

## 1. 基本的なアプローチ

押せるスタンプの種類と位置の中で、押した時に画像の評価値が最も高くなるものを貪欲的に採用していく。しかしスタンプのサイズが大きい場合、押せる場所全てについて評価値を調べると非常に時間がかかるため、このようなスタンプに関してはあらかじめ枝刈りを行う。この動作を初期画像が完成画像と同じ画像に変わるまで繰り返す。

## 2. 評価関数

評価関数は「スタンプを押すことで完成画像と一致するマスの数の増減」「完成画像と一致しないマスの集積度」「画像全体の完成画像と一致しないマスの密度」の三点について、実験的に得た最適な重みを乗じて足しあわせたものを計算し評価値とする。

## 3. 枝刈り

スタンプのサイズがある閾値以上の場合、スタンプの黒マスの密度と画像の密度を比較し、確率的にスタンプを押しても状態が改善しない可能性が高いと判断したら評価値を計算せずにそのスタンプは採用しないとする。

## 4. ユーザインタフェース

プログラムの起動等はプロンプト上で行うが、画像の状態を逐次画面に表示するアプリケーションを作成した。これにより問題の内容や計算状態を視覚的に確認することが可能となっている。

## 5. 開発環境

Vim - vi Improved 7.2

Gnu Compiler Collection 4.7.0