

部 門	競 技 部 門	No.1 登録番号	3 0 0 3 7
-----	---------	-----------	-----------

No.2	1) 予定開発期間：7 ヶ月間							
	2) 予定開発人数：3 人							
		4月	5月	6月	7月	8月	9月	10月
	問題分析	←————→						
	設計		←————→					
実装				←————→				
試用・トレーニング					←————→			

No.3	<p>実現方法</p> <p>1) 原画像の推察アルゴリズム</p> <p>(1)ピース同士の一貫性の判定 それぞれの断片画像の辺において、ピクセル毎の不一致度合いをその画像以外の断片画像の全ての辺において計算する。ここで「不一致度合い」は、辺の各ピクセルを LAB 色空間に変換してその差の 2 乗和の平方根(ユークリッド距離)を辺のピクセル数で割った値を指す。 そうしてピースの辺に対する不一致度合いが一番小さいピースの辺同士をペアとして管理し、復元操作を行う。</p> <p>(2)復元操作 位置が確定済みのピース集合 S と未確定のピース集合 T を考えて、最初に位置を決定したピースを S に追加する。そして S と隣接しそうな不一致度の低いピースを T から取り出し S に追加していく。この時、S に含まれるピースと隣接辺の多いマスから優先して S に追加するとし、それぞれ不一致度合いの合計が一番小さいピースやペアとして選んだピースを選ぶ。 S に含まれるピースと隣接している数が少ないマスを決める際には数ピース先まで調べて上記の評価を行うことでより精度を高める。</p>
	<p>2) 断片画像の並び替えアルゴリズム</p> <p>選択可能回数があることやコストが交換レートよりも選択レートの方が高い可能性がある為、なるべく 1 回の選択で並び替えることができるアルゴリズムを作ることを考える。 まず、1 回の選択による断片画像の並び替えはスライドパズルと近似しているため初期配置からゴール可能かどうかの判定を行うことができる。そこから 1 回の選択で移動させることができるピースを選択しそれぞれのピースの位置と原画像における位置とのマンハッタン距離がなるべく増えないように選択ピースを動かす。これは 4 方向にしか選択ピースは動くことができないので 4 方向について貪欲法を適用することで計算を行う。 また、選択していないピースを移動する際に 1 回の選択で移動を行う場合、例えば(0, 0)から(2, 2)に移動するものと(4, 4)で移動するものとはマンハッタン距離は同じだが実際の交換回数に違いがある為、ある座標からある座標に移動する際は最低必要手数をダイクストラの辺の重みにしてダイクストラ法を適用する。 このように交換を行い現画像の一番上の列から順に完成するように移動を行うと、徐々に埋めなければならない行が少なくなっていく 2×n(n は分割数)の形にできるためこの形になったら別の処理で移動を行うことでスライドパズルを完成させる。</p>
	<p>3) その他 (独創的なところ)</p> <p>GUI 等を用いて対話形式で原画像を完成させる。仮に製作したアルゴリズムで原画像が完成しなかった場合、少ない修正で原画像が完成する場合は人間が画像同士の位置変更を行うことができるようにし、大幅に修正が必要で人間が見ただけでは原画像が推察することができない場合は不一致度も変更を行うことができるようにする。これは一致しなかった辺に対して不一致度を十分大きくすることで次の原画像推察アルゴリズムを実行した際に前回と同じ結果を得ないようにするためである。</p>

No.4	<p>開発環境</p> <p>OS : Windows</p> <p>エディタ : Visual Studio, サクラエディタ</p> <p>使用言語など : OpenCV, C++, Python</p>
------	--