

プログラミング基礎演習レポート 2016-1

長谷川禎彦

注意点

レポートは以下の2つを提出すること

- プログラムのソースコード (C 言語)
- レポート本体 (doc, docx, pdf, odt). 英語でも良い.

プログラミングのレポートではプログラムソースや実行結果のみを送る人がいるが、プログラムのソースだけでは何をするものなのか分からないため、採点出来ない。プログラムの作成にあたってどのような工夫をして、それをどのように実現したのか、レポート本体に書くこと。レポートでは以下の点を書くことが一般的である。

- 導入
- 手法・結果
- 考察
- 参考文献 (あれば)

レポート本体の提出フォーマットは Microsoft Word (docx, doc), PDF, OpenOffice Writer のどれかで提出する。なお、レポート本体のページ数に上限はないが、無意味な結果の羅列による水増しは却って印象を悪くする。

- 提出〆切り：2016/12/19 の 23:59
- 提出方法：作成したソースファイル (ファイル名は自由。一つのプログラムを複数ファイルに分割しても良い) とレポート本体 (doc, docx, pdf, odt 等) を「学籍番号.zip」としてまとめる。作成した zip ファイルをホームページの提出フォームから提出する (「ファイル」と書かれた所から、ソースとレポート本体の zip ファイルを添付する)。その際、課題の選択を「レポート1」とすること (「小課題」ではない)。
- なお、全ての問題において大枠として題意を満たしていれば、方法・内容とも自由に変更して良い。また、問題の拡張や手法の改良を行って良い。

1. 課題 (必須課題)

与えられた画像にノイズを付加するプログラムを実装せよ。簡単のため白黒画像を扱う。各ピクセルの値は0が白で、1が黒を表す (後述の PBM フォーマット参照)。ノイズは、各ピクセルにおいて、確率 p で値を反転させる (図 1)。入力画像ファイル名や p の値は実行時引数や、標準入力等で指定できるようにせよ (つまり値をソースコードに書き込まない)。ノイズを付加した画像はファイルとして出力せよ。PBM の画像ファイル例は授業のホームページにアップロードしてある (同じ画像で PNG フォーマットも同時にある)。



図1 : 元の画像 (左) とノイズ $p = 0.1$ を付加した画像 (右). 画像は GATAG より利用.

x_1	x_2	\cdots	x_w
x_{w+1}	x_{w+2}	\cdots	x_{2w}
\vdots		\ddots	\vdots
$x_{(h-1)w+1}$	$x_{(h-1)w+2}$	\cdots	x_{hw}

図2 : 高さ h , 幅 w の画像のピクセルと変数 x_i の関係.

2. 課題 (自由課題)

課題 1 で作成したノイズ画像を, 逆に復元するアルゴリズムを実装してみる. ここではマルコフランダム場 (Markov Random Field : MRF) と呼ばれる方法を用いる. 今変数 x_i でノイズのない真の画像のピクセル値を表す (x_i と画像のピクセルの値は図 2 に示す). そして, y_i でノイズが付加された画像のピクセル値を表す ($x_i, y_i \in \{0,1\}$). そのため, ノイズ画像からの復元は, 与えられた y_i から最も適切な x_i を推定する問題となる. x_i は未知であるが, 当然 y_i とは近い値を持つはずである. 同時に, ノイズのない画像は, 隣同士のピクセルの色が近いという特徴を持つ. つまり, x_i は

- y_i となるべく近い値を取る
- 隣同士のピクセル x_i と x_j はなるべく近い値を取る

という条件を満たすように決定すればよい. この場合, 例えば以下のようなエネルギー関数を考え, このエネルギー関数を最小化させるような $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ を求めればよい.

$$E(\mathbf{x}, \mathbf{y}) = -\beta \sum_{\{i,j\}} (2x_i - 1)(2x_j - 1) - \eta \sum_i (2x_i - 1)(2y_i - 1). \quad (1)$$

式(1)で, 第一項の和 $\sum_{\{i,j\}}$ は隣り合うピクセル i, j について取るとする. また, β, η は非負の値を取るパラメータである ($\beta = 1, \eta = 2$ 程度の値で行ってみよ). 第一項は, 隣同士が同じ値を取る場合エネルギーが小さくなり, 違う値を取ると大きくなる. また, 第二項は, 真の画像 x_i と観測されたノイズ画像 y_i が同じ値の時エネルギーが小さくなり, 違う値を取ると大きくなる. 式(1)の x_i に対する最小化を以下の方法で行ってみる.

- ① 推定したい真の画像ピクセルの値を $x_i = y_i$ で初期化
- ② ピクセル x_i を一つ選択. 他のピクセル x_j ($i \neq j$) を固定した状態で, x_i の可能な二つの状態 $\{0,1\}$ のエネルギー $E(\mathbf{x}, \mathbf{y})$ を計算し, 小さくなる方の値を選択する
- ③ 一定の収束条件を満たすまで, ②を繰り返す

このアルゴリズムは **Iterated conditional modes (ICM)** と呼ばれるアルゴリズムである. ①でのピクセルの選択は x_i を順に選択する方法や, ランダムに選択する方法がある. また, ③の収束条件も適当に定義せよ (例: これ以上ピクセルの更新が出来なくなるまで, 等).

画像の境界は周期的条件等を用いるなどして適切に処理せよ. 出力はファイルへの出力でもよい

し、余力があれば画面への表示でも良い。

PBM (portable bitmap format) フォーマット

値と値の間をスペースで区切った白黒の画像フォーマットである。通常のテキストファイルなため、テキストエディタで開くことが出来る（そのまま開けない場合は、拡張子を **.pbm** から **.txt** に変更してみよ）。各値は、ピクセルの輝度を表し、**1** が黒、**0** が白となる。最初の 2 行は画像の情報を書き込み、その後に画像自体の値を書き込む (図 3)。PBM は **emacs** (コマンド: **emacs hoge.pbm**)、**GIMP** (<https://www.gimp.org/>)、**LibreOffice** で見る事が可能である。DocsPal (<http://www.docspal.com/viewer>) という Online の Viewer を使っても見る事が出来る。

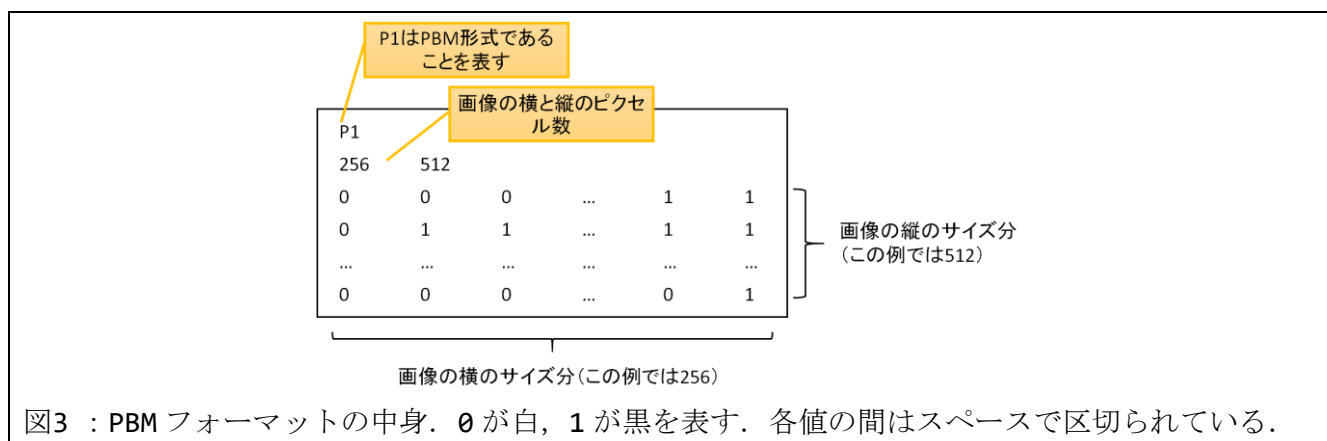


図3 : PBM フォーマットの中身. 0 が白, 1 が黒を表す. 各値の間はスペースで区切られている.

参考文献

- C. M. Bishop, "Pattern recognition and machine learning"