統合順序の履歴保持に基づく画像の領域分割

日大生産工（院）　○中尾一博
日大生産工　目黒光彦

1　はじめに

画像や動画像を，似た色や模様の領域ごとにまとめ，分割する処理は領域分割と呼ばれている．領域分割は，画像からの物体領域の抽出や，画像内容の認識処理などの前処理として使われる．領域分割が行われたあと，関連する領域を統合することによって，物体の抽出が可能となる．さらに，領域分割の結果は，画像の内容がコンパクトに記述されていることから，画像内容の特徴を表しているといえる．

領域分割を実現する手法の一つとして，Watershed がある．Watershed とは，画像の輝度値の勾配を表す輝度の微分値を標高と見立て，尾根の部分で囲まれる領域を一つの領域として分割する手法である．しかしながら，領域が過剰に分割される傾向にある．そのため，Waterhsed による領域分割を行った後，分割結果に対して，輝度値や色差の近い隣接する領域間を統合する領域統合法が提案されている [1],[2]．文献 [1] の手法はモノクロ画像に対して，文献 [2] の手法はカラー画像に対して，画像領域の分割・統合を行っている。しかしながら、[1],[2] の両手法とも，領域統合の過程を考慮せずに，最終の統合結果に対してのみ評価を行っていた．そのため，領域の誤統合，未統合があった場合でも，どの領域の処理が不適当であるか否かの判断を行うことができなかった．

本稿では，統合順序の履歴を保持し，どのように領域が統合されているかを可視化する手法を提案する．本手法も，文献 [1]，[2] と同じく，Watershed を用いて領域分割を行い，領域統合においても隣接する領域の色差を領域統合の判定に用いる．さらに，全ての領域間の隣接領域との関係を把握するために，各領域に隣接する領域を RAG(領域隣接) グラフにより，領域統合の過程を保持しておく．既存の手法でも，隣接領域間の関係をグラフとして処理していたが，本研究では，隣接領域の統合過程自体をグラフとして保持，図示化するものである．領域統合に関する情報を蓄えることにより，統合順序を確認し，どの領域が統合されているかを把握することが可能である．これにより，誤統合・未統合の問題を解決することができ，より正確な領域分割統合処理が実現される．

2　画像の領域分割・統合処理

2.1　Watershed アルゴリズムによる領域分割

Watershed アルゴリズムとは，画像の画素の輝度値を山の標高，輝度値の極小値 (以下，minima)，極大値 (以下，Dam) をそれぞれ谷の底，尾根とみなして領域分割するアルゴリズムである．はじめに，画素の輝度勾配から minima となる箇所を探索する．次に，探索した全ての minima に対して番号付けを行う．以降，アルゴリズムを図 1 を用いて説明する．番号の小さい minima から順に処理を行う．まず，minima1 の谷に水を注入し，Dam となる箇所を見付ける．図 1 では Dam1 に相当する．続いて，minima を挟む Dam の箇所が見付かるまで水を注入していく (図 1，Dam2)．minima を挟む Dam が見付かったら (図 1，minima1,Dam1,Dam2)，minima を挟む Dam 間が Watershed となり，一つの領域となる．このアルゴリズムを，番号付けした全ての minima に対して処理を行うことで領域分割する．図 1 では二つの領域に分割したことになる．図 2 に対して Watershed アルゴリズムを適用したのが図 3 である．

しかし，Watershed アルゴリズムの問題点として輝度の変化で領域分割するため，雑音や影などの微細な変化に対しても領域分割する．結果として，過剰分割になってしまうため，過剰分割を防ぐために，あらかじめ入力画像に対して平滑化処理を行うことで，雑音や影の影響を軽減しておく．

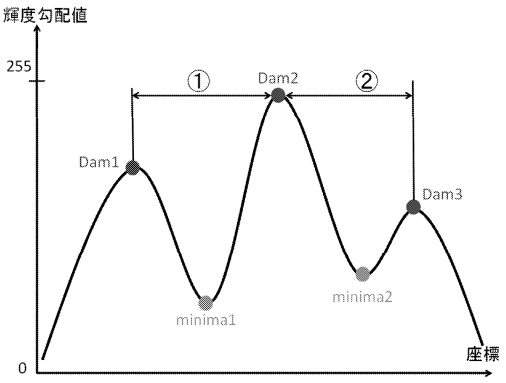


図 1: 輝度勾配による領域分割法

Image segmentation based on region merging by keeping connected order
Kazuhiro NAKAO and Mitsuhiko MEGURO

図 2: 原画像

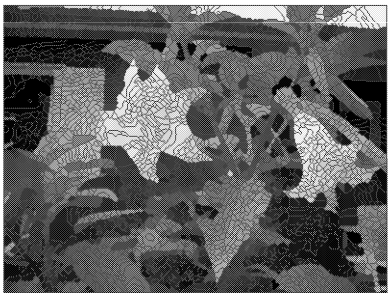


図 3: 領域分割結果

2.2 RAG と色差による領域統合

$L^*a^*b^*$ 表色系に基づく隣接領域間の色差 ΔE と RAG(Region Adjacency Graph:隣接領域グラフ) を用いた領域統合を行う.

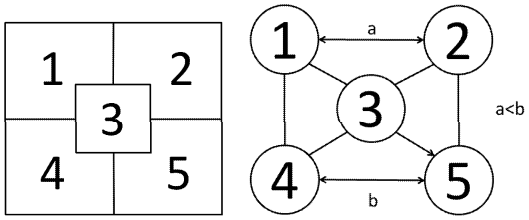
$L^*a^*b^*$ 表色系に基づく色差 ΔE とは，2 つの領域の色の平均値を (L_1^*, a_1^*, b_1^*) と (L_2^*, a_2^*, b_2^*) とすると，この 2 つの領域間の色差 ΔE^* は

$$\Delta E^* = \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$$

で与えられる.

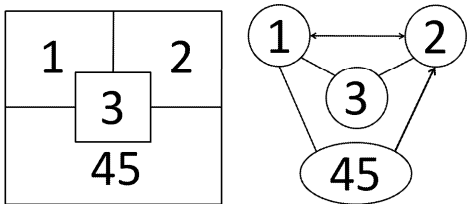
RAG とは Watershed で求めた各領域に対して，全ての隣接領域間の類似関係を線と矢印で表現したグラフである．RAG の求め方は，まず，分割された各領域について，領域内の $L^*a^*b^*$ 表色系に基づく平均色を求める．次に，各領域について，全ての隣接領域の色差を求める．その中で，最も色差が小さい隣接領域である類似領域を求め，その類似領域に向かって矢印を書く．さらに，隣接領域同士が類似領域となる矢印を向け合う状態をサイクルと呼び，これを探す．複数のサイクルの中で，最も色差が小さい 2 つの領域の組である最類似隣接領域を見付ける．図 4(a) のように領域分割された領域に対する RAG は図 4(b) となる．つまり，辺が隣接領域を表し，片方向の矢印が類似領域を表し，双方向の矢印がサイクルを表す．図 4 では，領域 1，2 のペアと領域 4，5 のペアがサイクルとなっている．この各ペアの色差を a, b としたとき，$a < b$ なので，領域 4，5 が最類似領域であり，統合対象と判断する．統合後 RAG グラフの様子を図 5 に示す．統合後には，領域 4，5 の色の平均値を更新し，新たな領域 45 の色の平均値とする．RAG が変更されたため，全ての領域に対して隣接領域を再設定する．統合後の RAG 表現を図 5(b) に示す．この一連の処理を最類似領域の色差 ΔE が閾値以下になるまで繰り返す.

また，Watershed を施した画像に対して，すぐに領域統合を行うのではなく，領域内のピクセル数が閾値以下の狭い領域については，最も色の似通った隣接領域と統合しておく．この前処理を行うことで，図 3 によって求めた領域数が 3357 だったのに対して，領域数を 2384 まで減らした状態から領域統合を行うことができる．図 6 に，図 3 の領域に対して領域統合を行った結果を示す.



(a) 領域分割例　　$(b)RAG$ 表現

図 4: 領域分割と RAG 表現



(a) 領域統合例　　(b) 統合後の RAG 表現

図 5: 領域統合後の RAG 表現

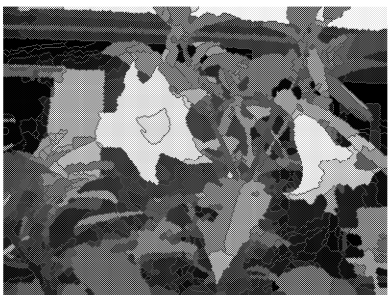


図 6: 領域統合結果

3 統合順序の履歴保持に基づく画像の領域分割

本稿では，統合順序の履歴保持をすることで，統合過程を可視化する手法を提案する．これにより，領域統合を行う毎に，どの隣接領域間の最類似領域が統合されたか，また，統合前後の隣接関係，色差などの情報を蓄えることによって，領域統合の履歴が保存され，領域統合を評価することができる．さらに，RAG を図示化することにより，どの領域がどの領域と隣接関係にあるかを確認するこ

とができる．領域統合に必要な情報を蓄えるためのデータ構造を構造体により以下のように定義する．

```
TYPEDEF STRUCT REGION{
    着目領域の要素数
    隣接領域の個数
    各領域内部の要素数
    着目領域と最も近い隣接領域との色差
    ....
    ....
    ....
  }REGION;
```

定義された構造体の中から統合過程の情報に必要な部分を抜き出す．以下が，その内容である．

```
MERGING NUMBER 1
THE PEAR SHOLD BE MERGED IS
```

$$(621, 602) \tag{1}$$

```
AREA OF THIS PEAR IS
```

$$(99, 325) \tag{2}$$

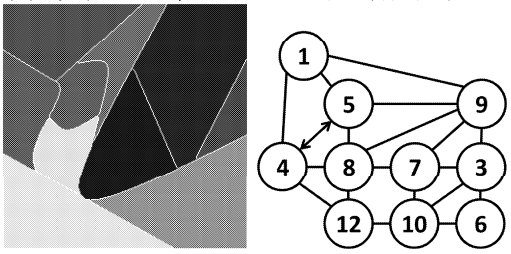
```
THE WEIGHT OF DISTANCE IS
```

$$0.054506 \tag{3}$$

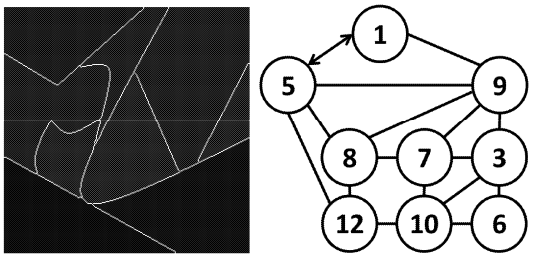
merging number とは統合の回数を表している．上記の例では統合 1 回目を表している．
(1) は，統合するペアの領域番号
(2) は，統合するペアの領域の面積
(3) は，統合するペアの色差
を表している．

次に，領域統合をする毎に，領域統合経過の画像出力と RAG の図示化により，どのように統合されているかの確認の例を，領域数 9，統合回数 5 のサンプル画像に対して，領域統合とその統合順序を示す．

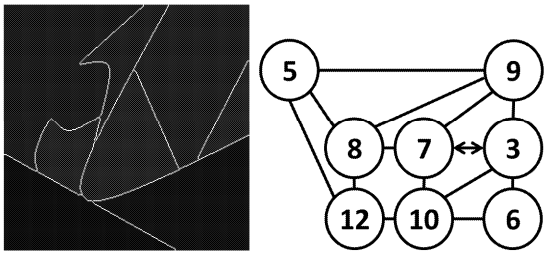
図 7(a) に Watershed などによる領域分割の結果を示す．図 7(b) は図 7(a) の RAG 表現である．図 8(a) から図 12(a) には図 7(a) の統合結果を示す．図 8(b) から図 12(b) には，統合の様子の RAG 表現を示している．双方向の矢印が次に統合すべき最類似領域を示し，図 13 が統合の最終結果である．



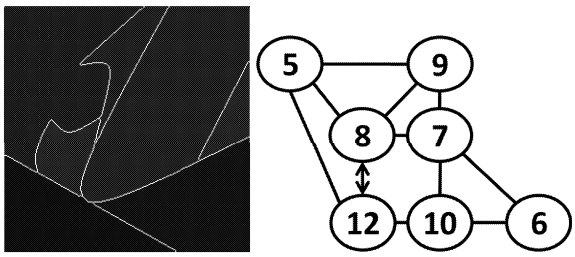
(a) 領域分割例　　(b)(a) の RAG 表現
図 7: 領域分割と RAG 表現



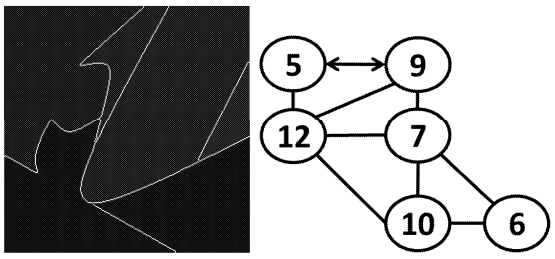
(a) 統合 1 回目　　(b)(a) の RAG 表現
図 8: 領域統合 1 回目と RAG 表現



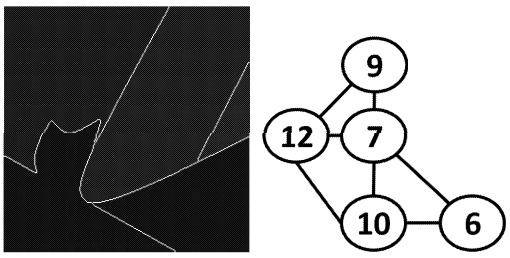
(a) 統合 2 回目　　(b)(a) の RAG 表現
図 9: 領域統合 2 回目と RAG 表現



(a) 統合 3 回目　　(b)(a) の RAG 表現
図 10: 領域統合 3 回目と RAG 表現



(a) 統合 4 回目　　(b)(a) の RAG 表現
図 11: 領域統合 4 回目と RAG 表現



(a) 統合 5 回目　　(b)(a) の RAG 表現
図 12: 領域統合 5 回目と RAG 表現

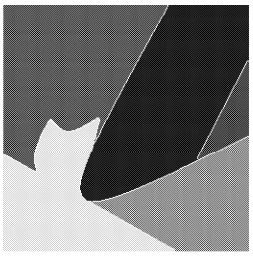


図 13: 統合終了

4　適用例

図 14 の領域分割の結果に対して図 16 の結果になるまでの統合過程の一部を図 15 で示す。
領域分割数は 829，統合前の事前処理で領域数 622，統合回数 549，最終領域数 76 の処理結果である．統合は色差が 5 以上になるまで繰り返す．

図 15(a) と (b) は 445 回目の領域統合では，ラケットの上の背景領域が統合されている．このときの色差は約 2.5 である．

次に図 15(b) と (c) は 446 回目の領域統合では，腕の下の背景領域が統合されている．このときの色差は約 2.0 である．

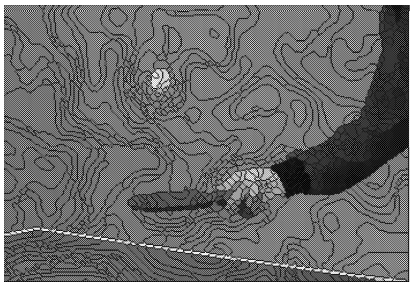
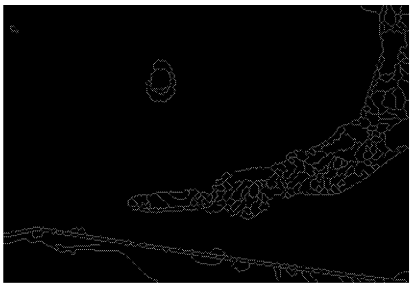


図 14: 領域分割結果

(a) 統合 444 回目

(b) 統合 445 回目



(c) 統合 446 回目

図 15: 領域統合過程 (1)

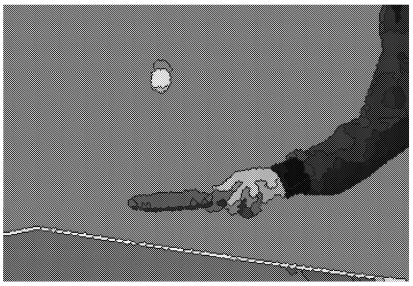


図 16: 領域統合結果

5　おわりに

本稿では，領域統合の統合順序を可視化することで，どうように領域統合されているかを確認する方法を提案した．今後の課題は，処理の高速化，および，誤統合があった場合に，RAG を使って自動的に一つ前の分割状態にもどり，適した領域統合の手順を探索するアルゴリズムを構築することである．

参考文献

1. K.Haris, et al., “Hybrid Image Segmentation Using Watershed and Fast Region Merging”, IEEE Trans. on Image Processing, Vol.7, No.12, 1998, pp.1684-1699.
2. 松崎慧介，目黒光彦，古閑敏夫, “変形輪郭モデルに基づく動画像から任意オブジェクトの抽出”, 電子情報通信学会, 信学技報 SIS2006-78, 2007-03, pp.34-39.