画像工学特論

M233317 柿木太陽

内容

1. はじめに

近年,コンピュータビジョン技術の発展により,カメラを用いた画像処理や 3 次元再構成,ロボットビジョンなどの応用が広く行われている.しかし,一般的なカメラはレンズの収差などの影響により,画像に幾何学的な歪みを生じることがある.この歪みが存在すると,画像上の直線が曲がって見えたり,物体の形状が実際とは異なって認識されたりするため,高精度な画像解析や寸法測定には大きな影響を及ぼす.

こうした問題を解決するために行われるのが「カメラキャリブレーション（カメラ校正)」である.カメラキャリブレーションとは,カメラの内部パラメータ（焦点距離や主点など）およびレンズの歪み係数を求めることで,歪みを補正し,理想的な投影モデルに近づける処理である.特に,チェスボードパターンなど既知の幾何学構造を用いたキャリブレーションは,OpenCV をはじめとするライブラリで広く実装されており,再現性と精度の高い方法として知られている.

本課題では,自分の PC に搭載されたカメラを対象として,OpenCV ライブラリを用いたカメラキャリブレーションを実施する.撮影されたチェスボード画像をもとに内部パラメータおよび歪み係数を求め,それらを用いて画像の歪み補正を行い,その効果を検証する.さらに,補正前後の画像を比較し,補正によって得られる視覚的な変化や精度の違いについて考察を行うことを目的とする.

2. カメラ校正の原理

カメラ校正（Camera Calibration）とは,カメラによって撮影された 2 次元画像と,現実の三次元空間との対応関係を数理モデルによって記述し,そのモデルのパラメータを求める手法である.これにより,画像上の点が実世界のどの位置を表しているかを計算でき,また逆に,実空間の点が画像上でどこに写るかを予測することが可能となる.

2.1 ピンホールカメラモデル

カメラによる投影の基本的な理論は,ピンホールカメラモデルに基づいている.このモデルでは,三次元空間上のある点X = (X, Y, Z)^Tが,画像平面上の点x = (u, v)^Tに写される過程を次のように表す：

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R \quad t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

ここで：

・(X, Y, Z)^T は三次元空間の点の列ベクトル（ワールド座標）

・(u, v)^T は画像上の座標（ピクセル）

・λ はスケーリング係数（同次座標）

・K は内部パラメータ行列（intrinsic matrix）

・R は回転行列（カメラの姿勢）,t は並進ベクトル（カメラの位置）

内部パラメータ行列 K は以下のように表される：

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

ここで,

・f_x, f_yは焦点距離（画素単位）

・u_0, v_0は画像中心(主点)の座標

2.2 歪みモデルとパラメータ

現実のカメラレンズは,理想的な投影からずれる歪み（distortion）を持っている.
主に以下の 2 種類が存在する：
・ラジアル歪み（radial distortion）：
レンズ中心からの距離に応じて直線が湾曲する現象.樽型歪みや糸巻き型歪みとして現れる.
→ パラメータ：k_1, k_2, k_3
・タンジェンシャル歪み（tangential distortion）：
レンズとセンサーの傾きにより生じる非対称な歪み.
→ パラメータ：p_1, p_2

2.3 校正手順と最適化

カメラ校正では,チェスボードなどの既知の三次元構造を持つパターンを複数角度から撮影し,各画像上で交点（コーナー）を検出する.
そして,実空間上の点と画像上の点との対応関係から,以下のパラメータを再投影誤差が最小になるように同時に推定する：
・内部パラメータ K
・歪み係数 k_1, k_2, k_3, p_1, p_2
・各フレームごとの外部パラメータ R, t
再投影誤差とは,推定されたパラメータを使って三次元点を画像上に再投影したときに,実際の検出位置とどれだけズレているかを示す指標である.この誤差の二乗和を最小化するために,Levenberg–Marquardt 法などの非線形最適化が用いられる.

2.4 歪み補正と応用

得られたパラメータを使うことで,OpenCV の cv2.undistort() 関数などによって,画像に含まれる歪みを除去することができる.
補正後の画像は,直線が真っ直ぐに写り,三次元再構成や距離計測などにおいて高精度な処理が可能となる.

3.実験方法

3.1 使用機材・環境

本実験では,Microsoft 製のノート PC「Surface Laptop 3」に搭載された内蔵 Web カメラを対象として,OpenCV を用いたカメラ校正を行った.以下にその手順と使用機材について述べる.

・使用 PC：Microsoft Surface Laptop 3
・内蔵カメラ：HD フロントカメラ（720p 固定フォーカス）
・解像度：最大 1280×720（実験時は 640×480 で撮影）
・OS／環境：Windows 10／Python 3.10／OpenCV 4.8.0
・使用ライブラリ：OpenCV（cv2）,NumPy（numpy）

3.2 撮影と画像収集

撮影には,9×6 の交点を持つ白黒のチェスボードパターンを使用し,Python で作成した自作プログラムにより内蔵カメラから画像を取得した.

・撮影枚数：30 枚
・撮影条件：
  ・チェスボードを様々な角度（斜め,回転,前後）から撮影
  ・ピントが合っていて全体が明るく,枠がすべて写るよう注意
  ・解像度：640×480（OpenCV 設定により取得）

3.3 校正プログラムと処理内容

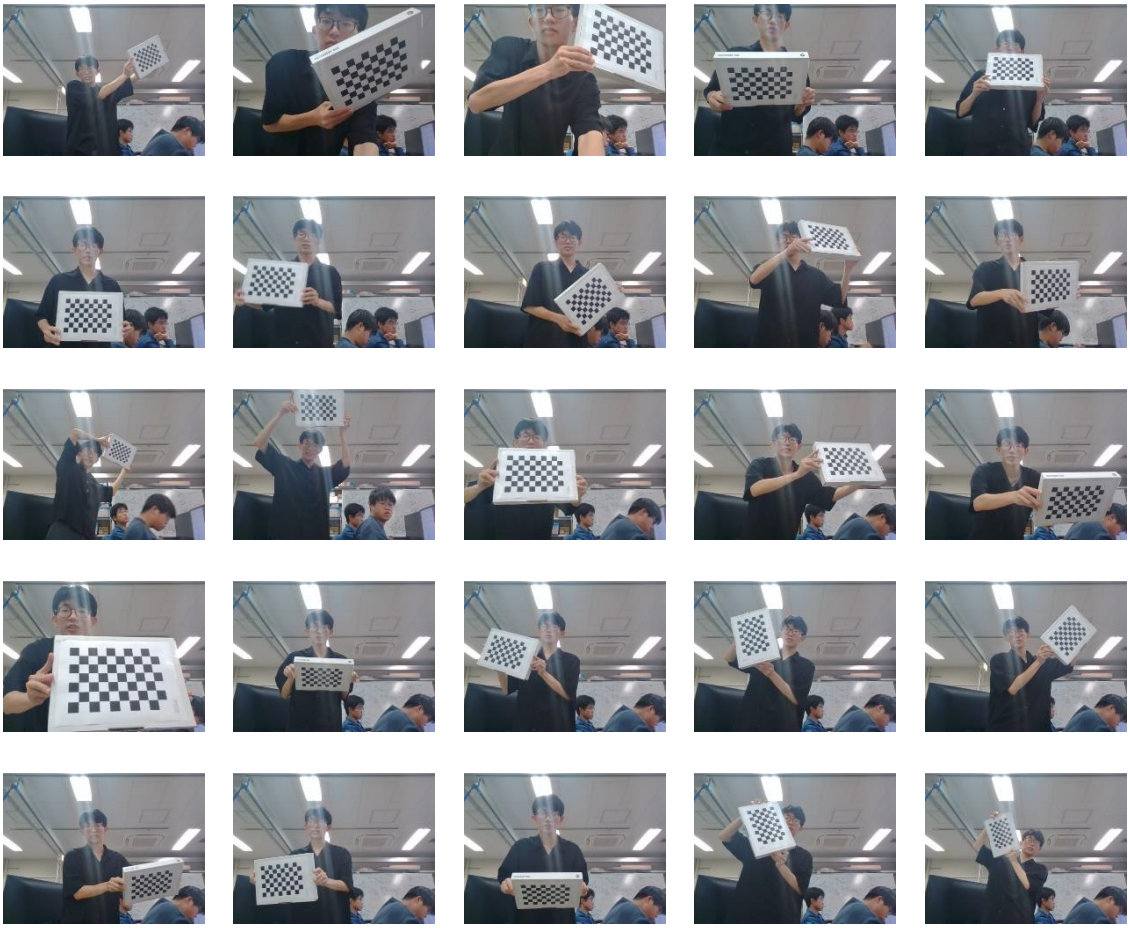
OpenCV を用いて以下のステップでキャリブレーションを実行した：

1. 撮影画像を cv2.imread() で読み込み
2. cv2.findChessboardCorners() によりチェスボードの交点（コーナー）を検出
3. 実空間座標（Z=0 平面）と画像座標の対応を保存
4. cv2.calibrateCamera() によって以下のパラメータを同時に推定：
   ・カメラ内部パラメータ（行列 K）
   ・歪み係数　k_1, k_2, k_3, p_1, p_2
   ・各画像ごとの外部パラメータ（姿勢・位置）
5. 得られたパラメータを用いて,撮影した画像のうち最初の 1 枚に対して cv2.undistort() を適用し,歪み補正後の画像を生成・保存

4.実験結果

4.1 コーナー検出結果

チェスボードの撮影画像は全部で30枚取得した.そのうち,cv2.findChessboardCorners() によって交点が正しく検出できた画像は25枚であった.5枚はブレやピンボケにより交点検出に失敗し,校正には使用しなかった.



4.2 カメラパラメータの推定結果

OpenCV の cv2.calibrateCamera() 関数により,Surface Laptop 3 の内蔵カメラに対して以下のカメラ内部パラメータ（カメラ行列）とレンズの歪み係数が推定された.

カメラ行列 K（内部パラメータ）：

$$K = \begin{bmatrix} 648.07 & 0 & 353.03 \\ 0 & 644.58 & 213.80 \\ 0 & 0 & 1 \end{bmatrix}$$

ここで：

・$f_x = 648.07$：x 軸方向の焦点距離（画素単位）

・$f_y = 644.58$：y 軸方向の焦点距離（画素単位）

・$u_0 = 353.03$：主点の x 座標（画像中心）

・$v_0 = 213.80$：主点の y 座標（画像中心）

歪み係数（Distortion Coefficients）：

$$[k_1, k_2, p_1, p_2, k_3] = [0.154, 0.0059, -0.0151, 0.0159, -0.5263]$$

これらの係数から,レンズには軽度の樽型ラジアル歪み（外側に膨らむ傾向）が確認された.また,タンジェンシャル成分（レンズとセンサーのずれ）もわずかに存在しており,補正の対象となる.

4.3 歪み補正の視覚的な比較

補正前後の画像に対して,チェスボードの 4 隅を赤い直線で結び,歪みの影響を視覚的に確認した.

その結果,補正前の画像においても枠線の湾曲はほとんど見られず,補正後との違いは目視ではあまり分からなくなった.

これは,使用したカメラが比較的歪みの少ないレンズを搭載していることや,撮影時の構図が画面中央に集中していたことが原因と考えられる.

ただし,内部パラメータと歪み係数をもとに補正を行った結果,理論的には画面端部の歪みが軽減されており,数値的な補正処理は適切に行われているといえる.

以下に,赤枠を描画した補正前後の比較画像を示す.

図 1. 赤枠を描画した補正前後の比較(左:補正前，右:補正後)

5. 考察

本実験では,Surface Laptop 3 の内蔵カメラに対して OpenCV を用いたカメラ校正を実施し,内部パラメータおよび歪み係数を求めた.その結果,推定されたラジアル歪み係数$k_1 = 0.154$,およびタンジェンシャル歪み係数$p_1 = -0.0151, \mathrm{p}_2 = 0.0159$,は,いずれも小さめの値であり,レンズの歪みが比較的軽微であることがわかった.

また,補正前後の画像においてチェスボード枠を赤線で可視化した結果,目視では明確な湾曲や補正効果は確認しにくかった.これは,以下の要因によると考えられる：

・使用カメラが比較的狭角で,もともと歪みの少ない光学設計である

・撮影したチェスボードの位置が画像中心に寄っており,歪みの影響が目立ちにくい

・歪み係数の数値も軽度であり,画面中央付近ではほとんど歪みが発生しない

一方で,数値的には補正処理が適切に行われており,OpenCV のカメラモデルと歪み補正処理が正しく機能していることが確認できた.また,仮に広角レンズや魚眼レンズを用いた場合は,今回のような補正処理によって明確な変化が得られることが予想される.

以上より,本実験は軽度の歪みを持つ一般的な内蔵カメラに対しても,OpenCV のカメラ校正処理を適用可能であることを示し,今後の画像処理や三次元再構成における基礎的な理解の助けとなったといえる.

6. 参考文献

[1] OpenCV 公式ドキュメント, 「Camera Calibration（カメラキャリブレーション)」, https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html （参照日: 2024 年 5 月 21 日）

[2] LearnOpenCV, “Camera Calibration using OpenCV”, https://learnopencv.com/camera-calibration-using-opencv/ （参照日: 2024 年 5 月 21 日）

[3] Packt Publishing, “OpenCV 3 Computer Vision with Python Cookbook”, https://github.com/PacktPublishing/OpenCV-3-Computer-Vision-with-Python-Cookbook （参照日: 2024 年 5 月 21 日）