Camera calibration by Zhang

Siniša Kolarić
<http://www.inf.puc-rio.br/~skolaric>

September 2006

Abstract

In this presentation, I present a way to calibrate a camera using the method by Zhang.

NOTE. This is accompanying material to my trabalhos for the course INF2064 "Tópicos de Computação Gráfica III - Realidade Aumentada e Cooperativa" held by prof. Marcelo Gattass, during the 2006.2 semestre.

The problem

Given a set of photos (either *real* photos — made with a real camera, or *virtual* photos — made with a "virtual"[1] camera), determine camera's:

- **Intrinsic parameters**
- **Extrinsic parameters**

[1] For example the one implemented with perspective transformation in OpenGL, or one implemented in a raytracer.

Camera's intrinsic parameters

- Scaling factors — s_x, s_y
- Image center (principle point) — (o_x, o_y)
- Focal length(s) — $f (f_x = f/s_x, f_y = f/s_y)$
- Aspect ratio (skewness) — s_h
- Lens distortion (pin-cushion effect) — k_1, k_2

Camera's intrinsic matrix (Trucco & Verri)

$$K = \begin{bmatrix} -\frac{f}{sx} & s_h & o_x \\ 0 & -\frac{f}{sy} & o_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -f_x & s_h & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

- f — focal length in [m]
- s_x, s_y — scale factors in the image's u and v axis. Can be interpreted as the horizontal and vertical size (in meters) of the pixels, in another words dimensionality of s_x, s_y is [m/pixel].
- f_x, f_y — focal lengths in [pixel].
- s_h — skewness of two image axes (dimensionless). Holds $s_h = \tan\delta \approx 0$ (because generally $\delta \approx 0$), where δ is the angle between axis y and verticals on the axis x.
- (o_x, o_y) — coordinate pair of the *principal point* (intersection of optical axis with image plane), expressed in [pixel]. Also called *image center*.

Camera's intrinsic matrix (Faugeras)

$$K = \begin{bmatrix} -f k_u & 0 & u_0 \\ 0 & -f k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Remarks:

- no skewness factor
- $k_u = s_x^{-1}, k_v = s_y^{-1}$

Camera's intrinsic matrix (IMPA folks)

$$K = \begin{bmatrix} fs_x & f\tau & u_c \\ 0 & fs_y & v_c \\ 0 & 0 & 1 \end{bmatrix}$$

Compared with Trucco/Verri and Faugeras, IMPA people have added the following changes:

- change of sign for diagonal elements k_{11}, k_{22}
- s_x, s_y are defined as inverted values of s_x, s_y in Trucco/Verri notation

Camera's intrinsic matrix

- Having images only, it's not possible to estimate individual values of f, s_x, s_y; only values f_x and f_y can be estimated
- However if the manufacturer supplied s_x, s_y with the camera, it's possible to derive f
- If we discover f_x, it will be expressed in [pixel]. So if we know the height H of the image (also expressed in [pixel]), we can calculate fov_y.

Camera's extrinsic parameters

- Placement of the camera (translation vector t)
- Orientation of the camera (rotation matrix R)

Complete chain of coordinate transforms

$$\text{pixels} \leftarrow \begin{bmatrix} \frac{1}{s_x} & s_h & o_x \\ 0 & \frac{1}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \leftarrow \text{image} \leftarrow \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \leftarrow \text{camera} \leftarrow \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \leftarrow \text{world}$$

Combining first two matrices we get:

$$\text{pixels} \leftarrow \begin{bmatrix} -\frac{f}{s_x} & s_h & o_x \\ 0 & -\frac{f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \leftarrow \text{camera} \leftarrow \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \leftarrow \text{world}$$

$$\text{pixels} \leftarrow \begin{bmatrix} -f_x & s_h & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \leftarrow \text{camera} \leftarrow \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \leftarrow \text{world}$$

Zhang's method

ZHANG()

1 take several($n \geq 3$) photos of your planar model's printout
2 detect features in photos using LoG, jvInterpret(), etc
3 calculate camera's extrinsic and intrinsic parms using closed-form solution
4 calculate coeffs for radial distortion solving linear least-squares
5 fine tune calculated parms using Levenberg-Marquardt
6 output calculated parms

There can be less than 3 photos, but only under the supposition that some intrinsic parameters are known, see below.

Zhang's method

- Firstly, the standard pinhole camera is being considered
- Then, radial distortion is being calculated on top of it

Zhang uses *planar* 3-D models

"Planar" means that we can flatten coordinate Z of every point of the model in the Zhang method, that is, consider Z to be 0.

Examples of planar 3-D models would be, for example, patterns of black rectangles with known dimensions, printed on a paper, glued to a hard-cover book, and photographed by a camera.

Therefore, $[X\,Y\,Z\,1]^\tau$ (a 3-D point of the model) can be treated as $[X\,Y\,1]^\tau$ in all subsequent calculations, since $Z = 0$ for all points.

General projective transformation can be simplified

Because of the simplification $[X\,Y\,Z\,1]^\tau \longrightarrow [X\,Y\,1]^\tau$, we can simplify the general projective transformation

$$[X\,Y\,Z\,1]^\tau \longrightarrow K[R\,t][X\,Y\,Z\,1]^\tau$$

as

$$[X\,Y\,1]^\tau \longrightarrow K[r_1\;r_2\;t][X\,Y\,1]^\tau$$

where r_1 and r_2 are the first two columns of rotation matrix R, t translation vector, and K intrinsic matrix. By this reduction, we can work with a simpler projective plane to projective plane transformation ($P^2 \longrightarrow P^2$) instead with the more general and more complex ($P^3 \longrightarrow P^2$) transformation.

Homography

Because it uses planar 3-D models, Zhang's method makes use of a homography (which is a map from projective plane P^2 onto itself):

$$[X\,Y\,1]^\tau \longrightarrow K[R\,t][X\,Y\,1]^\tau = \frac{1}{\lambda}H[X\,Y\,1]^\tau = [u\,v\,1]^\tau$$

where

- H is a homography from the model plane to the image plane $P^2 \longrightarrow P^2$ defined as $H = \lambda\,K[R\,t]$
- K is camera's intrinsic matrix, and R, t extrinsic matrices

Homography

- There is a factor λ in the definition of H because any homography is defined up to a factor.

The idea behind the Zhang method

Let M designate the set of 2-D model points, and M_i' set of 2-D points detected in image i. In a nutshell, the idea is first to extract n homographies H_i (3x3 matrices) from n pairs $\{M, M_i'\}, i = 1, \ldots n$:

$$\{M, M_1'\} \longrightarrow H_1 = \begin{bmatrix} {}^1h_{11} & {}^1h_{12} & {}^1h_{13} \\ {}^1h_{21} & {}^1h_{22} & {}^1h_{23} \\ {}^1h_{31} & {}^1h_{32} & {}^1h_{33} \end{bmatrix}$$

$$\{M, M_2'\} \longrightarrow H_2 = \begin{bmatrix} {}^2h_{11} & {}^2h_{12} & {}^2h_{13} \\ {}^2h_{21} & {}^2h_{22} & {}^2h_{23} \\ {}^2h_{31} & {}^2h_{32} & {}^2h_{33} \end{bmatrix}$$

$$\ldots$$

$$\{M, M_n'\} \longrightarrow H_n = \begin{bmatrix} {}^nh_{11} & {}^nh_{12} & {}^nh_{13} \\ {}^nh_{21} & {}^nh_{22} & {}^nh_{23} \\ {}^nh_{31} & {}^nh_{32} & {}^nh_{33} \end{bmatrix}$$

The idea behind the Zhang method

Then we use these newly-found coefficients of H_i (eight coefficients for each H_i, because all homographies have 8 DOF, that is, are determined up to a factor) to setup a linear system of $2n$ ($n =$ number of images) equations for five intrinsic parameters (unknowns) $s_x, s_y, \gamma, u_0, v_0$ — elements of K. Thus in this way, we end up finding (estimating) intrinsic matrix K.

With K in hand, we find extrinsics $R = [\vec{r}_1, \vec{r}_2, \vec{r}_3]$ and t for each image i:

$$\vec{r}_1 = \lambda K^{-1}\vec{h}_1 \qquad \vec{r}_2 = \lambda K^{-1}\vec{h}_2 \qquad \vec{r}_3 = \vec{r}_1 \times \vec{r}_2$$

$$\vec{t} = \lambda K^{-1}\vec{h}_3$$

The idea behind the Zhang method

Please note that n matrices R_i so calculated do not in general case satisfy the properties of rotation matrix (that is, columns and rows of R_i aren't unitary orthogonal vectors) due to inherent noise in data. There is a method, however, that allows us to find a rotation matrix that is most similar to R_i — see the article by Zhang.

Of course, noise also affects the other extrinsic parameter t_i.

The linear system

A couple of remarks about the aforementioned linear system for $s_x, s_y, \gamma, u_0, v_0$.

For each image i (and thus each homography $H_i = K[r_{1i}\ r_{2i}\ t_i]$), we have a *pair* of constraining equations:

$$h_{1i}^{\tau} K^{-\tau} K^{-1} h_{2i} = 0$$

$$h_{1i}^{\tau} K^{-\tau} K^{-1} h_{1i} = h_{2i}^{\tau} K^{-\tau} K^{-1} h_{2i}$$

where $H_i = [h_{1i}\ h_{2i}\ h_{3i}]$ and h_{1i}, h_{2i}, h_{3i} are columns of H_i.

Since h_{1i}, h_{2i}, h_{3i} are knowns, unknowns are six (6) coefficients of $B := K^{-\tau} K^{-1}$. Also, every such pair of equations gives constraints for two $(2 = 8 - 6)$ intrinsics, since a homography has 8 DOF and there are 6 extrinsic parameters.

The linear system

If:

- $n = 1$ — then we can only solve two intrinsic parameters, for example s_x and s_y. In this case, we set $u_0 = \frac{W}{2}, v_0 = \frac{H}{2}, \gamma = 0$, that is, u_0, v_0 are at the image center, and skewness is zero.

- $n = 2$ — then we can solve four intrinsic parameters, for example s_x, s_y, u_0, v_0. In this case, we set $\gamma = 0$.

- $n \geq 3$ — the linear system becomes overdetermined, so we have a unique solution (up to a factor) for all five intrinsics $s_x, s_y, \gamma, u_0, v_0$.

This was just an estimate

Finally, matrix K, matrices R_i, and vectors t_i so calculated, are just an *estimate* of the ground truth.

To improve the accuracy of results, matrix K, matrices R_i, and vectors t_i will be fed into a nonlinear-minimization solver. That is, matrix K, matrices R_i, and vectors t_i are treated as an *initial guess* for further refinement.

Minimizing a functional

In other words, now it's necessary to minimize the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|\vec{m}_{ij} - \hat{\vec{m}}(K, R_i, t_i, M_j)\|^2$$

where

- $\vec{m}_{ij}$ is an observed (detected) point j in image i, and
- $\hat{\vec{m}}(K, R_i, t_i, M_j)$ is the (re)projection of model point M_j onto image i using estimated K, R_i, t_i

Minimizing a functional

Therefore, $\hat{\vec{m}}$ is an estimate:

$$\hat{\vec{m}}(K, R_i, t_i, M_j) = H[X\,Y\,1]_j^\tau = K[r_1\,r_2\,t][X\,Y\,1]_j^\tau$$

The nonlinear minimization solver usually used for the aforementioned problem is Levenberg-Marquardt solver.

Camera's optical center in planar model's space

One of the tasks was to find the position of the camera's optical centre C expressed in the coordinate system defined by image i (that is, coordinate system defined by the pose of calibration pattern during which image i has been taken).

GETCOORDSOFCAMERASOPTICALCENTREINIMAGESYSTEM()

1 get extrinsics matrix $[R\ t]$ for image i, where $t = [t_1, t_2, t_3]^\tau$

2 **return** $R^\tau(-t)$

Camera's optical center in planar model's space

EXPLANATION. Let points O_{im} and O_{cam} be origins of the image system and the camera system, respectively, in an absolute system. Let $\vec{t}$ be the vector that gives the position of O_{im} in the camera system. Let $\vec{r}_1, \vec{r}_2, \vec{r}_3$ define the unitary base of the image system (in relation to the unitary base of the camera system). Because of this, matrix R which translates image's canonical base $\{\vec{i}', \vec{j}', \vec{k}'\}$ into camera's canonical base $\{\vec{i}, \vec{j}, \vec{k}\}$, has columns $\vec{r}_1, \vec{r}_2, \vec{r}_3$ and is an unitary matrix, $RR^\tau = R^\tau R = I$. In other words, matrix R rotates (but only after translation $\vec{t}$) image points into their corresponding camera points. Therefore, given a point with image coordinates $P_{im} = (X, Y, Z = 0)$, its coordinates P_{cam} in the camera system are equal to the coordinates of the vector

$$\overrightarrow{O_{cam}P_{im}} = \overrightarrow{O_{cam}O_{im}} + \overrightarrow{O_{im}P_{im}} \qquad \longrightarrow \qquad P_{cam} = \vec{t} + RP_{im}$$

Camera's optical center in planar model's space

Now using the fact that $RR^\tau = R^\tau R = I$, and multiplying $P_{cam} = t + RP_{im}$ by R^τ from the left side:

$$R^\tau P_{cam} = R^\tau t + R^\tau R P_{im} = R^\tau t + P_{im}$$

$$\longrightarrow P_{im} = R^\tau P_{cam} - R^\tau t$$

Therefore, given a point expressed in camera coordinates (P_{cam}), we can calculate its image coordinates (P_{im}).

Finally, in the special case $P_{cam} = O_{cam} = (0, 0, 0)_{cam}$, the formula above gives:

$$P_{im} = R^\tau P_{cam} - R^\tau t = -R^\tau t = R^\tau \cdot (-t)$$

and this way we get optical centre O_{cam} expressed in image coordinates.

Camera's optical center in planar model's space

Using homogenuous coordinates, matrix $[R\ t]$ transforms image coordinates into camera coordinates. More precisely:

$$\begin{bmatrix} R & t \\ 0_3 & 1 \end{bmatrix} P = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = P'$$

where $P = (X, Y, Z, 1)$ and $P' = [X', Y', Z', 1]$.

Calculating fov_y

How to calculate fov_y for image i, from the intrinsics matrix $K_i = \{k_{ij}\}$?

Holds

$$\tan \frac{fov_y}{2} = \frac{H/2}{f_y}$$

where H is the height of image expressed in pixels, and $f_y = k_{22}$ is the focal length in the y direction, also expressed in pixels. Now

$$fov_y = 2 \arctan \frac{H/2}{f_y} \quad \text{[rad]}$$

Finally, $fov_y \cdot \frac{360^\circ}{2\pi\,[rad]}$ is the field of view expressed in degrees $^\circ$.

References

- *A flexible new technique for camera calibration*, Zhengyou Zhang, Technical Report MSR-TR-98-71 (report last updated on March 25, 1999)