

透視投影画像を用いたカメラ位置・姿勢推定

M1 田川幸汰

1 概要

透視投影画像からカメラ位置・姿勢推定を行った。方法については前回の発表で説明した、直行射影の共線性と共面性を用いた方法 [1] で推定する。また、カメラ位置姿勢推定を行う際に全方位カメラの内部パラメータである焦点距離を使用する。そのため事前準備として全方位カメラのキャリブレーションを行った。

2 全方位カメラのキャリブレーション

2.1 キャリブレーション方法

全方位カメラのキャリブレーションは以下の手順で行う。

- チェッカーボードを全方位カメラで 10 枚以上撮影
- 全方位画像から、チェッカーボードの全体が写る視点で透視投影画像を生成
- キャリブレーションを行うプログラムを実行

透視投影画像を生成する際、解像度は変更しないようにする。また、キャリブレーションを行うプログラムは OpenCV のチェッカーボードの交点の座標を調べる関数 `indChessboardCorners`、3次元座標と2次元座標の対応から内部パラメータを計算する関数 `calibrateCamera` が用いられている。

2.2 実行結果

全方位画像から生成された透視投影画像のキャリブレーション結果の一部を図 1 に示す。また、出力されたカメラの



図 1: キャリブレーション結果

内部パラメータ行列 M を以下に示す。

$$M = \begin{pmatrix} 511.69918760786322 & 0.000000000000000 & 784.54139306531715 \\ 0.000000000000000 & 499.73234060481144 & 536.82834728101511 \\ 0.000000000000000 & 0.000000000000000 & 1.000000000000000 \end{pmatrix} \quad (1)$$

これより、チェッカーボードの交点を検出し、カメラの内部パラメータが推測されていることがわかる。また焦点距離 $f_x = 511.69918760786322$ 、 $f_y = 499.73234060481144$ となったが、今回焦点距離は各軸共通なので f_x と f_y の平均を用いる。

3 カメラ位置・姿勢推定の実行結果

3.1 入力

直行射影の共線性と共面性を利用したカメラ姿勢の推定の際に必要な入力を以下に示す。

- p_a : 世界座標系で表現された空間点の座標
- $v_a: p_a$ に対応する画像上の特徴点座標
- d_a : 世界座標系で表現された直線 L_a の方向ベクトル
- r_a : 世界座標系で表現された直線 L_a 上の点の座標
- $n_a: L_a$ に対応する画像上の直線のパラメータベクトル
- R^0 : 回転行列の初期値
- thd : 収束に用いるしきい値

p_a は3次元モデルの頂点の座標の一部を用いる。ただし透視投影画像に映らない点については、空間点の座標として用いない。3次元モデルについてはC棟エレベータ前の廊下のモデルを用いる。3次元モデルを図2に示す。

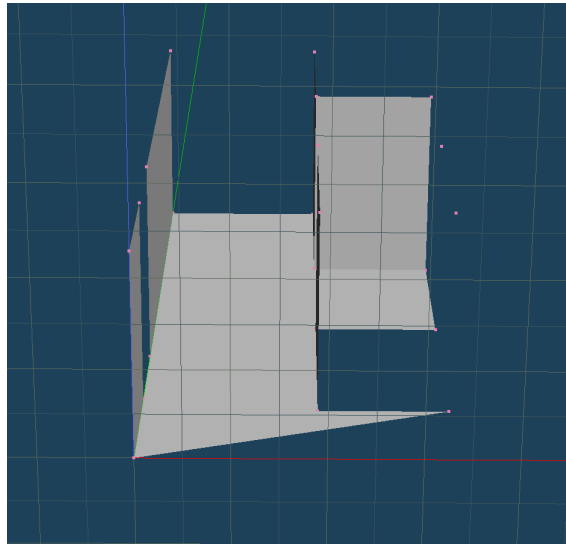


図 2: 3d モデル

v_a は透視投影画像から手動で選択して用いる。透視投影画像を表示し p_a に対応する画像の座標をクリックする。に追加される(図3)。また $v_a = (x_a, y_a, f)$ で、 (x_a, y_a) は光軸点を原点とした画像点の座標であるため、以下の式で平行移動する。

$$x_a = x - \frac{W}{2}, y_a = y - \frac{H}{2} \quad (2)$$

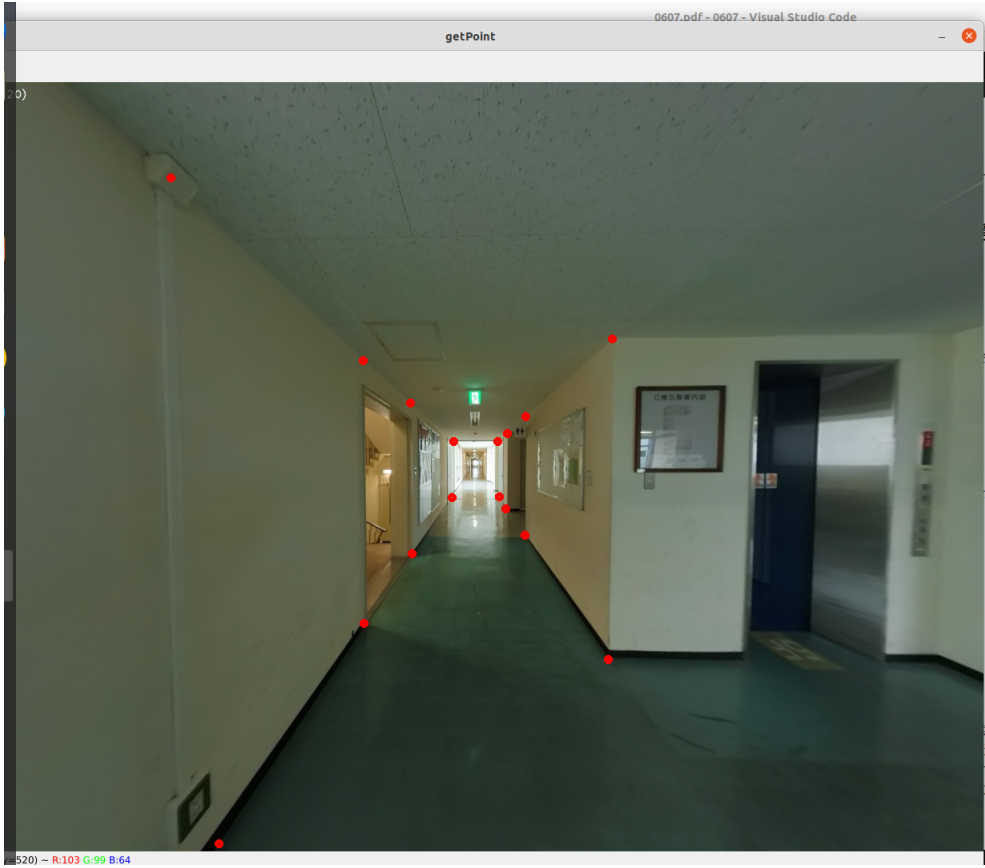


図 3: 画像特徴点を選択

さらに、焦点距離 f を座標の後ろに追加する必要がある。焦点距離 f については、全方位カメラのキャリブレーションを行い計算した。

$d_a = d_1, \dots, d_i, \dots$ については3次元モデルの頂点の座標の一部 (p_a で使用していない点) を用いて以下の式で求める。

$$d_i = \frac{p_{i+1} - p_i}{|p_{i+1} - p_i|} \quad (3)$$

r_a については3次元モデルの頂点の座標の一部 (p_a で使用していない点) を用いる。

$n_a = n_1, \dots, n_i, \dots$ については v_a の座標を用いて以下の式で用いる。

$$n_i = \frac{n_{i+1} \times n_i}{|n_{i+1} \times n_i|} \quad (4)$$

R^0 は世界座標系カメラ座標系の関係から以下の行列を設定する。また、 $thd = 1.0^{-8}$ とする。

$$R^0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \quad (5)$$

3.2 実行結果

出力された回転行列 R 、及びカメラ位置 c を以下に示す。

$$R = \begin{pmatrix} 0.99848126 & 0.01266343 & -0.05361733 \\ -0.05432481 & 0.06440953 & -0.99644379 \\ -0.00916493 & 0.9978432 & 0.06499965 \end{pmatrix} \quad (6)$$



$$\mathbf{c} = \begin{pmatrix} 0.99930194 \\ -2.46764452 \\ 1.18038819 \end{pmatrix} \quad (7)$$

回転行列、カメラ位置ともに想定したものと近い値が出力された。また、収束にかかった計算回数は、平均して22回程度だった。

4 今後の計画

今後の計画として、計算されたカメラ運動行列を用いてテクスチャの貼り付けに早急に取り掛かりたい。具体的には6月中にテクスチャを貼り付けた3次元モデルを完成させる。

参考文献

- [1] 菅谷保之, 「直交射影の共線性と共面性を用いたカメラ姿勢の推定」