ジグソーパズルのピース自動配置アルゴリズムの提案

Proposed Algorithm for the Piece Automatic Assignment of Jigsaw Puzzle

1941102　御幡　直熙
Naoki MIHATA

指導教員　秋葉 知昭

This study develops an automatic assignment algorithm using 2D image processing and the shape information of jigsaw puzzle pieces. Acquire the backside image of the piece using a scanner, and the image is resized and binarized. Then, the contour information of the piece is obtained, and the position information of the four corners of the piece is obtained by performing corner detection on the contour. The contour information of each of the four edges between the corners is obtained and feature extraction is performed for each edge. Classify all the pieces into groups of similar pieces based on the shape information of each edge. By matching the shape features of the pieces and identifying the connection relationship between the pieces, the pieces are placed using only the shape information of the pieces. The results of our placement algorithm show that the pieces are placed correctly.

1．緒　言

近年の高度情報化社会[1]では，画像処理技術は多くの産業で作業の自動化による省力化・省人化の実現や機械によって作業を行うことで作業標準化と精度向上,高速化など多くの恩恵を与えている.企業の DX[2]の推進などで業務の効率化・最適化が必要である現代の産業には欠かせないものとなっている.本研究は２次元画像処理技術を使い,ジグソーパズルのピースの形状情報に注目した配置アルゴリズムの開発を行う.

本研究における.開発工程のフローを図１に示す.

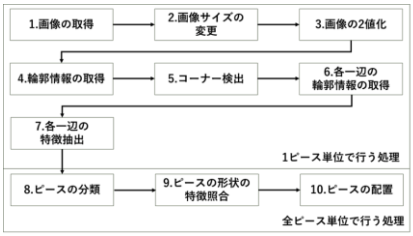

図１　開発工程のフロー

２．配置アルゴリズムの概要

2.1 ジグソーパズルのピースについて

本研究に使用するパズルのピースには３つの条件がある.

1. 同じ形が存在しないこと.
2. ピースの各辺は凸,凹,フラットのいずれかの特徴を持っている.
3. すべてのコーナーはピース画像を９つのエリアに分けたとき,4 隅のエリアに必ず存在している.

2.2 コーナー検出

ピース画像の４隅のエリアにある輪郭点に対し,コーナー検出を行う.エリア内の輪郭点 i から両隣の輪郭点 $i-1, i+1$ へのベクトルのなす角[3]を求め,1 つのエリアで最も鋭角であった輪郭点 i をコーナーとする.輪郭点 i から輪郭点 $i-1$ のベクトルを $\vec{a}$, 輪郭点 i から輪郭点 $i+1$ のベクトルを $\vec{b}$,ベクトルなす角を θ としたものが図２である.

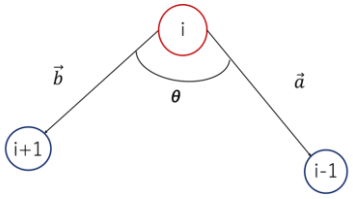

図２　ベクトルのなす角

2.3 ピースの各辺の特徴抽出

本研究で扱うピースの辺の特徴は２つある.

1 つは両端にある２つのコーナーを結ぶ直線と輪郭点の距離[4]である.図３のようにコーナーとコーナーを結ぶ直線を引き,その直線と輪郭点がそれぞれどれだけ離れているかを数値で表す. コーナーを $a(x1, y1)$, $b(x2, y2)$,2 つのコーナーの間にある輪郭点 i の１つを $c(x3, y3)$ とする.点 a から点 b と点 c を結ぶの２つのベクトルを u と v とし, u と v のなす角を θ とする.点 c から直線 ab へ引いた垂線の長さ L を求める.

L はベクトルの外積を使って求めることができる.三角関数の公式より L は $L = |v|sin\theta$,u と v の外積は $u \times v = |u||v|sin\theta$,外積の両辺を $|u|$ で割り, $|v|sin\theta = u \times v/|u|$ となり,このことから $L = u \times v/|u|$ と求めることができる.辺の全ての L を求め,隣り合う L の差分[5]の波形情報(diff)を辺の照合に使用する.

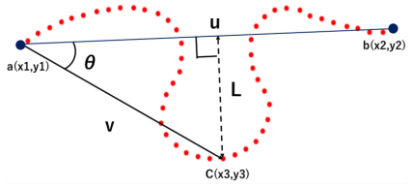

図３　直線と輪郭点の距離

千葉工業大学　社会システム科学部　経営情報科学科
2022 年度卒業研究概要

もう 1 つは各辺が凸,凹,フラットのうち,どの特徴を持っているかである.コーナーとコーナーの中央の座標と辺の中央の輪郭点の座標を比較することで,凸か凹かを判定する.フラットはピースの各辺のLの平均が任意の閾値以下のときに判定する.

2.4 ピースの分類

ピースが持つ 4 つの辺の特徴とその並びに注目して,ピースをグループ分けする.左上のコーナーを始点として,時計回りに辺の順番を考える.

2.5 ピースの配置

任意のフラットを 2 つ持っているピースを図 4 のブロック 1 に入れる. ブロック 1 に配置するピース次第でパズルが(縦,横)=(4,5),(5,4)と変わる.以降のピースの配置は図 4 の番号順であり,.配置を行いたいブロックのフラットの有無と配置を終えていて隣接するブロックの接している辺の特徴からグループを選び,そのグループのピースを候補とする.対応する辺と配置する候補のピースの辺のうち, diff のコサイン類似度[6]が最も高い辺を持つピースをそのブロックに配置する.コサイン類似度は 2 つのベクトルのリスト間の類似度を測定することができる. 2 つのベクトルのなす角のコサイン値のことであり,図 6 なら「0度で,とても似ている」, 0なら「90 度で,似ている／いない,のどちらにも無関係」, −1なら「180度で,似ていない」となる.

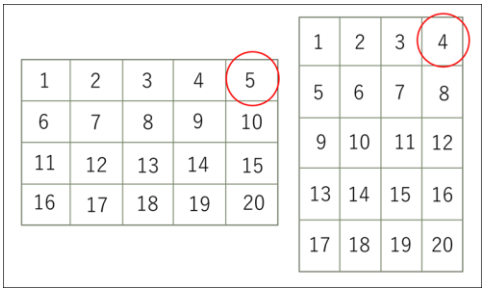

図 4　パズルのブロック分け

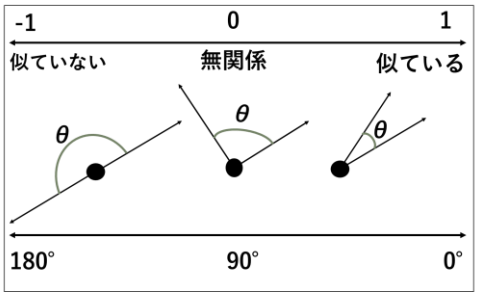

図 5　コサイン類似度

3．結果

配置の結果を図 6 に示す.

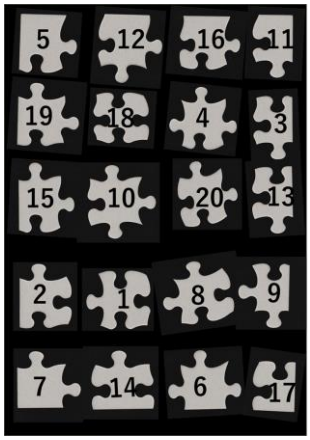

図 6　配置の結果

図 6 の結果,からパズルの組み立てがうまくいくことがわかった.

4．結　言

本研究では, ピースの形状情報を使用したジグソーパズルの配置アルゴリズムを開発した. 20 ピースのパズルでは正しく配置を行えることを示した. 今後の課題は多ピースのジグソーパズルを対象に本研究のアルゴリズムを使用できるように改良を行うことである.ピースの数が増えた影響で誤配置が起きた場合はピースの形状情報と色情報の特徴を組み合わせることで,多ピースのパズルも正しく配置することが可能であると考える.

文　献

[1] 高度情報化社会とは / 現代社会 by John Smith |マナペディア| https://manapedia.jp/text/999 (2023 年 1 月 23 日時点)

[2] DX (デジタルトランスフォーメーション) とは？定義や事例を紹介 https://www.smbc.co.jp/hojin/magazine/planning/about-dx.html(2023 年 1 月 23 日時点)

[3] python-numpy 指定した 3 点間での’なす角’を計算する方法！ https://www.higashisalary.com/entry/numpy-angle-calc(2023 年 1 月 23 日時点)

[4] 2 点を通る直線から離れた位置にある点までの距離を求める . https://tokibito.hatenablog.com/entry/20121227/1356581559 (2023 年 1 月 23 日時点)

[5] numpy.diff()：要素の差分 https://python.atelierkobato.com/numpy-diff/ (2023 年 1 月 23 日時点)

[6] Python NumPy コサイン類似度の求め方 https://qiita.com/Qiitaman/items/fa393d93ce8e61a857b1 (2023 年 1 月 23 日時点)