University of Burgundy

Masters in Computer Vision and Robotics

Autonomous Robotics

Camera Auto-calibration Lab

by

Vamshi Kodipaka

Supervisor:

Dr. Devesh Adlakha

CAMERA AUTO-CALIBRATION

The procedure of estimating internal parameters of a camera directly from multiple uncalibrated images of unstructured scenes is well known as “Camera auto-calibration”. Auto-calibration does not require any special calibration objects in the scene[1]. It is the classical approach, in which camera motion and parameters are recovered first, using rigidity, then structure is readily calculated.

In 3D Reconstruction, the retrieval of camera calibration parameters is important because it provides metric info about the observed scenes, e.g., measurements of angles and ratios of distances. This procedure of auto-calibration helps us to estimate the camera parameters without using a calibration patterns like checkerboard, by applying some simple constraints on the camera parameters, like constant intrinsics in multiple images, known principal point, known pixel shape, etc.

Given camera auto-calibration algorithms[2]:

1. Mendonça-Cipolla auto-calibration method.

2. Simplified and classical Kruppa's equations.

3. Dual Absolute Quadric method

EXPERIMENTATION AND RESULTS OBTAINED

1 . Mendonça-Cipolla Auto-Calibration method

This method is based on the use of rigidity constraint. Design a cost function, which considers the intrinsic parameters as arguments and the fundamental matrices as parameters.

Cost function to find instrincs for Mendonça-Cipolla auto-calibration method given below,

$$\mathcal{C}(\mathbf{A}_i, i = 1, \ldots, n) = \sum_{ij}^{n} \frac{w_{ij}}{\sum_{kl}^{n} w_{kl}} \frac{\sigma_{ij}^{(1)} - \sigma_{ij}^{(2)}}{\sigma_{ij}^{(2)}}$$

The method finds the property of essential matrix in which two identical singular values (third is zero). I used nonlinear least squares optimization to estimate the cost function.

errestim = costFuncMenCipolla(Fs, p)

Result:

```
------ Estimated Intrinsics by Mendonça-Cipollas method
   1.0e+02 *

   8.000000021832946  -0.000000001981655   2.559999984146586
                   0   8.000000023048889   2.559999995325478
                   0                   0   0.010000000000000
```

Fig. 1 Estimated Intrinsics from Mendonça-Cipolla method

2. Kruppa's method

With a minimum of three displacements, we can obtain the internal parameters of the camera using a system of polynomial equations due to Kruppa, which are derived from a geometric interpretation of the rigidity constraint.

2.1 Classical Kruppa's method

Assuming images are taken from same moving camera with same intrinsics.

Cost function to find instrincs for Classical Kruppa's auto-calibration method given below:

$$\mathcal{C}(\alpha_u, \alpha_v, \gamma, u_0, v_0) = \sum_{i,j} \left|\left| \frac{\mathbf{F}_{ij}\omega^{-1}\mathbf{F}_{ij}^T}{||\mathbf{F}_{ij}\omega^{-1}\mathbf{F}_{ij}^T||_{fro}} - \frac{[\ \mathbf{e}_{ji}\ ]_\times \omega^{-1} [\ \mathbf{e}_{ji}\ ]_\times^T}{||[\ \mathbf{e}_{ji}\ ]_\times \omega^{-1} [\ \mathbf{e}_{ji}\ ]_\times^T||_{fro}} \right|\right|_{fro}$$

where $||.||_{fro}$ refers to the Frobenius norm.

errestim = costFuncKClassical(Fs, p)

Problem: cost function optimization with tolerance parameters

Solution I proposed: setting low tolerance values to find minimum.

Result:

```
------- Estimated Intrinsics by Classical Kruppas method
   1.0e+02 *

   7.999999999998311   0.000000000000071   2.560000000000057
                   0   7.999999999998284   2.559999999999835
                   0                   0   0.010000000000000
```

Fig. 2 Estimated Intrinsics for Classical Kruppa's method

2.2. Simplified Kruppa's method

Simplified Kruppa uses SVD of fundamental matrix. Also Simplified Kruppa equation is given by:

$$\frac{r^2\mathbf{v}_1^T\omega^{-1}\mathbf{v}_1}{\mathbf{u}_2^T\omega^{-1}\mathbf{u}_2} = \frac{rs\mathbf{v}_1^T\omega^{-1}\mathbf{v}_2}{-\mathbf{u}_1^T\omega^{-1}\mathbf{u}_2} = \frac{s^2\mathbf{v}_2^T\omega^{-1}\mathbf{v}_2}{\mathbf{u}_1^T\omega^{-1}\mathbf{u}_1}$$

errestim = costFuncKSimplified(Fs, p)

Result:

```
----- Estimated Intrinsicss by Simplified Kruppas method
   1.0e+02 *

   7.999999999999724   0.000000000000140   2.560000000000402
                   0   7.999999999999845   2.560000000000349
                   0                   0   0.010000000000000
```

Fig. 3 Estimated Intrinsics for Simplified Kruppa's method

2.4 Dual Absolute Quadric Method

Firstly, find the plane at infinity by Dual Absolute Quadric and guess initially the intrinsics.

Given Eqn. $$\mathsf{M}_j \mathcal{Q} \mathsf{M}_j^T \simeq \omega^{-1} \quad \text{where} \quad \mathcal{Q} = \begin{pmatrix} \omega^{-1} & \omega^{-1}\mathbf{n}_\Pi \\ \mathbf{n}_\Pi^T \omega^{-1} & \mathbf{n}_\Pi^T \omega^{-1} \mathbf{n}_\Pi \end{pmatrix}.$$

here, M is camera matrix, Q is DAQ, w = AA^T and 'n' is normal to plane at infinity.

We solve this eqn by scale factor 'lambda' and by initially guess we find the plane at infinity & 'lambda'.

Step1: normal = getNormalToPlaneAtInf(PPM, A)

Results:

```
---- Estimated normal to plane at infinity by DAQ
  -0.625409742621828
  -0.231086754291671
  -2.988295709651895
```

Fig. 4 Normal to plane at infinity using DAQ

Step2: Finding Homography at plane at infinity and then finding 'w' from homography at plane at infinity. From that we find intrinsics 'K' using Cholesky factorization[3].

```
---- Estimated Intrinsics by DAQ
   1.0e+02 *

   8.702582716108395                   0   2.791007468913193
                   0   8.121786285338063   2.609300502222471
                   0                   0   0.010000000000000
```

Equn. For plane at infinity: $H_\infty = [e_{21}]_x F + e_{21} n^T$

where, F is fundamental matrix, e_{21} is epipole and 'n' is normal to plane at inifnity.

Step3: Optimize 'w' by, $H_\infty w H_\infty^T = w$

I am struck in optimization as minimum as at initial guess and could not proceed further even-though I vary the optimizing factorss such as 'Tolerance value', 'max no. of iterations' etc.

3. Conclusion

These auto calibration methods helps us in finding the Camera Intrinsics though we are not given pattern for camera calibration.

References

[1] Auto-calibration, Wikipedia

[2] Lecture notes of Dr. Adlane Habed & Devesh Adlakha .

[3] "Multiple View Geometry in Computer Vision", by Richard Hartley and Andrew Zisserman