

部 門	競 技 部 門	No. 1 登録番号	30026
-----	---------	------------	-------

No.2	タイトル	Renaissance -そのとき画像は復活されたのです-
------	------	-------------------------------

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

No.4	<p>&lt;概要&gt; 今回の競技は全体の処理を二つの手順に分けて処理を行う。</p> <p>&lt;原画像の推測&gt; 各断片画像の原画像における位置を推測するために以下の手順を行う。</p> <ol style="list-style-type: none"> <li>各断片画像がどの断片画像と隣接するかを検査する。 <ul style="list-style-type: none"> <li>各断片画像の各辺同士の色を比較し類似しているものを隣接しているとみなす。</li> <li>辺が類似した断片画像が多数ある場合の誤認識を防ぐことが求められる。</li> </ul> </li> <li>各断片画像の配置の推測。 <ul style="list-style-type: none"> <li>他の断片画像と繋がる可能性が低い断片画像を四隅の画像と仮定する。</li> <li>四隅それぞれの断片画像に隣接している断片画像を再帰的に配置し原画像全体における位置を導く。</li> </ul> </li> </ol> <p>&lt;復元操作&gt; 各断片画像を元の位置に復元する操作を以下の手法で求める。このとき選択可能回数以下でかつコストがより少なくなるような操作を求める。</p> <ul style="list-style-type: none"> <li>復元操作を経路探索問題としてモデル化する。 <ul style="list-style-type: none"> <li>問題をグラフとして解釈し、復元途中の問題画像を状態としてノードで表現し、選択・交換の操作をエッジで表現する。さらに一度の選択・交換に要するコストをエッジの重みとする。</li> <li>上記のグラフ上で問題画像をスタートノード、推測した原画像をゴールノードとし、その間の最短経路探索を行う。</li> </ul> </li> <li>最短経路の探索手法 <ul style="list-style-type: none"> <li>アルゴリズム A* を活用して最短経路を求める。</li> <li>ノード <math>n</math> に対するゴールまでの最小コスト予想 <math>h^*(n)</math> を活用し、探索の高速化を図る。</li> <li><math>h^*(n)</math> の精度を向上させることでさらなる高速化を目指す。</li> </ul> </li> </ul>									

No.5	開発環境 Python 2.7.6 Java 1.7.0									
------	------------------------------------	--	--	--	--	--	--	--	--	--