情報デザイン専攻

画像情報処理論及び演習II

-動画像処理-
基礎、連番画像の入出力

第10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

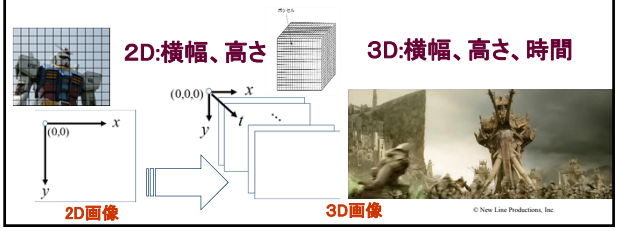
www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec21.pdf

1. 連番画像の入出力:今回から後期の終わりまで使う、
   1. 3D画像クラス.
   2. フォルダー内のBMP画像を名前でソートして入出力を行う方法.
2. 演習:　今日はプログラミングの話メイン.
   1. 3D画像クラスの作成.
   2. 連番画像の入出力.

重要:↑は次回レポートの内容なので頑張ってp(^ ^)q
＋今日作るプログラム(クラス)を次回以降の演習で使うので必ず来週までに作成してください！

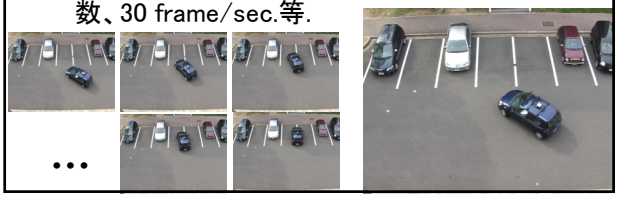
動画像の基礎

- ✓ 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, …
- ✓ 理論/数学的には1次元増えただけ⇒3D画像.



動画像の基礎2

- ✓ 講義では複数の2D画像の組で3D画像を扱う.
  - 画素:ピクセル(2D)→ボクセル(3D).
  - サイズ:(sx,sy)→(sx,sy,st).
  - 輝度値:2次元配列→3次元配列.
  - ループ:　2重→3重.
  - フレームレート: 単位時間のフレーム(2D画像)数、30 frame/sec.等.

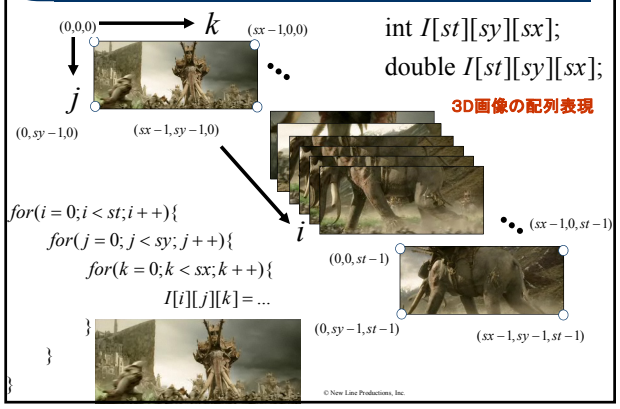


動画像の基礎3

- ✓ 複数2D画像ファイル⇔動画フォーマットの変換:
  - 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を行うコーデックが必要.
  - フリーのソフトを使うのが簡単で良い.
  - 例えばWinでは、AVIMaker(bmp→avi)やAviUtl(bmp⇔avi):
    http://www.vector.co.jp/soft/dl/win95/art/se121264.html
    http://spring-fragrance.mints.ne.jp/aviutl
  - http://www.vector.co.jpに色々な動画⇔画像ソフトがあるので、みんな独自のビデオを連番bmp画像にしてみましょう！
  - Linuxでは機能が多彩で難しい！画像・動画⇔動画: ffmpeg
  - 簡単！複数bmp⇔gifアニメ(Linux): convert
  - 動画へ「convert *.bmp 出力.gif」
  - 画像へ「convert 入力.gif 出力.bmp」

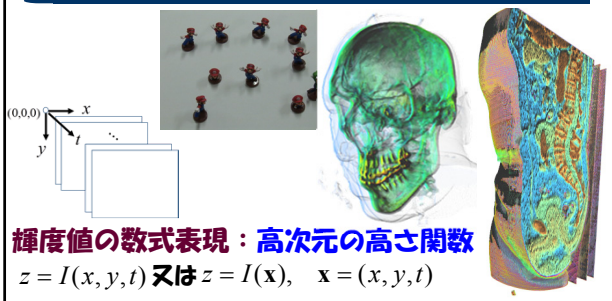
番号を揃えたい場合はCのprintfの表記と同じに
「convert 入力.gif 出力%0桁数d.bmp」とする. 例えば3桁なら
「convert 入力.gif 出力%03d.bmp」

動画像の配列表現



Shin Yoshizawa: shin@riken.jp

動画像の数式表現



輝度値の数式表現：高次元の高さ関数

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

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

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

Shin Yoshizawa: shin@riken.jp

3D画像クラスの作成

- ✓ 3D画像クラス: Image3DクラスをSimpleImage3D.hというヘッダーファイル名で作ってみる.
  www.riken.jp/brict/Yoshizawa/Lectures/Ex14.zip
- ✓ 必要なクラスのメンバー/メソッド:
  - 画像サイズ(int)で三つsx,sy,st.
  - 輝度値を格納するためのdoubleの３重ポインター.
  - コンストラクター二つ:
    - 引数無: サイズにゼロ、輝度値のポインターにNULLを代入する.
    - 引数画像サイズ: 輝度値の３重ポインターのメモリを確保して3次元配列にする.
  - デコンストラクター:クラスがdeleteしたとき輝度値の3次元配列をdeleteする.

Shin Yoshizawa: shin@riken.jp

C++クラスの基礎

```
class クラス名{   /* 設計図の様なものでクラス=新しい型 */
    public:   /* パブリックの場合は、クラスの外から参照可能 */
    メンバー変数   /* クラスが持っている変数、構造体、クラス内クラス */
    クラス名(){   /* コンストラクター:newされたときに呼ばれる. */
    }
    クラス名(引数){} /* コンストラクターは複数あってよい */

    ~クラス名(){   /* デコンストラクター:delete されたときに呼ばれる. */
    }
    戻り値 メソッド名(引数){} /* メソッドを作れる= */
     private: /* プライベートの場合は、クラスの外から参照不可 */
};
```

Shin Yoshizawa: shin@riken.jp

多重ポインターから多次元配列を作る方法

- ✓ １重ポインターから１次元配列を作る方法:

```
double *AAA = new double[N];
```

これで、A[0], A[1], …A[N-1]まで配列として使える.

- 使い終わったらメモリの開放が必要:`delete [] AAA;`

- ✓ 2重ポインターから2次元配列を作る方法:

```
double **AAA = new double *[N];
for(int i=0;i<N;i++)AAA[i] = new double[M];
```

これで、A[0][0], A[0][1], …A[0][M-1], A[1][0], A[1][1],…A[N-1][M-1]まで配列として使える.

- 使い終わったらメモリの開放が必要:

```
for(int i=0;i<N;i++) delete [] AAA[i];
delete [] AAA;
```

Shin Yoshizawa: shin@riken.jp

多重ポインターから多次元配列を作る方法2

- ✓ 3重ポインターから3次元配列を作る方法:

```
double ***AAA = new double **[st];
for(int i=0;i<st;i++){
    AAA[i] = new double *[sy];
    for(int j=0;j<sy;j++)AAA[i][j] = new double[sx];
}
```

これで、A[0][0][0], A[0][0][1], …A[0][0][sx-1], A[0][1][0], A[0][1][1], …A[0][sy-1][sx-1], A[1][0][0], A[1][0][1],…A[st-1][sy-1][sx-1]まで配列として使える. 同様にメモリの開放は以下:

```
for(int i=0;i<st;i++){
    for(int j=0;j<sy;j++) delete [] AAA[i][j];
    delete [] A[i];
}
delete [] AAA;
```

Shin Yoshizawa: shin@riken.jp

Image3Dクラスの使い方

- ✓ 使い方は今まで使ってきたSimpleImage.hのImageクラスとほぼ同じで、一次元増えただけ.

Image3D* 変数名 = new Image3D();
か
Image3D* 変数名 = new Image3D(サイズ);
例えば横500×縦256の画像が120枚あった場合に3D画像を

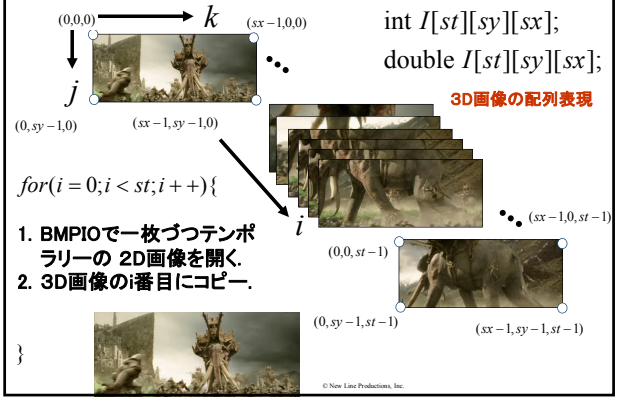
```
Image3D *AAA = new Image(500,256,120);とし
for(int i=0;i<120;i++)
    for(int j=0;j<256;j++)
        for(int k=0;k<500;k++)AAA->img[i][j][k]で輝度
```

値を参照する. カラーの場合は三つのImage3Dを使う.

使い終わったらdelete AAA;を忘れずに!

Shin Yoshizawa: shin@riken.jp

連番画像の入出力へ向けて



Shin Yoshizawa: shin@riken.jp

連番画像名の取得方法

- ✓ Ex14.zip内のImageSetIO.cxxを開いてください.入力としてフォルダー名を与えて、その中のBMPファイルをファイル名順にソートしたファイル名のリストを得るプログラムです.
- ✓ 今回の演習でやる方法は、
  ステップ1:Linux/UnixコマンドのlsとgrepをC/C++からシステムコール関数system()を使って、与えられたフォルダー名内のBMP画像ファイル名(複数)をテンポラリーのファイル(tmp_img_file_names.txt)に書き出す.
  - system()はstdlib.hが必要.
  - system(char*)で引数に書いたLinuxコマンドを実行出来る. 例:system("ls");

Shin Yoshizawa: shin@riken.jp

連番画像名の取得方法2

- ✓ 今回は以下のコマンドを用いる:
  "ls 入力フォルダー名 | grep .bmp > 出力ファイル名"
  ここで|と>はそれぞれ、パイプとリダイレクトと呼ばれてコマンドの結合とファイルへの出力を行える:
  - 「ls AAA」AAA内のファイル名・フォルダー名を出力する.
  - 「grep AAA BBB」 BBBの中からAAAがある行を抜き出す.
  - 「AAA | BBB」 AAAの結果をBBBに渡す.
  - 「AAA > BBB」AAAの結果をBBBに書き出す.
  - sprintf(格納先,printfの表記,変数)でコマンド内にメインの引数やテンポラリーファイル名をプリント.

与えられたフォルダー名内のlsの結果から.bmpが付いているファイル名だけ抽出して出力ファイルに書き出すコマンド.

Shin Yoshizawa: shin@riken.jp

連番画像名の取得方法3

- ✓ ステップ2:テンポラリーのファイル(tmp_img_file_names.txt)を開いて、一行づつfscanf()で呼び込み、vector<char *>へ格納する:
  - FILE *fp = fopen(ファイル名,"r");で開いたファイルポインターfpを使ってfscanf(fp,"%s",格納先)の戻り値がEOFでない間、繰り返しスキャンする.
  - vectorを使うには#include<vector>が必要.
  - vector<char *>へ代入するためにchar *をnewしてfscanf()の結果をコピーする.
  - .push_back()メソッドを使ってvectorへ格納する.

格納後はvectorなので配列の様に使える. 例えば、vector <char *> AAA;ならAAA[0]に最初のファイル名がchar *で入っており、以下AAA[1], AAA[2]と使える. サイズ(push_backした回数=ファイル名の数)は AAA.size()で得られる.

Shin Yoshizawa: shin@riken.jp

連番画像名の取得方法4

- ✓ ステップ3:std::sortを使ってvector<char *>に格納したファイル名をソートする. 例えばvector <char *> AAA;ならstd::sort(AAA.begin(),AAA.end());でソートされる.
  - std::sortは#include<algorithm>が必要.
- ✓ ステップ4:ソート後は、vector<char *>を配列の様に使いファイル名の操作を行い、実際の処理をする.
  - ImageSetIO.cxxは連番名の取得だけなので、実際の処理は無いが、演習ではVideoIO.cxxでソート後のファイル名を順番に開いて3D画像クラスに格納する. BMPIO.hを使って2D毎に入出力をファイル名の数だけ行う.
- ✓ ステップ5:new したchar *のメモリを解放する. 例えば、
  for(i=0;i<AAA.size();i++)delete [] AAA[i];

Shin Yoshizawa: shin@riken.jp

演習: 連番画像の入出力

www.riken.jp/brict/Yoshizawa/Lectures/index.html
www.riken.jp/brict/Yoshizawa/Lectures/Lec21.pdf
www.riken.jp/brict/Yoshizawa/Lectures/Ex14.zip

1. Lec21-1: 3D画像クラスをSimpleImage3D.hとして作成せよ.
2. Lec21-2: 連番画像の入出力を行うプログラムVideoIO.cxxをコメントを読みながら作成せよ. LV3_1.zipとLV3_5.zipを展開して入力フォルダーとして実行してみよ.

Lec21-3: ↑の1,2を使って、連番の各画像にBilateralフィルタ(Lec18-2)を計算して結果を保存するプログラムを作成してみましょう.

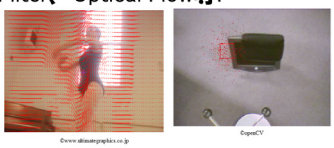
Shin Yoshizawa: shin@riken.jp

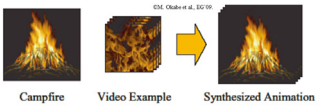
来週の予定

✓ 動画像処理その2(12/14).

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	計算Photography・Artistic Stylization
8回	
9回	
10回	動画像処理
11回	
12回	
13回	エッジ・形状・特徴抽出とパターン認識の基礎
14回	
15回	補講

内容(9－12)：動画像処理基礎、Animation**合成、**Particle Filter、Optical Flow**等.**





Campfire　Video Example　Synthesized Animation