などを与える。また、終了した時点の個体群の中で最も評価値の良い個体を最適解とする。



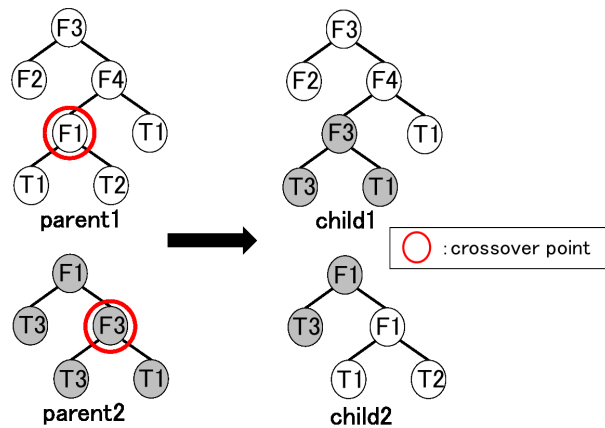


Fig. 2.6 GPの交叉

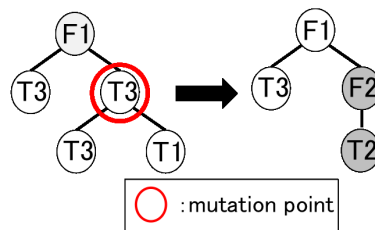


Fig. 2.7 GPの突然変異

## 2.4 選択手法

選択とは適合度を基に次世代の親となる個体を選ぶ操作である。本節で、ルーレット選択、ランキング選択、トーナメント選択、エリート選択について述べる。

### 2.4.1 ルーレット選択

ルーレット選択とは、Fig. 2.8 に示す評価値を基に作成したルーレット盤を用いてランダムに選択する手法である。

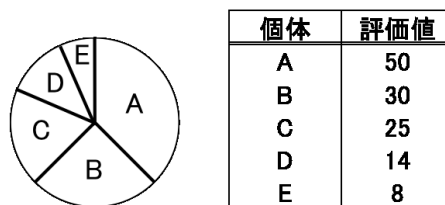


Fig. 2.8 ルーレット選択

各個体の評価値に比例した割合で個体を選択する。選択確率の高い個体は、複数回の交叉に参加するため、その個体は集団中に広がる。個体数を  $n$ 、ある個体  $f$  の評価値を  $f_i$  とすると、個体  $i$  にお

けるルーレットでの割合は適合度の総合計に対する各個体の割合で決まる。これを式 2.1 に示す。

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j} \quad (2.1)$$

### 2.4.2 ランキング選択

ランキング選択とは、評価値によって各個体に順番を付け、各順番に対して予め決められた確率で子を残す手法である。各個体は、評価値ごとにランキングされているため、選択確率は評価値には依らずに、ランキングに依存する。ランキング選択の例を Table 2.1 に示す。

Table 2.1 ランキング選択

ランク	個体	評価値	選択確率
1	A	50	0.4
2	B	30	0.3
3	C	25	0.15
4	D	14	0.8
5	E	8	0.7

### 2.4.3 トーナメント選択

トーナメント選択とは全ての個体群の中からランダムに定数分の個体を選び出し、その中で評価値が一番高い個体を次世代に残す手法である。この作業を母集団の大きさだけ繰り返し、必要な個体を確保する。定数分の値が小さいほど評価値の低い個体が次世代に残る可能性が高くなる。

### 2.4.4 エリート選択

エリート選択とは、評価値が高い  $n$  個の個体をそのまま次世代に残す手法である。この手法は、その時点で最も良い解が選択や突然変異で壊されないという利点がある。エリート選択は、評価値の低い個体を次世代に残さないため、局所的最適解に陥りやすく、他の選択手法を併用して使用する。

## 2.5 ブロート

GP のプログラム（遺伝子型）は探索過程でその長さが増大する傾向がある<sup>6)</sup>。この現象をブロートと呼ぶ。ブロートのおもな理由は、プログラムのイントロンや冗長部分が成長することである。イントロンとは、全体のプログラムの動作に無関係なコードのことである。無関係であるというのは、このコード部分が取り除かれても得られる結果に影響しないということを意味している。

このブロートが発生することにより、プログラム実行の遅延や人にとっての解のわかりにくさなどの問題が生じる。ブロートの発生を抑制するために、木構造の深さの制限や、構成されるノード数に比例して個体の評価値にペナルティを与えることが用いられている。

### 3 進化的画像処理

本章では、画像処理において進化的計算法を用いることで、画像処理を最適化問題として扱う進化的画像処理について述べる。

#### 3.1 概要

一般的に画像処理は、Fig. 3.1 に示すように、処理する前の元の画像である原画像に対して何らかの変換処理を行い、目的に合った画像を得る変換処理であると定義されている<sup>1)</sup>。現状の画像処理は、変換処理を人が考案し、処理プログラムを計算機に与えることによって行う。しかし、その処理は特定の目的のためだけに有効であり、処理対象が変わると役に立たないことが多いため、著しく汎用性に乏しいものが多い。そのため、処理目的が変化する度にアルゴリズムを考案する必要がある。そこで、画像処理において最適化を取入れ、画像処理は最適化問題であると捉えることで処理方法の考案を自動化できる。これが進化的画像処理の考え方である。

進化的画像処理では、与えられた任意の画像処理を組み合わせ、最適化問題として定式化し、その最適化問題を進化的計算法により解く。進化的計算法には、遺伝的アルゴリズム (Genetic Algorithm:GA) や遺伝的プログラミング (Genetic Programming:GP) などがあるが、本章では、GP を用いた進化的画像処理について述べる。

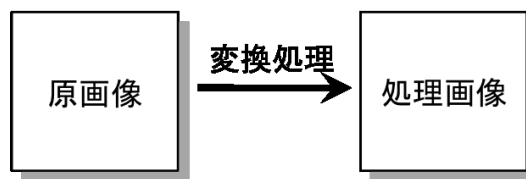


Fig. 3.1 画像処理

#### 3.2 処理手順

進化的画像処理の処理手順を以下に示す。

(1) 定式化：

画像処理を、GA や GP によって解きやすいタイプの最適化問題に変換する。ここで、画像処理を行う者はあらかじめ GA や GP について十分理解していることが望ましい。

(2) 求解：

定式化された最適化問題を、元が画像処理の問題であることを特に意識せずに GA や GP を用いて解く。

(3) 解の解釈：

得られた解は単なる数値列や木構造であるため、それを画像処理の結果として解釈する必要がある。

### 3.3 GPによる画像処理フィルタの自動構築

進化的画像処理の考え方をを用いることで、画像処理フィルタを自動構築することができる。本研究では、GPを用いて画像処理プロセスを Fig. 3.2 に示すような木構造フィルタと近似して、画像処理フィルタの最適化を行う。

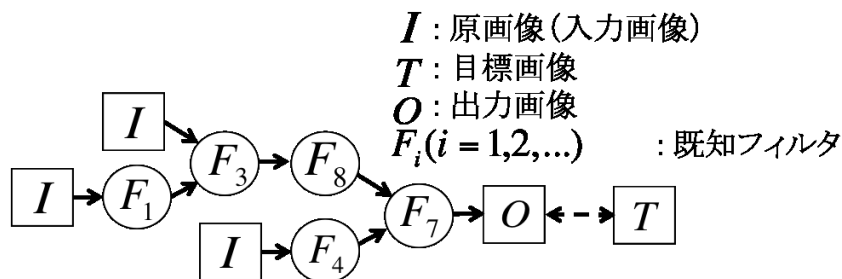


Fig. 3.2 木構造フィルタによる画像処理の自動化の原理

画像処理フィルタを構築する際の対象となる画像処理の事例が、原画像（入力画像） $I$ と、それに対する理想的な処理画像である目標画像 $T$ の組みとして与えられる。このとき、入力画像 $I$ から目標画像 $T$ に至る画像処理を同図のような木構造上フィルタによって近似する。この木の末端（葉）から原画像が入力される。節は、入力・出力がともに画像であり、入力が1つ以上、出力が1つのフィルタである。葉から入力された画像が複数系統で順次処理され、最終的に木の根（Fig. 3.2では $F_7$ ）から1つの画像 $O$ が出力される。木の出力画像 $O$ が、入力画像 $I$ に対する目標画像 $T$ と比較され、 $O$ が $T$ に近いほど、その木構造状フィルタが優れていると評価する。この評価を行うための評価関数については、3.4節で述べる。与えられた原画像（入力画像）に適用すると、目標画像とほぼ同じ出力画像を出力することができる木構造状フィルタを決定することが、ここでの最適化の対象となる。

この最適化問題にGPを用いると、終端記号を原画像（入力画像）とし、非終端記号をあらかじめ用意した既知フィルタ（これまでに提案された画像に対して何らかの効果を加えるフィルタ）とする木を、GPにおける1つの個体として進化をさせていく。既知フィルタは、入力系統数が1および2のものであり、1章で述べたような代表的なフィルタを用いる。本研究で使用する1入力の既知フィルタを Table3.1、2入力の既知フィルタを Table3.2 に示す。

### 3.4 評価関数

木構造フィルタの評価方法として、木の出力画像 $O(x, y)$ と目標画像 $T(x, y)$ の各階調値の差分が小さいものほど優れていると評価する。これは個体の適応度 *fitness*（0.0以上1.0以下）を次式で求めることを示している。

$$fitness = \frac{1}{K} \sum_{i=1}^K \left\{ 1 - \frac{\sum_{x=1}^{W_x} \sum_{y=1}^{W_y} |O(x, y) - T(x, y)|}{W_x W_y V_{max}} \right\} \quad (3.1)$$

ただし画像の大きさを $W_x \times W_y$ 、最大階調値を $V_{max}$ （通常は255）としている。ここで、 $K$ は事例（原画像+目標画像）の数を示す。複数の事例を用いて学習を行う場合は、それぞれの事例に対

Table 3.1 1入力1出力の非終端記号（既知フィルタ）

番号	フィルタ	機能（新しい階調値の決定方法）
F1	Sobel	1次微分値
F2	Laplace	2次微分値
F3	Min	近傍画素の階調の最小値
F4	Max	近傍画素の階調の最大値
F5	Median	近傍画素の階調の中央値
F6	Light	閾値画素より大きい画素を $V_{max}$
F7	Dark	閾値画素より小さい画素を 0
F8	LargeArea	閾値面積より大きい領域内の画素を 0, それ以外を $V_{max}$
F9	SmallArea	閾値面積より小さい領域内の画素を 0, それ以外を $V_{max}$
F10	Binary	閾値を基準に 0 か $V_{max}$
F11	Inverse	$V_{max} -$ 階調値

Table 3.2 2入力1出力の非終端記号（既知フィルタ）

番号	フィルタ	機能（新しい階調値の決定方法）
F12	BoundedSum	2つの階調値の和
F13	BoundedProd	2つの階調値の和 $- V_{max}$
F14	LogicalSum	2つの階調値の論理和
F15	LogicalProd	2つの階調値の論理積
F16	AlgebraicSum	2つの階調値の和 $-$ 積 $/ V_{max}$
F17	AlgebraicProd	2つの階調値の積 $/ V_{max}$
F18	Sub	2つの階調値の絶対値の差

する評価値の単純平均として評価を与える。複数の事例を用いることで、画像処理手法の最適化において汎用性を持たせ、未知の画像に対しても同様の画像処理が行えるようにする。

## 4 進化的画像処理手法を用いた画像処理フィルタ自動構築の検証

### 4.1 概要

本検証では、3章で述べた進化的画像処理の手法を用いて、原画像から目標画像へ近似する画像処理フィルタの自動構築を行う。本検証では、文書画像中に手書きメモが書かれた原画像から手書きメモを除去することを対象問題とした画像処理フィルタ構築の検証を行う。この検証により、進化的画像処理手法を用い、与えられた事例に沿う画像処理フィルタの自動構築の確認を行う。

本検証において、GPで生成する木構造の非終端ノードは既知フィルタ、終端ノードは原画像に対してグレースケール化を施したものである。GPで使用するパラメータをTable4.1、実装した既知フィルタは3章で述べたTable3.1、Table3.2に示したものである。既知フィルタは、F1～F11が1入力1出力、F12～F18が2入力1出力のフィルタとなっている。「F6」、「F7」、「F10」のフィルタの閾値は判別分析法、「F8」、「F9」のフィルタの閾値は平均面積から決定される。なお、GPで生成される木構造の深さは、「シミュレーテッドアニーリングを用いた自動プログラミング<sup>7)</sup>」を参考にし、最大で17としている。これを超えた木においては、深さ17にあるノードを強制的に終端ノードに置き換える。また構築された画像処理フィルタの性能は、3.1式で示した評価式に対して、個体のノード数×ペナルティを引いた値を評価値とする。

Table 4.1 Parameter of GP

Parameter	Value
Generation	400
Population	100
Way of Selection	Tournament
Crossover Rate	0.9
Mutation Rate	0.1
Creation Tree Depth	4
Elite Size	1
Max Depth	17
Penalty	0.001

### 4.2 手書きメモを除去する画像処理フィルタ構築の検証

本節では、進化的画像処理手法によって目標に近似するための画像処理フィルタが自動構築が可能か検証を行う。用いる対象問題は、Fig. 4.1に示す文書画像中に手書きメモが書かれた原画像から手書きメモを除去することとする。Fig. 4.1(a)の原画像は、文書画像中に筆者がシャープペンシルを用いて記入を行ったものである。

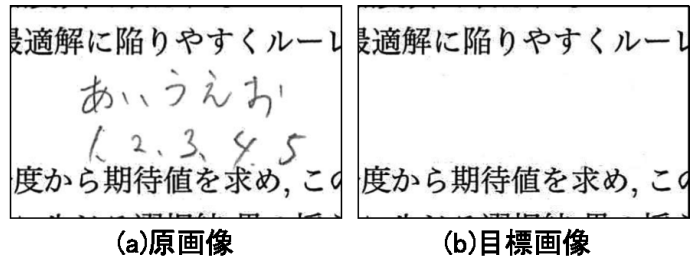


Fig. 4.1 学習に用いる単一の事例（手書きメモ除去）

Fig. 4.1 に対して GP によって自動構築された画像処理フィルタを Fig. 4.2 に示す. Fig. 4.2 の画像処理フィルタを Fig. 4.1(a) に適用した結果を Fig. 4.3 に示す. Fig. 4.2 の各非終端ノード番号は Table3.1, Table3.2 と対応させており, I は入力画像, O は出力画像となっている.

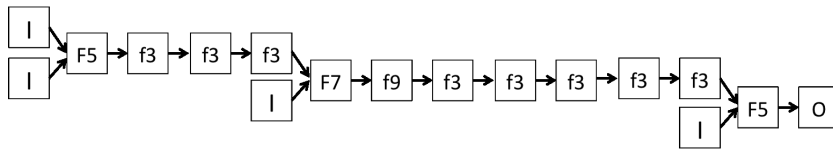


Fig. 4.2 自動構築された画像処理フィルタ（手書きメモ除去）

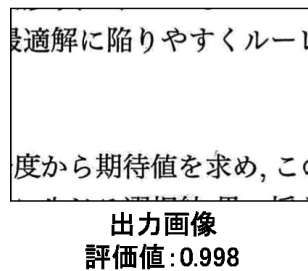


Fig. 4.3 出力結果（手書きメモ除去）

Fig. 4.3 の出力画像は, Fig. 4.1(b) に示した目標画像とほとんど同じであることがわかる. これは Fig. 4.1(a) から (b) に至る手作業による画像処理が, 既知フィルタを組み合わせた木構造状フィルタによって自動化されたことを示している.

このときの GP における世代交代に伴う個体集団中の最大評価値の推移を Fig. 4.4 に示す. 同図から, ランダムに生成された初期個体集団では低い評価値が, 世代交代を重ねるにつれて高くなり, 最終的には実用解とみなせる高い値に収束している様子がわかる. このときの世代数は 282 世代であった. また, このときの世代交代におけるいくつかの世代での, 最も優れた個体による出力画像を Fig. 4.5 に示す. 世代交代とともに, 目標画像により近い出力画像が得られるようになることがわかる.

GP により自動構築された Fig. 4.2 の画像処理フィルタの汎用性を確かめるため, 学習に用いなかった未知画像に適用した結果を Fig. 4.6 に示す. Fig. 4.6 に示すように, 学習に用いていない未知の

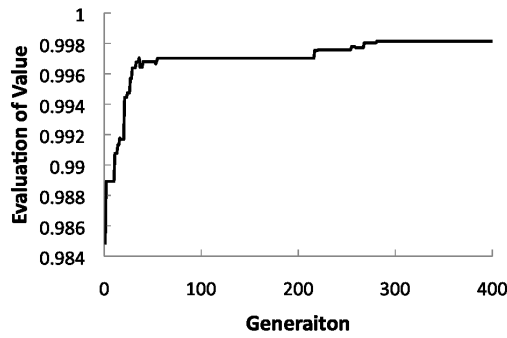


Fig. 4.4 世代交代に伴う個体集団中の最大評価値の推移

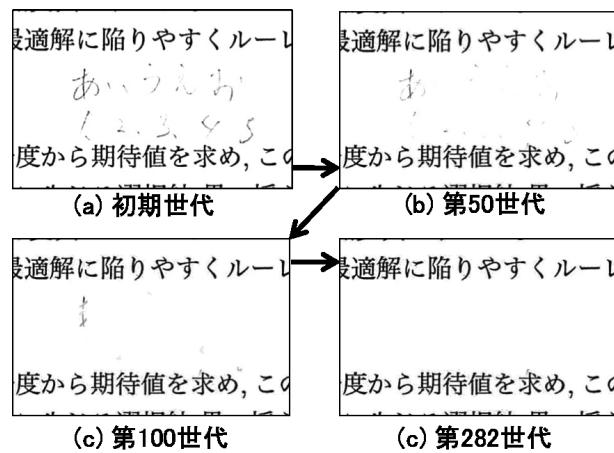


Fig. 4.5 各世代での最大評価値をもつ個体による出力画像

画像においても良好な出力画像が得られる結果となった。

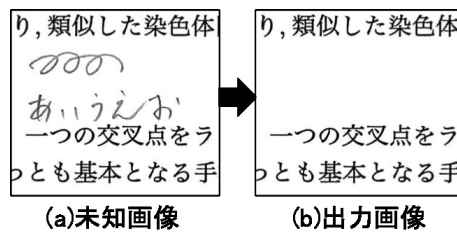


Fig. 4.6 未知画像に適用した出力画像 (手書きメモ除去)

このように、進化的画像処理手法を用いることで、原画像から目標画像へ近似する処理を行う画像処理フィルタを自動構築することができた。



## 謝 辞

本研究を遂行するにあたり、多大なるご指導そしてご協力を頂きました、同志社大学生命科学部の廣安知之教授に心より感謝申し上げます。また、様々な指摘、助言をして下さいました、そして、同じ医用画像班であり、一年間コーチとして研究に関する指導等をして頂いた、知的システムデザイン研究室所属の藤田宗佑氏に深く感謝しております。同じく医用画像班の野田徹氏、南谷祥之氏に御礼申し上げます。また、本論文を校正して下さいました、横田山都氏、藤田宗佑氏に感謝致します。

最後に、知的システムデザイン研究室のみなさま、医療情報システム研究室のみなさまには多くの議論や助言をして頂きました。また、皆様のおかげで、1年間すばらしい研究生活を送ることができました。この場を借りて厚く御礼申し上げます。

## 参考文献

- 1) 長尾智晴. 進化的画像処理. 昭晃堂, 2002.
- 2) 内村圭一. 上瀧剛. 実践画像処理入門. 培風館, 2007.
- 3) 大津展之. 判別および最小 2 乗基準に基づく自動しきい値選定法. No. 4, 1980.
- 4) 田村秀行. コンピュータ画像処理. オーム社, 2002.
- 5) J.Koza. *Genetic programming, on the programming of computers by means of natural selection*,. MIT Press, 1992.
- 6) 伊庭斉志. 遺伝的プログラミング入門. 東京大学出版会, 2001.
- 7) 藤田佳久, 三木光範, 橋本雅文, 廣安知之. シミュレーテッドアニーリングを用いた自動プログラミング. No. SIG 15, 2007.