

画像処理 レポート

氏名: 木下直樹
学籍番号: 09425521

提出日: 2015 月 11 月 30 日

1 lsq.c に現れる関数について

lsq.c はパノラマ画像で用いる似まいの画像の対応点から pano0.c で用いる変換行列を生成するプログラムである。lsq.c に現れる関数について説明する。

- **Matrix**

行列を管理する構造体。H, W: $H \times W$ 行列であることを表す。data: 要素を格納した double 型の配列。(i,j) 要素は data[i*W+j] に格納される。

- **Elem(Matrix*mt, int i, int j);**

行列 mt の (i,j) 要素の取得, および, 代入に使う。s+=Elem(mt,i,j); や Elem(mt,i,j)=0; のような書式が可能。

- **double*Row(Matrix*mt, int i);**

行列の第 i 行ベクトルを表すポインタを取得する。double*mi=Row(mt,i); を行ったら後は,mi[j] が Elem(mt,i,j) と同じ意味になる。ループの内側で,i が固定,j が変化する時には, 一度ポインタを取得しておく方が良い。

- **Matrix*MatrixAlloc(int H, int W);**

$H \times W$ 行列を作成する。確保された領域の値は未定義。

- **void MatrixClear(Matrix*mt);**

行列の全て要素に 0 を代入する。

- **void MatrixCopy(Matrix*mtD, Matrix*mt);**

行列 mt を mtD にコピーする.mtD は mt と同じ大きさで確保されている必要がある。

- **void MatrixCopyT(Matrix*mtD, Matrix*mt);**

行列 mt の転置を mtD に代入する.mtD は正しく確保されている必要がある。

- `void MatrixMultT(Matrix*D, Matrix*A, Matrix*B);`
 $D = AB^T$ を計算する. D は確保されている必要がある. A と B の大きさが正しく対応している必要がある.
- `void MatrixPrint(Matrix*mt);`
 行列を `stdout` に出力する.
- `void MatrixQRDecompColMajor(Matrix*mtR, Matrix*mt);`
 mt を上三角行列と行直行列に分解する. mtR は正しい大きさと確保されている必要がある.
- `void MatrixSimeqLr(Matrix*mtB, Matrix*mtR);`
 $mtBmtR^{-T}$ を計算し, mtB を上書きする.

2 実行結果

以下の対応点で `lsq.c` を実行した.

```
xy[][2]={          // from 0.jpg          uv[][2]={          // from 1.jpg
    111,212,          230,227,
    169,354,          287,365,
    454,455,          573,467,
    451,221,          568,231,
}
```

実行結果は以下の様になった.

```
0.922122 0.007513 123.332960 -0.038207 0.960041 25.029107 -0.000106 -0.000000
```

これに 1 を合わせた行列を `m10` として `pano0.c` を実行し, 以下の画像が得られた.

前回よりパノラマ画像のクオリティが高くなっている. これは `lsq.c` によるものでなく, よりよい対応点の選別によるものであり, 実際の画像合成アルゴリズムは前回と同様である.



図 1: 前回の出力画像



図 2: 選んだ対応点から得られた行列を使用した出力画像