情報デザイン専攻

画像情報処理論及び演習II

-画像ファイルフォーマット-
前期の復習と後期の予定・BMP

第1回講義
水曜日１限
教室6218

吉澤　信
shin@riken.jp, 非常勤講師
大妻女子大学　社会情報学部
独立行政法人 理化学研究所

今日の授業内容

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec15.pdf

1. 講義について.
2. 後期の予定.
3. 画像ファイルフォーマット、圧縮・符号化.
4. 演習：BMPの入出力.

今日の演習は後期全ての演習・レポートで使う内容なのでみなさん頑張ってくださいねーp(^^)q

自己紹介

✓ **講師：　吉澤　信　（よしざわ　しん）**
- **本務：(独)理化学研究所　研究員**
- **専門：デジタル幾何学・CG/CAD・画像処理**
- **E-Mail：shin@riken.jp**
- **URL：**www.riken.jp/brict/Yoshizawa/

✓ **TA：　丸山　典宏　（まるやま　のりひろ）**
- **所属：東京大学 大学院 博士課程1年**

よろしくお願いします！

本講義について

✓ 目的：デジタル画像処理の基礎知識と技術の習得
- 画像処理の楽しさを知る.
- 役に立つ事を知る.　前期と同じ、ただし、より」を重視.
- **画像処理の基礎的なプログラミングを習得.**

✓ 教科書：なし、毎回講義資料と演習課題を印刷して渡します.

✓ 参考書：
- 「ディジタル画像処理」、CG-ARTS協会、2006.
- 「画像処理アルゴリズム」、斉藤恒雄著、近代科学社、1993.
- 「Digital Image Processing」, R. Gonzalez & R. Woods著, Pearson Edu. Inc., 2008.

本講義について

✓ 講義のHP：

www.riken.jp/brict/Yoshizawa/Lectures/index.html

- 講義資料.
- 演習課題・プログラムの雛形.
- レポート・提出先.

✓ 前期のHP：後期のHP↑からリンクを張っています.

www.riken.jp/brict/Yoshizawa/Lectures/Semester1.html

- 前期講義を取ってない人や、忘れちったr(^ω^*)という人は↑をよく復習しておいてください.

本講義について：授業の進め方

✓ 講義：画像処理の背景・理論・アルゴリズム・プログラミング・応用に関する講義.

✓ 演習：講義の内容をプログラミング（基本的にLinux環境でC言語＋Java言語）.

✓ 課題：講義と演習の内容をより理解するための課題を解き、レポートとして提出.

✓ 評価方法：
- 出席40%：遅刻は少し減点(0.8倍).
- レポート60%：2～3回に1回・次々週までに提出.
- テスト：なし.　前期と同じ!

Shin Yoshizawa: shin@riken.jp

本講義について：その他コメント

- ✓ 1限ですが、頑張って授業に来て下さい.
- ✓ 分からないところは遠慮なく質問してください.
  - 講義で話している途中でも可.
  - 授業後でも可、メールでの質問も可：shin@riken.jp
  - 授業に関しての意見も可.
- ✓ 課題や演習は他の学生さんと相談してもOK、でもコピーはダメです：
  - レポートやプログラムのコピーは(少し変えても)すぐに分かります.

前期と同じ!

Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

シラバス		少し変えます
1回	フィルターと雑音処理、輪郭の抽出	1回 画像フォーマット・I/O
2回	フィルターと雑音処理、輪郭の抽出	
3回	フィルターと雑音処理、輪郭の抽出	3回 周波数分解
4回	フィルターと雑音処理、輪郭の抽出	
5回	フィルターと雑音処理、輪郭の抽出	
6回	動画プログラム jpeg,Mpegの基礎	4回 フィルタ処理・エッジ強調
7回	動画プログラム	
8回	動画プログラム	1回 HDRI・計算Photography・Artistic Stylization
9回	動画プログラム	
10回	動画プログラム	
11回	3Dの基礎とプログラム ワイヤーフレームモデル 陰線処理と光線	4回 動画像処理
12回	ワイヤーフレームと移動、回転	
13回	ソリッドモデル	
14回	アナグリフと3D プログラムの作成	2回 エッジ・形状・特徴抽出とパターン認識の基礎 +補講?
15回	第14回の続き	

Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

1回 画像フォーマット
2回
3回 周波数分解
4回
5回 フィルタ処理・エッジ強調
6回
7回
8回
9回 計算Photography・Artistic Stylization
10回 動画像処理
11回
12回
13回
14回 エッジ・形状・特徴抽出とパターン認識の基礎 +補講
15回

内容(1)：画像フォーマット
1：画像ファイルフォーマット、符号化、圧縮、BMP等.

BITMAPFILEHEADER 構造体
ビットマップ情報ヘッダ
(下記のいずれかの構造体)
BITMAPINFO
- BITMAPCOREHEADER
- BITMAPINFOHEADER
- BITMAPV4HEADER (Windows 95/NT4.0)
- BITMAPV5HEADER (Windows 98/Me/2000/XP)

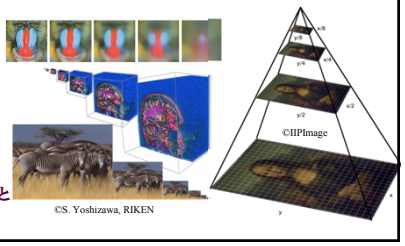
カラーマスク (BI_BITFIELDS の場合のみ)
カラーテーブル (1/4/8 BPP では必須、16/24/32 BPP ではオプション)
(ここに隙間がある場合もある. (ない場合は packed bitmap と呼ばれる.))
ビットマップデータ
プロファイルデータ (オプション)

Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

1回 画像フォーマット
2回
3回 周波数分解
4回
5回 フィルタ処理・エッジ強調
6回
7回
8回
9回 計算Photography・Artistic Stylization
10回 動画像処理
11回
12回
13回
14回 エッジ・形状・特徴抽出とパターン認識の基礎 +補講
15回

内容(2－4)：周波数分解
- ✓ **フーリエ変換と周波数操作、多重解像度解析.**
- ✓ **Gaussianフィルタ等.**

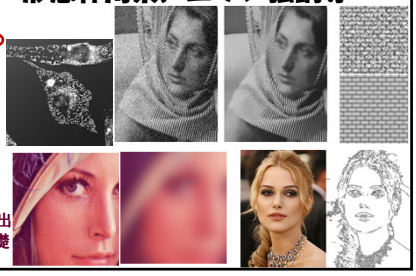


Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

1回 画像フォーマット
2回
3回 周波数分解
4回
5回
6回 フィルタ処理・エッジ強調
7回
8回
9回 計算Photography・Artistic Stylization
10回 動画像処理
11回
12回
13回
14回 エッジ・形状・特徴抽出とパターン認識の基礎 +補講
15回

内容(5－8)：
フィルタ処理・エッジ強調
ノイズ除去、平滑化、画像復元、形態作用素、エッジ強調等.

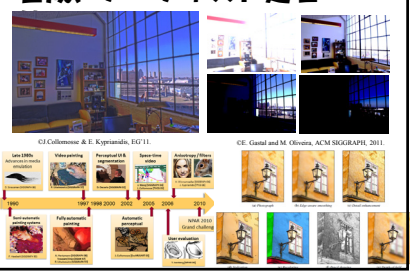


Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

1回 画像フォーマット
2回
3回 周波数分解
4回
5回
6回 フィルタ処理・エッジ強調
7回
8回
9回 計算Photography・Artistic Stylization
10回 動画像処理
11回
12回
13回
14回 エッジ・形状・特徴抽出とパターン認識の基礎 +補講
15回

内容(9)：HDRI・計算Photography・Artistic Stylization
合成、アーティスト処理・NPR



Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

内容(10－13)：動画像処理基礎、スタイル化合成等.

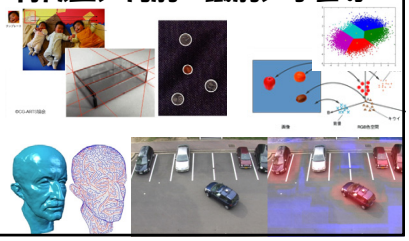
1回 画像フォーマット
2回
3回 周波数分解
4回
5回
6回 フィルタ処理・エッジ強調
7回
8回
9回 計算Photography・Artistic Stylization
10回
11回 動画像処理
12回
13回
14回 エッジ・形状・特徴抽出とパターン認識の基礎＋補講
15回

Shin Yoshizawa: shin@riken.jp

講義・演習内容の予定

内容(14－15)：エッジ・形状・特徴抽出とパターン認識の基礎 微分幾何学の基礎、形状検出、特徴量、判別・識別、学習等.

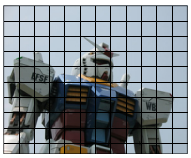
1回 画像フォーマット
2回
3回 周波数分解
4回
5回
6回 フィルタ処理・エッジ強調
7回
8回
9回 計算Photography・Artistic Stylization
10回
11回 動画像処理
12回
13回
14回 エッジ・形状・特徴抽出とパターン認識の基礎＋補講
15回



Shin Yoshizawa: shin@riken.jp

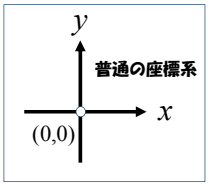
復習：デジタル画像とは？

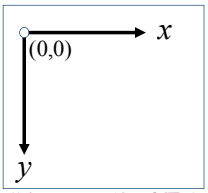
- ✓ **デジタル画像(Raster)：コンピュータ内で表現されたデータ付正規直交格子(画素の集まり).**
- ✓ **画素：格子の最小構成要素：格子1個.**
  - 2次元：ピクセル(Pixel).
  - 3次元：ボクセル (Voxel).
- ✓ **画素値：明度や色の数値.**
  - グレースケール画像：明るさ(明度).
  - カラー(色)画像：RGB, CMY等.
- ✓ **画素値のビット数：色数.**
  - 8bit画像：2の8乗で256色、グレースケールの場合は0から255までの256段階の明度.16bit画像なら2の16乗で65536段階. RGB毎に8bitなら256の3乗で16777216色.



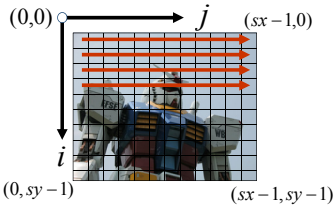
Shin Yoshizawa: shin@riken.jp

復習：デジタル画像の座標と配列





画像処理でよく使う座標系



輝度値の配列表現：

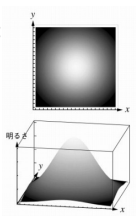
```
int I[sy][sx];
double I[sy][sx];
for(i = 0; i < sy; i++){
  for(j = 0; j < sx; j++){
    I[i][j] = ...
  }
}
```

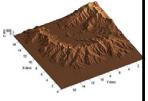
Shin Yoshizawa: shin@riken.jp

復習：デジタル画像の数式表現

輝度値の配列表現：

```
int I[sy][sx];
double I[sy][sx];
```





輝度値の数式表現：高さ関数

$z = I(x, y)$ **又は** $z = I(\mathbf{x}), \quad \mathbf{x} = (x, y)$

カラー画像： $\mathbf{z} = \mathbf{I}(x, y) = (R(x, y), G(x, y), B(x, y))$

又は $\mathbf{z} = \mathbf{I}(\mathbf{x}) = (R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x})), \quad \mathbf{x} = (x, y)$

Shin Yoshizawa: shin@riken.jp

画像フォーマット

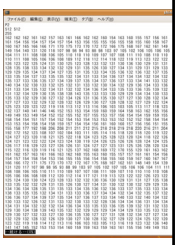
- ✓ 数百～の画像フォーマットがある！
- ✓ 代表的な画像フォーマット:
  - BMP, PNG, PNM(ppm,pgm), GIF, JPEG, TIFF, PS, EPS,…
- ✓ 医用画像フォーマット(CT, MRI等):
  - DICOM, Acr/Nema, Analyze(SPM), Concore/μPET, CTI ECAT, NIfTI-1, InterFile…
- ✓ 動画像フォーマット:
  - ASF(wmv等), AVI, MPEG (mpg,mp4等), DVD, RealVideo, DviX, Flash(flv), QuickTime, MP4,…
  - Animated Gif, multipage TIFF, 3次元画像…

Shin Yoshizawa: shin@riken.jp

復習：pnm画像フォーマット

- ✓ 一番簡単な画像フォーマットです：
  - グレースケール画像は「.pgm」、カラー画像は「.ppm」でテキスト形式とバイナリー形式があります.
  - グレースケール(.pgm):

1行名: テキストで「P2」
2行目： 画像サイズ(横：width 縦：height)
3行目：画素の階調(最大値) 8bitの場合は255
4行目から: integerで画素値スペース画素値…



  - カラー(.ppm):

1行名: テキストで「P3」
2行目： 画像サイズ(横：width 縦：height)
3行目：画素の階調(最大値) 8bitの場合は255
4行目から: integerでR G B R G B R G B…

Shin Yoshizawa: shin@riken.jp

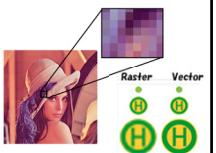
画像フォーマット2

- ✓ 符号化(encode)：データに暗号化、圧縮、バイナリーデータ化等の変換を行う事.
- ✓ 復号(decode): 符号化されたデータを復元する事.
  - Codec: 符号化方式(ファイルフォーマット)を用いてデータのencode/decodeを行う装置・ソフト.
- ✓ データ圧縮： 重要な情報を保持しながらデータ量を減らす符号化. 逆の操作をデータ解凍とも呼ぶ.
  - 可逆(lossless)符号化：圧縮されたデータから元のデータを完全に復元出来る方式: ランレングス符号化、ハフマン符号化等.
  - 不可逆(lossy)符号化： 圧縮されたデータから元のデータを完全には復元出来ない方式: DCT等の周波数成分の除去(「周波数分解」の講義で解説します).

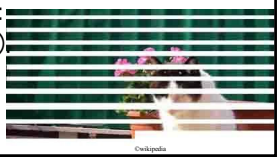
Shin Yoshizawa: shin@riken.jp

画像フォーマット3、復習(前期Lec03.pdf)

- ✓ ラスター vs ベクター.
- ✓ 色深度(量子化の解像度=bit数).
- ✓ インデックスカラー(限定色).
- ✓ 透明度(アルファチャンネル).
- ✓ メタデータ・拡張データ.
- ✓ インターレース： 描画・転送の高速化などの為にスキャンラインを飛び飛びに走査・処理する事.



- ✓ HDR(High Dynamic Range): 高階調(HDRの講義でやります)
- ✓ CMS(Color Management System): 色補正設定.
- ✓ animation, multipage, etc.



Shin Yoshizawa: shin@riken.jp

可逆符号化

- ✓ 連長圧縮(Run Length Encoding):
  - 連続したデータに対して、同じ符号が連続していくつ並んでいるかを記述する方法(TIFF, BMPの一部等).
  - 例：「AAABBCCCCAAA」のデータ列なら「A3B2C4A3」と記述. 同じ符号が連続している程圧縮率が高い.
- ✓ 拡張:
  - 同じ符号が並んでいる部分だけ適用: 「ABCDDD」なら「A1B1C1D3」ではなく「ABCD3」.
  - Pack Bits: 連続するデータが現れるまでの数を記述：「AAABBBBCDF」なら「3A4B-3CDF」、-3は三つ連続しない符号有りの意味.
  - Switch Run Length: Pack Bitsと通常方法の組み合わせ: 「ABCCCCDDD」なら「3ABC31D3」.

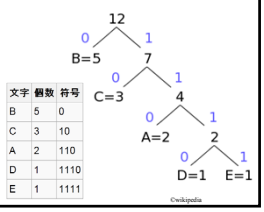
Shin Yoshizawa: shin@riken.jp

可逆符号化2

- ✓ ハフマン符号化(Huffman Encoding):
  - 連続したデータに対して、同じ符号の出現頻度を求めて木構造(n進数ならn分木)の葉(leaf)にして木構造を構成しデータを符号化する方法(PNG, JPEGの一部等).
  - 最初に頻度を計算しておく静的ハフマン法と木構造を符号が入力される度に更新する動的ハフマン法がある.

1. 出現頻度作成.
2. Leafにデータ&頻度格納.
3. 頻度最小の節を繋げ頻度の和を格納を繰り返す.

- 例「DAEBCBACBBBC」→



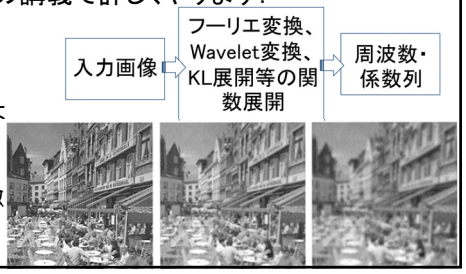
文字	個数	符号
B	5	0
C	3	10
A	2	110
D	1	1110
E	1	1111

Shin Yoshizawa: shin@riken.jp

予習：周波数分解と圧縮(JPEG)

- ✓ 不可逆画像圧縮で最も用いられているのが、フーリエ変換やWavelet等の周波数分解と高周波の除去: JPEGはDCT (Discrete Cosine Transform)、周波数分解の講義で詳しくやります.

- 簡単に言うと平滑化、人間の目は高周波(細かいエッジ)の削除に敏感でない→



Shin Yoshizawa: shin@riken.jp

BMPフォーマット

- ✓ BMP(Microsoft Windows Bitmap Image)、又はDIB(Device Independent Bitmap)はWindows(Microsoft)とOS2(IBM)に分かれる前に共同開発されたフォーマット.
  - ビットマップとは通常ラスター画像全般を指すので、BMPだけがビットマップではない.
  - 通常圧縮されていない.
  - WindowsとOS2の複数のVer.によるバイト形式.
  - 演習のBMPIO.hに記述.

Shin Yoshizawa: shin@riken.jp

演習: BMP

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec15.pdf

BMP画像の入出力・フォーマット変換
www.riken.jp/brict/Yoshizawa/Lectures/Ex07.zip

- ✓ 再来週までに必ず今日の演習(ppmとBMPの相互変換)が出来る様にTryしてください!
- ✓ わからなかったら直接 or e-mail等で遠慮なく質問してください.
- ✓ 今日の内容を復習・Tryしないと、たぶん後期の単位は取れないですp(≧口≦)q.

Shin Yoshizawa: shin@riken.jp

復習:重要

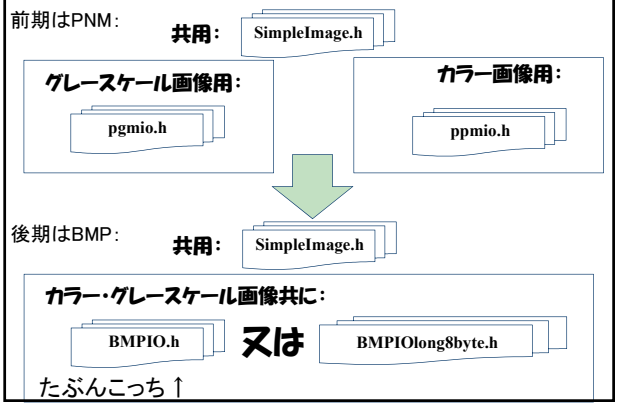
必ず使える様になってね!:ls、cd、pwd:
端末(コンソール)にて打ち込みエンターキーで実行.

- cd: ディレクトリー(フォルダー)の移動.
  「cd ディレクトリー名」
- ls:ディレクトリー内のファイル名・フォルダー名を表示.「ls ディレクトリー名」、「ls ./」「ls ../」.「ls -lh」、「ls -alh」
- pwd: 現在のディレクトリーを表示.「pwd」

- ✓ ファイル名・ディレクトリー名に日本語はダメ!
- ✓ プログラムのソースコードにコメント以外では、日本語は使わない事!

Shin Yoshizawa: shin@riken.jp

後期はグレースケールもカラーもBMPを使います



Shin Yoshizawa: shin@riken.jp

復習:Imageクラス

SimpleImage.h: 2次元配列で一色の画像を表すImageクラス.

#include"SimpleImage.h"**した後の使い方例:**

宣言・メモリ確保(allocation):

```
Image *in = new Image();
Image *out = new Image(in->sx,in->sy);
```

処理:

```
/* Only Copy */
for(i=0;i<in->sy;i++)
 for(j=0;j<in->sx;j++){
   out->img[i][j] = in->img[i][j];
 }
```

注:↑は画像を読み込んだ後!

- ✓ 画像サイズ:縦:sy、横sx.
- ✓ **(座標(i,j)での)画素値:img[i][j]**

メモリの開放:

```
delete out;
delete in;
```

SimpleImage.h

Shin Yoshizawa: shin@riken.jp

演習:longのバイト数をチェック

www.riken.jp/brict/Yoshizawa/Lectures/Ex07.zip

1. ↑をダウンロードしてください.
2. 適当なフォルダーにEx07.zipを展開してください.
3. Ex07内のプログラムをmakeでコンパイルしてください.
4. 端末にて./testBMPIO lena.bmp lena_test.bmpを実行.
5. displayでlena_test.bmpを確認.

- ✓ ex07.cxxを編集してBMPIO.hとBMPIOlong8byte.hどちらが使えるかチェックしてみましょう!
- ✓ ヒント:
  - printf("size of long is %d¥n",sizeof(long));等を使ってlongのバイト数を表示してみましょう.
  - ¥は「Back space」の左にあるバックスラッシュ記号.

演習：Ex07の説明1

- ✓ BMPIO.h: BMPファイルの入出力(longが4バイトのOS、sizeof(long)==4).
- ✓ BMPIOlong8byte.h: BMPファイルの入出力(longが8バイトのOS、sizeof(long)==8).
  - 基本全てカラー画像として扱う、グレースケールの場合はR=G=B=Grayで入出力.
  - カラーパレット、圧縮等には対応していない: readBMPSize()の戻り値がtrueの場合にfilenameで指定したBMPファイルの入力が可能.
  - OS2/Windows 12byte, 40byte, 108byte, 124byteに対応.
- ✓ bool readBMPSize(int *sx,int *sy, char *filename)
  - filenameで指定したBMPファイルのサイズを縦(sy)、横(sx)に代入する関数. 対応していないBMPファイルを開こうとしていると戻り値がfalseになる(convertなどで変換するればOK).

演習：Ex07の説明2

- ✓ void readBMP(Image *R, Image *G, Image *B, char *filename)
  - filenameで指定したBMPファイルをImageクラスに入力.
  - 注意：ImageクラスはSimpleImage.h及び前期の演習01を参照.
- ✓ void saveBMP(Image *img, char *filename)
  - filenameで指定したファイルへグレースケール画像をBMPフォーマットで保存.
- ✓ void saveBMP(Image *R, Image *G, Image *B, char *filename)
  - filenameで指定したファイルへカラー画像をBMPフォーマットで保存.
- ✓ testBMPIO.cxx
  - BMP画像を開いてBMP画像としてセーブするプログラム.
  - 引数3：入力BMP 出力BMP(カラー) 出力BMP(グレースケール)

Imageクラス＋BMPの流れ

- ✓ testBMP.cxxをemacsで開いてみてください.

① BMPIOクラスをnew.
② readBMPSize()で画像サイズを確保.
③ 画像クラスを取得したサイズでnew.
④ readBMP()で画像を読み込む.

⑤ 処理….

⑥ saveBMP()で画像を保存.
⑦ new したオブジェクトをdelete.

注：グレースケールに変換する部分は省いてあります.

```
int main(int argc,char *argv[]){

  BMPIO *mybmp = new BMPIO();  ①
    Image *R= NULL;
  Image *G= NULL;
  Image *B= NULL;
  int sx,sy;

  if(mybmp->readBMPSize(&sx,&sy,argv[1])){  ②

    R = new Image(sx,sy);
    G = new Image(sx,sy);  ③
    B = new Image(sx,sy);

    mybmp->readBMP(R,G,B,argv[1]);  ④

    //処理
    int i,j;
    for(i=0;i<R->sy;i++)
      for(j=0;j<R->sx;j++){  ⑤
        //処理
      }

    mybmp->saveBMP(R,G,B,argv[2]);  ⑥

  }

  delete mybmp;  ⑦

  if(R!=NULL)delete R;
  if(G!=NULL)delete G;
  if(B!=NULL)delete B;

  return 0;
}
```


演習07-1：ppmとbmpの変換

- ✓ ex07.cxxを編集して以下の二つのプログラムを作ってみましょう！
  - bmp2ppm: bmp画像を読み込んでppm画像としてセーブするプログラム.
  - ppm2bmp: ppm画像を読み込んでbmp画像としてセーブするプログラム.
  - ヒント：ppmの入出力はppmio.hを使う(ex01_2.cxx又は前期演習01を参照).
  - カラー画像で確認する事.
- ✓ ↑が出来た人はpgm2bmpとbmp2pgmも作ってみてください.
- ✓ Makefileを編集して上記4つのプログラムがmakeでコンパイル出来る様にしてみましょう.

第1回レポートは↑を含むので頑張って一p(^^)q

演習：出来ちゃった人

- ✓ BMPの入出力が出来ちゃった人は前期のLec14.pdfにある「出来る人のための課題」をBMPを使ってやってみてください. 同様の演習はフィルタ処理でやりますし、レポートに出します.

再来週の予定

1回	画像フォーマット
2回	
3回	周波数分解
4回	
5回	フィルタ処理・エッジ強調
6回	
7回	
8回	計算Photography・Artistic Stylization
9回	
10回	動画像処理
11回	
12回	
13回	
14回	エッジ・形状・特徴抽出とパターン認識の基礎＋補講
15回	

内容(2－4)：周波数分解

- ✓ **フーリエ変換と周波数操作、多重解像度解析.**
- ✓ Gaussian**フィルタ等.**