Self Calibration - Lab

ASRAF ALI Abdul Salam Rasmi
Masters in Computer Vision
University of Burgundy

June 10, 2019

1 Introduction

Any problem in Computer Vision will always be dependent directly or indirectly on the following equation of 2D-3D Transformation.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = [K]_{(3\times3)} \times [C]_{(3\times4)} \times [R|t]_{(4\times4)} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

where,

- $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ is the 2D Point in Homogeneous Coordinates.
- $\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ is the 3D Point in Homogeneous Coordinates.
- K, C, $R|t$ are the Intrinsics, Camera Matrix and Extrinsics respectively (in short, Camera Parameters).

Calibration is the Process of finding the Camera Parameters given a set of 2D Image Points and corresponding 3D Scene Points, whereas in Self Calibration, we need to find the Camera Parameters with only 2D Image Points (Scene points are unknown). Three Algorithms of Self Calibration are Implemented here. The goal is to find the **True Intrinsics** given the **Appoximate values of Intrinsics.**

```
Approximate Intrinsics (Given)
   1.0e+02 *

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

Figure 1: Approximate Intrinsics

2 Mendonça-Cipolla's Method

2.1 Procedure:

For each pair of Images,

1. Compute the Essential Matrix using the given Fundamental Matrix and approximate Intrinsics.
2. Compute SVD of Essential Matrix to get two singular values $\sigma_{ij}^{(1)}$ & $\sigma_{ij}^{(2)}$ (third is 0).
3. Compute Cost Function using one among the following Expression,

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

$$or$$

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

   where, w_{ij} is the weights (considered as 1 in our case.)

4. Solve by Nonlinear Least Squares Optimization of the Cost Function to get True Intrinsics.

```
Intrinsics computed by Mendonca & Cipolla`s Self Calibration(Method 1)
  1.0e+02 *

  8.000000021817968  -0.000000001980040   2.559999984145359
                  0   8.000000023036041   2.559999995326293
                  0                   0   0.010000000000000
```

Figure 2: Intrinsics computed by Mendonca & Cipolla's Self Calibration(Method 1)

```
Intrinsics computed by Mendonca & Cipolla`s Self Calibration(Method 2)
   1.0e+02 *

   8.000000015871684  -0.000000002202020   2.559999984057947
                   0   8.000000018050471   2.559999997784594
                   0                   0   0.010000000000000
```

Figure 3: Intrinsics computed by Mendonca & Cipolla's Self Calibration(Method 2)

Note: The result of Optimization depends on the Parameters we choose (see for *lsqnonlin* in MATLAB Documentation).

3 Classical Kruppa's Method

3.1 Procedure:

For each pair of Images,

1. Compute the Image of Absolute Conic ($\boldsymbol{\omega}^{-1}$) using the given approximate Intrinsics.

2. Compute Cost Function using the following Expression,

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

   where, F_{ij} is the Fundamental Matrix and e_{ji} is the Epipole Vector. Here, $||.||_{fro}$ is the Frobenius Norm. ## **Note:** $[\mathbf{e}_{ji}]_\times$ is the Skew Symmetric Matrix of the Epipole Vector.

3. Solve by Nonlinear Least Squares Optimization of the Cost Function to get True Intrinsics.

```
Intrinsics computed by Classical Kruppa`s Self Calibration
   1.0e+02 *

   7.999999999998101  -0.000000000000030   2.560000000000096
                   0   7.999999999998176   2.559999999999732
                   0                   0   0.010000000000000
```

Figure 4: Intrinsics computed by Classical Kruppa's Self Calibration

4 Simplified Kruppa's Method

4.1 Procedure:

For each pair of Images,

1. Compute the Image of Absolute Conic ($\boldsymbol{\omega}^{-1}$) using the given approximate Intrinsics.

2. Compute SVD of Fundamental Matrix.

3. Compute Cost Function using the following Expression,

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

   where, $\mathbf{u}_1, \mathbf{u}_2$ and $\mathbf{u}_3$ are the column-vector of $\mathbf{U}$; $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ are the column-vector of $\mathbf{V}$; r and s are the Singular Values of SVD.

4. Solve by Nonlinear Least Squares Optimization of the Cost Function to get True Intrinsics.

```
Intrinsics computed by Simplified Kruppa`s Self Calibration
   1.0e+02 *

   7.999999999998261   0.000000000001012   2.559999999999724
                   0   7.999999999997073   2.560000000000374
                   0                   0   0.010000000000000
```

Figure 5: Intrinsics computed by Simplified Kruppa's Self Calibration

5 Dual Absolute Quadric Method

5.1 Procedure:

For each pair of Images,

1. Compute the Image of Absolute Quadric ($\boldsymbol{\omega}^{-1}$) using the given approximate Intrinsics.

2. Compute the Normal to the Plane at Infinity, $\mathbf{n}_\Pi = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ by solving the following Expression of Dual Absolute Quadric,

$$\mathrm{M}_j \mathcal{Q} \mathrm{M}_j^T \simeq \omega^{-1}$$

   where, $\mathcal{Q} = \begin{pmatrix} \boldsymbol{\omega}^{-1} & \omega^{-1}\mathbf{n}_\Pi \\ \mathbf{n}_\Pi^T \boldsymbol{\omega}^{-1} & \mathbf{n}_\Pi^T \boldsymbol{\omega}^{-1} \mathbf{n}_\Pi \end{pmatrix}$ and M_j is the Projection Matrix.

3. Compute Homography of the Plane at Infinity using the following Expression,

$$\mathbf{H}_{\Pi\infty} \simeq \left[\mathbf{e}_{ji}\right]_{\times} \mathbf{F}_{ij} + \mathbf{e}_{ji}\mathbf{n}_{\Pi}^{T}$$

4. Compute Cost Function using the following Expression,

$$\mathscr{C} \ = \ \mathrm{M}_j \mathscr{Q} \mathrm{M}_j^T - \omega^{-1}$$

5. Solve by Nonlinear Least Squares Optimization of the Cost Function to get True Intrinsics.

Note: In this Method, the Scale is lost while performing Optimization, and so I regained it by setting the last element of the computed Intrinsic Matrix to 1 .

```
Intrinsics computed by DAQ Self Calibration
   1.0e+02 *

   8.182431864517078   0.086451049819890   2.677558864403179
  -0.108930283376241   7.637130831845974   2.499891577077524
  -0.000243082577869  -0.000245023219465   0.010000000000000
```

Figure 6: Intrinsics computed by DAQ Self Calibration Method