・提出された原稿をそのまま印刷しています。

1 Unityで切り開く明るいプロコン競技部門

宇　部

國弘　颯真(3年) 手嶋ひかる(3年)
中村虎之助(2年) 江原　史朗(教員)

1. はじめに

今回の競技は、早く枠にピースを並べて、パズルを完成させる物である。全ピースの追加情報を活用して、座標データから頂点ごとの角度と、時計回りの辺の長さを抽出しパズルを完成させる。

2. パズルの解法・アルゴリズム

(1) 追加情報からピースデータ(第1 ヒント)を取得し、ピースに優先度を付与する。
外枠の凹凸に着目し、辺の長さが一致するピースの組み合わせを探索する。その組み合わせが見つかれば、枠と結合させる。

(2) (1)の工程を、枠の全ての辺について行う。
(1),(2)の工程で、枠に隣接するピースを埋める。

(3) ピースを2 つ選択し、それぞれのピースの内角の和が 180 度になる箇所を探索する。
2 ヶ所以上見つかった場合は、角と角の間の辺の数が両ピースとも等しいこと、またそれらの辺の長さが一致し、頂点が存在する場合、各頂点の内角の和が 360 度になるならば、2 つのピースはその個所を区切られた1 つのピースとして扱う。

(4) (3)の工程をピースの大きさが最大(ピースの数が最小)になるまで繰り返す。
それぞれのピースは人の目で見て並べられるようにUnity の画面上に描画する。

3. 開発環境

Windows10/8.1/7

Visualstudio2017/2015/2013/2008

Unity

2 パズルDaisuke

明　石

井上　勢大(1年) 前野　聖太(1年)
大谷　直輝(1年) 佐村　敏治(教員)

1. はじめに

ピースのデータ入力、パズル完成の速さと正確さを重視させたい。また、QR コードでの形状座標、位置座標の配布も考慮に入れつつ、複数の端末で並列処理する。

2. ピースの入力方法

ピースの入力はタブレット端末のカメラで読み取る。また、1.5 cmのグリットに対応できるよう、カメラの位置を調整する。また、QR コードを読み取るプログラムも作成し、上手く読み取れない場合は使用する。

3. ピース選択アルゴリズム

「ピースを合わせて 180° になる角」や「ピースの同じ長さの辺」となるピースを選択する(図を参照)。相性の良い図形(枠)はそれ自体を1つのピースとして扱い、更に相性の良いものを選択する。以上の処理を繰り返す。

4. ピース組み合わせアルゴリズム

4.1 逐次報告型

相性の良い図形、枠の組み合わせを見つけ、より多くのピースが入る解法を逐次出力する。ピースは中心から埋めていく。

4.2 探索型

4.1 の方法からピースの組み合わせ傾向や位置などを参考にしつつ、パズルが完全に埋まるまで探索を繰り返す。

4.3 機械学習型

事前にピースの組み合わせを学習させてそれを利用して、パズルを埋めていく。

5. 環境

言語：python/ライブラリ：OpenCV

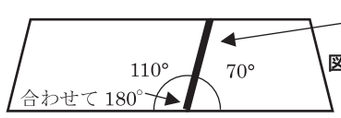


図:ピースの組み合わせ例

3 快速カケライナー競プロ方面・わく行き

呉

今村　圭(4年) 長松　幹太(3年)
水本　直希(2年) 藤井　敏則(教員)

1. システム概要

問題の QR コードをカメラで読み取り、ピースの形を取得しデータとして取得する。ピースを配置するアルゴリズムを用いたプログラムを実行しグラフィックで解を表示する。求めた解のように実物のピースを配置する。

2. データの取り扱い

2.1 取得

Web カメラなどを使い QR コードの画像の取得をし、ピースの形状情報を得る。得られたデータを保存する。

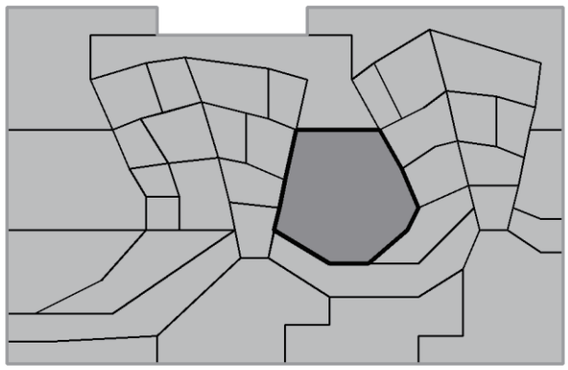
3. 機能

3.1 機能

自動配置、自動マージなど

3.2 GUI

並び終えたピースの情報を画像で表示するようにして、コンピュータで得られた解から現物のピースを探しやすくする。



4. 開発環境

言語 ： C/C++,C#

IDE ： Visual Studio

LIB :Qt,OpenCV

4 解になりたいPUZZLIUM

新居浜

井上　遼(5年) 森野　誠也(5年)
高岡　康平(5年) 占部　弘治(教員)

1. システム概要

配布されたピースと枠を Web カメラで撮影し、その画像を元にピースと枠のデータを生成する。生成したピースデータを枠データ内に配置し、その途中経過を逐次出力する。競技者は、この途中経過、または配置完了後の画像を用いて、配布されたピースを配置する。

2. 画像処理

撮影した画像から、ガウシアンフィルタでノイズ除去し、R1/4,G1/2,B1/4 の重みでグレースケール化。その後、分散共分散行列の固有値を求め、頂点を検出。最後に、回転させて、最もグリッドに近い置き方を求める。

3. 解法

枠の図形をなす角の 1 つに注目し、その角と同じ角度の角を持つピースをピース群から検索する。このとき複数のピースが該当するならば、その角をなす 2 辺の長さが一致するという条件を追加し、ピースを厳選する。これでも複数のピースが該当するならば、さらに条件を追加する。このように、枠のある角に対し、条件に最も合ったピースを検索し、そのピースを枠に配置することを繰り返し、全ピースの配置を狙う。また、ピースの配置順を記録していき、間違ったピース選択をしてしまった場合には、その箇所まで処理を戻すことができるようにする。

4. 開発環境

言語: C++

IDE: Visual C++2015

ライブラリ: OpenCV

5 パズルフレンズ

熊　本（熊本）
川俣　大喜(4年) 森川みどり(4年)
末永　和大(3年) 孫　　寧平(教員)

1. はじめに

今回のパズルも前回同様画像処理とパズルを解く 2 つの部分に分けて考えることができる.

画像処理については, ヒントを利用して確実にピースの形状情報を取得し, 次の処理につなげる.

パズルの解析には, 全探索とビームサーチを利用して両方を 2 つのパソコンを利用して, 様々な場面を想定する.

2. パズル解法

全探索とビームサーチを利用する理由については, 例えばピース数が少ない場合全探索が先に処理を終了する可能性が存在し, ピース数が多ければビームサーチを利用する方が早く処理を終了することができると想定する.

3. ビジュアライザ

ビジュアライザには今回工夫を施した(図 1).

正常に解析が進んでいるのか判断できるように GUI に進捗状況を表示. また, ピースを組み立てる場合, 枠の位置は変えられないため, できる限り早くピースを並べるため, ピースの形状と番号を表示する支援方法を考えた.

この GUI で全探索とビームサーチどちらかを選択できるようプログラムを作成し, 2 つのパソコンで処理を行う.

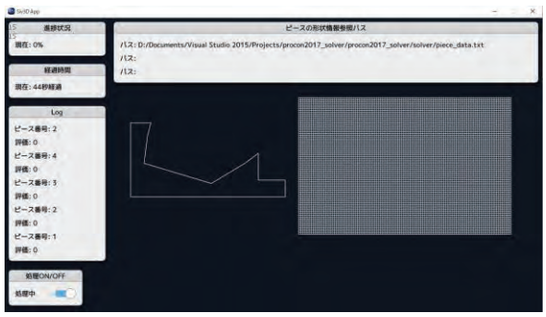


図 1　現在開発中の GUI

4. 開発環境

C++, Visual Studio 2015, Siv3D, Windows10 など

6 再試:高専の情報処理パズル基礎 I

長　野
清水　勇太(3年) 池上　蒔典(3年)
芳賀　七海(2年) 鈴木　　宏(教員)

1. システム概要

ピースの形状情報を用いて、ピース同士を結合しながらパズルを解く。システムには GUI を実装し、これを操作しながら、視覚的に、かつ素早くパズルを完成させる。また、Raspberry Pi をサーバとし、PC 間のデータ中継を行う。

当プログラムは、全ての場合における、有限時間内での完全解導出の保証を持たない。したがって、解答開始時に問題の性質をよく見極めた上で、プログラムによる導出か、人による導出かを選択する。

2. データの扱い

精度・要する時間などの観点から、画像認識は行わずに、ピースの形状情報を全て利用する。

得られた形状情報を、頂点を結ぶベクトルの集合として、考えうる裏返し・回転を含め扱う。

3. パズルを解く

小さな唯一解を組み合わせていくことでピースの数を減らしていき、最終的に一つの大きな唯一解を導く。

動的計画法を用いて、取りうる結合を計算し、ピースの結合候補を大幅に絞りながら探索を行う。

結合候補に挙がったピースを合体させて、できあがったピースの結合候補を探索する。この作業を繰り返し、解を完全なものへと近づける。

4. 開発環境

OS：Windows, Linux mint

言語：C/C++, Python, Common Lisp 等

ライブラリ：Siv3D

IDE：Visual Studio 2015

7 枠の中に、ピースが、ある。解きにいくぉ!アッこの探索、深いッ

熊　本（八代）
扇塚　和希（2年）村上　史高（4年）
西﨑　友輔（4年）小島　俊輔（教員）

1.　はじめに

今回の問題では，実物で与えられたピースを数値化する．そのデータから，ピースの各頂点が必ずグリッド点に存在することに焦点を当てたアルゴリズムと，ヒントを入力するためのプログラム，解を出力するためのプログラムから構成される．

2.　画像処理

まず，スキャナーを用いて各ピースの画像を取得する．取得した画像を元に OpenCV ライブラリを用いて各頂点の座標を求める．

3.　アルゴリズム

格納された各頂点を回転させることによって，全て頂点がグリッド点に存在する向きを求め，各頂点の dx，dy をハッシュテーブルに格納する．dx，dyから各辺の傾き・長さを O(1)を求め，深さ優先探索によって傾き・長さの順で一致するピースを組み合わせていく．ただし，全体解を求めようとすると計算量が莫大になってしまう可能性があるため，部分解を求め，手動で組み合わせて最終的に全体解を求める．

4.　ヒントの入力

アルゴリズムにおいて探索が難しくなってきた場合，ヒントを活用する．QR コードをウェブカメラで読み込むことによってピースの形状情報・位置情報を取得し，アルゴリズムに渡す．渡されたピース情報を中心に探索を再開させる．

5.　解の出力

アルゴリズムによって求めた解を，各頂点の座標を直線で結ぶことによって出力する．その際，対応したピース画像も同時に表示することにより，ピースの場所を把握しやすくする．

6.　開発環境

OS　：　Windows10，OS X
IDE　：　Visual Studio 2015 / 2017，Xcode
言語・ライブラリ　：　C++，OpenCV 3.3

8 さぁ!次は101×65グリッドよ!もうやめてお姉さーん（泣）

米　子
寺西　勇裕（3年）梶本　翔（2年）
山本　一樹（2年）徳光　政弘（教員）

1. はじめに

今回の問題は、ピースの入力、パズルの計算、実際にピースをはめる、といった 3 つの処理に分けることができる。

2. ピースの入力

減点を避けるため、QR コードは使用しない。黒いマットの上にピースをばらまき、カメラで撮影して判別分析法により二値化する。今回はピースの頂点がグリッド状に位置するため、グリッド点が最も近い点をピースの頂点とする。このようにすることで、入力誤差が大幅に小さくできる。

3. パズルの計算

ピースはわくの頂点に対してすべて調べる。枠の辺・角・面積に対して、どのピースをはめればよいかは部分和問題として解くことである程度絞ることができる。また、存在することのできない閉領域が生じた場合に枝刈りを行う。この要領で探索していく。

4. 実際にピースをはめる

枠にどの位置にどのような向きでどのピースをはめる、というのはとてもわかりにくい。そこで、ピースと枠それぞれにプロジェクションマッピングを行い、どのようにはめるのかを直感的にわかるようにした。

5. 開発環境

言語：C++
IDE：Visual Studio 2017
ライブラリ：OpenCV

9 grow grass

秋田
石川亜留都(5年) 松渕 勇真(4年)
熊谷 有莉(4年) 竹下 大樹(教員)

1. はじめに

実物のピースを撮影しデータ化することや、すべて人力でやることは時間がかかり、パズルを完成できる可能性が低い。そこで、形状情報を用いてプログラムでパズルを解いていく。

2. データの入力

カメラでQRコードを撮影し、データ化する。

3. パズルの解法

3.1 パズルのシミュレート

プログラム上で回転/反転したピースを置いたときに枠や配置済みのピースと重ならないか判断したのち、そのピースを配置する。

3.2 アルゴリズム

枠の一つの角に対して、ピースの配置可能なパターンを全て配置する。続けてそのピースに隣接するよう同様に全パターンを配置し、これを繰り返す。枠に沿って一周するようなピースの置き方を探索し、その後内側を埋める。

3.3 評価関数

前述したような幅優先探索では計算量が膨大になり制限時間に解答できないため、評価関数を用いて、枝切り、もしくは探索の優先度の決定を行う。

4. 完成図の出力

カメラで回答台を撮影しながらARを用いて枠に合わせて完成図を表示する。

5. 開発環境

言語 : C++ / C#

IDE : Visual Studio

OpenCV / Unity / ZXing

10 大きな島の愉快な仲間たち

大島商船
今津 拓哉(5年) 秋山 誠賀(5年)
藤川 光浩(3年) 北風 裕教(教員)

システム概要

本システムは，スキャナで「わく」と「ピース」を取り込み，「わく」「ピース」の頂点検出する抽出部、検出された頂点をもとに解を求める探索部、および探索結果を元にGUIに出力する出力部の３つから成る。

1. 各システムの概要

1.1. 抽出部

抽出部では、競技開始と同時に持ち込んだスキャナに、配布された「ピース」と「わく」をすべて並べてスキャンする。その後、頂点の検出を行い、同時に線分検出も行う。抽出された頂点と線分を用いて多角形の辺のつながりを確定させる。最後に、グリッド点上にすべての頂点を重ねる。また、ピースの抽出画像を図１に示す。

1.2. 探索部

①抽出部によって得られた座標をもとに、「ピース」ごとのグリッド上であることを保つ回転となりうる辺の傾きを計算によって求める。

②内枠の頂点と辺を一つ選択し、その辺と同じ傾きの辺を回転で持ちうる「ピース」をすべて取得する。

③ ②で取得したピースを部分和問題や枠と共有する辺の数などで評価をして並べる。

④ ③で並べた「ピース」を新たな内枠とし、②から④を繰り返す。

1.3. 出力部

探索部で求められた解を、人間がわかりやすく、また、「わく」にはめやすいようにGUOを用いて出力する。

2. 開発環境

OS: Windows 10 / IDE: VS2017 / Library: Opencv3.1

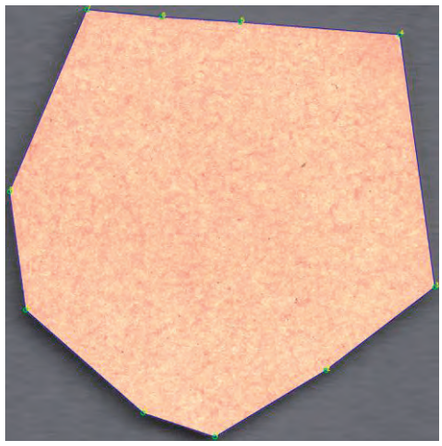


図1. ピース抽出画像

11 パズルソルバー

鳥羽商船

出口　綾音(2年) 土屋　大地(2年)
土田　隼之(教員)

1. はじめに

今回の競技はピースデータの入力とパズル完成の速さ、正確さを競うものである。

2. 枠・ピースの入力方法

枠・ピース情報を QR コードで読込み、空き領域に 0、パズル存在領域に素数(ピース毎に異なる素数、図 1 では 3)を格納する。

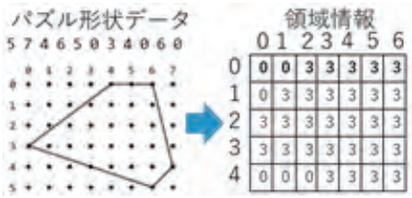


図 1 ピースの内部表現例

3. パズルの組立アルゴリズム

ピースと枠の内部表現データを重ね合わせてピースをはめることが可能かを確認する。グリッド内のピース同士の重なりはピース素数の掛合わせで表現し、ピースはめこみ処理は枠の内側左寄りの上辺から順に行う。全てのピースをはめるまでこの処理を繰り返す。複数ピースをはめることが可能な場合には分岐ポイントとして記録し、後段処理ではまらないピースが発生した際には、記録しておいた箇所でのピース選択を変更して処理を再実行する。

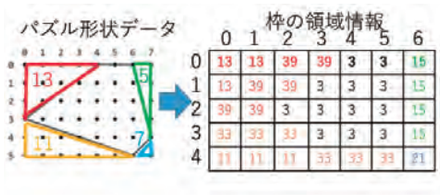


図 2　ピースはめこみの処理例

12 ダークホース (大穴とは言ってない)

鶴　岡

土田　陽平(4年) 成田　祐貴(4年)
加藤健太郎(教員)

1. はじめに

今回の競技は、ピースの画像処理による取得、または、実際に競技を行っている際に受け取れるピースの形状情報・配置情報から得られたピースの情報から、プログラムを用いてパズルを組み立て、それを実際のピースで組み立てる競技である。

2. ピースのデータ取得方法

タブレットまたはスマートフォンにより撮影したピース画像を OpenCV および python を用いて、2 値化を行い、その後、ピースのエッジ検出を行う。そして最終的に座標を取得する。

3. 解法について

今回配布されたサンプルピースではピースの座標により、ピースの辺の長さを算出し、それぞれのピースの辺を比較し、2 つ以上の同じ辺の長さを持つピース同士を探し、組み合わせていく。2 つ以上の同じ辺のピースがなくなった場合は同じ辺の長さが 1 つだけのピースの組み合わせを探し、その組み合わせもなくなった場合、組み合わせた 2 つのピースの辺の合計の長さと同じ辺の長さを持つピースを探し、組み合わせていく。ピース同士の組み合せがなくなった場合、ほぼパズルが完成に近づいているので、手作業でパズルを完成させていく。

4. 開発環境

Python3.6.1

Anaconda4.4.0

OpenCV3.1.0

13 パズル&コレクション ～パ･ズ･コ･レ～

徳　山
黒木　駿矢(3年) 田中　祐豪(3年)
杉本　　航(2年) 力　　規晃(教員)

1. システム概要

本システムは、大きく分けてピースのデータ化、探索、及びパズルの組み立ての 3 段階に分かれている。

2. ピースのデータ化

スキャナを利用し、ピースの画像をスキャンする。スキャンした画像から辺の線分情報を得ることで、頂点と頂点の角度を求める。 画像から得られたピースの情報が正確でないと思われる場合は、配布された形状情報を利用する。

3. 探索

評価値を利用した最良優先探索を行う。評価値は、角度と辺を利用して求める。この際、角度毎や長さ毎に分別しておくことで高速化を図る。 探索が一定時間以上終わらない場合、追加情報を随時利用していく。

4. パズルの組み立て

スキャンしたピースの画像を利用し、各ピースに識別番号を振る。その後、識別番号とピースを重ねて GUI で表示する。 探索終了した際は、探索結果の配置図と識別番号を図 1 のように GUI で表示する。

最終的に、識別番号を比較し、枠にピースを埋めていく。

図 1 探索結果の表示

5. 開発環境

以下の環境で開発を行う。

- 言語: Java
- IDE: Eclipse / NetBeans
- ライブラリ: OpenCV

14 ぱずふぃっと

津　山
小橋　　優(3年) 天野　拓海(4年)
三宅　智也(2年) 川波　弘道(教員)

1.我々のシステム

枠に対し各ピースの評価点を生成する。
例えば,角または辺が等しい場合には評価点を上げ,等しくない場合には評価点を下げる。

評価点の高いピースを優先的に嵌めていくことで,より迅速に解答を探索することが可能になる。また,複数のピース同士を結合させ,一つのピースとして扱うことによって全体の試行回数を減らすことが可能になる。

2.解法の手順

2.1 パズルの情報化

スキャナーによって各ピースの情報を読み取り,形状情報に生成する。

2.2 解法の手順

各ピース同士に関しても評価点を作成する。ピース同士の評価点を用いて,複数のピースを結合し,一つのピースとして扱う。

形状情報を用いて,枠に対し各ピースの評価点を生成する。生成した形状情報を用いてピースを嵌めていき,解答を生成する。

2.3 解答の保存

生成した解答を画像として保存する。その解答の画像を参考に解答台のピースを嵌めていく。

3. 開発環境

- VisualStudio2015Community
- OpenCV_Version2.4.17
- Github
- zbar
- Windows 7,10

15 おいでました、魅力のない県から

茨　城
大沢　武流(4年) 小坂部恭輔(3年)
佐藤隆太郎(専1) 安細　　勉(教員)

1. はじめに

わくやピース、QR コードを画像データにするために A4 スキャナを使用する。

2. 解法の手順

2.1 画像処理

二値化処理を施して輪郭座標を求め、曲率の高い要素を頂点座標とみなす。検出した頂点座標はグリッド上に存在するように補正し、誤検出は GUI で修正する。

データに誤差がある場合、形状情報を利用する。

2.2 探索

幅優先探索を用いるが、多角形の構成要素から評価値を計算して枝刈りをする。多角形同士の演算は時間がかかるため、衝突判定の事前計算を行う。

予め定めた時間を超えた場合、配置情報を利用する。

3. 回答方法

図 1 に、処理結果が表示された GUI の例を示す。

GUI に完成したパズルを表示し、それを見てピースを並べる。この作業を効率よく行うために、実物のピースとデータ上のピースを対応付ける番号を振る。

（A）頂点の検出結果　　（B）解の探索結果

図 1　GUI

16 探索が深いことを事前に察したボーちゃん

阿　南
宮川　大樹(5年) 橋本　綾斗(3年)
渡部　悠真(3年) 平山　　基(教員)

1. システム概要

ピースの写真を撮り、画像処理で頂点座標データを取得する。それを使ってパズルの解を探索し、探索の情報は逐一 GUI 出力することで人による解答提出をサポートする。

2. ピースのデータ化

写真撮影の際は、複数のピースをまとめて写真を撮る。画像処理については、まずノイズ除去をし、次に島探しの再帰アルゴリズムを使用してピースごとに画像を切り分ける。そして頂点検出を行った後、1 グリッドの画像上の長さを用いてその頂点座標に補正をかけることでデータ化を完了する。

3. 解の探索アルゴリズム

3.1 ピースの並べ方

探索途中でのピースの仮置きの方法と条件を次に示す。各ピースを図形として、90° ごとの回転、反転、平行移動を行い、角度、辺の長さのいずれかについて以下の条件を満たすとき、仮置きを行う。角度については、1 つないし複数のピースの角(かど)がなす角度が、探索途中でピースを置いていない領域の持つ角と一致する場合、辺の長さについては、1 つのピース、ないし複数のピースの集合体の辺の長さが探索途中でピースを置いていない領域の持つ辺の長さと等しい場合、それらのピースを仮置きする。しかし、当然ピース同士が重なったり、枠からはみ出したりするようなときは候補として除外する。

上記の判断条件を満たす組み合わせを探索する。なお、探索方法としては深さ優先探索を用いる。

3.2 枝刈り

探索する組み合わせ数を削減するため、ピースの回転を 90° に制限し、各辺の長さが全て等しいと判断したピースにおいては反転の動作は行わない。

4. パズルの GUI 出力

テキストファイルに随時書き込まれていく、探索途中または探索が完了したパズルの座標データを読み込み、GUI に出力する。人間はこの GUI をもとに、枠にピースを設置する。GUI は、テキストファイルのタイムスタンプが更新されるたびに自動的に更新されるため、パズルが組まれていく様子がリアルタイムで確認できる。

17 Sixth sense is solution.

福　井
山本　雄太(4年) 岡部　勇希(4年)
武村　航平(4年) 村田　知也(教員)

1. 概要

本競技は、実物のピース・枠からその形状情報を取得し、プログラムを用いて解く競技である。本システムでは、パズルを解く処理を「入力」「処理」「出力」の３つに分けて行う。

2. 入力：画像処理について

スキャナよりピース・枠のモノクロ画像を取得、ここから「OpenCV」を用いて頂点座標を求め、ヒントとして与えられる QR コードと同様のフォーマットでテキストファイルに出力する。

3. 処理：解法アルゴリズムについて

得られたピース、枠の情報をもとにピースに優先順位を付けながら探索を行う。角度、大きさをもとに複数のピースを仮想的に合成し、ピース数を減らすことで探索数の削減を狙う。

4. 出力：UI について

得られたピースの配置情報を元に、プログラムから結果をグラフィックで表現、パソコンのディスプレイに出力する。

5. 開発環境

Visual Studio 2017

- C++版 OpenCV 3.1.0

18 ヒラメキパズル ～OTAの不思議なノート～

一　関
太田　拓海(5年) 佐藤　一輝(5年)
永原　基也(4年) 管　　隆寿(教員)

1. 概要

システムは、Web カメラを使用し追加情報を読み込む「QR デコーダ」、解を求める「ソルバー」、読み取ったピースの識別（形状と実物のピースとの対応付け）や、ソルバーからの回答を表示し、ピースの配置を支援する「ビジュアライザ」から構成される。

ピースとわくの形状は、与えられる形状情報を利用して取得することを前提としている。

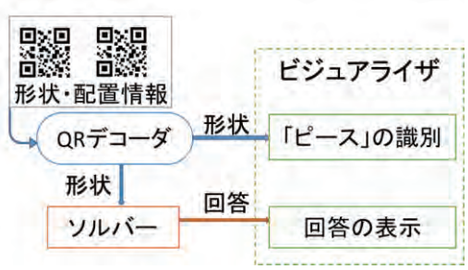


図 1　システム構成

2. 解法

① わくから１つの頂点を選択する。
② 辺の傾きが一致するピースを配置候補とする。
③ １つ以上のピースを組み合わせて、選択した頂点に収まりかつ他のピースやわくに衝突しないように配置する。
④ 以上の操作を繰り返し行いすべてのピースが配置できたケースを解とする。

3. 開発環境

言語：C++、C#
IDE：Visual Studio、Qt Creator
ライブラリ：OpenCV、ZBar

19 しゅらのくにちほー

北九州

上田健太郎(3年) 豊田　誠弥(3年)
都甲　真仁(2年) 松久保　潤(教員)

1. システム概要

本システムは，パズルデータ格納，配置アルゴリズム，GIU を利用したパズル描画からなる．

2. パズルのデジタルデータ化方法

各辺の始点と終点をデータとして格納する．その後，アルゴリズムで解き始める前に以下のデータをあらかじめ計算し格納しておく．これにより，アルゴリズムにて解く際に計算量を減らすことができる．

・各頂点の角度
・考えられる回転の角度での回転後の座標
・上下左右反転後の座標
・各辺の長さ

3. 配置アルゴリズム

枠の中の未設置部のうち，左上の頂点に合わせてピースを設置し当たり判定を行う．設置が可能な場合は続けて設置，不可能な場合は回転や反転の組み合わせを確かめ，それでも不可能であれば別のピースで確かめる．これの繰り返しでは総当たりとなり計算時間がかかるため，設置により発生した辺や角の評価等，工夫し枝切りを行う．

4. GUI を利用したパズル描画

図のように，各ピースに番号を振り分けて回答表示を行うことで手動での組み立てを支援する．

5. 開発環境

C++, Siv3D, Visual Studio 2015, Windows10

20 冴えないコードの育てかた#

群　馬

砂賀　開晴(2年) 時澤亮一郎(2年)
清水　怜央(2年) 木村　真也(教員)

1. はじめに

今年も去年に引き続き、どれだけ早く「わく」に「ピース」を収めるかを競う競技である。システムは「ピース」を認識する入力部と、アルゴリズムにより「ピース」を「わく」に当てはめる処理部、そして結果を随時表示する出力部の３つによって構成される。それぞれについて解説する。

2. 入力

Web カメラでピースの頂点を抽出する。カメラ認識のみでは精度が足らず完全に認識することができないため、人力で修正する。

3. 処理

「ピース」の辺の長さ及び頂点の角度から、合体できると思われる「ピース」の組を選び出し、合体させる。合体を繰り返して大きな「ピース」のまとまりを複数作る。それらを「わく」にはめる。

4. 出力

3 の処理を実行しながら現在の「ピース」の状態を随時表示し、いつでも「わく」にはめ始めることができるようにする。完成時には「わく」及び「ピース」を全て正しい位置に表示する。

5. 開発環境

【開発言語】C++/C#/Python

【開発環境】Microsoft Windows 10/Mac OS・Visual Studio2017

【ライブラリ】OpenCV

21 じょーほーふれんず

函　館
織田　智矢(5年) 布施　詠政(5年)
安部　龍馬(5年) 小山　慎哉(教員)

1. アプローチ

(1)初めに、コーナー検出を行ってコーナーを特定し、すべてのコーナーからの差が最小になるようにグリッドを合わせる。そして画像から特定したコーナーを修正する。これによって正確に「すべてのピースのなす角」「コーナー間の辺の長さ」が求められる。

(2)枠の隅（左上から時計回りに各枠の隅を探索）の角度に合うピースを探索する。このとき角度にあうピースの組み合わせすべてを求める。そしてその状態を評価関数で評価し、上位の結果を保存しておく。

(3)すべての枠の隅のピースが決まったら、その隅から枠にそっての長さに合うピースの組み合わせをすべて探索し、実際に組み合わせて評価関数で評価する。

(4)(3)でピースが確定した場合はそのピースを枠画像に加算処理し、新たな内枠として(2)の手順へ戻る。

(5)残りのパーツがなくなった状態を解とする。この時残りパーツが余り、パズルが完全に組み上げられていない場合は手順を戻し再計算できるようにする。

2. 回答の表現方法

1．でも述べたとおり、現在の状態を動的に画面表示していき、それをもとに実際にパズルを組んでみる。 画面表示する際には最初に自動的に振った番号をピースの画像の上に表示しわかりやすくする。

22 ピースチャンプルー

沖　縄
垣花　　周(3年) 高良　昇吾(2年)
比嘉　陸人(2年) 正木　忠勝(教員)

1. はじめに

今回の問題は完全回答，全てのピースを埋めなければ点数を得ることが出来ない．よって，全探索をベースとした解法を使い，時と場合によってヒントを駆使し，完全回答を目指す戦法を採用した．

2. QR コードのデコード

ヒントを使用するために，Web カメラで QR コードを撮影しデコードを行う．撮影に OpenCV を使用し，デコードには ZBar を使用した．また，言語は高速な C++を採用した．

3. パズルの解法アルゴリズム

全てのピースに対し，置くことの出来るパターンを列挙する全探索をベースに，適宜ヒントを使い完全回答を求める．しかし，全探索だと競技時間内に回答が求まらないため，適切な枝刈りを行うことで時間を短縮する．例えば，全てのピースの座標はグリッド点上にあることから，あるピースの回転した図形が限られた数だけ存在することが分かる．また，ピースのある角の角度や辺の長さに基づいて探索を行う．

4. GUI の機能

パズルソルバから受け取った情報を可視化し，GUI 上にどのような結果になったかを表示出来るようにした．ピースを並べる際，人間が今並べようとしているピースを画像認識によって，ピースを読み取りどの位置に置くべきかを分かるようにし，ピース配置の高速化を行う．

5. 開発環境・動作環境

[OS] Mac OS Sierra,Windows10

[Language] C++,Java,Python3

[IDE・Editor] Vim,Visual Studio 2017,IntelliJ IDEA

[Library] OpenCV,ZBar,Boost,OpenSiv3D

23 グリッドタスカル

広島商船

大坪　尚希(4年) 徳満　悠太(4年)
中島　友喜(4年) 大高　洸輝(教員)

1. 概要

システムを入力部、探索部、出力部の 3 部門に分ける。特徴として、探索部では、隣り合うピースを探索する過程で統計データを利用する。また、出力部においては、マウス操作によるピース配置を補助する情報の提示を行う。

2. システムの構成

2.1 入力部

形状情報を読み取り、ピースの各辺の長さと角度を計算する。読み取った情報を基に、わくとピースの形状を出力部で描画しやすいように編集する。また、配置情報を利用する際は、その読み取りと編集を行う。

2.2 探索部

探索部では、ピースの各コーナーの角度と長さに加え、統計データを利用して隣接するピースの候補を列挙する。統計データとして、隣接するピース同士のコーナー角度や辺の長さを分類したものを用いる。このデータを基に、隣り合う傾向にある条件のピースに優先度をつける。選択されたピースのコーナーの角度および辺の長さ、統計データによる優先度を参考に、隣接するピースを探索する。

2.3 出力部

入力部および探索部において、編集、探索された、わくやピースのデータをユーザに提示する。また、マウス操作でピースを配置する際の補助機能として、配置してあるピースの角を選択すると、探索部で得られた隣り合うピースの候補を出力する。その候補とともに、統計データで得られた各ピースの優先度と各コーナーの角度、各辺の長さを提示し、ピース配置作業を補助する。

3. 開発環境

Processing

Open CV

24 SAME

仙　台（広瀬）

佐々木結大(4年) 酒井　玲弥(2年)
五十嵐　覚(2年) 園田　　潤(教員)

1. はじめに

今回の競技は、用意されているパズルの数値データを利用して、パズルの組み立てを行う競技である。

パズルデータを用いずに挑戦する場合、組み合わせが爆発的に増加して解くことが非常に難しくなると考えられるので、データを活用したパズルソルバーの設計を考える。

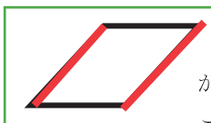
2. QR コードの読み込み

OpenCV というライブラリを利用して Web カメラから QR コードの読み込みを行う。

3. パズルソルバーのアルゴリズム

基本となる考え方は全探索だが、同じ傾きの辺のみが解としてくっつき得ることから効率的な枝狩りを行う。

辺の傾きや配置を記録した辞書を用いて、可能性のないピースの組み合わせを早期に排除する。



ほかの図形に赤い辺と同じ傾きの辺が存在しない場合、これ以上探索してもこの辺を消すことができない。

4. パズルソルバーのアルゴリズム以外の工夫点

C++ AMP による並列処理や複数コアの利用、SIMD などを利用した高速化を行う。

5. パズル組み立て用 UI

QR コードの読み取りとパズルソルバーのリアルタイムな連携を行うユーザインタフェースを製作し、問題の難易度などに合わせて操作する。

6. 開発環境

Visual Studio 2017/C++/Boost/OpenCV/C++ AMP

25 神戸高専電算部

神戸市立
高田　直希(4年) 山田　峻矢(4年)
石原　　実(2年) 朝倉　義裕(教員)

1. パズル完成の実現方法

1.1 パズルのデータ化

カメラを用いて取り込んだ画像を２値化し，各ピースに識別番号を与え，頂点角度・辺の長さをデータとしてデータ化する．その際にはグリッドを活用する事でノイズの影響を減らす事を考えている．

一定面積以下のピースの数，凸四辺形などの単純なピースの数など，探索に不利と思われるパラメータが閾値を超えた際，追加情報を低レベルから順次用いる．

1.2 パズルの解法探索

各ピースの角の角度を用いてグリッドとのマッチングを行う．各ピースと 2.5㎜ 間隔のグリッドを同サイズで取り込み，あるピースのすべての角がグリッド上に存在するかどうかでピースの回転状態と反転を含めた８状態まで絞り込む．この操作により，無限ともいえるピースの回転状態を，大幅に絞ることが期待される．

鈍角を持つ，グリッドに対する位相が特異である辺を持つ，などのユニークなパーツに一定の重みを与え，優先的に探索する．

また，角同士のマッチングを行う．鈍角と，その鈍角に組み合わせて 360° となるような鋭角を探索する．候補が複数存在する場合，辺長などで優先度を決める．

さらに，辺同士のマッチングを行う．長さを使い探索を行い，辺の結合に関しては，グリッドを２次元平面とみて，同位相の辺の集合を得る手法を使う事を考えている．

結合が確定したピースには，新たな番号を与え，データを再度取得し，また探索を再帰的に行うアルゴリズムを行う．

追加情報を用いる際は，探索中にリアルタイムで入力できるようにし，手動で解法が見つかった等の状況の変化に適応できるようにする．

また，測定誤差も考慮し，一定の閾値を定めて誤差による誤検知を減らす事を考えている．また，アルゴリズムに沿った上で誤答が発生することや，追加情報使用時点で解が追加情報と不整合であることも考えられるので探索の巻き戻しもできるようにする．

1.3 パズルの組み立て支援システム

画面上に読み取った枠，ピースを描画する，またピース結合による更新をリアルタイムで反映できるようにしそれを確認することで間違った探索結果が出力された際に人力で修正が可能となる．画像認識時点での修正も可能となる．また，ピースには識別番号を付与し，実際のピースとの対応をつけやすくする．

2. 開発環境

2.1 IDE，ライブラリ

- Visual Studio 2017
- OpenCV
- Boost

2.2 使用言語

- C++
- Python3

26 TMCITアラカワ

都　立
(荒川)
天野　優樹(4年) 菊池　信志(4年)
鄭　　煐都(4年) 斎藤　敏治(教員)

1. システム概要

パズルのピースの情報を視覚化し、競技参加者がより早く正確にパズルを完成できるように補助を行う。角の大きさや辺の長さの一致するピース・一致する箇所を特定しやすくすることを主目的とする。

2. 想定手順

2.1 ピース・枠の QR コードを読み取る

読み取りは持ち込む web カメラを計算機に有線接続して行う。

2.2 ピース・枠のデータを表示できるように処理。

2.3 ピース・枠の形状を PC 上に線で描画。

2.4 ピース・枠の辺の長さ、ピース・枠の角の大きさ、それぞれの条件を照合・比較して各ピースの正確な位置情報を探索。

2.5 ピース同士またはピースと枠の辺・角度を比較。

2.6 すべてはめ込み出来れば終了。

3. 使用環境

言語 Ruby/C

IDE Visual Studio

ライブラリ DxRuby

27 ソフ研

福　島

加藤　大貴(3年) 佐藤　健喬(3年)
小助川克也(4年) 小泉　康一(教員)

1. システム概要

パズルを解くCUIプログラムとパズルの配置を示すGUIプログラムで問題を解く。今回の競技のルールより、すべてのピースを並べたチームが上位であるため、必要に応じてヒント情報を使用し、すべてのピースを並べることを目指す。

2. パズルの解法

2.1 ピースの読み込み

ピースの写真を撮り、画像を二値化し、頂点検出を行う。精度があまり良くなければ、ヒントのピース形状情報を使用する。

2.2 ピースのはめ込み

アルゴリズムに従って順番にピースを枠に当てはめ、枠からピースを引き算していく。３台のPCを使用し、それぞれ異なるアルゴリズムを使用して、最も良い答えを使用してパズルを並べる。

使用するアルゴリズムは機械学習、総当たり、面積が大きいもの優先、人力等を予定している。

2.3 解情報出力

枠とピースの配置情報を文字列として出力する。

3. パズルの表示

パズルを解くプログラムが出力した文字列を入力とし、枠とピースの配置をGUIプログラムが表示し、それに従ってパズルを並べる。

4.開発環境

言語:VisualC++,VisualC#,VisualBasic

IDE:VisualStudio

ライブラリ:OpenCv

28 人機共闘

弓削商船

田頭　直樹(2年) 平木　大輝(1年)
森上　時静(1年) 前田　弘文(教員)

1. はじめに

今回、本システムでは3台のPCを使用した。それぞれのPCに役割を与え、ソフトウェア的にもハードウェア的にもそれぞれを切り離すことで、独立した開発環境を構築した。これにより、システム全体の開発における効率化を試みた。

2. システムの概要（図1）

2.1 QRコード読み取り用PC

QRコードの読み込みには、Webカメラを使用し、OpenCV + ZBarによって処理する。

2.2 問題解答用PC

問題解答用PCではgnuplotを用い、解答を表示する。

2.3 オーバーレイ用PC

ARToolKitを用いて、モニター越しに「わく」の上に解答をオーバーレイ表示（バーチャル表示）し、組み立ての支援を行う。

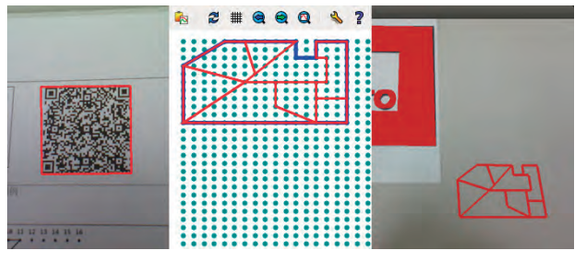


図1 動作画面（左：QRコード読み取り画面、中：問題解答表示画面、右：オーバーレイ表示画面）

3. 開発環境

開発OS	：Ubuntu 14.04LTS
開発言語	：C++
使用ライブラリ	：OpenCV、ARToolKit、ZBar

競技部門

29 HOOC−COOH

長　岡

田島　知宙(4年) 片桐　武史(3年)
布施　智進(2年) 竹部　啓輔(教員)

1. はじめに

去年と同様にピースを画像として読み込む場合、処理を行う際に誤差が生じ、時間もかかってしまうため、完全解答を求めるのは困難と判断し、形状情報使用による減点の度合いを見て、形状情報より解答を求める方法にするか、手動＆配置情報を用いて解答するかを決める。

2. データの読み取り

Unity で作成した QR コードリーダーを使う。
読み込んだデータを形状情報、位置情報の配列に格納する。

3. 処理

配列に取り込んだデータをもとに、個々のピースの形状（一辺の長さ、内角、n 角形、(面積)）を求め、そのデータをもとに枠の角と一致するピースを辺の長さが長いものから探索していく。

また、これとは別に、辺の長さが等しく、組み合わせた角度が 180° や 360° になるなど、隣接しそうなピースの探索も行う。

4. GUI

前節の処理によって整理された座標データをもとに、各ピースの座標を結んだ線を描画する。

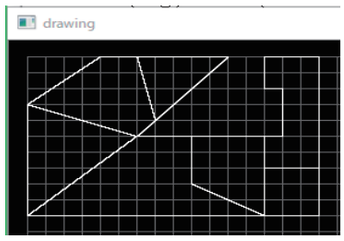


図 1　実行時の画面

5. 開発環境

言語：C++、C#

IDE：Unity5、Visual Studio 2017

ライブラリ：Zxing、OpenCV

30 さてはお前、パソコンを使うんだな？

香　川
（詫間）

三宅健太郎(3年) 小松　聖矢(5年)
人見　　俊(2年) 宮武　明義(教員)

1. はじめに

3 台の PC で回答を導出する。1 台目の PC でデータを処理した後それを 2 台目の PC で共有する。1 台目は辺の組を重視し、2 台目は角度を重視して探索を行う。残りの 1 台はサーバーとしてデータの管理を行う。また回答データの表示にも用いる。

2. 問題データの取得

カメラでピース、及び枠を撮影する。撮影台には黒点がプリントされた用紙を設置し、その黒点の撮影前後の状態からピースの存在位置を推定する。QR コードの読み取りには専用のアプリを開発し、読み取ったデータを自動成型し PC に送信する。

3. 回答の導出

深さ優先探索により回答を導出する。すべてのピースを 1 つ 1 つ探索を行うと回答時間内に解が出ない可能性がある。そのため組み合うピースを予測しそれを 1 つにすることで検索数を減らす。

4. ピースの敷き詰め作業

回答データはタブレット 1 台と敷き詰める人が持つ携帯端末に転送する。その画像を見ながらピースの敷き詰めを行う。なお一定以上の評価値がついたものについては部分解の時点で画像を出力する。これは完全解が時間内に出なかった場合でも、部分解の組み合わせで完全解が導出できるようにするためである。

5. 開発環境

OS　　Windows10, Android7.0
使用言語　C/C++, Java
使用ソフト　Microsoft Visual Studio 2017 Community
　　Android Studio 2.3
使用ライブラリ OpenCV, Boost, Eigen

31 パズルをPCで解くプログラム

旭　川
新田　　陸(5年) 近江　雄哉(3年)
伊勢谷賢司(3年) 嶋田　鉄兵(教員)

1. システム概要

PC に接続されたカメラを用いて形状情報が記載された QR コードを読み取り、複数のピースと枠の形状を得る。その後ピースの適切な位置を求める探索を行い、探索結果を表示する。表示された結果によって探索を再実行、パズルが完成した場合は人の手で現実のピースを操作し実際のパズルを完成させる。

2. QR

PC に接続されたカメラを用いて QR コードを撮影、パズルの形状情報を得る。得られた情報は全ての PC に共有され、後の探索処理を並列でおこなう。探索の結果に応じて追加情報の QR を開示し、探索のさらなるヒントとして使用する。

3. 探索

ピースの辺の長さや角の角度から判断して、相性の良い2つのピースを見つけ1つのピースとして扱う。この手順を繰り返し、ピースの数を一定数まで減らす。

その後、枠の内側を埋めるように、枠の頂点に対して結合されたピース頂点を合わせるような配置を試行し、全てのピースが正しい配置を行うような変形行列を求める。

4. GUI

探索によって得られた結果を表示する。ピースごとに位置を表示でき、結果によっては追加情報を読み取って再探索を行う。また、探索におけるピースの結合に不具合が生じた時、GUI によって手動で解除、再び結合を再選定する。

5. 開発環境

コンパイラ　　：Clang3.5 , VC++2015
フレームワーク：Qt 5.90
エディター　　：Visual Studio 2015 , Vim

32 人力は恥だが役に立つ

都 城
原　　翔耶(3年) 江本　紳豪(3年)
坂元　康平(3年) 丸田　　要(教員)

1. はじめに

本システムはパズルの解を求めるために画像処理部、演算部、出力部の３工程で構成されている。

2. 解法について

2.1 画像処理部

スキャナを使い実際のピースを画像データとして PC に取り込む。その画像データからピースの頂点を算出しテキストデータとして書き出す。この時、同時にピースに番号を割り振った図１のような画像データも出力しておく。

2.2 演算部

画像処理部で出力したテキストデータを基に、ピース情報として各ピースに関する角度や辺の長さなどを算出する。そのピース情報を用いて枠の最小角から深さ優先探索を行うことで、パズルの解を求める。この時可能な限り試行回数を減らすため、ピース同士の結合や枝刈り等を必要に応じて行っていく。

2.3 出力部

パズルがある程度完成に近づいたら現在できているパズルを描画する。それを基に実際のパズルを組み立てることで、プログラムによるパズルの完成と実際の組み立ての時間差を少なくする。

3. 開発環境

［言語］C++、［IDE］Visual Studio 2017 Community
［ライブラリ］OpenCV3.2/cvui/DX ライブラリ

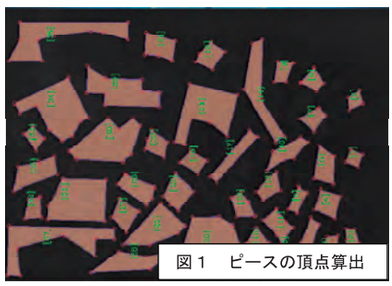


図１　ピースの頂点算出

33 ３人よれば阿修羅の腕

大阪府大
加賀　正樹(専2) 帖佐　克己(専2)
岩﨑　悠斗(3年) 窪田　哲也(教員)

1. はじめに

競技部門の問題は，形状認識とパッキングの２つの問題に分けて考えることができる．ルール上全ての「ピース」を並べる必要があるため，近似解ではなく厳密解を求めることを意識したアルゴリズムを用いる．

2. アルゴリズム

2.1 形状認識

パズルの画像と形状情報を変数とする誤差関数を定義し，これが最小になるように形状情報を推定する．また，形状情報の初期値を決める際に画像特徴量を利用する．

2.2 パッキング

厳密解を求める必要があるため，全探索を行う．ただし，過去に探索した盤面と一致する盤面では探索を打ち切り，無駄な処理を省く．

3. UI

プログラムが解いた結果を競技者に伝えるために，図１のような結果画像を出力し画面に表示する．必要に応じて「ピース」の順序も表示する．

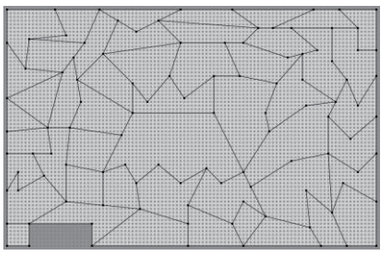


図１．結果画像例

4. 開発環境

C++，OpenCV

34 タコイカ

佐世保
小西　宏樹(3年) 田中　　樹(3年)
今井　壮志(3年) 嶋田　英樹(教員)

1.概要

開発するシステムは、形状情報のQRコードをカメラで撮影・解析し、記載された情報からピースの辺の長さ、角度などを計算しピース形状データを作成する。このピース形状データを用いてパズルの組み立てを行う。

2. QRコード解析，パズルデータ化

競技開始時に提供される QR コードをカメラを用いて撮影し、QR コードを解析後、記載された情報を保存する。また、ヒントに関しては、パズル解法時に適宜使用していく。一方、ピースおよびパズルのデータ化については、QR コードから解析された情報を元に、ピースの長さ、角度、面積等を計算し、ピース情報として保存する。

3.パズルの解法

枠にはめられているピースの辺の長さ、または枠の辺の長さと等しい長さを持った別のピースをリストアップし、互いを合致させる。このときピースが枠からはみ出していないか、他のピースと重なっていないか、角度の和が360°を超えていないか注意し、ピースの位置を確定する。

配置情報の QR コードを参照する場合、記載された情報からその図形の全ての辺の長さを計算する。計算で得られた全ての辺の長さと等しいピースを保存したデータ内からリストアップし、見つけたピースを枠にはめる（図参照）。

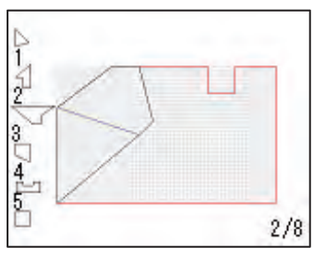


図　ピース位置の確認画面

4.開発環境

言語：C , Java

IDE：Visual Studio 2015 , Eclipse

35 枠外から失礼するゾ＾～（人力）（QRで差をつけろ）

有　明

居石　桃夜(4年) 谷　　廣(3年)
石川　順平(4年)
ゴーチェ ロヴィック(教員)

1. システム概要

スキャナーを用いてピースと枠のデータを PC に取り込み、画像処理を用いた組み合わせ探索プログラムによりパズルの解答を求める。

ピースのスキャンはスキャナに収まる数同時に行い、複数回に分けて行う

2. パズルの解法

2.1 アナログ的ピースのナンバリング

ナンバリングスタンプを使用し、ピースを手動でナンバリングする。似た形状のピースを視覚的に区別し、出力された解を並べやすくする。

2.2 アルゴリズム

まず、スキャンデータからピースと枠の各辺の長さおよび各角の角度を取得するために画像処理を行う。

OpenCV を用いてスキャンデータをグレースケール化、閾値処理等を行い、ピースと枠の輪郭を検出し、直線近似する。

検出した輪郭から、ピースと枠の画像を切り出し各辺の長さ及び各角の角度を取得する。取得したデータはピースと枠の各辺の長さ、各角の角度、切り出した画像を保持するクラスに格納する。

次に、ピースの並びを求めるためにピースの組み合わせ探索を行う。枠の一辺に着目し、その長さと等しくなるピースの辺の長さの組み合わせを探索する。

枠に隣接する部分を埋めたのち、枠に隣接するピースで構成される一辺に着目する。その一辺の長さと、等しくなるピースの辺の長さの組み合わせを探索する。

辺の長さによる探索が難しい場合、角度の組み合わせ探索に切り替える。枠に隣接するピースによって、新たな枠が構成されるため、その枠の角に対応するピースの角度の組み合わせを探索する。

3. 結果の出力

探索した組み合わせおよび切り出した画像を基に、アプリケーション内で解を表示する。

表示された解を基に、手作業でピースを並べる。

4. ヒントの使用

製作したプログラムによる結果の探索が機能しない場合、もしくは時間がかかりすぎる場合は配布された形状情報、配置情報を使用する。なお、QR コードの読み取りは Web カメラを用いる。

その際は別のプログラムを用いて表示を行う。

5. 開発環境

OS: Windows10
言語: C++, C#, Python
IDE: Visual studio, Spyder
ライブラリ: OpenCV, Unity
ハードウェア: スキャナー, Web カメラ

36 ノギス計測のパイオニア

八　戸

照屋　雄斗(3年) 山一　竜光(3年)
小倉　直弥(2年) 細川　　靖(教員)

1. システム概要

スキャナを用いてピースとフレームの頂点データを取得し、複数のパソコンで異なるアルゴリズムを適用して解を求める。

2. パズルの解析

2.1 ピース・フレームのデータ化

スキャナを用いてピースとフレームをスキャンし、頂点の位置に点を打って位置ベクトルを求める。また、データ化するときに、データ化した順にピースの区別をするための番号を決める。QR コードを読み取る際には USB カメラを用いてデータを取得する。

2.2 解析アルゴリズム

一致する辺・角の数などでフレームにはまるピースの候補に優先度をつけて解を見つける。計算時間を短縮するために、同じ状態が重複したときに枝刈りを行う。

2.3 解の表示

フレームにはまったピースの位置を最初に決めた番号と共に描画する。

2.4 GUI

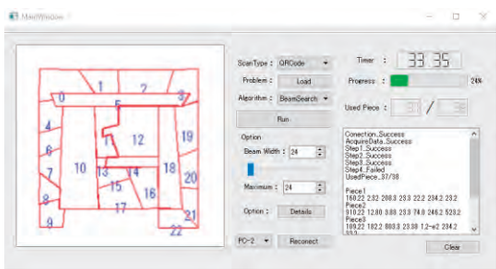


図 GUI のイメージ

GUI を用いてスムーズ且つ視覚的に解析する。

3. 開発環境

言語：C/C++　　IDE：Visual Studio/Qt Creator

ライブラリ：Boost/Qt/OpenCV/OpenMP/ZBar

37 いわゆるじょうほうかのこうせんせいたち

沼　津
緒方　健人(2年) 山本　拓未(2年)
小澤　　慶(1年) 鈴木　康人(教員)

1. システムの特徴

私達のプログラムは、論理演算を利用してピースの重なりを判定するという処理を特徴としています。

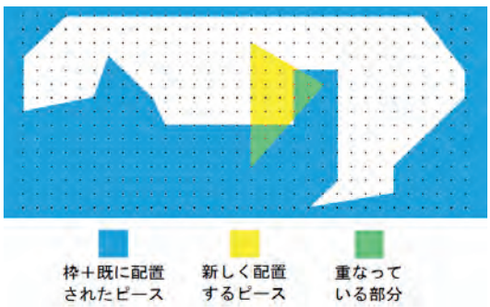
2. この論理演算を利用するプロセス

① グリッド点同士を重なりやすくする為にグリッド点の座標の値を 10 倍にします。

② ピースとピースが配置されているグリッド点、枠の占めるグリッド点に「真」、枠内でピースが配置されていないグリッド点に「偽」を記憶させておきます。

③ 配置した新しいピースのグリッド点の真偽と配置された部分の元のグリッド点の真偽に論理積演算をかけ、その全ての論理積結果に論理和演算をかけます。

④ 結果が「真」なら配置したピースが重なっていて、逆に「偽」であればピースは重ならず、配置出来ます。



3. このシステムの発想

コンピュータがパズルを解くというのは、目をつぶり、ピースの形を想像せずにパズルを解く、ということに似ています。この条件ではパズルが見えず、パズルも形を考えられないので、勿論ピース同士や枠が重なった事には気付けません。そこでピースや枠の形、ピースの位置を座標で表し、コンピュータが扱いやすい真理値を用いてパズルを安易に解きやすく出来ます。

38 自分優勝イイっすか?W

大　分
河野　稜斗(3年) 佐藤　凌誠(3年)
吉海　皓貴(3年) 徳尾　健司(教員)

1. システム概要

①カメラを用いてピースを読み込む、②ピースを組み合わせる、③遺伝的アルゴリズムによって最適解を求めるという 3 つの処理を行い、パズルを完成させる。さらに、ピースの読み込み精度を補うために、必要に応じてヒントからピースや枠の情報を読み込むプログラムを使用する。

2. システム構成

2.1 ピースの読み込み

格子状の紙の上に置いたピースをカメラで撮影する。その撮影した画像上のピースの頂点をクリックするとピースの座標情報を取得することができる。

2.2 ピースの組み合わせ

ピースの辺の傾きと長さが等しい組み合わせをプログラムによって探し、さらにピースの結合の仕方により評価値を付け、一番評価値の高い組み合わせで最終的に結合する。また、その結果をディスプレイ上に表示する。図 1 は結果の一例である。

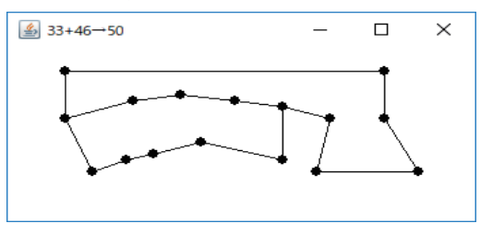


図 1. ピース ID 33 と 46 が結合してピース ID 50 へ

2.3 遺伝的アルゴリズム

2.2 で得られたピースの組み合わせを遺伝子として表現し、遺伝的アルゴリズムを用いて良個体を残していき、最適解を求める。

3. 開発環境

言語：Java

統合開発環境：Eclipse

ライブラリ：OpenCV

39 てんぱ組

都　立（品川）
和田　靖広(3年) 高松　　健(3年)
波多野　陸(1年) 福永　修一(教員)

1. はじめに

今回の競技は、長方形の「わく」から切り出した「ピース」を「わく」に収めることを目的とした競技である。構成するシステムは「わく」と「ピース」をデータ化する入力部分と「ピース」を「わく」に収めるように並べるアルゴリズムの処理部分、結果を表示する出力部分によって構成される。それぞれの処理については以下に説明する。

2. 入力:データの取得

今回はデータの入力に QR コードが配布されるので、この形状情報を入力とする。QR コードは Web カメラを用いることで PC に画像を取り込みデータ化する。

画像処理も併用し、精度のよい結果が得られた場合は、このデータを使用する。

3. 処理:アルゴリズムについて

今回は入力のデータが QR コードで与えられるので正確なデータを使用して処理する事ができる。そのため深さ優先探索を用いて「わく」の全頂点に対して「ピース」を置いていく。ただし、「ピース」が多い場合には枝刈りを行うことで探索時間の短縮を図る。

4. 出力:UI について

今回の競技では問題の解を求められたあとの「わく」に「ピース」を並べる時間も非常に重要になるため、データと現実の「ピース」との紐付けをわかりやすくする必要がある。そのため、解を表示する GUI を用意し、人間はその GUI に沿って現実の「ピース」を「わく」に配置する。

5. 開発環境

言語:C, C++, Java

環境:Sublime Text 3, Eclipse

ライブラリ:OpenCV 2.4.12, PDFBox 2.0.2

40 カピさん

豊　田
宇井　花純(3年) 長坂　光将(2年)
畳谷　晴那(3年) 藤原　賢二(教員)

1. データ読み込み

準備として予め配られていた 6 グリッドの正方形を数回に渡り読み取って平均値を出し、1 グリッドの基準値を割り出しておく。

必要に応じてヒントの QR コードを Web カメラを用いて読み取り、データを配列に入れる。次に枠と各ピースを複数回に分けてスキャナで読み取り、読み込んだ画像を二値化してエッジ検出を行う。

各ピースの頂点座標を求め、用意してあるグリッドを用いて辺の長さ、数、角度を求めて配列に入れる。そして、ピースを識別できるように番号をふり、画面の左上から順番に表示させる。その後、配布された木製のピースに番号を書き込む。

2. アルゴリズム

まず、「QR コードで読み込んだピース」と「スキャナで読み取ったピース」が一致するものを探す。配置情報もある場合はグリッド上に配置する。次に小さなピースの中に 4 グリッド（1 cm四方）の正方形を入れることでピースの向きを絞る。大きなピースは頂点から頂点にかけて線を引き三角形に分けてその内の 1 点をグリッド上に配置し、他の 2 点もグリッド上に合うよう角度を調節する。多角形も中に三角形を作り、分割して先述したことを繰り返す。そして三角形の余った一つの頂点に合うグリッドを探していく。

3. 開発環境

開発環境: Visual Studio2017　　開発言語: C,C++

OS: Windows10

使用機器: Canon LiDE210, Logicool HD Pro Webcam C920r

41 中国山地のパズドラゴン

松　江
鈴木　　豪(2年) 青木　蓮樹(2年)
川上　直人(専1) 橋本　　剛(教員)

1. はじめに

回答プログラムと支援プログラムに分かれている. 回答プログラムは QR コードで提供される形状情報を用いてパズルを解く. 支援プログラムは人間がピースを素早く配置できるように補助する. また, これら 2 つのプログラムは異なるプロセスで実行され, WCF を通してリアルタイムに情報をやり取りする.

2. プログラム概要

2.1 回答プログラム

「結合度」という, 2 つ多角形を辺で結合してできた図形の評価値を基に完全回答を目指す. 結合度の計算には, 辺の一致数などを用いている.

2.2 支援プログラム

2 つの機能がある. 1 つは, QR コードを Web カメラで読み取り, 情報をソルバに送信する機能,もう 1 つは, Web カメラに映ったピースをどこに置くか画面上で指示する機能である.

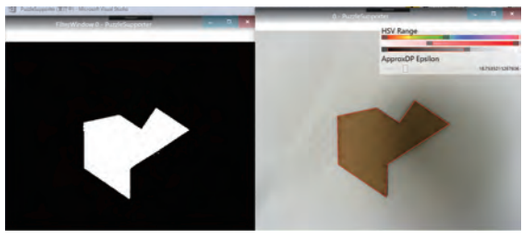


図. 支援プログラムの実行画面

3. 開発環境

言語　　: C#, XAML
エディタ　: Visual Studio 2015/2017
OS　　: Windows 10

42 MLP—My Little Puzzle

高　知
岡田　太陽(4年) 楠目　啄也(3年)
岡本　和樹(2年) 谷澤　俊弘(教員)

1. 導入

我々はただ合理性だけを追うのではなく、GUI を効果的に使用し、誰が使用しても競技部門で優勝できるソフトの作成を目指す。

2. パズル解法の概要

前回の大会の経験から、このパズルをすべてコンピューターで解こうとすることはあまり効果的ではないと考えた。そのため、本大会では、我々はソフトを補助的なものとして扱い、パズルを解く。

以下が解法の手順である。

1. ある一定の大きさを基準とし、ピースを二種に分ける。
2. サイズが大きいピース群を二名が組み合わせてゆく。
3. 二名がパズルを解く間にもう一名がサイズの小さいピース群を計測し、開発したソフトを使い、長さの合うものを表示させる。
4. 出された結果をもとにピースを枠に収め、完成させる。

3. プログラム概要

ピースをスキャナーで読み取り、その画像を OpenCV によって二値化処理した後、エッジ検出などを行って、各辺の長さを計測する。

その後、連想コンテナを用いてブルートフォース探索により長さの合うピース同士を組み合わせて表示する。

4. 開発環境, 使用言語, ライブラリ

OS:Windows10
IDE:VisualStudio2017
使用言語:C++,C#
使用ライブラリ:WINAPI,DIRECTX11,OpenCV

43 B's

和歌山
大川　竜生(3年) 椋梨　　廉(3年)
吉川　　匠(3年) 森　　徹(教員)

1. はじめに

今回の競技は、より早く枠にピースを並べて、パズルを完成させる競技である。ピースに裏表や色などの特徴がないことからピースの型をデータに直して解くことにした。

2. データの取得

黒印を用いて白紙にピースの型をとり、それを撮影する。撮影した画像データをパソコンに送り、パソコン上で枠の位置を固定し、そこの角とピースとのそれぞれの角をひとつ選択し、ピースの角度を取得し、ピースのすべての角度を自動で取得する。

3. 探索

あてはまるピースを適当に選択、そのピースを枠の角度に合うように外側から配置する。そして一周するたびに内側へと探索を続ける。

上記の方法でピースがほかのピースと重なるまたは枠から飛び出る場合は、そのピースを取り除く。配置が可能な場合はそのピースを配置する。

それを繰り返し行い、配置することのできるピースがなくなれば、最後に配置したピースを取り除き、任意の角にあてはまる、他のピースを配置し、探索を再び行う。

4. 終了判定

終了判定は、3通りとする。１つ目は、すべてのピースを配置することができれば、探索を終了し、その結果を出力する。２つ目は、制限時間数分前になれば、探索を終了し、配置したピースが一番多いものを、結果として出力する。３つ目は、すべてのパターンを探索終了した場合、配置したピースが一番多いものを、結果として出力するという以上３つの方法を用いる。

5. 開発環境

C++, Open CV

44 高専に鬼はいま…せん

石　川
宮崎　航輔(4年) 山下　寛人(4年)
古北　昂己(3年) 小村良太郎(教員)

1. はじめに

今回の競技はルールは昨年と似ているが、昨年とは違いピースデータがヒントとして与えられるため、画像処理による取り込みが必須ではないという相違点がある。今回はこれを利用し、ピースデータがあれば確実に解答できるシステムを開発する。

2. ピースデータの読み取り

今回のピースのデータは、形状情報として紙に書かれたQR コードにより与えられる。それを PC に接続したドキュメントスキャナにより読み取る。得られたデータは OpenCV 内のライブラリよりテキストファイルとして保存し、探索アルゴリズムで利用する。

3. 探索アルゴリズム

辺長を重視して探索をする。今年のルールでは頂点がグリッド上にのみあるため、同じ辺長となる組み合わせが昨年に比べ大幅に少なくなる。これを利用し、ピース同士を接続していき、最終的に枠と一致する一つの大きいピースにするのが理想である。

4. 解答の表現

Processing を利用した GUI を作成する予定である。パズルを早く並べる工夫としては、ピースの面積を小さい順にソートし、番号を振って GUI に表示する。

5. 開発環境

言語：C、C++、Java、Processing

IDE：VisualStudio、Intellij idea

Library：OpenCV

45 俺たちはプロコンに行ったらまず真っ先にバグを確認する

奈　良
森田　悟大(4年) 福本　大輔(4年)
三野　天羽(3年) 山口　賢一(教員)

1. ピース・枠の入力方法

ピース、枠の形状情報の入力は配布される QR コードをウェブカメラで読み取り、枠の左上頂点を原点とした座標系として解釈することで行う。また、配置情報についても、必要に応じて同様に読み取る。

2. 解法

① 探索前に、自明に隣接するといえるピース同士は結合してひとつのピースとして扱うことで、探索を簡易化する。

② 各ピースの辺情報から、隣接しうるピースを列挙し、探索する。隣接しうるかの判定には、頂点の角度の和と、辺の長さを用いる。このときに、ピースのある辺がグリッド点を通過していない場合には、その辺上にほかのピース（または枠）の頂点が現れることはないという性質を利用して、探索を高速化する。また、途中で配置情報を入力できることとし、入力があった場合には、それに従ってピースを配置する。

③ すべてのピースがわく内に収まった場合は探索を終了する。

3. 表示方法

以下に示す図は、形状情報をピースの画像として表示させたものである。このような画像を見ながら、実際のパズルを組み立てる。

4. 開発環境

プログラミング言語 ： Java

開発環境 ： Eclipse

46 ミラノ風パズル

小　山
並木　　涼(4年) 田村　　崚(4年)
高井　淳光(3年) 平田　克己(教員)

1. はじめに

今回の競技部門の問題に対して、わくとピースの認識部、ピースの置き方を調べる探索部、探索結果を表示する出力部からなるシステムを作成した。

2. 問題のデータ化

スキャナを用いて取得した画像から、わくとピースの頂点を認識し、頂点座標のリストとして出力する。頂点の認識結果が誤差を含む場合も考えられるため、認識結果を確認するための GUI アプリケーションを用意し、その場で認識結果の修正を行うことができるようにした。

3. ピースの敷き詰め

わくの頂点の内角に対してピースをはめていき、わくの空き領域を繰り返し更新することで再帰的にパズルを解くソルバを作成した。

実際には、ピースとピースの頂点数により探索空間が非常に大きくなると考えられる。したがって、探索前に結合できるピースは 1 つのピースとする前処理を施し、その後盤面を評価しながらビームサーチを用いた探索を行うことで、探索量を削減するようにした。また、探索に関わる動作をパラメータとして持たせ複数のマシンで解くことで、探索に多様性を持たせられるようにした。

4. 探索結果の表示

探索結果は、競技者にとってできるだけわかりやすい形で出力する必要があるが、今回はパズルを解いて得られるわくとピースの辺と、グリッドを重ね合わせた画像を出力する。

5. 開発環境

言語: Visual C++, C#

ライブラリ: DxLib, OpenCV

IDE: Visual Studio

47 京都高専に改名しようの会

舞　鶴

中井　隆智(3年) 久万颯一郎(3年)
河原　未侑(2年) 舩木　英岳(教員)

1. システム概要

1.1 データの入力

スキャナを用いてピースを画像データとし、入力する。

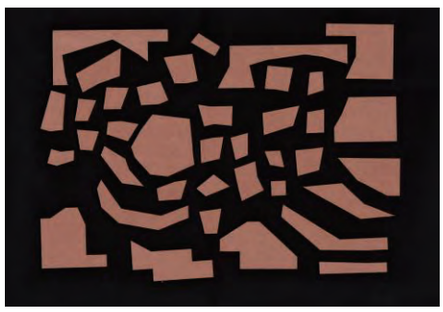


図 1　取得した画像

1.2 ピースの形状情報の取得方法

画像からピースおよびわくの各頂点の座標を取得しテキストデータとして保存する。これはヒントを用いて座標を取得した場合との互換性を保つためである。取得した座標の位置関係や三角関数を用いて数学的な計算により辺の長さおよび角度を取得する。

1.3 パズルの解法アルゴリズム

以下に解法アルゴリズムを記す。

1. わくの辺を 1 つ選択しその「辺の長さと両端角が一致する」ピースを探す。
2. 一致するピースが 1 つだけの場合わくにピースを埋める。
3. 1 および 2 の操作を繰り返す。また、行き詰った場合、1 で探すピースを「辺の長さと一端の角が一致するもの」、「辺の長さが一致するもの」、「一端の角が一致するもの」と順に変え、探索する。

1.4 データの出力

わくにピースを埋めた状態の画像をパソコンの画面に表示する。

48 30文字以内という字数制限に挑戦し続ける鹿児島工業高等専門学

鹿児島

中原　　護(3年) 畑山紘一朗(4年)
湯之上　航(4年) 原　　崇(教員)

1. はじめに

今年の競技内容は、与えられた問題を解析する処理と、実際にピースを並べる処理に分け検討した. また、問題によって、ピースを並べるプログラムの適性が異なることが推測される為、異なるアルゴリズムを用いて複数のソルバを作成し並列に動かし、最も良い解を提出会とする.

2. 問題の解析

この処理では、まずカメラを用いて、ピース形状情報の QR コードを撮影し、ピースの形状をデータ化する. このとき、ピースの各辺の長さや角の大きさを求める。

3. ピースを並べる

ピースの敷き詰め処理では、問題の解析処理で得たデータを元に、動的計画法を始め、ビームサーチや遺伝的アルゴリズムなど複数の探索アルゴリズムを用いてピースの敷詰め処理を行う. まず、求めたピースの角度を用い、隣り合う可能性のあるピースをグループ化する. 更に、ピースの組み合わせなどからグループを絞り込み、最終的に残ったグループから、あまりのピースができるだけ少なくなるように敷き詰める.

4. 開発環境

OS: Windows7, Mac OS X
Editor: Visual Studio 2015/Code, Vim
Language: C++, C#
Library: Open CV

49 Cult of the Party Parrot

東　京

木下　嵩裕(3年) 中野　良春(2年)
佐野　友哉(2年) 山下　晃弘(教員)

1. 処理の流れ

スキャナにわくとピースを置き、それぞれをスキャナで二回に分け撮影する。撮影した画像を、線分検出を用いてデータに変換し、後述のアルゴリズムで解答を導く。

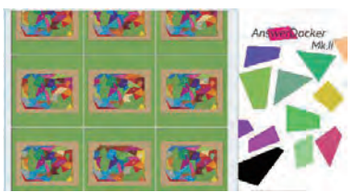


図 1. 解答の出力例

今回も最終的に人の手によって解答するので、人が見やすい GUI をディスプレイに出力し、ピースの配置を確認しながら組み立てていく（図 1）。

2. 設計

問題の傾向によって最適なアルゴリズムが異なると考えられるので、異なる手法やしきい値を用いて複数のアルゴリズムを並列で実行させる。そのため 4 台のパソコンをサーバーとクライアントに分けて動作させる。

3. アルゴリズム

今回は Beam Search を使用する。

まず、わくに対するピースの置き方をいくつか挙げる。次に評価関数を使い、それぞれの置き方から評価値が高いものを何個か選びだす。選び出した置き方によってできた空き領域を新たなわくとして、再度わくに対するピースの置き方を挙げる。この繰り返しでピースを埋めていき、パズルの解答を導く。

評価関数とはある置き方の一つをわくやピースがもつ角の大きさや辺の長さから、自然な当てはまり方をする組み合わせの評価を高く、不自然な当てはまり方をする組み合わせの評価を低く評価する関数である。

4. 開発環境

Arch Linux, C++14, Qt, Boost, OpenCV

50 独立解答法人 Team　WAH

香　川（高松）

丸山　裕雅(専2) 中野　将生(3年)
桑村　真生(2年) 柿元　　健(教員)

1. 概要

持ち込んだ PC の内 1 台をデータ読み込みに用い、他の PC でソルバーを実行する。最終的に出てきた解を、ディスプレイに実物大の大きさと同じになるように表示する。追加情報の使用は試合開始時に問題を見てから判断する。

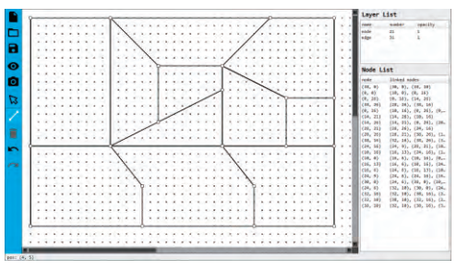
2. システムについて

2.1 入力

スキャナーを用いてピースとフレームを読み込み、頂点情報としてデータ化する。必要に応じて QR コードを読み込み、解析したデータを修正していく。

2.2 ソルバー

各ピースの頂点が合致するように置いていく。辺の長さ、頂点座標を基にした評価値を求め、より良い評価の解を優先的に評価し探索していく。完全解が見つからず解の候補が複数見つかる場合もあるが、その場合は人が判断する。



画面構成図（イメージ）

2.3 出力

完全解に限らず、評価の良い解であれば表示する。具体的には解の探索途中に発見した、その時点での最善解を出力し、リアルタイムに更新していく。

3. 開発環境・言語

言語：C++, D 言語, Python3, OCaml

エディタ：Visual Studio 2017, VSCode, Vim

ライブラリ等：Qt, ZBar, OpenCV

51 WAKUDEKA PING

岐阜

岩田 直樹(5年) 柴野 和樹(3年)
利根 悠司(3年) 廣瀬 康之(教員)

1. システムの設計

全体のシステムを、問題を取得する、解答を作成する、競技者に解を伝える部分の 3 部構成としてとらえ、システムの設計を行った。

2. システムの詳細

2.1 問題取得

問題は木製のピースや枠として与えられるため、スキャンを行うことでデータ化する。スキャンを行う際は、複数のピースを同時にスキャンし、ピースの各辺の長さや各角の大きさの値を取得する。

スキャンが困難な場合は QR コードを利用してピースの形状を取得する。

2.2 解の作成

辺の長さや角の長さを基準とし、隣り合いそうな 2 つのピースを合成し、1 つの大きなピースとして扱う。これを繰り返し、枠の形となる 1 つの大きなピースを作る。

枠に 1 つずつピースを当てはめていき、深さ優先探索を用いて解を探索する。計算量が大きくなることが予想されるため、配置可能領域の面積が最小ピースの面積を下回ったときに枝刈りする。

2.3 解の出力

GUI 出力と同時に、問題取得時に通し番号を振っておき、解答の補助とする。ピースを異なる色で表示し、競技者が円滑に解答としてパズルを組み立てられるようにする。

3. 開発環境

Microsoft Visual C++

NetBeans

52 Let ' s高専STYLE

苫小牧

藤田 春貴(3年) 石川 陸(2年)
中村 太一(2年) 三上 剛(教員)

1. システム概要

Web カメラでピースと枠の画像を取得しそれを解析してデータに加工、または QR コードからピースの情報を取得し、全探索によって解き、回答に近いと思われる出力を GUI に出力しそれを人間が実際の回答台に並べる。

2. ピースのデータ化

Web カメラで撮影した画像を OpenCV のエッジ検出にかけ、さらにそのデータを利用し頂点を検出する。

Web カメラによるピースの形状取得がうまくいかない場合は QR コードから頂点の情報を取得する。

このようにして得た頂点と辺のデータをもとに各ピースを各々の角の大きさと辺の長さにまとめたデータに加工する。

2.4 パズルの解答

枠の内角とピースの外角が一致するかで幅優先探索を行いパズルの解答を求める。このときピースが枠に重ならないかなどで枝刈りを行い、探索結果が一番深い物を答えとして出力する。

また、競技者が十分な解答結果を得られなかったと考えた時は QR コードの撮影に戻り、配置情報を得てから再探索を行えるようにする。

2.5 ソフトウェアの操作について

ピース撮影モード、QR 撮影モード、結果表示モードを用意しすべてのモードに GUI でアクセスできるようにする。

2.6 開発環境

言語:C++, Python

ライブラリ等:OpenCV, Numpy

53 でぃーぷフレンズ

鈴　鹿
和泉　友人(2年) 小谷将太郎(2年)
中川　　悠(2年) 森島　　佑(教員)

1. はじめに

今年の競技部門は昨年度までの課題とほとんど同じで，人力という運要素もあるが，それには頼らず，プログラムでのパズルの完全回答，もしくは完全に近い回答を出すことを目指す．現在製作中のプログラムについて簡単に説明する．

2. 問題解決へのアプローチ

2.1 探索

パズル回答については木探索による探索をベースとし，枝刈りを行う評価に関しては，深層学習の畳み込みニューラルネットで盤面の状態や，次におかれる候補のピースを学習させたものを用いる．

2.2 学習

ランダムでパズルを生成させるプログラムを作成し，小さい規模のパズルから学習させ，重みを継承しながら本選同様の規模のパズルを最終的に学習させる．

2.3 回答

完全回答，もしくはそれに近いものが求まった場合，GUI上にそれを表示する．それを見ながら実際にパズルを並べていく．

54 新パズル島(大嘘)

釧　路
西谷　洋哉(3年) 畠山　襟子(3年)
関村　浩志(3年) 天元　　宏(教員)

1.はじめに

今年の競技は、第２７回プログラミングコンテストに続いて「ピース」を「わく」にはめるパズルを使用した対抗戦で、パズルを完成させる「早さ」と「正確さ」を競う。配布された実物の「ピース」を「わく」内に並べて、パズルを早く完成させたチームが勝利となる。パズル完成までのプロセスは以下のとおりである。

2.パズル完成までのプロセス

2.1 パズルのデジタル化

パズルの枠とピースを並べ撮影し、画像をコンピュータに送る。受け取った画像からパズルのピースの輪郭を検出し、点ベクトルとして格納する。水平・垂直・斜めの線分を圧縮して端点のみを残し、輪郭の全部の点を出力する。

2.2 パズルの解法

撮影した写真から輪郭を出し、求めだされた輪郭をもとに黒線で描画する。複数のピースの中からランダムに二つ選び出し、ピースとピースを合わせ、二つのピースが合ったらそれを一つのピースとする。これを繰り返し、すべてのピースが一つとなったらパズル完成とする。

3.パズルの組み立て支援システム

解答データを画像で表示し、その隣に原画像を表示する。原画像側でピースを選ぶとそのピースとそれに対応した解答画像側の色が変わる。使用したピースは原画像側では淡色表示、解答画像側のピースは濃色表示になる。ピースを選ぶ際、マウス操作での時間ロス対策としてタッチ操作にも対応させることを目指す。

55 ファーストプラーイ's

久留米

菅野　暁(4年) 前田　南樹(4年)
堤　幸太朗(3年) 堺　研一郎(教員)

1. 基本的な戦略

ピースの形状情報を使うと減点されてしまうため、なるべくそれを使用せずにデータ化を行い、完答を目指す。

2. ピースのデータ化

去年と違って枠の大きさがA4サイズに収まるようになったため、スキャナを使用してピースをデータ化する。プログラムが扱いやすいように、ピースは多角形として正確に取り込む。データ化の手順としては、独自に作成したAI(人工知能)に任せる。専用のGUIも作成し、細かい修正などがあったり不具合が起きたりした場合に備える。人間である我々が事前に特別な訓練を積み、あらゆる不測の事態に対応できるようにする。

3. パズルを解く

枠の内側にピースをくっつけるように配置して、それを新たな枠とする方針で探索しパズルを解いていく。具体的な手順としては、独自に作成したAI(人工知能)に任せる。専用のGUIも作成し、細かい修正などがあったり不具合が起きたりした場合に備える。人間である我々が事前に特別な訓練を積み、あらゆる不測の事態に対応できるようにする。

4. パズルを並べる

ここでは主に人間を使用する。ピースを置く場所についての位置を、AIがGUIを介して指示し、人間はそれを見ながらすばやく正確にピースを配置する。ピースを置く場所を把握し配置する作業には高度な技術が必要なため、我々自身が事前に学習を行う。

56 筋肉優先探索

仙　台（名取）

北島　彰(4年) 鈴木　暢真(3年)
渡邉　天翔(1年) 北島　宏之(教員)

1. はじめに

課題に対して、QRコードの読み取りのためのプログラム、読み込んだ頂点からパズルを完成させるアルゴリズムを考案、プログラムを開発した。更に、完成したパズルを出力するUIを作成した。

2. アルゴリズム

2.1 QRコードの読み取りについて

カメラを用いて読み取る。

2.2 ピースの組み立て

事前に、完全に合致するであろうピース同士を一つのピースとして扱うことでピースの絶対数を減らし、計算時間の高速化を図る。その際、ピースの各辺の長さや辺のなす角度を参考にする。

2.3 パズルの組み立て

実際に枠にピースをはめていき、出来た盤面に対して評価値を与えてビームサーチを行い、最終的な解を求める。

図1. 探索途中の様子

2.4 開発環境

C++、Boost、Visual Studio 2015

57 ガーディアンズオブ単位ー

富　山（射水）
新山　響生(3年) 岩田　有喜(1年)
渡辺　孔英(5年) 山口　晃史(教員)

1. 概要

本プログラムではピースもしくはQRコードを撮影し、形状情報を取得した後、複数のPCを用いてそれぞれ異なる解法を適用してパズルを完成させる。また、並列処理を行うことで途中のヒントの入力と高速化を実現する。

2. 解法

これらの操作をもとにピースと枠をグループ化していく。最終的に枠に完全におさまるピースのグループを作成するか、人力でも組み立てられる一定の大きさに到達するグループを探索する。

2.1 角度によるグループ化

ピースの角度の合計が360°もしくは180°になり、かつ物理的に角を密着させられるピース同士をグループ化する。

2.2 辺によるグループ化

枠や他のピースと同じ辺の長さを持ち、かつ物理的に密着させられるピース同士をグループ化する。

2.3 正方形の集合への近似によるグループ化

ピースを局所的に与えられたグリッド上の最小の正方形の集合とみなし、近似的に接着できるピースをグループ化する。

3. GUI

QRコード解析結果や求めた完成図をGUIで表示する。パズルの組み立てを容易にするため、ピースをwebカメラで撮影し完成図上の候補位置を画面上に表示する。

4. 開発環境

言語：C++

IDE：Visual Studio 2017 Community

58 KCB

木更津
小島　五大(2年) 青山　大地(2年)
岩井祐一郎(2年) 岩田　大志(教員)

1. プログラム概要

プログラムは座標を入力し、その座標を「@」および「x」「＊」で示すものである。二次元配列を用いて11x17の配列を作成しその中の値を変えることで座標一つ一つの扱いを変えている。例を挙げると、現在出力中の図形は値を「1」とし「@」で表示とする。また、手順的に前に表示させた図形は他の表示させた図形との区別がつくようにそれぞれ色の変更等の工夫で対処している。上記プログラムの視覚的動作を図1.1に示す。

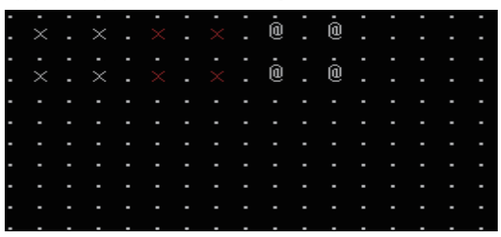


図1.1 プログラムの視覚的な挙動

2. 詳細な視覚的な挙動

具体的に人間の視覚には動作は縦11横17の図形で表示する。出力希望図形は各頂点の座標を「@」で示し、出力済み図形は各頂点の座標を各色の「x」で示す。

また、出力途中で重なってしまった座標はバックグラウンドの色を変更するようにしている。

3. パズル完成までの大まかな流れ

流れとしては最初に与えられたヒントを上記プログラムに組み込み一部のピースをはめ込む。すると残りのピースの位置が理解しやすくなるので残りは人力ではめ込む。

4. 開発環境

開発言語　C言語

開発環境　Windows下での仮想マシン（Linux）

59 あまつばめ

サレジオ
埒　尚太朗(3年) 平山　智己(3年)
吉野　　瑠(3年) 清水　哲也(教員)

1. システム概要

本システムは試合開始時に配布される QR コードを読み込み, パズルの形状情報データのみを取得することを前提としたプログラムである。動作を以下に示す。

2. わくとピースの結合（マージ）

わくとピースの辺の長さを比較し一致したもの同士を結合させる。またこの動作の事をマージという。

3. ピースとピースの結合（ボンド）

ピース同士の辺の長さを比較し一致したものを結合させる。また, この動作の事をボンドという。

4. マージとボンドの利用方法

初期状態のわくをルート, 一回マージしたものを一つ下のノードとしたヒープ木構造を考え, より評価の高いノードを優先する最良優先探索を行う。マージができなくなった場合にはボンドの評価値の一番高いものを採用しマージの動作に戻る。

5. 評価

評価の方法としてはマージ、ボンドをした時に一致した頂点と辺の数が多いほど評価を高くする。

6. 出力

求めた回答を画面上に原寸大に表示し形の似ているピースを判別しやすくする。

7. 開発環境

OS: MacOS, iOS

言語: Swift

IDE: Xcode

60 Lucky Clover

ハノイ国家大学
Nguyen Thi Huyen, Tong Ly Trinh
Nguyen Van Vinh

1. Solution idea

Our idea is looking for two pieces with matching angles, merge them together, then treat it as a piece and continue to do so until the completion of matching all the pieces. Each step, we draw to the screen to track the match of the image to be able to give up or accept.

2. Find two matching pieces

2.1 Find the right angles

We will use the algorithm to find the angles of each other with high heuristic: angles different from 90 degree, angles with the least number of matching cases, the angles with the most matched vertices. We can use the algorithm A *.

2.2 Put two priority polygons together

We will use the translate function to make the two corners coincide, and then rotate and test the two overlapping images to put the two together. Then, we use the function of the border of the grafted image and considered a new polygon. When merging two polygons together, we added the gpc.h library.

2.3 Export the screenshots

Each step, we can see, accept or remove the image is not reasonable.

3. programming language

Our team uses C ++ and Javascript to design the project.

4. Environment and application

The program will run in Window 10 or Window 7, Mozzilla Firefox. We write the project on Codeblock and Sublime text.

61 mustSolve

モンゴル科学技術大学
ERDENEBAYAR Dovchindorj,
ENKH-AMAR Gantulga,
KHUDER Altangerel

1. Introduction

Our program will be written in javascript. Javascript has really good coding eco-system which is rich open-source library, algorithm implement is traditional C alike and object oriented features and easy to integrate with user interface.

2. Methodology

Our team considered we must use multiple algorithmic approaches to solve this problem.

2.1 Evaluation function

This algorithm considers each piece starting from biggest piece and tries to find the most appropriate position for the piece. There will be an evaluation function which calculates points (closer to the correct solution, more points). The piece will be fixed on the position with maximum evaluation points.

2.2 Normalization algorithm

This algorithm will normalize every edge of every piece to define its angle with OX axis. We will consider top left corner coordinates are (0, 0). Then we will draw all pieces on computer screen. Edges with same angle will have same color. That will greatly help us to combine pieces together.

3. Discussion

First algorithm may fix a piece at wrong position.

Second algorithm will not help much if many pieces are similar with each other.

When we will get the pieces we will decide which algorithm to use.

If all our algorithmic approach seems to fail or takes too much resource, we will get all 4 hints and use our bare hand to complete the puzzle.

62 UTP

ペトロナス工科大学
Nurul Insyirah MUHAMMAD AMIRUL CHANG,
Nuraina Fathinie FAHRURRAZI,
Ahmad Izuddin ZAINAL ABIDIN

1. Introduction

Our software is designed to fill up puzzle pieces of irregular shape onto a designated frame. The software aims to find a solution with the most efficient way to solve the puzzle within an acceptable time limit. In matching each piece, we analyze the shape and use a method to detect corners with each piece is being approximated with a polygon.

2. Software Details

2.1 Technical Specification

Our solution uses Python that is lightweight, yet it efficiently supports multithreading features. The version of Python that is used is Python 3.5.2. The IDE that we used is NetBeans IDE 8.1 that is also used for editing purposes.

2.2 Requirement

For functional requirements, whenever a set of inputs (*e.g.* puzzle pieces) are received, our solution will arrange the pieces so that they will fit into the designated frame. The non-functional requirements would prevent any invalid input based on the pre-defined conditions to ensure no runtime error occurs.

2.3 Performance goals

In our implementation, we aim to minimize the processing time by applying multithreading, allowing a task to be divided into smaller tasks. Our software is expected to complete the puzzle at minimum possible time.

63 VTC Engineering

香港 VTC
HUANG Jingxin,
CHAN Sui Fu Albert,
CHAN Sze Yin,
LEE Kin Sang Timothy

1. Introduction

We use smartphone ‘s camera to take image of the QR code and all program base on Android studio and Java language. The main function of the program is reconstructing the shape image with the information in QR code. Each level of puzzle piece location information will be display on screen which speed up the puzzle completion time.

2. Methodology used in the program

The main objective of the program is shortening the puzzle assembly time and assist user to find puzzle piece correct location. To retrieve location information of puzzle piece. It is necessary to capture QR code image and decode the QR code information. However, the location information is a series of numbers which are difficult to read and interpret by user during the contest. As a result, a graphical user interface is used to convert location information into geometry graphic with co-ordinations.

2.1 Revealed location information

The program consists of three modules to reveal the location information.

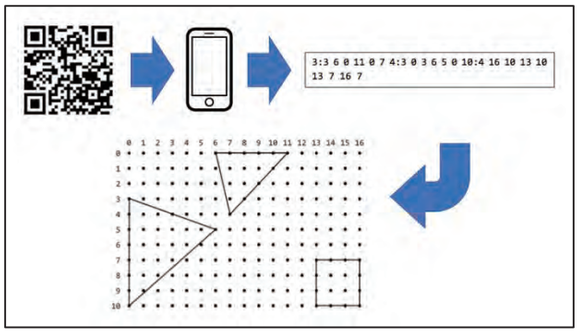


figure 1: Concept of the program reveal puzzle piece location

64 徳川&今川

豊橋技術科学大学
須和田与春, 吉渡　匠汰,
小原　祐斗, 梅村　恭司(教員)

1. はじめに

今回のルールではパズルのパターンに制約が加えられたこと、スキャナの使用が可能なことから、ピースの形状を精度よくデータ化できると考えられる。各問題で形状情報や配置情報が提供されるが、これらは可能な限り用いずに完答をめざす。

2. ピース形状のデータ化

スキャナで取り込んだピース画像を 2 値化し、確率的ハフ変換を用いたアルゴリズムによって線分検出を行う。線分情報からピース頂点の相対位置を推定し、回転させたときに全ての頂点がグリッド上に乗るパターンを列挙する。

3. パズルの解法

ピースを、枠や別のピースに接するように置いていき、その状態をノードとしてビームサーチを行う。また、配布されたサンプルからパズルの切り方の傾向がある程度推定できるので、この情報もヒューリスティックとして用いていく。

4. 出力

ピースごとに ID を割り当て、パズルの完成画像とスキャン画像のそれぞれに ID を書き込んだものを GUI 画面に出力する。これらを参照してパズルを組み立てる。

5. 開発環境

C++14

OpenCV3. 2