Camera Calibration

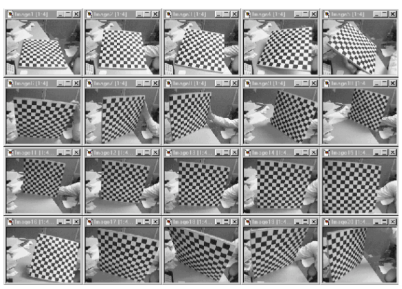
Based on Zhengyou Zhang's method

http://research.microsoft.com/~zhang/Calib

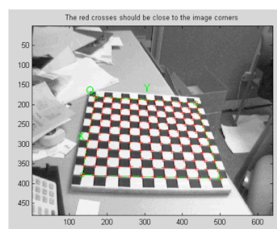
Basic Procedure

1. Print a pattern of known geometry and attach it to a planar surface.
2. Take a few images of the pattern from different angles by moving either the pattern or the camera (or both).
3. Detect the features (corners) in the pattern.
4. Determine the intrinsic and extrinsic parameters first from matched points.
5. Estimate the radial distortion coefficients.
6. Refine all the parameters by non-linear optimization.

Sample images of registration pattern



Feature detection



- We use a checker pattern so that
  - Features are just corners that are easy to detect.
  - The image coordinates of the corners can be optimized.
  - Radial distortion is easy to see and correct.
  - The scale is derived from the size of the squares.

Calibration algorithm

- In each image, we have many matched pairs of image point m (x, y) and its corresponding 3D point M (X, Y, Z).
- All the pairs are related through A{R, t}, i.e., m = A{R, t}M => m = H M.
- H = is 3x3 and called a homography. It can be computed from at least 4 matched pairs.
- Once H is known, we can express A as a function of H using the two constraints on {R}.
- With at least two images, we have two H's and can solve for the intrinsics A without skew c, which is invariant w.r.t. images.
- With A solved, {R, t} of each image can be derived.

Where is distortion?

- Distortion is difficult to solve concurrently with the other parameters.
- This creates a chicken-and-egg problem.
- So, we ignore it first, i.e., assume distortion-free images.
- Once the image center is determined, we use a separate process to estimate distortion.
- Once distortion is estimated, we re-estimate the camera parameters (intrinsics and extrinsics)

Calibration algorithm

- In each image, we have many matched pairs of image point m (x, y) and its corresponding 3D point M (X, Y, Z).
- All the pairs are related through A{R, t}, i.e., m = A{R, t}M => m = H M.
- H = is 3x3 and called a homography. It can be computed from at least 4 matched pairs.
- Once H is known, we can express A as a function of H using the two constraints on {R}.
- With at least two images, we have two H's and can solve for the intrinsics A without skew c, which is invariant w.r.t. images.
- With A solved, {R, t} of each image can be derived.

m and M through A{R, t}

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & \overset{t}{\cancel{T}} \\ & \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}.$$

m = A T{R, t} M

m = A{R, t}M

I.e., each pair of image and space points (m, M) are related by a linear mapping.

Calibration algorithm

- In each image, we have many matched pairs of image point m (x, y) and its corresponding 3D point M (X, Y, Z).
- All the pairs are related through A{R, t}, i.e., m = A{R, t}M => m = H M.
- H = is 3x3 and called a homography. It can be computed from at least 4 matched pairs.
- Once H is known, we can express A as a function of H using the two constraints on {R}.
- With at least two images, we have two H's and can solve for the intrinsics A without skew c, which is invariant w.r.t. images.
- With A solved, {R, t} of each image can be derived.

Homography H

- What is it?
  - Points in two 3D planes are related by a 3x3 matrix H, called its homography.
- How is it derived? (Important!)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}[\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{A}\ [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathbf{H}\tilde{M}$$

This is okay!

Calibration algorithm

- In each image, we have many matched pairs of image point m (x, y) and its corresponding 3D point M (X, Y, Z).
- All the pairs are related through A{R, t}, i.e., m = A{R, t}M => m = H M.
- H = is 3x3 and called a homography. It can be computed from at least 4 matched pairs.
- Once H is known, we can express A as a function of H using the two constraints on {R}.
- With at least two images, we have two H's and can solve for the intrinsics A without skew c, which is invariant w.r.t. images.
- With A solved, {R, t} of each image can be derived.

Two constraints on H

$$\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3] = \mathbf{A}\ [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t}]$$

- The first two columns are perpendicular.

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1}\ \mathbf{h}_2 = 0$$

- And they are of same length.

2 scalar constraints from 1 image

$$\mathbf{h}_1^T \mathbf{A}^{-T} \mathbf{A}^{-1}\ \mathbf{h}_2 = \mathbf{h}_2^T \mathbf{A}^{-T} \mathbf{A}^{-1}\ \mathbf{h}_2$$

- Remember that **H** is known and **A** contains the intrinsics.

Calibration algorithm

- In each image, we have many matched pairs of image point m (x, y) and its corresponding 3D point M (X, Y, Z).
- All the pairs are related through A{R, t}, i.e., m = A{R, t}M => m = H M.
- H = is 3x3 and called a homography. It can be computed from at least 4 matched pairs.
- Once H is known, we can express A as a function of H using the two constraints on {R}.
- With at least two images, we have two H's and can solve for the intrinsics A without skew c, which is invariant w.r.t. images.
- With A solved, {R, t} of each image can be derived.

Solving camera calibration

- Linear solution to intrinsics
  - From each image we get two equations on the intrinsic parameters.
  - If we limit ourselves to four intrinsic parameters, we need two images at different poses.
  - If we have more than two images, we can estimate the skew and obtain the least square solution to all five intrinsic parameters.
- Closed-formed solution to the extrinsics

$$\mathbf{r}_1 = \mathbf{A}^{-1}\mathbf{h}_1, \mathbf{r}_2 = \mathbf{A}^{-1}\mathbf{h}_2, \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2\, \mathbf{t} = \mathbf{A}^{-1}\mathbf{h}_3$$

Optimizing the parameters

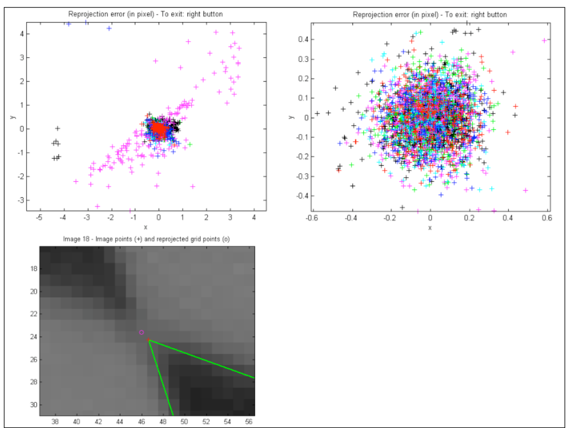
- The 3D points M are reprojected to the images m by the camera's intrinsic and extrinsic parameters, and the parameters are optimized by minimizing the overall reprojection error.

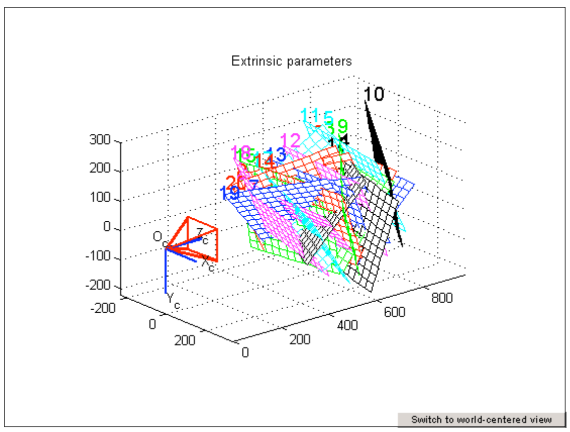
$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left\| m_{ij} - \hat{m}(\mathbf{A}, \mathbf{R}_i, \mathbf{t}_i, M_j)\right\|^2$$

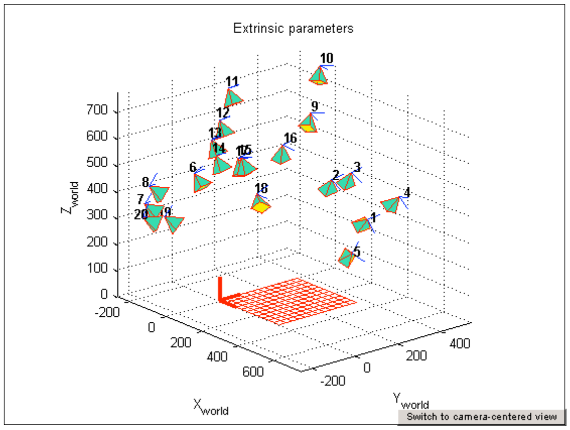
real corner locations

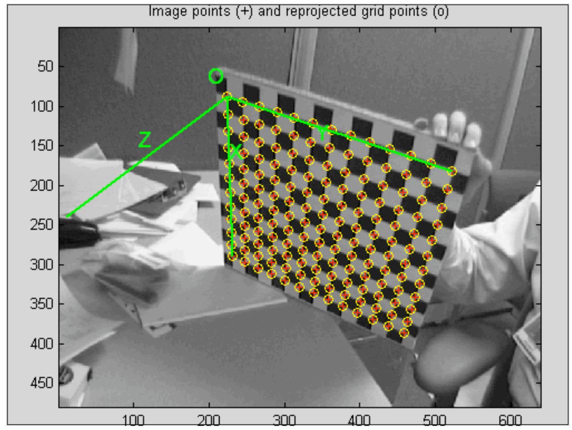
reprojected corner locations

```
Calibration parameters after initialization:

Focal Length:          fc = [ 671.13759   680.77186 ]
Principal point:       cc = [ 319.50000   239.50000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 20
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 661.67001   662.82858 ] ± [ 1.17913   1.26567 ]
Principal point:       cc = [ 306.09590   240.78987 ] ± [ 2.38443   2.17481 ]
Skew:             alpha_c = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.
Distortion:            kc = [ -0.26425   0.22645   0.00020   0.00023  0.00000 ] ± [ 0.00934
Pixel error:          err = [ 0.45330   0.38916 ]

Note: The numerical errors are approximately three times the standard deviations (for reference
```










```
Extrinsic parameters:

Translation vector: Tc_ext = [ -94.617156        -184.010867      766.209711 ]
Rotation vector:   omc_ext = [ -1.451113         -1.827059        -0.179105 ]
Rotation matrix:    Rc_ext = [ -0.043583         0.875946         -0.480436
                               0.765974          0.338032         0.546825
                               0.641392          -0.344170        -0.685684 ]
Pixel error:           err = [ 0.10156   0.08703 ]
```

Rotation in the extrinsic parameters is given in Rodrigues Form

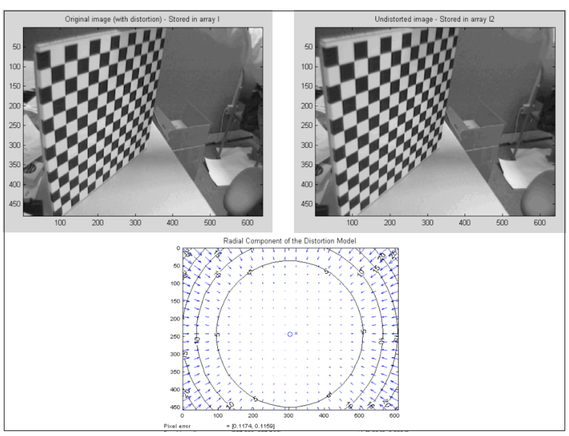
$$Rot(\mathbf{w}, \theta) =$$

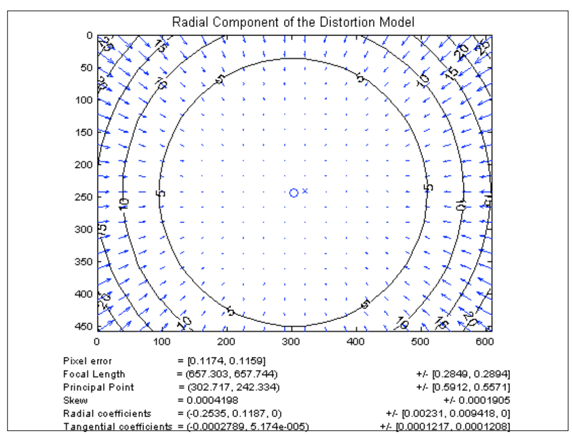
$$\begin{bmatrix} \cos\theta + \omega_x^2(1-\cos\theta) & \omega_x\omega_y(1-\cos\theta) - \omega_z\sin\theta & \omega_y\sin\theta + \omega_x\omega_z(1-\cos\theta) \\ \omega_z\sin\theta + \omega_x\omega_y(1-\cos\theta) & \cos\theta + \omega_y^2(1-\cos\theta) & -\omega_x\sin\theta + \omega_y\omega_z(1-\cos\theta) \\ -\omega_y\sin\theta + \omega_x\omega_z(1-\cos\theta) & \omega_x\sin\theta + \omega_y\omega_z(1-\cos\theta) & \cos\theta + \omega_z^2(1-\cos\theta) \end{bmatrix}$$

Distortion correction

- With m corners and n images, we can generate a system of mxn equations linear in the radial distortion coefficients (k_1, k_2).
- The system is solved with least squares.
- The complete system (intrinsics, extrinsics, and radial distortion) can be re-estimated by optimizing:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left\| m_{ij} - \hat{m}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \mathbf{t}_i, M_j)\right\|^2$$





Additional Remarks

- Nonlinear optimization converges fast in a few iterations.
- The images must have different orientation for them to provide independent constraints.
- Errors in the intrinsics grow linearly with noise.
- When the number of images is greater than 5, there is no significant change in results.