CA-002

遺伝的プログラミングに基づく物体の外枠抽出ルール生成法
Region Border Extraction using Rule-Based Generic Programming

上田　凌*　小野　景子†
Ryo Ueda　Keiko Ono

1　はじめに

遺伝的プログラミング (Genetic Programming:GP) は進化計算手法の一つであり，局所最適解が複数ある問題や設計変数間に依存関係がある問題など，大域最適化を得ることが難しい問題に対する高い解探索能力と汎用性を持つ．またＧＰは遺伝子を木構造で表すため，制御ルールの生成やクラスタリング問題など，木で表現できる問題クラスに適用可能であり，現在までにロボットの行動制御，回路設計，ゲーム制御，画像の識別など多くの分野において利用されている [1, 2].

画像のエッジ抽出のための画像フィルタは木構造で表わすことが可能であるため，GP を用いた手法が提案されている [3, 4] ．画素単位での画像セグメンテーションやエッジ抽出は輝度変化に基づくエッジやフィルタバンクの各フィルタ応答に基づいて行われるが，雑音や多数の局所解による影響を受けやすい．GP は学習によりフィルタを最適化するため，雑音を考慮したフィルタを構築することができる．Zhang らはエッジフィルタを関数同定問題として捉え，ターミナルに定数とノードに算術演算子を用いた手法を提案し，高い性能を示している [5]．一方で，定数と算術演算子をベースにした個体表現は可読性が悪く，フィルタ性能に影響する特徴を抽出することが難しい．

そのため，本研究では，画像セグメンテーションの一歩として領域境界の抽出のための If-Then ルール構築法を提案する．算術演算子に基づく部分木は上下の部分木の影響を受けるため，部分木のみで機能を判断することが難しいが，If-Then ルールを用いることによって，解釈可能なサブルールの頻出パターンを抽出可能になる．

2　提案法

本研究では，遺伝的プログラミングに基づく物体の領域境界抽出ルールを生成する手法を提案する．画像処理において 多く用いられる画像形状に基づいたエッジフィルタは，輝度勾配などに基づく為，物体内部の模様などを抽出する．本研究では，GP により領域境界の抽出を目指すため，領域境界のみの学習データを使用することにより，外枠だけ抽出が出来るルールを生成する．既存研究では，異なる種類の画素情報の算術演算は意味がないため，グレースケール画像の輝度値を終端ノードに用いることが多いが，ルールを作る場合，画素情報が異なる指標の混在は意味があるため，本研究では輝度，RGB，LUV の画素情報を用いる．

2.1　LUV 表色系

CIE LUV 表色系とは国際証明委員会（CIE）が定めた色を定量的に記述するための表色系の一種である．CIE XYZ 表色系は，画素の RGB 値に基づいているが，人が識別できる色差が色の領域によってかなり異なることが知られおり，色度図では色差と図上の離隔距離が一致しない．一方，CIE LUV 表色系はできるだけ人間の感覚に近い均等な色空間として定義された．

2.2　アルゴリズム

複数の学習画像から一定の割合でカットアウトを切り出し，汎用的な境界抽出フィルタを作成する．提案法のアルゴリズムを以下に示す.

Algorithm 1 提案法

Step1: If-Then ルールに基づいたツリー構造の初期個体を生成

Step2: 学習画像よりランダムサンプリングしたカットアウトを用いて個体を評価

Step3: 遺伝操作を適用

Step4: 学習期間内は Step2，学習期間終了後は Step5 へ移動

Step5: 評価値に基づき最良個体を抽出

Step6: テスト画像に対し sliding window を用いて最良個体のルールを適用し，境界を抽出

2.3　ノード設計

2.3.1　終端ノード

木構造の終端記号には，{edge，nonedge} と設定する．edge は画像の注目ピクセルが境界である，nonedge は境界では無いと判断する．

2.3.2　非終端ノード

画像から抽出できる特徴をもとにサブルールの組み合わせて全体ルールを構築するため，非終端記号は，

$$\mathrm{If}(L < \alpha)\{A \ \ \mathrm{else}\{B\}\}$$

$$\mathrm{If}(D < \alpha)\{A \ \ \mathrm{else}\{B\}\}$$

とする．ここで，L は輝度値も基づく特徴量であり，D は RGB と LUV の色情報に基づく特徴量である．また，A と B には終端ノードまたは非終端ノードが入る．また α は閾値を示す．L または D の詳細は，以下のとおりである．括弧の中は領域かどうか判断する注目ピクセルのウィンドウの形を示しており，正方形と長方形を用意した．

L：輝度値
maxL：最大輝度（5×5）minL：最小輝度（5×5）
ave3L：輝度平均（3×3）ave5L：輝度平均（5×5）
std3L：輝度標準偏差（3×3）std5L：輝度標準偏差（5×5）
std2×5L：輝度標準偏差（2×5)
std5×2L：輝度標準偏差（5×2）

D：RGB 値または LUV 値
ave3D：色平均（3×3）ave5D：色平均（5×5）
std3D：色標準偏差（3×3）std5L：色標準偏差（5×5）
std2×5D：色標準偏差（2×5)
std5×2D：色標準偏差（5×2）

また，ave3D は，注目ピクセルの RGB 値もしくは UV 値から，3×3 ウィンドウ内のそれぞれのピクセルとの色空間上

*龍谷大学 理工学部 電子情報学科
†龍谷大学 理工学部 電子情報学科

の距離を計算し，平均を求めている．他の記号も ave3D と同様である．これらを用いた木構造の例を図 1 に示す．

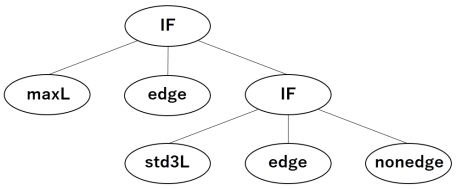


図 1: 木構造の例

また，今回の実験での画像データは，輝度値のみと輝度値＋ RGB 値，輝度値＋ CIE LUV 表色系の UV 値の 3 種類において比較する．

2.4 目的関数

本研究では評価関数 $f(x_i)$ を最大とする x_i の最大化問題を考える．評価関数は，進化された木構造によって，注目ピクセルが境界であるかないかの判断を行い，正解データとい一致するか否かで，一致する場合のみに，評価関数値を一つ増加させる．ここで，対象とする正解データの画像 I にはピクセル $p_j; j = 1, 2, 3, \ldots,$ n があり，各ピクセルには，エッジ，エッジ以外というラベル $l(p_j)$ が付加されている．ここでラベルは 0 か 1 の 2 値とする．このとき，ＧＰを用いて生成した個体 x_i に対する評価関数は

$$f(x_i) = \sum_{j=1}^{n} g(p_j|x_i),$$

$$g(p_j|x_i) = 1 \quad \text{if } l(p_j) = x_i(p_j),$$

$$g(p_j|x_i) = 0 \quad \text{otherwise,}$$

と定義する．n はピクセル総数である．GP により抽出した境界が正解データに近づくにつれ，評価関数は増加する．

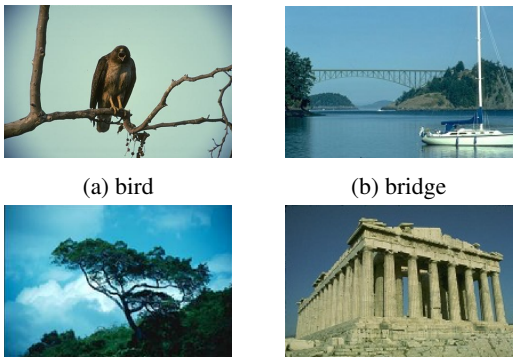
3 評価実験

3.1 パラメータ設定

本研究では，各パラメータは以下のように設定した．個体数=500; 世代数=50; 交叉率=0.9; 突然変異率=0.1; 終了条件=適応度 (=98000) が最大になった，もしくは，最大世代数に到達した場合とした．また，条件文の閾値 α の標準偏差は輝度と RBG が 50 程度，その他は 200 程度，LUV は最大が 1 のため標準偏差は 0.01 程度，その他が 0.1 程度になるようにランダムに設定した．木の最大の深さは設定せず，初期個体は木の深さが 3 から 5 の間になるように，ランダムに生成した．

3.2 画像セット

実験では，エッジ抽出を行う際に The Berkeley Segmentation Dataset and Benchmark より，20 枚の学習画像と 4 枚のテスト画像を使用した [6]．テスト画像は 200×133 ピクセルである．学習画像は， 1 枚の画像より 70×70 ピクセルを切取った画像を用意した．図 2，5，6 にテスト画像と学習画像，学習画像の正解データをそれぞれ示す．



(a) bird (b) bridge

(c) tree (d) greece

図 2: テスト画像

3.3 実験結果

3.3.1 単一画像での学習結果

はじめに，輝度以外の色情報を用いる効果を検証するために，単一画像で学習させた場合の結果を図 3 に示す．図 2 のテスト画像のうち，bridge を用いた．図 3 より，輝度値のみで学習した場合より，RGB 情報を付加した場合の方が境界を多く捉えられていることがわかる．この結果より，If-Then ルールを用いて境界を検出する場合，RGB 値が有効に働くことがわかった．

(a) Gray (b) RGB

図 3: 単一画像での学習結果

3.3.2 学習結果

学習に関する結果を示す．表 1 に初期個体と最終世代の適応度を示す．初期世代の適合度に差はないが，最終世代では輝度と輝度+LUV が同程度の学習性能を示した．また，図 4 に学習の履歴を示す．横軸が世代数，縦軸が適合度である．この結果より，学習過程において探索が停滞することなく，学習が進んでいることがわかる．

表 1: 学習における適合度

世代	輝度	RGB	LUV
初期世代	90558	90558	90557
最終世代	90777	90760	90772

3.3.3 テスト結果

Gray，RGB，LUV のデータを用いた学習結果，つまり，最終世代の最良個体を抽出し，それぞれを用いて 4 枚のテスト画像に対して境界を抽出した．結果を図 7 に示す．左から順に図 2(a)，(b)，(c)，(d) に対する結果である．また，表 2 にテストにおける適合度を示す．これらの結果より，Bird と Tree の適合度が高く，Bridge と Greece の結果が低いこと

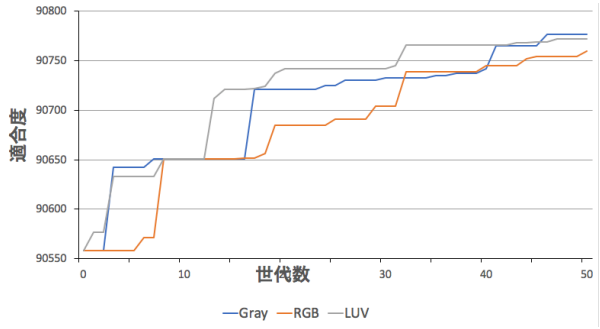


図 4: 世代数に対する適合度の変化

がわかる．また，RGB 値を用いた方が輝度値のみの場合より若干，性能が高くなった．

表 2: テストにおける適合度

画像	輝度	RGB	LUV
bird	25353	25357	25324
bridge	24358	24323	24356
tree	25303	25314	25305
greece	23682	23679	23675

GP で作成したルールに基づき領域が抽出可能である理由と RGB の効果を分析するために，輝度と RGB を用いた場合の最良個体のルールを分析する．以下のルールの 1 つ目が輝度の結果であり，2 つ目が RGB の結果を示す．輝度の個体を分析すると，輝度の平均値や最大値および最小値より，標準偏差 (std) が多く用いられていることが分かった．これは，sliding window 内の輝度変化が大きい場所を境界として認識していることを示している．また，2×2 や 5×5 の正方形の window より 2×5 や 5×2 の長方形の window を多く利用していることが分かった．これは画素の縦方向や横方向の連続的な変化を長方形の window の方がより捉えやすいことを示唆している．

(std2x5L>70){(std5x2L>40){(maxL>230){(std5x2L>60)
{(std5x2L>60){(std3L>100){(std5x2L>40){nonedge,(std2x5L>50){
edge,nonedged},edge},nonedged},(std5x2L>40){edge,nonedged},
nonedged},nonedged},edge},nonedged},(std2x5L>50){
(std5x2L>60){(maxL>230){(std2x5L>50){(ave3L>150){
(std2x5L>70){edge,nonedged},edge},nonedged},(std2x5L>50){
edge,nonedged},(std2x5L>70){nonedged,nonedged},(std2x5L>70){
(std5x2L>60){nonedged,nonedged},(std5x2L>40){(std5x2L>60){
(std5x2L>60){(minL>30){edge,edge},nonedged},(std2x5L>70){
(std5x2L>60){edge,nonedged},nonedged},nonedged},edge},
nonedged}

次に RGB の結果を分析する．RGB は輝度情報に RBG 情報を加えてルールを抽出している．*D* が RGB 情報を示し，*L* が輝度情報を示す．個体から，RGB 情報のみでなく，多くの輝度情報が用いられており．輝度と RGB の結果が大きく変わらなかったのは，両方を用いているためだと考えられる．また，window 形状は輝度と同様で長方形の window が多く個体に存在していることが分かる．以上の結果より，提案法により，If-Then ルールにより領域の境界を抽出することが可能であること，複数の終端ノードと非終端ノードを用意しているが，進化の過程で長方形の window が残り，正方形より境界周辺の変化を捉えることが分かった．

(std5x2L>60){(std2x5L>70){(std5x2D>30){edge,(ave3D>80){
(std2x5D>90){(std5L>100){(ave3D>80){(std5L>40){nonedged,
nonedged},(ave3D>80){nonedged,(std5L>40)},edge},nonedged},
(std3L>40){edge,nonedged},nonedged},nonedged},(ave3L>190){
(std2x5D>90){edge,nonedged},(std2x5L>50){(ave3L>150){
(L>170),edge},nonedged},(std2x5L>70){(std2x5D>90){edge,
nonedged},edge},nonedged},nonedged}

4 まとめ

本研究では，遺伝的プログラミングに基づく物体の領域境界の抽出ルールを生成する手法を提案した．実験結果より，シンプルな終端ノードおよび非終端ノードを設計するのみで，領域の境界をある程度抽出可能なルールを生成できることが分かった．非終端ノードの設計は性能に非常に大きな影響を与えるため，今後，改良する必要がある．例えば，テスト画像に含まれる色彩量に合わせて，輝度と RGB のどちらか優先して境界判断に用いることを可能にする，sliding window の形状のバリエーションを増やす，またその形状を探索過程で学習するメカニズムを導入するなどが考えられる．

参考文献

[1] John R Koza. *Genetic Programming II, Automatic Discovery of Reusable Subprograms*. MIT Press, Cambridge, MA, 1992.

[2] William B Langdon and Riccardo Poli. *Foundations of genetic programming*. Springer Science & Business Media, 2013.

[3] Christopher Harris and Bernard Buxton. Evolving edge detectors with genetic programming. In *Proceedings of the 1st annual conference on genetic programming*, pp. 309–314. MIT Press, 1996.

[4] N Senthilkumaran and R Rajesh. Edge detection techniques for image segmentation–a survey of soft computing approaches. *International journal of recent trends in engineering*, Vol. 1, No. 2, pp. 250–254, 2009.

[5] Krzysztof Krawiec, Daniel Howard, and Mengjie Zhang. Overview of object detection and image analysis by means of genetic programming techniques. In *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*, pp. 779–784. IEEE, 2007.

[6] Pablo Arbelaez, Charless Fowlkes, and David Martin. The berkeley segmentation dataset and benchmark. *http://www.eecs. berkeley. edu/Research/Projects/CS/vision/bsds*, 2007.

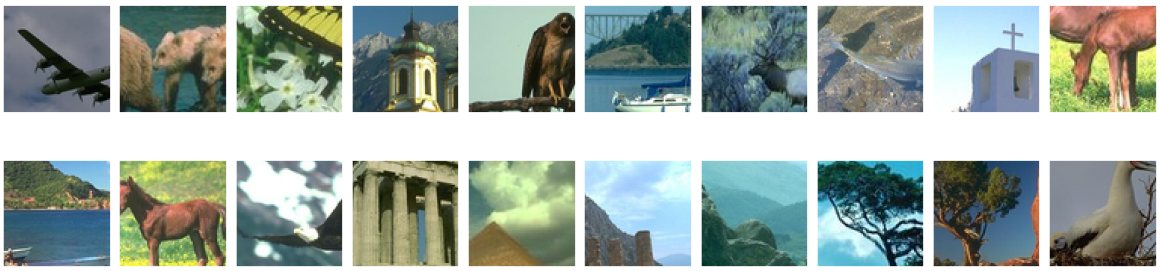


図 5: 学習画像

図 6: 学習画像の正解境界画像

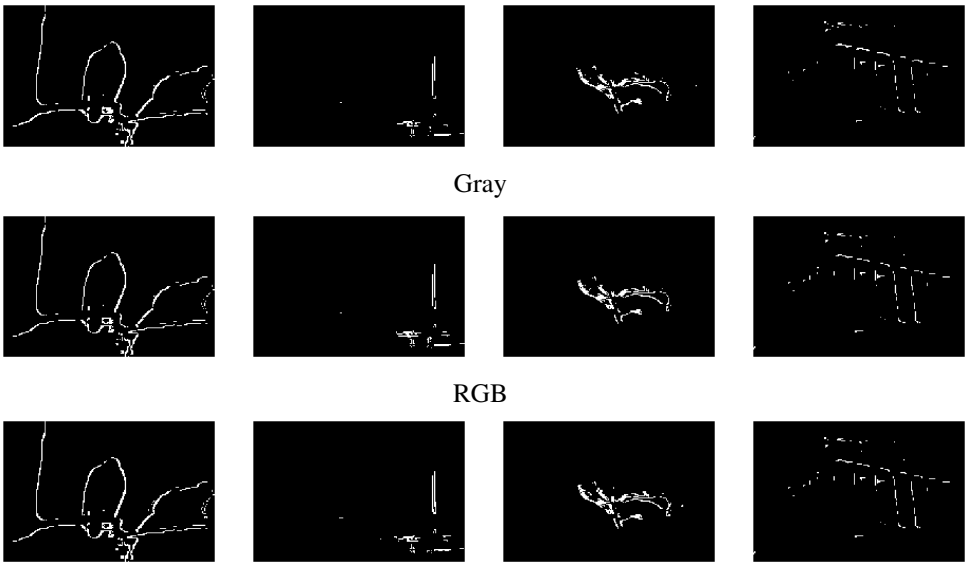


図 7: テスト結果