

病理画像を用いた画像処理手法の自動構築における GP と SAP の比較

藤田 宗佑

1 はじめに

近年、癌などの増加に伴い、日本での病理医が極端に不足しているのが現状である。そのため、癌など病理診断するための支援システムに対するニーズが今後増加していくと考えられ、病理診断を自動化し、医師の病理診断を支援する病理画像診断支援システムが日本電気株式会社 (以下 NEC) で考案されている^{1, 2)}。しかし、病理画像診断支援システムに必要とされる病理画像を解析するための画像処理手法を人手によって考案することは困難である。そこで本研究では、病理画像を用いた画像処理に着目し、病理部位抽出フィルタの自動構築を行うことを考える。

本研究では、病理画像から癌胞巣を抽出することを対象とし、癌胞巣抽出フィルタの最適化を行う。また、この自動プログラミング手法に遺伝的プログラミング (Genetic Programming: GP)^{3, 4)} および、SAP (Simulated Annealing Programming: SAP)⁵⁾ を用いて、癌胞巣抽出フィルタの構築における GP と SAP の性能比較を行う。

2 病理診断の現状

近年、癌の検診が増大している一方で、日本では病理診断を行っている医師が極端に不足している。一般的に、病理診断は病理医によって主観的に行われているため、その人の経験、体調などに左右されてしまう可能性がある⁶⁾。そのため、病理画像を対象に、癌組織中の癌構成要素の抽出を行い、癌の病理診断のための新たな指標を検討するための研究が数多く行われている。本研究では、病理画像を用いた画像処理に着目し、画像処理手法の自動構築を行うことを考える。本研究では、Figure 1 に示すような組織学的研究や、病理診断などで最も一般的に用いられている染色法である HE 染色により染色した低倍率画像 (10 × 画像) を扱うことにする。

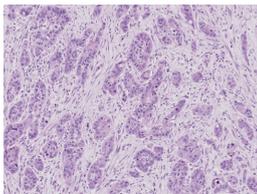


Fig.1 病理画像の一例

3 自動プログラミング手法による画像処理フィルタの構築

画像処理フィルタの自動構築を行うために、長尾らは進化的画像処理を提案している⁷⁾。一般的に画像処理は既知の単純な画像フィルタ (基本フィルタ) の組み合わせ

せとして表現可能であり、進化的画像処理は、自動プログラミング手法を用いて最適な基本フィルタの組み合わせを導く手法である。本研究では、画像処理フィルタを Figure 2 に示すような木構造状フィルタとして近似し、GP、および SAP の自動プログラミング手法を用いて画像処理フィルタの最適化を行なう。

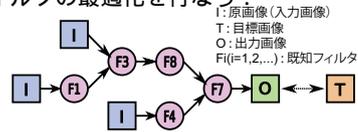


Fig.2 木構造状フィルタによる画像処理の自動化の原理

出力画像は、まず原画像を構築した画像生成フィルタの終端ノードに入力し、木構造の各ノードに格納された基本フィルタ (既存の画像処理) を終端ノードから順番に実行することで作成する。

各個体の適合度は、式 (1) に示すように、画像処理フィルタからの出力画像 $O(x, y)$ と目標画像 $T(x, y)$ の差分によって求め、1.0 を最適解とする最大値問題とする。

$$fitness = \frac{1}{k} \sum_{k=1}^K \left\{ 1 - \frac{\sum_{x=1}^{W_x} \sum_{y=1}^{W_y} w_i(x, y) |O_i(x, y) - T_i(x, y)|}{\sum_{x=1}^{W_x} \sum_{y=1}^{W_y} w_i(x, y) \cdot V_{max}} \right\} \quad (1)$$

ここで、 K は学習用画像セット数、 $w(i, j)$ は重み画像の値、 V_{max} は最大階調値を示す。

4 自動プログラミング手法

本稿では、自動プログラミングの手法として、GP と SAP を用いる。以下に、それぞれの自動プログラミング手法について述べる。

● 遺伝的プログラミング (GP)

GP は、生物の進化を模倣した遺伝的アルゴリズム (Genetic Algorithm: GA)³⁾ を構造的な表現 (木構造, グラフ構造) が扱えるように拡張した自動プログラミング手法である。GP では、選択, 交叉, 突然変異といった遺伝的オペレータを繰り返し行うことで、問題に適した木構造を生成する。

● シミュレーテッドアニーリングプログラミング (SAP)

SAP は、金属の焼き鈍しを模倣したシミュレーテッドアニーリング (Simulated Annealing: SA)⁸⁾ を、木構造が扱えるように拡張した自動プログラミング手法である。SAP では、生成処理, 受理判定, 状態遷移, 冷却を繰り返し行うことで、問題に適した木構造を生成する。

5 数値実験

本章では、GP と SAP の性能比較を行ない、それぞれの手法の探索における特徴について検証する。

5.1 実験概要

本実験では、病理画像から癌胞巣を抽出する画像処理フィルタを GP, SAP を用いて構築し、それぞれの自動プログラミング手法の特徴について検証する。また本実験では、単一画像による学習を行う。GP および, SAP のパラメータを Table 1, Table 2 に示す。なお, GP においてのみ木構造のフィルタ数の制限を 50 とする。Figure 3 に、本実験で用いた学習用画像セットを示す。今回用いた重み画像は抽出対象領域が 1 に対し、非抽出対象を 0.4 の比率にするように設定した。

Table1 GP のパラメータ

Parameter	Value
Generations	200
Populations	100
Selection Method	Tournament
Crossover Rate	0.9
Mutation Rate	0.1
Max number of filters	50

Table2 SAP のパラメータ

Parameter	Value
Number of Evaluations	20000
Max Temperature	1.44
Min Temperature	0.00015



Fig.3 単一画像の学習に用いた学習用画像セット

5.2 学習性能

学習性能とは、目的を達するためのプログラムを生成する能力である。それぞれの自動プログラミング手法で得られた評価値の結果を Figure 4 に示す。次に、それぞれの手法で構築した癌胞巣抽出フィルタに対して、Figure 3 の原画像を適用して得られた出力結果を Figure 5 に示し、この出力画像から作成した癌胞巣を抽出した画像を Figure 6 に示す。この癌胞巣を抽出した画像は、原画像と出力画像との論理和をとることで作成したものである。

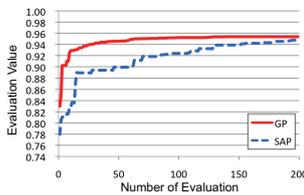


Fig.4 解探索における GP と SAP の比較

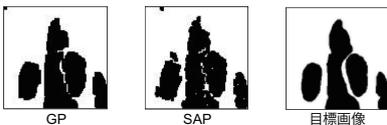


Fig.5 出力画像

Figure 4 より、学習性能では、GP の方が少ない探索回数で収束することが可能だとわかる。また Figure 5, Figure 6 からわずかながら GP の方が抽出対象の輪郭がなめらかに抽出されており、より目標画像に近いことが視覚的にも確認できる。



Fig.6 出力画像から作成した癌胞巣を抽出した画像

5.3 プログラムサイズ

プログラムサイズとは、個体の全ノード数であり、プログラムサイズが大きいほど、全体の処理に時間がかかる。Figure 7 に探索における木構造のフィルタ数の推移を示し、Figure 8 に探索を行なった際の計算時間の推移を示す。

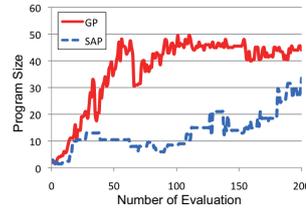


Fig.7 木構造のフィルタ数における GP と SAP の比較

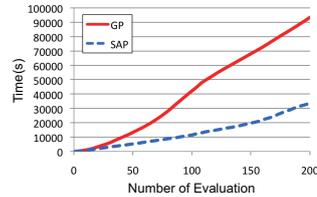


Fig.8 計算時間における GP と SAP の比較

SAP では、探索終盤わずかながらプログラムサイズが増加したが、GP よりはプログラムサイズを抑えた探索が可能であり、GP よりも少ない計算時間で探索が可能であることがわかった。

5.4 イントロン除去

GP, SAP の両手法で構築したフィルタに対して、イントロン除去を行なうことで、イントロンの有無について調査を行った。今回は以下の手順で検出・除去することとする。

1. 注目する個体に対し、ランダムに第 1 ノード N_i , N_i 以下の部分木からランダムに第 2 ノード N_j を選択する。
2. N_i, N_j の各ノードの類似度を下記の式で求める。

$$similarity = \frac{1}{k} \sum_{k=1}^K \left\{ 1 - \frac{\sum_{x=1}^{W_x} \sum_{y=1}^{W_y} |O_{N_i}(x,y) - O_{N_j}(x,y)|}{V_{max} \cdot W_k \cdot H_k} \right\}$$

3. 各ノードが極度に類似する場合 (類似度が 0.999 以上の場合), N_j の親ノードから N_i のノードまでの部分木を削除する。

上記の処理を GP, SAP によって得られた解に対して 50 試行行ない、最もプログラムサイズを抑えることができたフィルタを採用する。そして、この一連の処理を 3 回繰り返し、最大 3 つのイントロンを除去する。

Table 3 に GP, SAP それぞれのイントロン除去数の結果を示す。

Table3 インترون除去結果

	イントロン除去前	イントロン除去後	除去数
GP	45	28	17
SAP	31	29	2

Table 3 より、イントロン除去を行なうことで、GP のプログラムサイズを減少させることができ、SAP と同程度のプログラムサイズのフィルタが構築できることがわかった。

5.5 ロバスト性の検証

本節では、イントロン除去を行なって得られた GP、SAP で構築したフィルタに対して未知画像を適用し、ロバスト性について検討する。本実験では未知画像として、(a) 同じ検体で別画像の一部、(b) 濃淡の濃い画像、(c) 濃淡の薄い画像の 3 枚を用いた。GP、SAP で得られた結果をそれぞれ Figure 9、Figure 10 に示す。

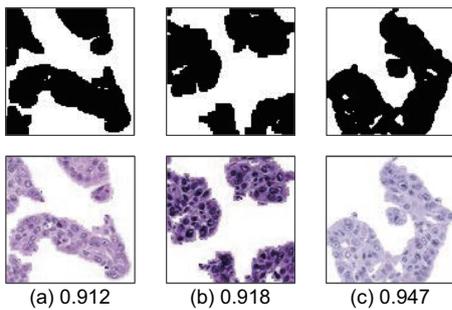


Fig.9 GP の出力画像

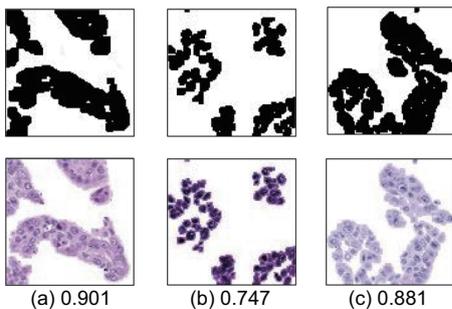


Fig.10 SAP の出力画像

Figure 9 より、GP は全体的に評価値も高く抽出対象領域が抽出されていることが視覚的にもわかる。一方 Figure 10 より、SAP は (b)、(c) の画像において、抽出対象の一部しか抽出できなかった。このような差が生まれたのは、2 入力フィルタで用いた論理和等の和演算が原因であると考えられる。Table 4 に各フィルタに含まれる和演算処理と積演算処理の平均値を示す。

Table4 GP、SAP の両フィルタに含まれる和演算処理の数

	和演算処理の数	積演算処理の数
GP	2.0	5.5
SAP	7.0	7.2

Table 4 より、SAP での解更新は突然変異のみでランダムに木構造が構築されるため、和演算と積演算の数が高

くば等しくなっていることがわかる。和演算が多くなると階調値が大きくなり、白色領域が増加するため、SAP では未知の画像に適用した際に上手く抽出できていない部分が生じたと考えられる。一方、GP では積演算に比べ和演算の方が少ないことから、未知の画像においても抽出対象が除去されずに結果として学習画像と同等の抽出結果が得られた。これは GP では、突然変異だけでなく交叉を用いて解更新が行なわれたためであると考えられる。探索が進むにつれて、良い評価の解で母集団が形成され、その中でさらに交叉が行なわれることでさらに良い解が生成されたためだと考える。今回の実験では、GP と SAP が構築する木構造の組み合わせ方に違いが生じることがわかった。

6 まとめ

本研究において、GP ではプログラムサイズが増大する一方で、イントロン除去を行なうと SAP と同程度のプログラムサイズに減少することが可能であり、ロバスト性においては SAP より高い性能を示すことがわかった。今後は、GP のロバスト性についてさらに検討すること、GP を改良することでより抽出精度を上げ、既存の癌抽出アルゴリズムよりも精度の高い手法を構築することを目標とする。

参考文献

- 1) NEC. 病理画像解析. <http://www.nec.co.jp/solution/bio/rd/>.
- 2) 小椋真貴, 齋藤彰. 癌診断支援のための病理画像解析システム. 病理と臨床, Vol. 24, No. 4, pp. 411-415, 2006.
- 3) 伊庭斉志. 遺伝的プログラミング. 東京電機大学出版局, 1996.
- 4) J.Koza. *Genetic programming, on the programming of computers by means of natural selection*. MIT Press, 1992.
- 5) 藤田佳久, 三木光範, 橋本雅文, 廣安知之. シミュレーションアニーリングを用いた自動プログラミング. 情報処理, Vol. 19, pp. 89-92, 2007.
- 6) 小車頭吾, 高橋正信. 初期肝細胞癌診断支援のための特徴量の考案. *IEICE*, p. 244, 2008.
- 7) 長尾智晴. 進化的画像処理. 昭晃堂, 2002.
- 8) Rosenbluth A. Rosenbluth M. Teller A. Teller E Metropolis, N. Equation of state calculation by fast computing machines. *Journ. of Chemical Physics*, Vol. 21, pp. 1087-1092, 1953.
- 9) P. Keller R. Banzhaf, W. Nordi and Francore. *Genetic Programming: An Introduction*. Morgan Kaufmann Publishers Inc, 1998.
- 10) Angeline. *Subtree Crossover Causes Bloat*. Proc. 3rd Int'l Conf. on Genetic Programming, 1998.