

囲碁における盤面識別システムの検討

橋本 燎 (指導教員 井上 優良)

令和 3 年 1 月 dd 日

A Study of System for Extracting Go Stone Positions

RYO HASHIMOTO (ACADEMIC ADVISOR YUSUKE INOUE)

概要：本研究では、コンピュータと現実での囲碁の対局面場との関係をより親密なものにするために、囲碁盤の画像から碁石の配置を識別するシステムを検討する。システムは、指定した四隅の座標をもとに碁盤を切り抜き、グレースケール変換を行った後に、石を設置できる座標の色情報から石の検出を行う。

キーワード：画像処理、射影変換、変更確認用に追記

1. 緒言

囲碁は、2 人のプレイヤーが、「碁石」と呼ばれる白黒の石を、通常 19×19 の格子が描かれた「碁盤」と呼ばれる板へ交互に配置するボードゲームである。他のボードゲームと比較すると、ルール上の制約が極めて少ないといった特徴を持ち、他のボードゲームよりも可能な局面の数は膨大になる (約 $2.081681994 \times 10^{170}$ 通り [1])。

コンピュータの世界においては、囲碁の研究は 1960 年代から行われていた。最も古い研究は、1962 年の Remus による研究 “*Simulation of a Learning Machine for Playing GO*” [2] とされている。1969 年には、Zobrist が世界で初めて囲碁が動作するコンピュータプログラムを作った [3]。2005 年には、モンテカルロ木探索を実装した囲碁プログラム “*Crazy Stone*” [4] が様々な大会で優秀な成績を記録し、以降のコンピュータ囲碁プログラムでも同様のアルゴリズムを採用するなど、コンピュータ囲碁の研究開発に大きな進歩をもたらした。2008 年には、モンテカルロ木探索を採用した囲碁プログラム “*MoGo*” が初めて、公の場でプロ棋士に対して 9 路盤 (9×9 の囲碁盤で行われる対局) で勝利を収めた [5]。2015 年には、Google DeepMind 社が開発した、ニューラルネットワークを実装した以後プログラム “*AlphaGo*” が、初めてプロ棋士に対しハンデ無しの対局で勝利した。コンピュータが人間に打ち勝つのは難しいとされていた分野で勝利を果たしたことは、人工知能の有用性を広く知らしめるものとなった。

本研究では、コンピュータと現実での囲碁の対局をより親密なものにするために、対局中の囲碁の画像から碁石の配置を識別するシステムの検討を行う。具体的には、盤面を含む画像から射影変換を用いて盤面を切り抜き、ノイズ処理を適用して盤面上の線を曖昧なものにした後に、輝度値をもとに碁石の配置を識別するシステムの構

築を行う。その後、複数の画像に対してシステムを試し、結果を考察する。

本論文の構成は次の通りである。第 2 章では先行研究についての説明を行う。第 3 章では実験で用いる射影変換とメディアンフィルタ、しきい値の評価で用いた感度と特異度についての説明を行う。第 4 章ではシステムが行う処理についての説明、具体的には射影変換を利用した画像の変形、メディアンフィルタを用いたノイズ除去、石を置くことができる座標の色情報をもとにした碁石の検出を行う。第 5 章では複数の画像に対してシステムを適用し、結果を考察する。最後に第 6 章で本論文をまとめ、今後の課題について述べる。

2. 先行研究

棋譜の自動生成に関する既存研究に着目すると、1 手前の着手と比較することで碁石の配置を検出する研究がある。芝らは、着手ごとのグレイ画像と 1 手前のグレイ画像との、碁盤領域内の差分をとることで、碁盤上の碁石の位置 (碁石座標) を検出する手法を提案した [7]。

本研究では、直前の着手の情報を使用せず、1 つの着手の画像だけで碁石の位置を検出することを目的とする。

3. 理論

3.1 変換

3.1.1 同次座標

座標 (x, y) に対し、その要素を 1 つ増やした座標 (ξ_1, ξ_2, ξ_3) を、以下の関係式を満たすように定義する。

$$\begin{aligned} x &= \frac{\xi_1}{\xi_3} \\ y &= \frac{\xi_2}{\xi_3} \end{aligned} \quad (3.1.1)$$

ただし、 ξ_1, ξ_2, ξ_3 のうち、少なくとも 1 つは 0 ではない

とする．このように定義される座標を同次座標と呼ぶ [8] ．

同次座標においては， $\lambda \neq 0$ なる任意の λ に対して， (ξ_1, ξ_2, ξ_3) と $(\lambda\xi_1, \lambda\xi_2, \lambda\xi_3)$ は，通常の座標に直したとき，ともに $(\xi_1/\xi_3, \xi_2/\xi_3)$ となるため，同じ点を表している．つまり，同次座標による表現では，定数倍をしても変わらないとみなすことができる．このような表現を同値であるとして，これを式では以下のように表す．

$$\begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} \sim \begin{pmatrix} \lambda\xi_1 \\ \lambda\xi_2 \\ \lambda\xi_3 \end{pmatrix} \quad (3.1.2)$$

ここで，記号 \sim が同値関係を表し，定数倍の違いを許して等しいことを意味する．

3.1.2 射影変換

同次座標を利用することにより，一般的な変換を表現することができる．これは以下の式で表現されるもので，射影変換と呼ばれている．

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.1.3)$$

あるいは，これをベクトルと行列の記号を用いて，以下のように表現することもできる．

$$\vec{x}' \sim H\vec{x} \quad (3.1.4)$$

H は任意の 3×3 の行列である．

式 (3.1.1) の関係を用いて，(3.1.3) から座標 (x', y') を求めると以下ようになる．

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{aligned} \quad (3.1.5)$$

射影変換においては，線分の直線性は保たれるものの，平行性は失われる．別の言い方をすると，任意の四角形を別の任意の四角形に移すような変換であるといえる．

3.2 メディアンフィルタ

領域内の画素値の中央値（メディアン）を出力とするフィルタをメディアンフィルタと呼ぶ．このフィルタは，コントラストの差がある輪郭部分がぼやけにくい特徴を持つ．特定の画像に対して，領域内の画素値の平均を出力するフィルタ（平均化フィルタ）とメディアンフィルタを施した一例を，それぞれ図 3.2.1，図 3.2.2 に示す．

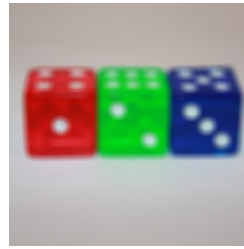


図 3.2.1 平均化フィルタ



図 3.2.2 メディアンフィルタ

3.3 感度

検査で検出したい信号や疾患を有するもののうち，検査が正しく陽性と判断したものの割合を指す．真陽性率（TPF: True Positive Fraction）とも呼ぶ．

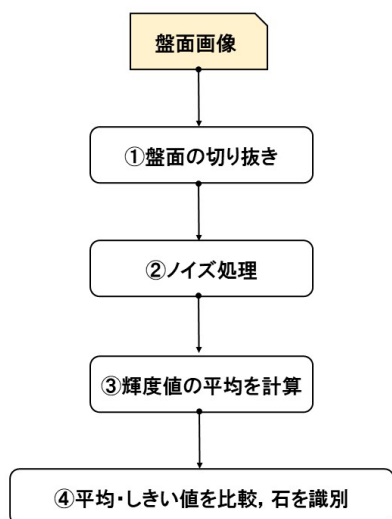


図 4.0.1 フロー図

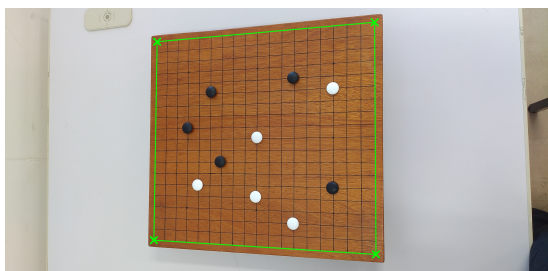


図 4.1.2 射影変換前

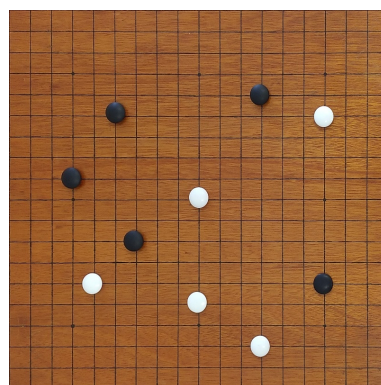


図 4.1.3 射影変換後

3.4 特異度

検査で検出したい信号や疾患を有さないもののうち、検査が正しく陰性と判断したものの割合を指す。真陰性率 (TNF: True Negative Fraction) とも呼ぶ。

4. 盤面識別システム

本研究で検討したシステムでは、盤面を含むグレースケール画像から盤面の切り抜きを行い、ノイズ処理を行った後に、輝度値をもとに石の検出を行う。システムの流れを図 4.0.1 に示す。これらの処理には、オープンソースのコンピュータビジョンライブラリである OpenCV[9] を利用した。

4.1 盤面の切り抜き

はじめに、`getPerspectiveTransform()` 関数を用いて、盤面を含む画像から盤面の四隅の頂点座標と、正方形の頂点座標から、変換行列を生成する。次に、`warpPerspective()` 関数を用いて、画像と変換行列を指定することで射影変換を行い、盤面部分を正方形に変換する。

変換前の図に、正方形に変換する四角形を重ねたものを図 4.1.2 に、変換後の図を図 4.1.3 に示す。

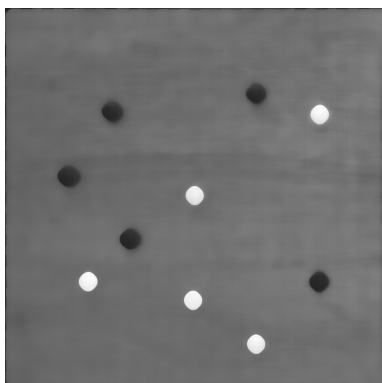


図 4.2.4 ノイズ除去後の盤面

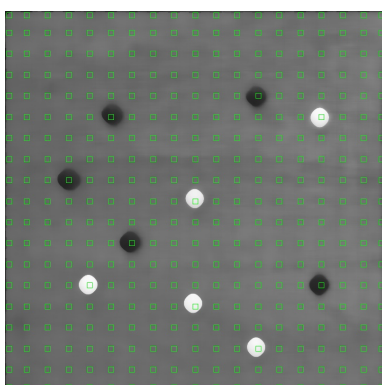


図 4.3.5 領域を可視化した盤面

4.2 ノイズ除去

4.1 章で生成した画像 (図 4.1.3) に対し, 碁盤上の線を消すために, メディアンフィルタ処理を行う `medianBlur()` 関数を適用する. 処理結果を図 4.2.4 に示す.

4.3 領域の付与

4.2 章でノイズ処理を施した画像 (図 4.2.4) に対し, 石が配置可能な座標の画素の情報を得るために, 等間隔に 19×19 個の小さな領域を付与する. 領域を可視化した図を図 4.3.5 に示す.

4.4 識別

4.3 章で与えた領域から, 領域内の輝度値の平均 (0 ~ 255) を取得し, 設定した閾値をもとに領域の座標を「黒石がある」「白石がある」「何も無い」の 3 状態に分類する. 閾値の決定については, 5.1 章で述べる.

5. 実験と考察

この章では, システムの有効性を検証するため, 実際に盤面を含む画像を使ってシステムを適用し, 結果を考察する.

5.1 しきい値の決定

今回の実験で用いるしきい値は, この盤面画像 (図

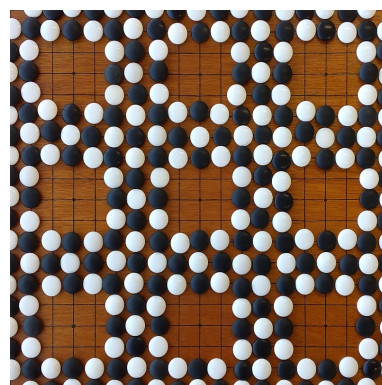


図 5.1.1 閾値の決定に用いた盤面

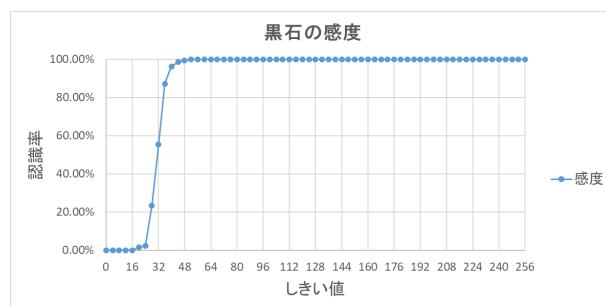


図 5.1.2 黒石の感度

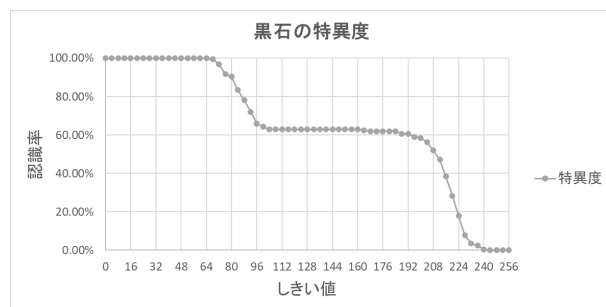


図 5.1.3 黒石の特異度

5.1.1) に対し, さまざまな閾値に応じた感度と特異度をもとに, 最適な閾値を決定した. 黒色のしきい値に対する黒石の感度と特異度のグラフを図 5.1.2, 図 5.1.3 に, 白色のしきい値に対する白石の感度と特異度のグラフをそれぞれ図 5.1.4, 図 5.1.5 に示す.

図 5.1.2 と図 5.1.3 より, 黒石のしきい値は 52 ~ 64 の範囲で感度 100%, 特異度 100% を記録したため, 今回の実験ではその範囲の中央値である 58 を使用した.

図 5.1.4, 図 5.1.5 より, 白石のしきい値は 104 ~ 160 の範囲で感度 100%, 特異度 100% を記録したため, 今回の実験ではその範囲の中央値である 132 を使用した.

5.2 事例 1

正しく取得できた場合について説明

画像を多用してページ数を稼ぎたいところだけど, ただでさえ画像が多いからこれ以上増やすと見栄えが悲惨

なことになるそう.....

5.3 事例 2

異常が多発する (反射・配置ズレが多い) 場合について説明

6. 結言

本文

まとめと、今後の課題とか？

参考文献

- [1] Number of legal Go positions
- [2] Remus, H. : *Simulation of a Learning Machine for Playing Go*, Information Processing, pp. 192-194 (1962).
- [3] Zobrist, A. L.: *A Model of Visual Organisation for the Game of Go*, Proceedings of AFIPS Spring joint Computer Conference, Vol. 34, pp. 103-112 (1969).
- [4] Rémi Coulom 「Crazy Stone」
- [5] 美添一樹 (2008) 「モンテカルロ木探索 コンピュータ囲碁に革命を起こした新手法」, 『情報処理』, 49(6), pp.686-693
- [6] DeepMind 「AlphaGo」
- [7] 芝 浩二郎・古屋 保・西 省吾・森 邦彦 (2006) 「画像処理による囲碁棋譜自動生成システム」, 『電気学会論文誌 C (電子・情報・システム部門誌)』, 126(8), pp.980-989.
- [8] デジタル画像処理 [改訂第二版], 松阪 喜幸 (2020)
- [9] OpenCV

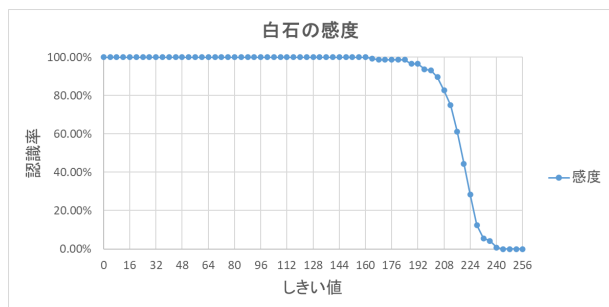


図 5.1.4 白石の感度

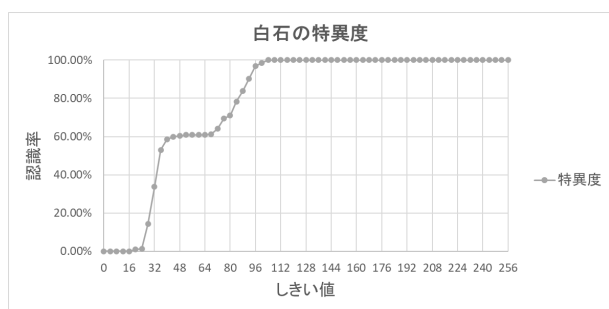


図 5.1.5 白石の特異度