

情報通信工学専門実験 A

画像処理プログラミング課題レポート

学籍番号:08D23091 氏名: 辻孝弥

2025 年 6 月 10 日

1 目的・理論

1.1 目的

本実験では、PGM-RAW フォーマットの画像ファイルに対して、エッジ強調処理および大津の方法による自動二値化処理を実装することを目的とする。具体的には以下の 3 つの課題に取り組む：

1. エッジ強調プログラムの作成（2 種類以上の手法を実装）
2. 大津の方法による自動二値化プログラムの作成
3. エッジ強調と二値化の組み合わせ処理の実装

1.2 理論

1.2.1 Prewitt フィルタ

Prewitt フィルタは、画像のエッジを検出するための空間フィルタの一種である。x 方向と y 方向それぞれについて、以下の 3×3 のオペレータを用いる：

$$P_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad P_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad (1)$$

エッジ強度の計算には以下の 2 つの方法がある：

$$g(x, y) = \sqrt{g_x^2 + g_y^2} \quad (\text{式 2: 平方根演算}) \quad (2)$$

$$g(x, y) = |g_x| + |g_y| \quad (\text{式 3: 絶対値演算}) \quad (3)$$

ここで、 g_x 、 g_y はそれぞれ x 方向、y 方向の Prewitt フィルタを適用した結果である。

1.2.2 大津の方法

大津の方法は、画像の濃度ヒストグラムから最適な閾値を自動的に決定する手法である。クラス間分散を最大化する閾値を求める。

全画素数を N 、階調値 i の画素数を n_i 、出現確率を $p_i = n_i/N$ とする。閾値 k でクラス 0 ($0 \sim k$) とクラス 1 ($k+1 \sim L-1$) に分割した時のクラス間分散 $\sigma_B^2(k)$ は：

$$\sigma_B^2(k) = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \quad (4)$$

ここで、 ω_0 、 ω_1 は各クラスの出現確率、 μ_0 、 μ_1 は各クラスの平均、 μ_T は全体の平均である。

2 課題 1～3 への取り組み

2.1 実装したエッジ強調手法

本実験では、Prewitt フィルタを用いた以下の 2 つの手法を実装した：

1. Prewitt フィルタ + 平方根演算 (式 2)
2. Prewitt フィルタ + 絶対値演算 (式 3)

2.2 プログラムの拡張と工夫

2.2.1 モード選択機能の実装

元のプログラムはネガポジ反転のみの機能であったが、以下の 3 つのモードを選択できるよう拡張した：

- edge: エッジ強調のみ
- binary: 二値化のみ
- both: エッジ強調→二値化の順次処理

2.2.2 境界処理の工夫

3 × 3 フィルタの適用において、画像の端部では近傍画素が存在しない問題がある。この問題に対して、以下の対策を講じた：

Listing 1: 境界処理の実装

```
1 for(y=0; y<height-2; y++)
2 {
3     for(x=0; x<width-2; x++)
4     {
5         // フィルタ処理
6     }
7 }
```

この実装により、画像の端から 1 画素分の領域を処理対象から除外し、メモリ保護違反を防いでいる。

2.2.3 値域制限処理

フィルタ演算の結果が 0～255 の範囲を超える可能性があるため、以下のクリッピング処理を実装した：

Listing 2: 値域制限処理

```
1 int result = sqrt(x_sum*x_sum + y_sum*y_sum);
2 if(result > 255)
3 {
4     result = 255;
5 }
6 resultImage->data[x+resultImage->width*y] = result;
```

2.2.4 大津の方法の実装

自動閾値決定のため、大津の方法を実装した。ヒストグラム計算からクラス間分散の最大化まで、理論に忠実に実装している：

Listing 3: 大津の方法の主要部分

```
1 // ヒストグラム計算
2 for(y = 0; y < height; y++) {
```

```

3     for(x = 0; x < width; x++) {
4         ni[originalImage->data[x + originalImage->width * y]]++;
5     }
6 }
7
8 // クラス間分散の最大化
9 for(int k = 1; k < L; k++) {
10     //  $\omega_0, \omega_1, \mu_0, \mu$  の計算1
11     double sigma_B = omega_0 * (mu_0 - mu_T) * (mu_0 - mu_T) +
12                     omega_1 * (mu_1 - mu_T) * (mu_1 - mu_T);
13
14     if(sigma_B > max_sigma_B) {
15         max_sigma_B = sigma_B;
16         optimal_k = k;
17     }
18 }

```

2.2.5 ヒストグラム計算の最適化

従来の教科書の実装では、階調値 k を 0-255 まで変化させる 2 重ループ `for(k){for(i){...}}` が用いられるため、画像を 256 回走査することになる。本実装では以下のコード 4 のように画像全体を 1 回走査して `ni[i]` を更新することで、計算量を $O(N)$ に削減した。

Listing 4: 効率的なヒストグラム計算

```

1 // の計算：各階調値の出現回数をカウントni
2 for(y = 0; y < height; y++) {
3     for(x = 0; x < width; x++) {
4         ni[originalImage->data[x + originalImage->width * y]]++;
5     }
6 }

```

この最適化により、リアルタイム処理に近い応答性を実現できた。

2.3 プログラムの構造化

可読性向上のため、以下の関数に分割して実装した：

- `filteringImageByPrewittWithSquareRoot`: 平方根演算による Prewitt フィ

ルタ

- `filteringImageByPrewittWithAbsolute`: 絶対値演算による Prewitt フィルタ
- `binarizationWithThreshold`: 閾値による二値化
- `calclateThreshold`: 大津の方法による閾値計算
- `PrewittAndBinarization`: エッジ強調→二値化の組み合わせ処理

3 実行結果

3.1 使用方法

プログラムは以下のコマンドで実行する：

```
./program <mode> <input.pgm> <output.pgm>
```

3.2 処理結果

複数のサンプル画像に対して 3 つのモード (edge、binary、both) での処理を実行した。以下に実行結果の概要を示す。

3.2.1 エッジ強調処理

Prewitt フィルタを用いたエッジ強調処理により、全 13 サンプルにおいて画像の輪郭部分が効果的に強調された。特に sample1-11 では明確なエッジが検出され、物体の境界が鮮明に抽出されている。

3.2.2 二値化処理

大津の方法による自動二値化により、各画像に対して最適な閾値が決定された。閾値の範囲は 1-162 と画像によって大きく異なり、画像の特性に応じた適応的な処理が行われていることが確認できる。

3.2.3 組み合わせ処理

エッジ強調後の二値化処理により、元画像のエッジ成分のみが効果的に抽出された。この手法により、物体の輪郭情報を強調した二値画像を生成できることが実証された。

詳細な処理結果画像については、7 章に示す。

3.3 全サンプルの閾値結果

実行日時：2025 年 5 月 31 日土曜日 19 時 58 分 32 秒 JST

大津の方法により自動決定された閾値を表 1 に示す。

表 1: 各サンプル画像の二値化閾値

サンプル	直接二値化の閾値	エッジ強調→二値化の閾値
sample1.pgm	134	125
sample2.pgm	111	127
sample3.pgm	105	125
sample4.pgm	155	112
sample5.pgm	128	128
sample6.pgm	127	109
sample7.pgm	107	114
sample8.pgm	136	105
sample9.pgm	162	133
sample10.pgm	103	126
sample11.pgm	117	100
sample12.pgm	1	1
sample13.pgm	1	1

3.4 結果の分析

表 1 から、以下の特徴が観察される：

- **画像間の閾値の差**: sample12, 13 を除く画像では、閾値が 100-162 の範囲で分布しており、画像の濃度特性に応じた適応的な処理が行われている。
- **処理方法による閾値の違い**: 元画像の直接二値化と比較して、エッジ強調後の二値化では異なる閾値が選択される。これは、エッジ強調により画像の濃度分布が変化するためである。
- **特殊ケース**: sample12, 13 では閾値が 1 となっており、これらの画像が特殊な濃度特性を持つことが示唆される。

特に sample4 では直接二値化の閾値 155 に対してエッジ強調後は 112 と大きく異なり、処理による濃度分布の変化が顕著に現れている例である。

4 考察

4.1 エッジ強調手法の比較

実装した 2 つの Prewitt フィルタ手法について：

- 平方根演算（式 2）：より正確なエッジ強度を計算できるが、計算コストが高い
- 絶対値演算（式 3）：計算が高速で、実用的な結果が得られる

実際の処理では、絶対値演算の方が一般的に使用される傾向がある。

4.2 境界処理の影響

今回の実装では、画像の端 1 画素分を処理対象から除外した。これにより：

- 利点：メモリ保護違反を確実に防止
- 欠点：端部の情報が失われる

より高度な境界処理として、ゼロパディングや対称拡張などの手法が考えられる。

4.3 大津の方法の有効性

大津の方法による自動閾値決定は、ヒストグラムが二峰性を示す画像に対して特に有効である。単峰性のヒストグラムを持つ画像では、適切な閾値が得られない場合もある。

4.4 エッジ強調と二値化の組み合わせ

エッジ強調後に二値化を行うことで、元画像のエッジ成分を効果的に抽出できることが確認された。この手法は、物体の輪郭検出や形状解析に有用である。

5 まとめ

本実験では、PGM-RAW フォーマットの画像に対するエッジ強調および二値化処理を実装した。Prewitt フィルタによる 2 種類のエッジ強調手法と大津の方法による自動二値

化を組み合わせることで、効果的な画像処理システムを構築できた。

実装において、境界処理や値域制限などの工夫により、安定した動作を実現した。また、プログラムの構造化により可読性と保守性を向上させた。

今後の発展として、他のエッジ検出手法（Sobel フィルタや Laplacian フィルタ）の実装や、より高度な境界処理手法の導入が考えられる。

6 課題 4：コンピュータビジョン研究の調査

6.1 選定論文の概要

- 論文名： *An End-to-End Approach for Handwriting Recognition: From Handwritten Text Lines to Complete Pages*
- 著者： Dayvid Castro, Byron L. D. Bezerra, Cleber Zanchettin
- 掲載： CVPR 2024 Workshops

本研究は手書きページ全体を分割せずに一括認識する **DANCER (Document Attention Network for Computationally Efficient Recognition)** を提案する。Octave Convolution と Gated Depthwise Separable Convolution で冗長な空間情報を圧縮した後、FlashAttention と Memory-Efficient Attention を組み合わせた Transformer デコーダで複数行を並列デコードする設計である。ICFHR READ 2016 データセットで、従来法と同等以上の精度を保ちつつ **GPU メモリを最大 65 % 削減し、バッチサイズを 3 倍** に拡大できることを示した。

6.2 背景——「ノートモ」と課題設定

私は大学生が紙ノートを撮影して共有・検索・再利用できる循環型プラットフォーム「ノートモ」を開発・運営している。ユーザが投稿するのはページ全体の画像であり、前処理なしで高速・高精度に *OCR* できる技術がサービス価値を高める重要課題である。

6.3 興味を持った理由（厳選）

1. ノートモのニーズに直結 DANCER はページ分割を不要とし、ユーザが撮影したノート画像をそのまま *OCR* できる。投稿フローの簡素化と全文検索精度の向上に直結する。
2. 計算効率の高さメモリ 65 % 削減・推論高速化により、スマートフォンや We-

bAssembly でのオンデバイス推論が現実的になる。プライバシー保護を重視するノートモの方針と整合する。

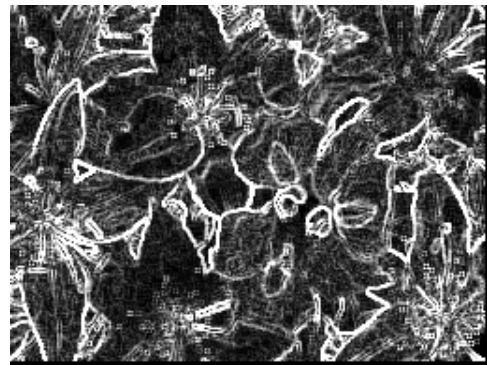
3. レイアウトタグ付き出力文字列内に領域タグを埋め込んで出力できるため、本文・注釈・図形などを UI 上で区別表示しやすく、高度な検索フィルタの実装にも活用できる。

7 実験結果画像

本章では、全 13 サンプルの処理結果画像を示す。各図は 2×2 レイアウトで、左上から時計回りに元画像、エッジ強調、エッジ強調→二値化、二値化の結果を配置している。



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 134)



(d) エッジ強調→二値化 (閾値: 125)

図 1: sample1.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 111)



(d) エッジ強調→二値化 (閾値: 127)

図 2: sample2.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 105)

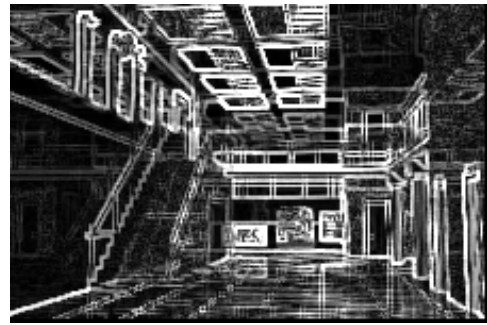


(d) エッジ強調→二値化 (閾値: 125)

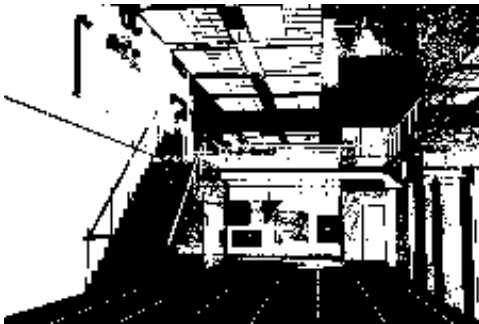
図 3: sample3.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 155)

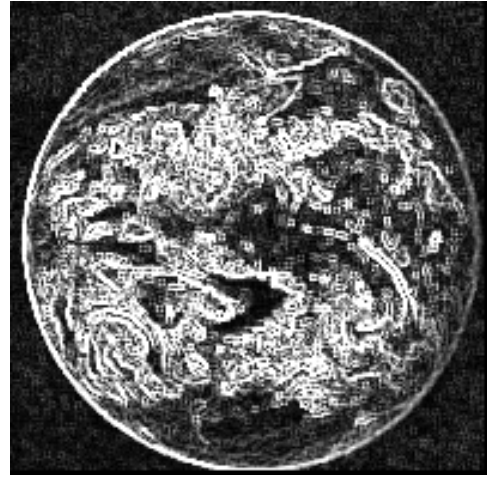


(d) エッジ強調→二値化 (閾値: 112)

図 4: sample4.pgm の処理結果



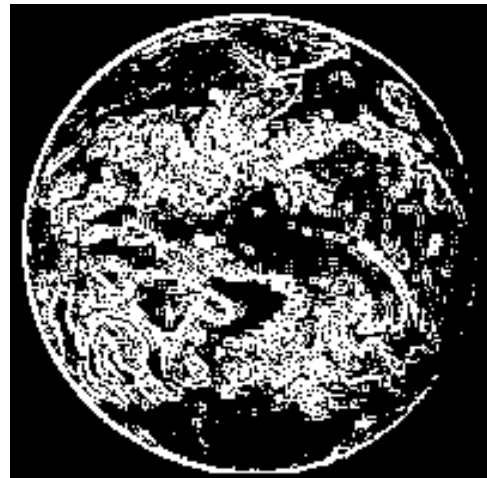
(a) 元画像



(b) エッジ強調

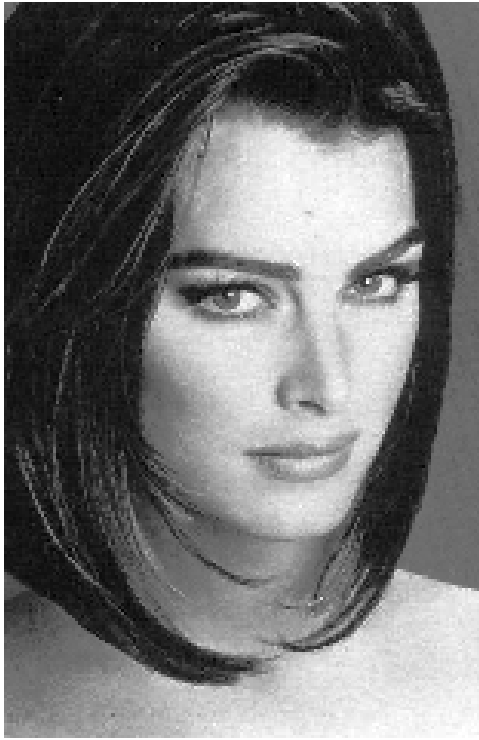


(c) 二値化 (閾値: 128)



(d) エッジ強調→二値化 (閾値: 128)

図 5: sample5.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 127)



(d) エッジ強調→二値化 (閾値: 109)

図 6: sample6.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 107)



(d) エッジ強調→二値化 (閾値: 114)

図 7: sample7.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 136)



(d) エッジ強調→二値化 (閾値: 105)

図 8: sample8.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 162)



(d) エッジ強調→二値化 (閾値: 133)

図 9: sample9.pgm の処理結果



(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 103)



(d) エッジ強調→二値化 (閾値: 126)

図 10: sample10.pgm の処理結果



(a) 元画像



(b) エッジ強調

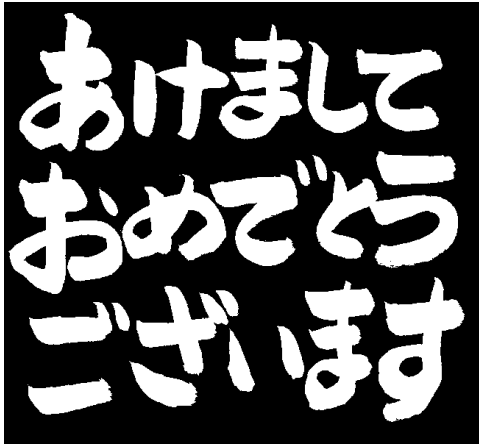


(c) 二値化 (閾値: 117)

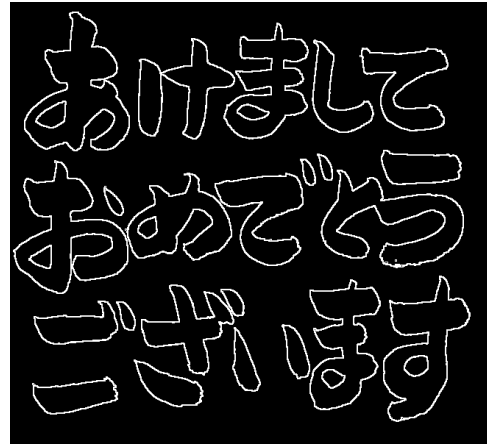


(d) エッジ強調→二値化 (閾値: 100)

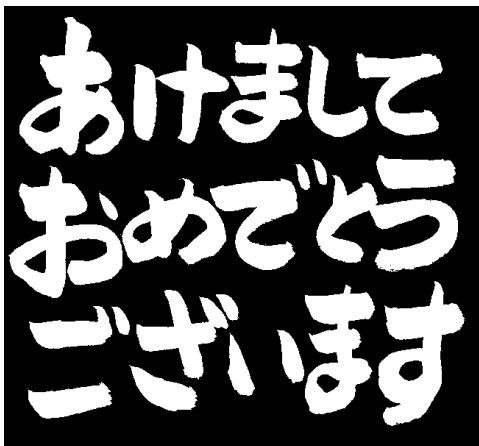
図 11: sample11.pgm の処理結果



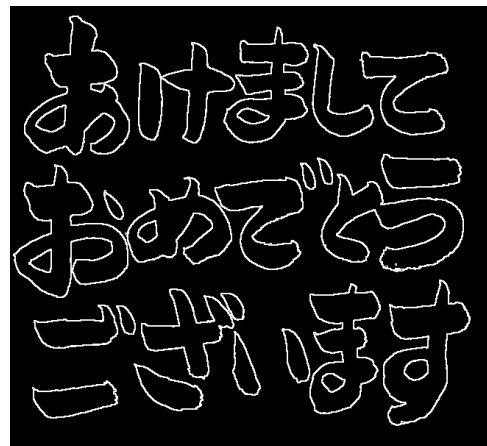
(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 1)

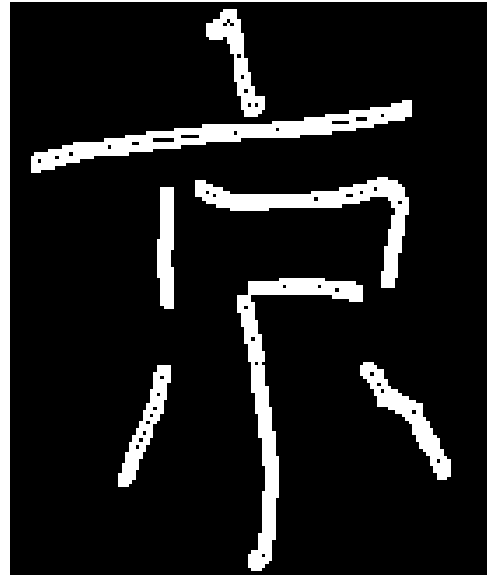


(d) エッジ強調→二値化 (閾値: 1)

図 12: sample12.pgm の処理結果



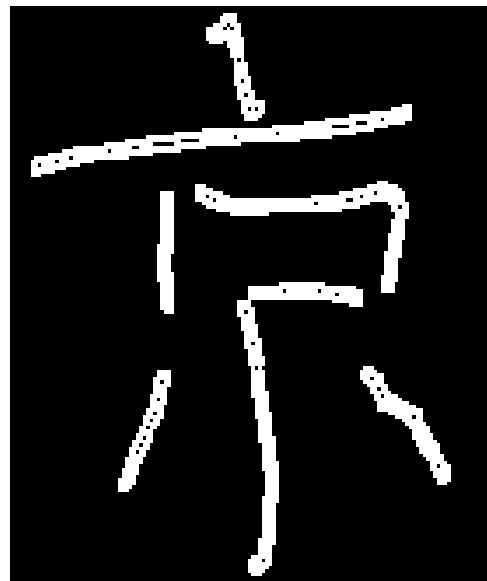
(a) 元画像



(b) エッジ強調



(c) 二値化 (閾値: 1)



(d) エッジ強調→二値化 (閾値: 1)

図 13: sample13.pgm の処理結果