

カメラを用いた麻雀自動得点計算システムの構築

松井雪治[†]澤野弘明[‡]水野慎士^{††}

愛知工業大学

1 はじめに

本稿では、画像を用いて麻雀の自動得点計算を行うためのシステムについて提案する。麻雀は将棋やリバーシといったボードゲームに比べ、ルールが複雑である。中でも得点計算は、役の種類と翻数(ハンスウ)と呼ばれる役の価値の大きさ、及び符点を合わせた計算を行う必要があるため、初心者がプレイするには敷居が高い。それを支援するためのシステムとして、自動点数計算麻雀卓 [1] がある。これは、IC タグを用いて麻雀牌(以下「牌」)の認識及び点数計算を自動で行うことができ、既に製品化されている。しかし、麻雀卓が重いので持ち運びが難しく、決められた場所でしかプレイすることができず一般的ではない。一方で、自動得点計算を行うための携帯アプリケーション [2] が提案されている。しかし、このアプリケーションでは牌を手動で選択するため、時間と手間がかかる。そこで本研究では、麻雀のあがりの形をカメラで撮影し、自動で牌を認識することで、得点計算を自動に行うシステムを提案する。なお、本システムでは携帯端末に移植することを想定している。

2 麻雀自動得点計算システム

本システムでは、麻雀のあがりの形をカメラで撮影し、テンプレートマッチング法により牌を認識する。認識結果に基づいて得点を計算し、計算結果をプレイヤーに表示する。処理の流れを図1に示す。

初めにカメラを用いてあがりの形を撮影し、その画像を入力画像とする。入力画像から手牌領域を抽出するため、ROI (Region Of Interest) を用いる。入力画像を二値化した画像から輪郭を抽出し、得られた輪郭を構成している点の座標から手牌を包含する最小の矩形をROIとして、その内部を手牌領域画像とする。

そして、予め用意しておいた図2の34種類の牌の画像を読み込み、テンプレート画像とする。テンプレートマッチングを行う際、手牌領域画像内の牌とテンプレート画像の牌のサイズが異なると類似度が低くなり、認識されない場合がある。そこでテンプレート画像として $a \times b$ [pixels] から $c \times d$ [pixels] まで1ピクセルずつ

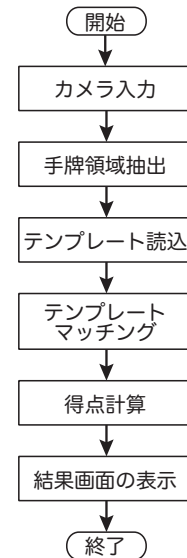


図1: 処理の流れ



図2: テンプレート画像

拡大させた n 枚の画像を用意することでスケール変動に対応し、精度向上を図る。それらの画像を用いてテンプレートマッチングを行い、類似度が最も高い牌を抽出し、手牌として認識する。この処理を繰り返し行い、全ての手牌を認識する。

認識結果に基づき、得点計算を行う。得点計算には、麻雀得点計算のプログラム『麻雀得点計算C++ソース(ver1.0)』[3]を用いる。このプログラムを用いて、手牌として認識された牌の組み合わせから、符点計算及び役の種類と翻数を判別し、得点を計算する。そして、計算結果をプレイヤーに表示する。

A Score Calculation System for Mahjong with a Camera

†Yukinao MATSUI ‡Hiroaki SAWANO ††Shinji MIZUNO
Aichi Institute of Technology



図 3: 入力画像

3 実験と考察

本システムを用いて基本的なあがりの形である牌 14 枚を含んだ画像を入力画像 (図 3) として実験した。入力画像のサイズは 640×480[pixels] とした。そこから手牌領域を抽出し、探索対象 (図 4) とした。テンプレートマッチングの類似度に基づいて、手牌の 14 枚を認識し、得点計算を行った。結果画面を図 5 に示す。牌の認識及び得点計算では、意図した結果が得られた。

今回の実験では基本的なあがりの形である 14 枚の牌を対象としていたが、牌を回転させる場合や牌の枚数が増える場合 (図 6) も考えられる。牌の回転及び増加を考慮した実験では、正確な牌の認識ができず、得点計算を行うことができなかった (図 7)。

実装は、2.4[GHz] Intel Core 2 Duo CPU, Memory 2[GB] の PC 上に、C++ と画像処理ライブラリに OpenCV を使用しており、平均的な処理時間は 3.7[s] であった。牌の回転及び牌の枚数増加や、携帯端末に移植した場合のリソース制約によって、処理時間の増大が考えられるため、処理の高速化を行う必要がある。

4 おわりに

本稿では、牌画像からの自動得点計算を行うためのシステムについて提案した。実験の結果、基本的なあがりの形である 14 枚の牌の認識、及び自動得点計算により意図した結果が得られた。そのため、本システムを使うことにより、得点計算の苦手な人でも麻雀を手軽にプレイでき、かつ円滑なゲーム進行が期待できる。今後の課題として、牌の回転やスケール変動による未認識や誤認識を防ぐロバスト性の確保が挙げられる。



図 4: 手牌領域画像

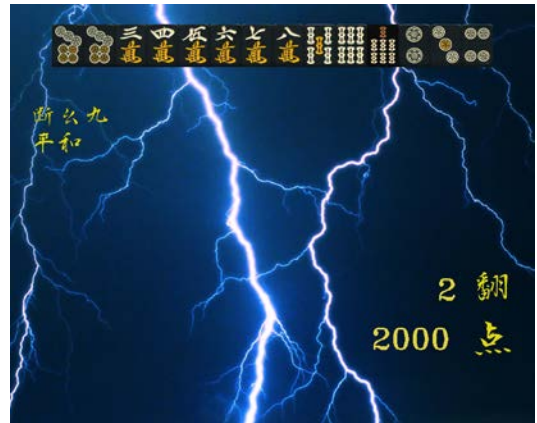


図 5: 結果画面



図 6: 失敗例 (牌の回転及び増加)

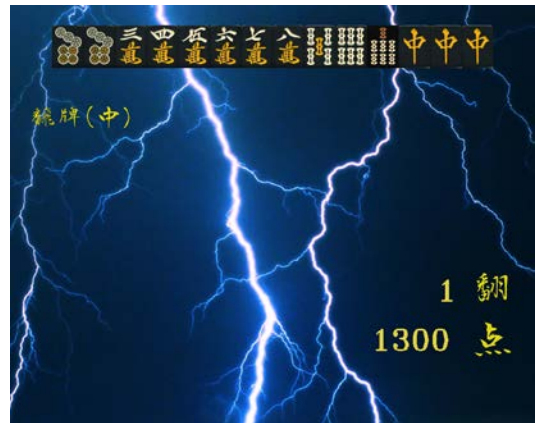


図 7: 失敗例 (結果画面)

参考文献

- [1] 株式会社ジョイス: “自動点数計算麻雀卓「パイレーダー」”
- [2] 株式会社エクスラント: “麻雀これ何点?”
- [3] 麻雀 C 言語プログラム集 『麻雀得点計算 C++ ソース (ver1.0)』 <http://cmj3.web.fc2.com>