

# 複数に分割された画像の復元

濱本 高志・金出地博之\*・大倉 充\*\*・塩野 充\*\*

岡山理科大学大学院工学研究科博士課程システム科学専攻

\*株式会社オーユーシステム

\*\*岡山理科大学工学部情報工学科

(1998年10月5日 受理)

## 1. ま え が き

本研究では、1枚の画像を複数に分割した後に、元の画像に復元することを試みた。原画像の分割はランダムに行い、分割された画像（以下、ピース画像と記す）の形状には制約を設けず、また任意形状の画像に対応するアルゴリズムの構築を目指し、ジグソーパズルの組み上げに関する研究<sup>1)</sup>で一般的に行われる、外枠を最初に組み上げる方法は採用していない。本研究では、任意のピース画像を基準として、輪郭形状および色情報を基に他のピース画像との接合箇所を検出し、順次、接合画像の生成を行っており、この探索処理にはバックトラック法を採用した。

## 2. 実験に使用した画像

ピース画像の作成は、背景領域を明確にするため、分割された写真等をスキャナで取り込むのではなく、フィルムスキャナを用いて入力した画像をコンピュータ上で分割する方法で行った。また、単調な色使いにならないように原画像には情景画像を使用する。原画像から作成するピース画像の形状は任意であるが、本研究では、ピース領域が単一の輪郭線で構成されていることを条件とし、割り貫いた形状には分割しないものとする。

実験に使用した原画像は、形状が矩形、画像サイズ340×228、RGB各256階調のカラー画像である(図1参照)。原画像からのピース画像の作成は、画像編集ソフトウェア(Adobe Photoshop 3.01J)を用い、本研究では、まず任意の形状に原画像を8分割した(図2参照)。次に各分割画像ごとに任意の回転角で回転処理を施し、得られた画像をピース画像とする。図3にピース画像を示す。

## 3. 特徴量の抽出

各輪郭線上の特徴を基に接合処理は行われるが、ピース画像は原画像からの分割後に回転処理を施して作成されているので、ピース画像の輪郭線上の形状や色の特徴が明確でない箇所が存在する。そのため、輪郭線を補正した後に輪郭線上の画素の特徴量を抽出する



図1 原画像

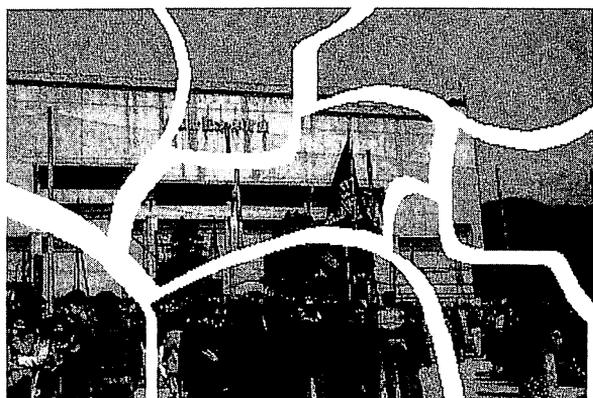


図2 分割された原画像

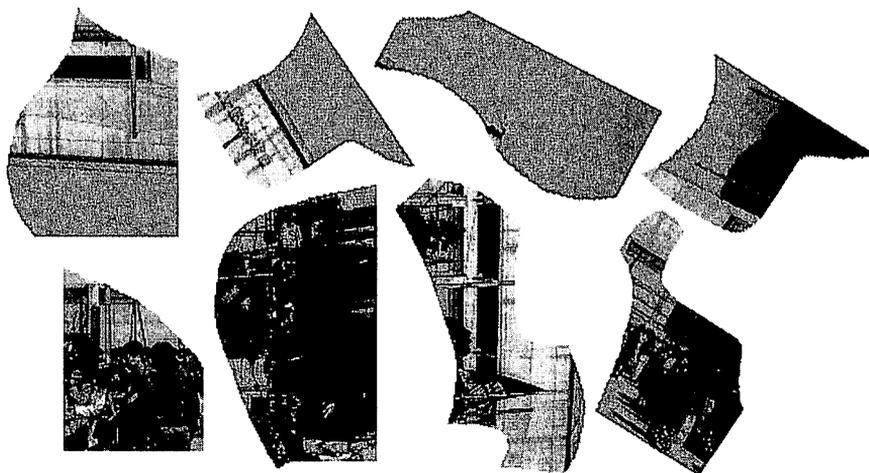


図3 ピース画像

必要がある。

まず、背景領域として指定した RGB 値でラベリング処理<sup>2)</sup>を行い、同じ RGB 値を持つピース領域と背景領域を明確に区別した上で、2 値化を行いピース領域を定める。得られた 2 値画像に対して膨張・収縮処理<sup>2)</sup>を行い、輪郭上の細かな凹凸を除去する。ピース領域と背景領域の境界付近の画素は両領域の RGB 値が混ざり合った不明瞭な値を持つ場合があるので、再度ピース領域に対して収縮処理を行い、最外周の画素を除去する。次に輪郭線追跡処理<sup>2)</sup>を行い、輪郭線を構成する画素を定める。膨張処理を行っているため、輪郭線上の画素の中にはピース画像上で背景領域に相当する画素も存在する。そのため、輪郭線上の画素の RGB 値は、その輪郭線上の各画素と同じ位置にあるピース画像の画素を中心とした一定半径を持つ円を考え、その円内におけるピース領域に含まれる各画素の RGB 値の平均値を使用する(図 4 参照)。ここで求められた平均値より、色相、彩度および明度を求め、その輪郭線上の画素の特徴量とする。

#### 4. 接合箇所の探索

全ピース画像の中より、任意に一つのピース画像を選び基準ピース、他の全てのピース画像を対象ピースとする。基準ピースの輪郭線上の画素(基準画素と記す)を任意に定め、対象ピースの輪郭線上の画素(対象画素と記す)との間で特徴量の比較を行う。以下に示す条件 1 を満たす対象画素が検出された場合には、条件 2 および 3 の判定を行い、これらが満足された場合には、検出された対象画素を 1 画素輪郭線上を左回りに移動させて同様の条件判定を続ける。これらの探索処理を全ての対象ピースに対して行った後に、基準画素を 1 画素輪郭線上を右回りに移動させて同様の探索処理を続ける。

条件 1

式(1)により得られる値  $c$  が一定のしきい値以下になる。

$$c = \sqrt{(S_i \cos H_i - S_j \cos H_j)^2 + (S_i \sin H_i - S_j \sin H_j)^2 + (V_i - V_j)^2} \quad (1)$$

ここで、 $H$  は色相値、 $S$  は彩度値、 $V$  は明度値を表し、添え字の  $i$  は基準画素、 $j$

RGB 値の平均値を  
求める領域

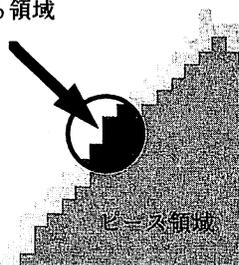


図 4 各 RGB 値の平均値を取得する範囲

は対象画素を示す。

#### 条件2

最初に全ピース画像の中から最も短い輪郭線の長さを求め、1つのピース画像に対して10ピース以上接合することはないという考えから、最短の輪郭線の1/10の長さ $l$ を最低限接合するために必要な輪郭線の長さとして定める。次に、基準画素と基準画素から輪郭線上を右回りに長さ $l$ だけ移動した画素を結んだ線分 $d$ を求め、対象ピースについては、対象画素から輪郭線上を左回りにほぼ長さ $l$ だけ移動した付近に存在し、線分 $d$ と等しくなるような画素を検出する。各ピースにおいて、線分 $d$ を数等分した各位置より輪郭線までの垂直距離を求める。各ピースの対応する垂直距離を比較し、全ての距離がほぼ等しくなる場合、条件を満たすとする(図5参照)。更に長さ $l$ を少し延ばした位置にある画素に対して、同様の比較処理を垂直距離に違いが見られるまで行い、接合候補箇所の最長距離の輪郭線分を求めておく。

#### 条件3

条件2で得られた基準ピースおよび対象ピースにおける輪郭線分上の全ての対応する画素間で、式(1)により得られる値を求め、これらの値が全てある一定のしきい値以下になる。

### 5. 接合候補箇所の絞り込み

接合箇所の探索処理で得られた接合候補箇所は多数存在するため、有力な接合候補箇所のみを残し候補数を減らすために次の処理を行う。

まず、基準ピースおよび対象ピースの接合候補箇所が他の接合候補箇所の領域内に完全に含まれる場合は候補から外す。次に、基準ピースおよび対象ピースの接合候補箇所のうちどちらか一方でも他の接合候補箇所と過度に重複する場合、実際に接合を行った際に重複する画素の少ない候補箇所を残す。

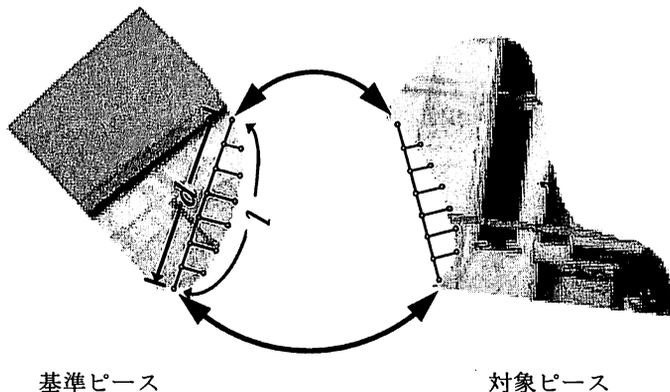


図5 直線距離と垂直距離の関係

最後に、残された接合候補箇所に対して、輪郭線分が最も長くなる候補から順に順位付けを行う。

### 6. 復元処理

得られた最も高い候補順位にある対象ピースをアフィン変換<sup>2)</sup>で回転・移動させ、基準ピースと接合候補箇所<sup>3)</sup>で接合を行う(図6参照)。接合した箇所<sup>4)</sup>で重複する領域<sup>5)</sup>が大きい場合には、候補順位の次の順位に当たる箇所と同様の接合処理を行い、重複領域があるしきい値以下になるまでこの処理を繰り返す。重複領域があるしきい値以下で接合した画像を新たな基準ピースと定め、接合箇所の探索処理を同様に行う。もし、全ての接合候補箇所<sup>6)</sup>で接合が行われない場合、現在の基準ピースに誤りがあると判定し、この基準ピースが生成される一つ前の段階に戻り復元処理の過程を続ける。

以上の処理を繰り返し、対象ピースが全て接合され、最終的に得られた画像がほぼ矩形の状態になった時点で終了する。

### 7. 実 験

実験に使用したコンピュータのCPUはPentiumII 233MHz(intel)、メモリは128Mbyteである。その環境の基で、C言語で作成したプログラムにより実験を行った結果、復元に

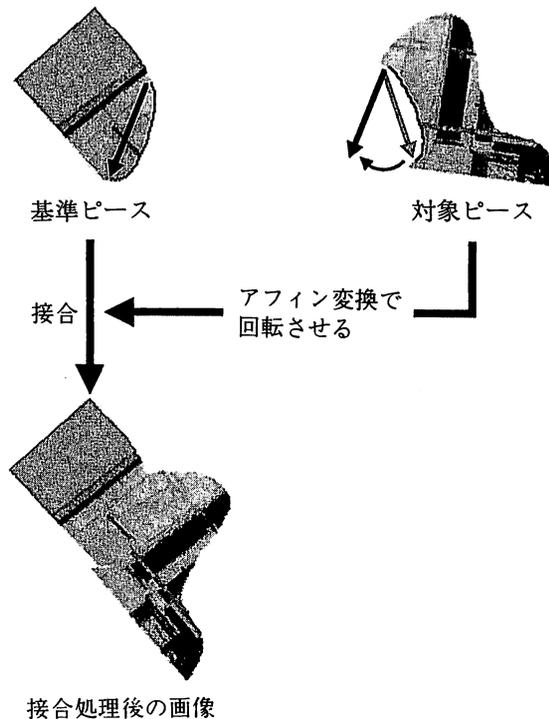


図6 接合方法

は約92時間要し、図7に示す復元画像が得られた。

復元処理過程における画像の接合時の若干のずれが蓄積され、復元処理が進むに連れて接合箇所探索で本来の接合箇所が検出されにくくなっていった。また、同じようにずれの影響で接合箇所が検出された場合でも接合が行われないという現象が生じ、既に接合された画像が外される場合もあった。原画像中に色変化の少ない領域が多いために、例えば図8に示すように、ほぼ同色の領域を有するピース画像が優先的に接合される結果となり、バックトラック法により修正されるまでには膨大な時間が必要となることも明らかとなった。

## 8. む す び

本研究では、ランダムに分割された画像の復元を試みた。接合時におけるずれの蓄積等により復元された画像には所々に隙間が見られるが、全く異なった箇所に接合されていることはなく、ほぼ正しい接合箇所の検出は行われたと考えられる。しかし、本実験では接合処理における回転処理等で生じる若干のずれを補正することは考慮していないので、ずれの蓄積による影響が大きく現れることとなった。また、バックトラック法を用いてるた



図7 復元された画像

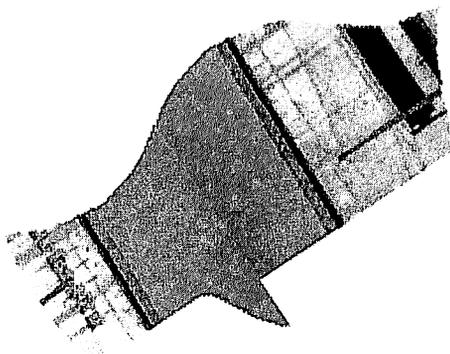


図8 誤った接合箇所の一例

め最初の接合で誤った場合、処理時間に多大な影響を及ぼすことになった。今後は、次の段階の接合処理で影響を及ぼさないように接合時にずれの補正を行うこと、接合候補箇所の適切な絞り込みと順位付け、および処理速度の向上が必要であると考えている。

#### 参考文献

- 1) 宮道壽一, 村田真人: “輪郭線情報を用いた多数ピースジグソーパズルの計算機解法”, 信学論 (D), J70-D, 6, pp. 1210-1217 (1987-06).
- 2) 長谷川純一, 興水大和, 中山 晶, 横井茂樹: “画像処理の基本技法”, 技術評論社, 1986.

## The Restoration of a Fragmented Image Using the Backtrack Method

Takashi HAMAMOTO, Hiroyuki KANADECHI\*, Mitsuru OHKURA\*\*  
and Mitsuru SHIONO\*\*

*Doctoral Program in System Science,  
Graduate School of Engineering,  
Okayama University of Science,*

*\*OU System Co.,*

*\*\*Department of Information & Computer Engineering,  
Okayama University of Science.*

(Received October 5, 1998)

In this paper, we attempted to restore an original image that was divided into several fragments, with the divisions made at random and the shape of the fragments having no restrictions. Rather than adopting a method based on the image frame that is often used when a jigsaw puzzle is assembled, we designed an algorithm that could be used for any shape of image. Firstly, one fragment was chosen arbitrarily as a basic image and a connected image was made by searching for an adjacent fragment using specific areas of fragment contours which matched those of the basic image. Secondly, the connected image was taken to be the basic image and the next adjacent fragment was searched. If an incorrect fragment was attached to the basic image, the backtrack method was employed and the incorrect fragment replaced with the correct one. Although the final connected image had overlapping areas and splits, the algorithm was successful in detecting the correct adjacent fragments.