レポート課題

- 以下に挙げたヒストグラムの類似度・相違度について調査
  - 相関
  - カイ2乗
  - 交差
  - Bhattacharyya 距離， Bhattacharyya 係数
- Lenne 画像のヒストグラムを作成し，gnuplot でそのグラフを表示する（次ページスライド 02_01 のケースで良い）
  - 色やフォント，凡例等を設定する gnuplot のスクリプトを作成

- TeX で作成して，ソースとPDFを提出
- 提出方法は，これまでと同じです

問題：ファイル名 02_01_histogram.cpp

ヒストグラムの作成	
機能	画像のヒストグラムを作成する
関数名	void ipr_calc_histogram(Mat_<uchar> &image, Mat_<double> &hist,)
引数	hist: サイズが 256 x 1 の Mat 型 hist image: モノクロ画像
戻り値	なし
備考	ヒストグラム hist は，関数に入る前に領域を確保して，ゼロで初期化． 後の処理の関係から，ヒストグラムは double 型の配列とする．

※ OpenCV の関数 calcHist は使わず自作する．
なぜなら，使い方が複雑なのと，特殊なヒストグラムの計算ができないので

問題：ファイル名 02_02_normalize_hist.cpp

ヒストグラムの正規化	
機能	ヒストグラムを正規化する（¥sum_{d} hist[d] = 1.0）
関数名	void ipr_normalize_histogram(Mat_<double> &src, Mat_<double> &dst)
引数	元のヒストグラム src と正規化後のヒストグラム dst
戻り値	なし
備考	ヒストグラムのビンの数は，Mat のメンバ変数から取得

※ OpenCV の normalize 関数を使う方法と，
Mat のメソッド convertScale を使う方法も，
併せて実装

問題：ファイル名 02_03_save_histogram.cpp

ヒストグラムの保存	
機能	ヒストグラムをファイルに保存する
関数名	void ipr_save_histogram(Mat_<double> &hist, const char path[])
引数	ヒストグラム hist と保存ファイル名 path
戻り値	なし
その他	ファイルが開けなかったときは，メッセージを表示して終了． ファイルのフォーマットは，一列目に画素値を二列目に頻度を出力 0 0.00000e+00 ... 100 2.22000e-05 ...

問題：ファイル名 02_04_calc_mode.cpp

最頻値の算出	
機能	ヒストグラムから最頻値を算出する
関数名	`int ipr_calc_mode(Mat_<double> &hist)`
引数	ヒストグラム hist
戻り値	最頻値
その他	最頻値が複数ある場合は，最も小さい値を返す

※ OpenCV の関数（minMaxLoc）を使う方法も，併せて実装

問題：ファイル名 02_05_comp_hist.cpp

二つのヒストグラムを比較	
機能	二つのヒストグラムを比較して，類似度を算出する
関数名	double ipr_compare_histogram(Mat_<double> &hist1, Mat_<double> &hist2, int method)
引数	ヒストグラム hist1, hist2 比較の種類 COMP_CORREL: 相関 COMP_CHISQR: カイ2乗 COMP_INTERSECT: 交差 COMP_BHATTACHARYYA: Bhattacharyya 距離
戻り値	比較した結果（類似度もしくは相違度）
その他	**比較の種類について，それぞれどのような特徴があるか調査すること**

※これで，画像間の類似度が定義できるので，3枚の画像間で比較するプログラムを作成