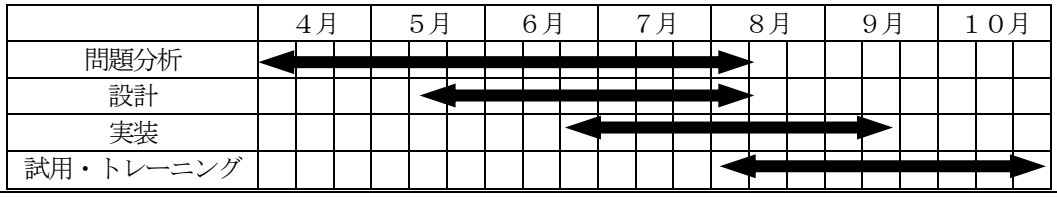
部　　門	競 技 部 門	No.1 登録番号	30039

No.2

1) 予定開発期間： 6ヶ月
2) 予定開発人数： 2人



No.3

実現方法【パズル完成までのプロセス】

1)　パズルのデジタルデータ化方法（追加情報を活用する場合はその旨記載）

パズルをスキャナーで読み込み、取り込んだ画像を背景とピースに分け、辺を探す。その後、見つけた辺に対して hough 変換を行い辺に沿った直線を出力する。そしてその辺をつなぎその交点を頂点とする。以上の手順でピースをデジタルデータ化する。最初から頂点を探索しない理由は、頂点の周りは加工の際についた誤差が多くついているからである。一方、辺には頂点ほどの誤差はないので、辺の交点を探すほうが誤差が出ないと期待できる。

今後は、hough 変換自体の精度を上げることで頂点の精度を上げていくことが必要である。

2)　パズルの解法（追加情報を活用する場合はその旨記載）

探索方法の基本方針は、枠の角を 1 つ適当に決めて、そこから深さ優先探索をベースとした全探索をして、ピースを順番に埋めていく。また、ピースの頂点はグリッドに必ず乗っているという条件に当てはまらない場所などは枝刈りしていく。もし、パズルを正しくデータ化できない場合は、データが少し間違っているのを考慮して、他のピースとの重なりを考慮することや、枠やピースの角を基準にして置いていかない方法などの探索に切り替え問題を解く。

今後は正しくデータ化できなかった時の探索の工夫が課題となる。

3)　パズルの組み立て支援システム

上記のアルゴリズムによりピースの配置が決定した後、その配置は JAVA 言語で実装した GUI により表示を行い、その出力を見ながら手動でピースを枠に配置していく。ただし、画面上のピースから現実のピースを特定するのが難しいため web カメラでピースを読み込み、そのピースはどこに置くべきかを GUI 上に表示することでピースの特定を容易にする。

4)　その他（独創的なところ）

探索ではうまくピースをデータ化できなかったときにピースの重なりを考慮する。また、他の方法としてピースの形の候補を複数提示して探索を行う。この際探索での処理が増えてしまい時間がかかってしまうので、GPU を搭載した PC を使用し並列化することで処理時間を短くする。

また、パズルのデータがうまくデータ化できた場合とできなかった場合で異なるアルゴリズムでピースの配置をする。その際にパソコンを複数台使用することで処理が遅くなることを回避する。

回答をするときには、web カメラにピースを映しそのピースを置くべき場所をパソコン上に表示させるなど回答が早くできるようにする。

No.4

開発環境
【OS】Windows7 / 8.1 / 10,　Ubuntu 16.04
【Compiler】GNU Compiler Collection,　JDK
【Language】C,　C++,　Java,　CUDA
【Library】OpenCV
【Editor】Sublime Text 3,　Eclipse