

医用画像工学

1回目 ガイダンス

<http://www.maebashi-it.ac.jp/~odagaki/ImageProc/>

医工学プログラム

小田垣

odagaki@maebashi-it.ac.jp

目 的

- 2次元画像処理技術の基礎、および濃淡画像の画像処理によるパターン認識までの処理技術を習得する

キーワード

- デジタル画像, 濃淡値画像処理, 二値化画像処理, 画像認識, パターン認識

履修して欲しい人

- プログラミングが苦手だけど、もう一度基礎からプログラミングを勉強したい！
- 画像処理は独学で学べるので、プログラムが得意な方は時間の無駄になるかもしれません

スケジュール案(7回目まで)

- 第1回: ガイダンス
- 第2回: 画像の入出力
- 第3回: 画像の表現, 二値化処理
- 第4回: 明るさ, コントラスト変換(レポート)
- 第5回: ヒストグラム
- 第6回: C#を用いた画像処理ソフトの作成
- 第7回: 平滑化, 微分フィルタ

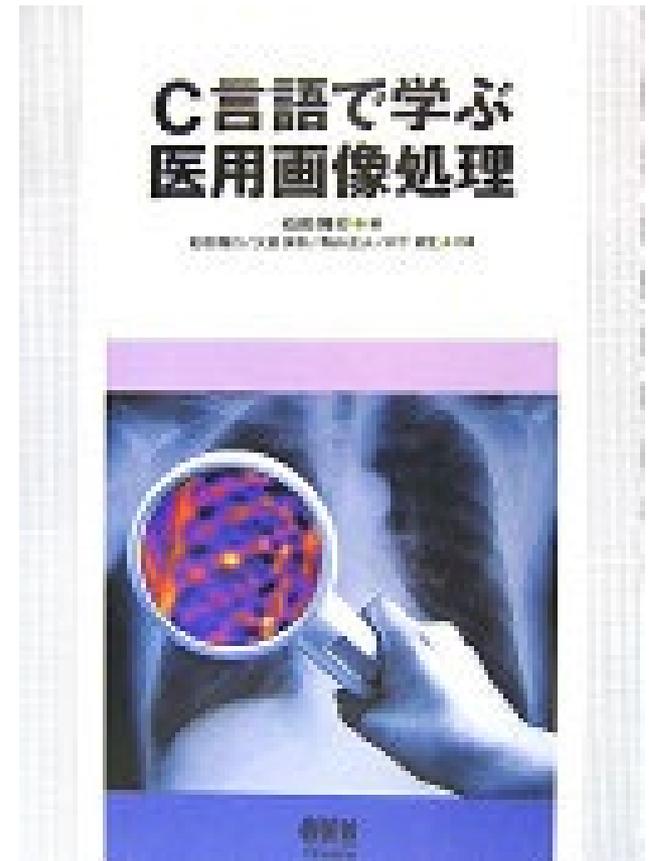
スケジュール案(8回目以降の予定)

- 第8回:二値化画像処理(輪郭追跡など)
- 第9回:ラベリング(レポート)
- 第10回:フーリエ変換
- 第11回:フーリエ変換
- 第12回:アフィン変換
- 第13回:文字・図形の認識
- 第14回:動画画像処理(レポート)
- 第15回:試験

参考書



C言語による画像処理プログラミング入門
長尾 智晴著



C言語で学ぶ医用画像処理
大倉保彦, 青山正人, 川下郁生 著

評 価

- 各回課題の課題を完成させる. 欠席した場合は, 理由を明記のうえ報告すること(事後でも良い)
- 3回程度のレポート提出
- 最終回に行う試験(50%)
 - 授業で扱うアルゴリズムが正しく説明できるか?
 - プログラムの知識を問う

レポートについて

- 以下のフォーマットでレポートを執筆する
 1. 目的
 2. アルゴリズム
 3. プログラムリスト
 4. 結果, 考察
 5. まとめ

参考書

- C言語で学ぶ医用画像処理 大倉・青山・川下
著 オーム社
- C言語による画像処理プログラミング入門ーサン
プルプログラムから学ぶ, 長尾 著

レポートについて

- レポートは出せば良いということではなく“質”を要求する
- インターネットからのコピー&ペーストは一切認めない
- 考察は、結果から考えられることを述べること。“結果が妥当であるか否かの論理立て”，“うまくできない原因”，“どのような工夫が考えられるか”などを書く

プログラミングのすゝめ

- 他人のプログラムを写すだけの人が多い
- それでもSEとして就職する学生が多い

- プログラミング言語の勉強は“語学”と一緒に
- プログラムは、自分で考えて、バグを修正しながら、根気強く勉強するのが一番良い

すぐに教員に頼るのではなく自分たちで考えバグ取りしながらプログラムを作ってほしい

デジタル医用画像

- 医用X線画像：
 - フィルムを用いたアナログ画像



- イメージングプレートや平面検出器を用いたデジタル画像
 - 利点：ネットワークを介したデータ転送する
 - 画像処理や解析することによって診断へ応用する

DICOM画像

- Digital Imaging and Communications in Medicine : 医療用画像規格
- MRIやCTなどの医療画像はこの規格に準拠している
- 画像データのほか検査日付や患者名前等の情報も含まれている(個人情報取り扱いに注意が必要)

DICOM Viewer

| | |
|-------------------------|----------------------|
| MR Acquisition Type | "3D" |
| Magnetic Field Strength | 3.000000(Tesla) |
| Manufacturer | "GE MEDICAL SYSTEMS" |

Acquisition Time "170543"
Acquisition Time "170543"

| | |
|-----------------------------|------------------------|
| Patient's Birth Date | |
| Patient's ID | "ZA4526e92a" |
| Patient's Name | "ZA-TEST" |
| Patient's Sex | |
| Percent Phase Field of View | 90.000000(%) |
| Percent Sampling | 100.000000(%) |
| Photometric Interpretation | "MONOCHROME2" |
| Pixel Bandwidth | 122.070000(Hz/pixel) |
| Pixel Representation | 1 |
| Pixel Spacing | 0.449200, 0.449200(mm) |

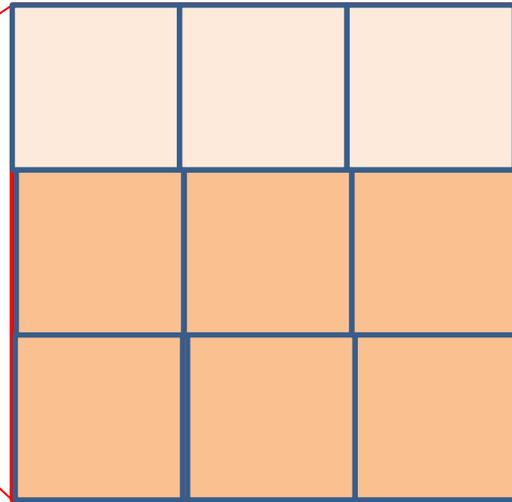
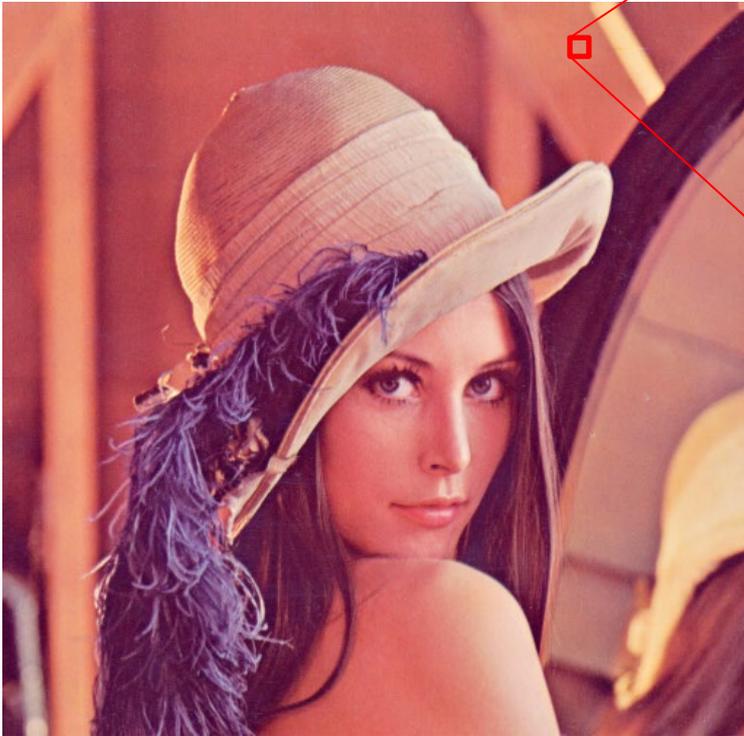
Pixel Bandwidth 122.070000(Hz/pixel)
Pixel Representation 1

リスト非表示



画素

- 画像を構成する最小単位
- 単位は pixel



一画素毎に輝度データによって色が構成されている
(カラー画像ならR, G, B)

カラー画像とグレースケール画像

- カラー画像 : R,G,Bのデータ
- グレースケール画像 : 白から黒のデータ



デジタル画像において輝度情報をどのように表すか？

濃淡分解能

- 8bit ならば 0～255までの256階調の画素値を持つことが出来る
- C言語の “unsigned char” のデータ型に相当する
- このデータサイズは,

```
printf(“%d”, sizeof(unsigned char));
```

とすれば確認できる.

C言語の復習

- ファイル入出力

文字列“123”をテキストファイルに書き出すプログラムを作成する

ファイル入出力

- ファイルオープン処理

```
FILE *fp;
```

```
if (( fp = fopen("file.txt","w"))== NULL){  
    printf("ファイルオープンエラー")  
    return 0;  
}
```

fopen関数

```
FILE *fopen(const char *filename, const char *mode)
```

const char *mode

| モード | モードの説明 |
|------------|--|
| r | 読み込みモードで開く。 |
| w | 書き込みモードで開く(ファイルが既に存在する場合は上書きする。) |
| a | 追加(append)モードで開く(書き込みは、ファイルの末尾に追加される)。 |
| r+, w+, a+ | 更新モードで開く(このモードでは、読み込みと書き込みの両方が可能)。 |

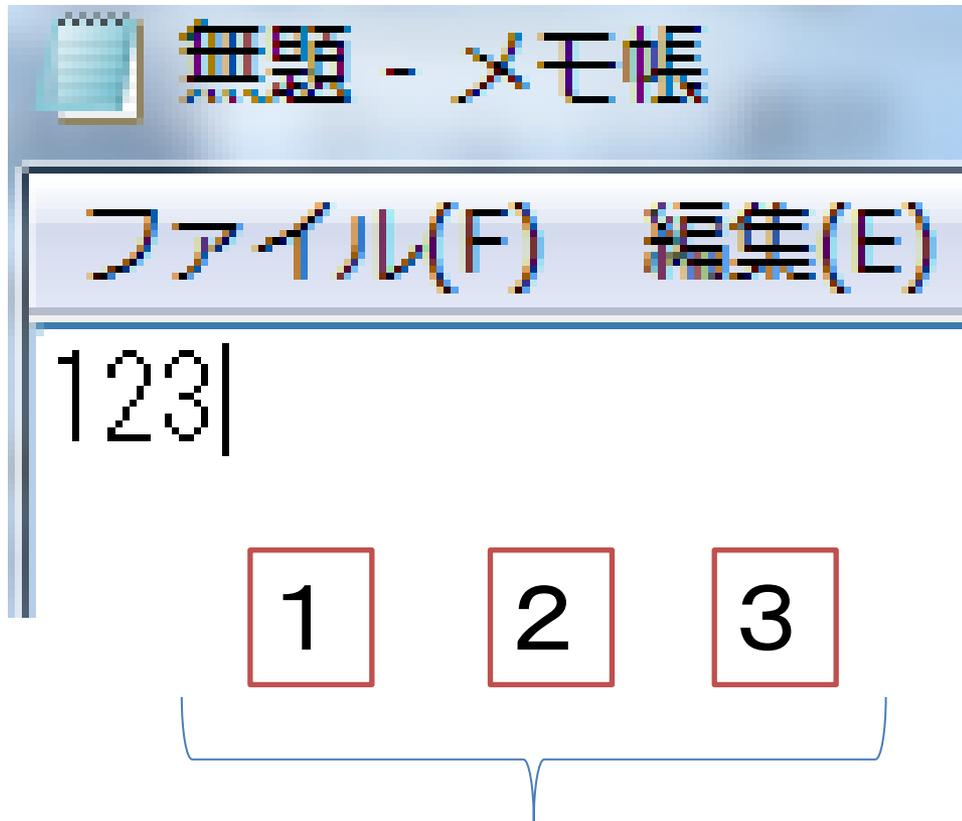
例

例えば、テキストモードでファイルを開き、

```
fprintf(fp, "123");
```

と文字列“123”を書き出す

テキストモード



ASCIIコードの'1', '2', '3' = 3文字分

testのプロパティ

全般 セキュリティ 詳細 以前のバージョン

test

ファイルの種類: テキスト ドキュメント (.txt)

プログラム: メモ帳 変更(C)...

場所: C:\Users\odagaki_lab\Documents\画像処理

サイズ: 3 バイト (3 バイト)

ディスク上のサイズ: 4.00 KB (4,096 バイト)

作成日時: 2013年7月27日 11:00:04

更新日時: 2013年7月27日 11:00:04

アクセス日時: 2013年7月27日 11:00:04

属性: 読み取り専用 (R) 隠しファイル (H) 詳細設定(O)...

OK キャンセル 適用(A)

サイズ: 3 バイト (3 バイト)

ディスク上 4.00 KB (4,096 バイト)

バイナリデータの扱い

rb 読み込み用のバイナリファイル

wb 書き込み用のバイナリファイル(ファイルが既に存在する場合は上書き)

課題1

- バイナリファイルの入出力について理解する
- バイナリモードでファイルをオープンし,

```
fprintf(fp,"%c",(unsigned char)123);
```


もしくは

```
fputc((unsigned char)123,fp);
```


としてファイルを書き出す
- 出来上がったファイルサイズが1バイトであることを確認する

fputc((unsigned char)123,fp)の意味

- キャスト: int型の整数 123を符号なしchar型に変換する
- ASCIIコード123に対応する文字は'{'

課題2

- バイナリファイルに

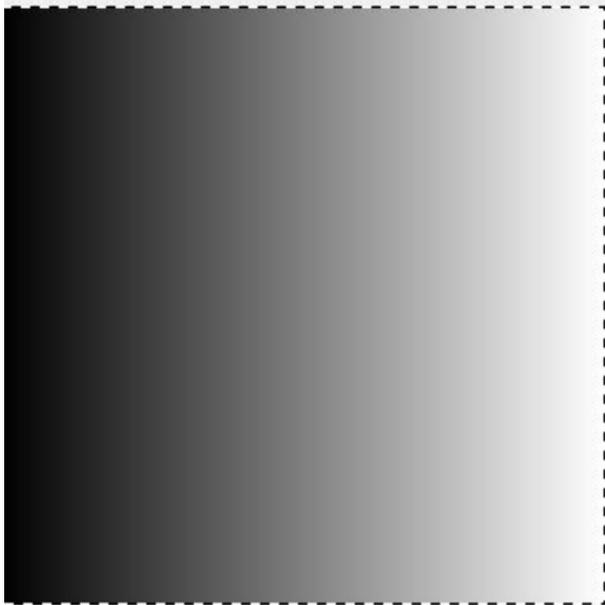
```
for(int i=0;i<256;i++){  
    for(int j=0;j<256;j++){  
        fputc((unsigned char)j,fp);  
    }  
}
```

とファイル出力する。

画像データの確認

- 完成したらGIMPでファイルを開く
 - 画像の種類をインデックスとする
 - 幅,高さをどちらも256とする

原点(0,0)



Raw形式画像が作成できる

輝度値

- 8ビット(256階調)の画像における輝度値と出力される色の関係

0

255



課題3

- 横方向のグラデーションではなく、縦方向のグラデーションに変更して画像ファイルを出力する
- 黒から白へのグラデーションではなく、白から黒へのグラデーションへ変更する

まとめ

- 画像ファイルは各画素の色情報が集まったものである
- 画像データはバイナリデータなので"wb"や"rb"等のモードでファイル入出力を行う
- 256階調のデータはunsigned char型の変数で扱うことができる