CSE 252B: Computer Vision II

Lecturer: Serge Belongie
Scribe[1]: Martin Stiaszny and Dana Qu

LECTURE 10
Camera Calibration

10.1. Introduction

Just like the mythical frictionless plane, in real life we will seldom, if ever, encounter a truly ideal pinhole camera. Real cameras have focal lengths that can change or get knocked out of adjustment. Thus, in any real system, the first thing one must usually do is camera calibration. We deal with two ways of doing so in this lecture. The first is calibration from special camera motions (such as pure translation or pure rotation), and the second is by the use of a calibration "rig" – i.e., an object with known and desirable properties placed in the scene.

10.2. Radial Distortion

Real cameras have lenses, and lenses have imperfections. Usually, the most significant kind of distortion is radial lens distortion, which increases as lens

[1] Edited Spring 2005 by Shay Har-Noy.

[1] Department of Computer Science and Engineering, University of California, San Diego.

April 28, 2004

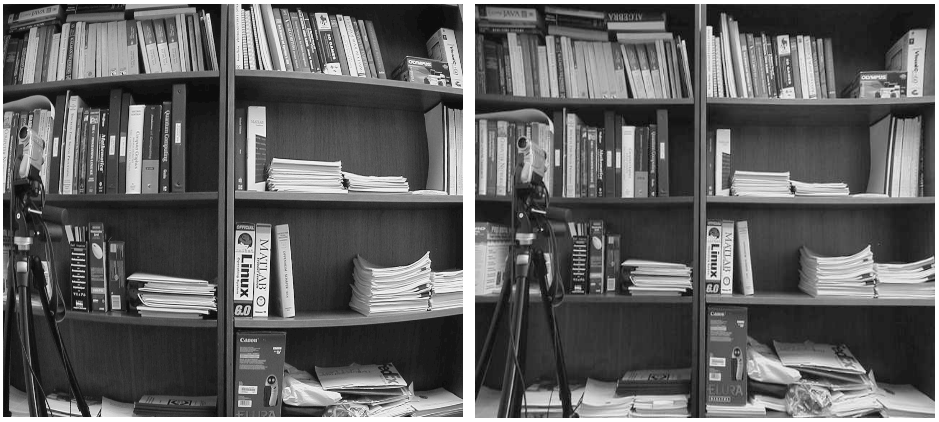

Figure 1. Example of radially distorted image (left) and its rectified version (right). (From MaSKS)

size and cost decrease. We can model radial distortion as

$$\boldsymbol{x} = \boldsymbol{c} + f(r)(\boldsymbol{x}_d - \boldsymbol{c})$$

$$f(r) = 1 + ar + a_2 r^2 + a_3 r^3 + a_4 r^4$$

where $\boldsymbol{x}_d = (x_d, y_d)^\top$ is the distorted image coordinates, $r^2 = \|\boldsymbol{x}_d - \boldsymbol{c}\|^2$, and $\boldsymbol{c} = (c_x, c_y)^\top$ is the center of the distortion (which is not necessarily the center of the image). You can think of applying $f(r)$ in concentric circles. Solving for a_1 through a_4 and $\boldsymbol{c}$ is a non-linear optimization problem that will not be dealt with in this lecture, but there are several good toolboxes that can take care of it; see the links on course webpage. However, regardless of the method you eventually use, it is very important to do distortion correction. Figure 1 shows a vivid illustration of this.

10.3. Calibration from Pure Rotation[1]

(MaSKS example 6.10) Recall the general image formation equation:

$$\lambda_2 \boldsymbol{x}'_2 = KRK^{-1}\lambda_1 \boldsymbol{x}'_1 + K\boldsymbol{T}$$

Under pure rotation, $\boldsymbol{T} = \boldsymbol{0}$, so

$$\lambda_2 \boldsymbol{x}'_2 = KRK^{-1}\lambda_1 \boldsymbol{x}'_1$$

for some $\lambda_1, \lambda_2 \in \mathbb{R}_+$. The 3×3 matrix KRK^{-1} is the uncalibrated counterpart to the rotational homography. As in the calibrated case, we can

[1] This would be applicable using a Pan-Tilt-Zoom (PTZ) camera with fixed zoom, for example.

eliminate the unknown depths algebraically by crossing both sides with $\boldsymbol{x}_2'$:

$$\lambda_2 \boldsymbol{x}_2' \sim KRK^{-1}\boldsymbol{x}_1' \quad \Longrightarrow \quad \widehat{\boldsymbol{x}_2'}KRK^{-1}\boldsymbol{x}_1' = \mathbf{0}$$

Letting $C = KRK^{-1}$, we can estimate C using the 4 point algorithm. Hartley's calibration method requires two such C's for R's around different axes.

Theorem 10.1. *(MaSKS Theorem 6.9) Given $C_i = KR_iK^{-1}$, for $i = 1, 2$ where $R_i = e^{\widehat{\mu_i}\theta_i}$ with $\|\mu_i\| = 1$ and $\theta \neq k\pi$, $k \in \mathbb{Z}$ then $S^{-1} - CS^{-1}C^\top = 0$ (with $S^{-1} = KK^\top$) has a unique solution for S^{-1} iff $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are linearly independent.*

Stated informally, this theorem says if you take two images with "random" camera rotations, you can estimate K.

Algorithm

Given $C_i, i = 1, \ldots, m$ for $m \geq 2$, we can rewrite the m equations

$$S^{-1} - C_iS^{-1}C_i^\top = 0$$

as

$$\chi \boldsymbol{s} = \mathbf{0}$$

where $\chi \in \mathbb{R}^{6m\times 6}$ is the design matrix and $\boldsymbol{s} \in \mathbb{R}^6$ contains the stacked upper triangular entries of the symmetric matrix S^{-1}. Since S^{-1} is symmetric, we can recover the remaining 3 entries using the 6 entries in $\boldsymbol{s}$. As usual, recover $\boldsymbol{s}$ using the SVD and reshape it to form S^{-1}. Then you can get K from the Cholesky decomposition of S^{-1}.

10.4. Calibration using a rig

We now look at how to recover the camera calibration parameters from a calibration object (or "rig"). We will look at two cases: a single view of a cube and multiple views of a plane.

10.4.1. Example 1: Checkerboard Covered Cube (MaSKS 6.5.2)

Figure 2 shows a checkerboard-covered cube for which all the geometry is known. Let $\boldsymbol{X} = (X, Y, Z, 1)^\top$ be the the coordinates of some known point p on the rig. Its image $\boldsymbol{x}'$ is given by

$$\lambda \boldsymbol{x}' = \Pi \boldsymbol{X} = K\Pi_o g\boldsymbol{X}$$

Let $\mathbf{\Pi}_1, \mathbf{\Pi}_2, \mathbf{\Pi}_3 \in \mathbb{R}^4$ denote the rows of Π:

$$\Pi = \begin{bmatrix} \mathbf{\Pi}_1^\top \\ \mathbf{\Pi}_2^\top \\ \mathbf{\Pi}_3^\top \end{bmatrix} \in \mathbb{R}^{3\times 4}$$

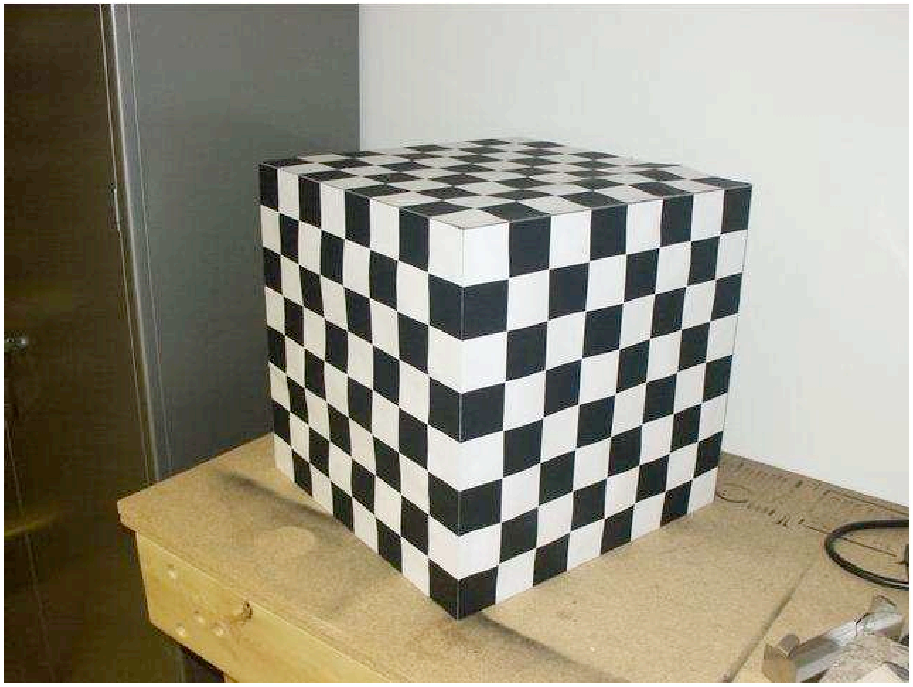

Figure 2. Cube-shaped calibration rig (from MaSKS).

Then for each point p^i on the rig, we can write

$$\lambda^i \begin{bmatrix} (x^i)' \\ (y^i)' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{\Pi}_1^\top \\ \mathbf{\Pi}_2^\top \\ \mathbf{\Pi}_3^\top \end{bmatrix} \begin{bmatrix} X^i \\ Y^i \\ Z^i \\ 1 \end{bmatrix}$$

The third coordinate implies that $\lambda^i = \mathbf{\Pi}_3^\top \boldsymbol{X}^i$. From this, we get the two equations:

$$(x^i)'(\mathbf{\Pi}_3^\top \boldsymbol{X}^i) = \mathbf{\Pi}_1^\top \boldsymbol{X}^i \quad \text{and} \quad (y^i)'(\mathbf{\Pi}_3^\top \boldsymbol{X}^i) = \mathbf{\Pi}_2^\top \boldsymbol{X}^i$$

Note that X^i, Y^i, Z^i, $x^{i\prime}$, and $y^{i\prime}$ are all known. We can put this into the standard form

$$M\mathbf{\Pi}^s = \mathbf{0}$$

with $\mathbf{\Pi}^{\boldsymbol{s}} = \Pi(:)$ and $M \in \mathbb{R}^{2\times 12}$. (The exact entries of M are left as an exercise.) Using the SVD, we can get a linear (suboptimal) estimate of Π. As a sidenote, the process of recovering Π from an object with known geometry is known as *resection.*

Note that $\Pi = K[R, \boldsymbol{T}] = [KR, K\boldsymbol{T}]$. The first 3×3 block of Π is KR, so use the QR decomposition to separate K from R. Finally, $\boldsymbol{T} = K^{-1}\Pi(:, 4)$.

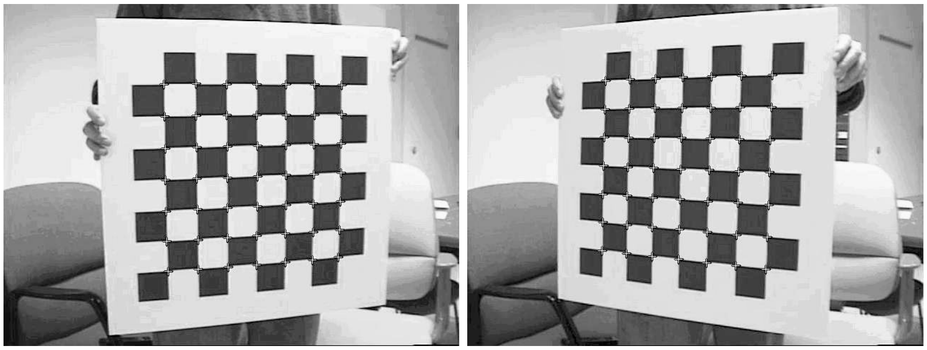

Figure 3. Two views of a planar calibration grid (from MaSKS).

Note that this method breaks down if the calibration pattern is planar. Since planar calibration grids are much easier to construct, the next example addresses that case.

10.4.2. Example 2: planar checkerboard[2]

Consider the planar checkerboard shown in Figure 3. For simplicity, choose the world reference frame to lie on the board. Then all points have the form $\boldsymbol{X} = (X, Y, 0, 1)^\top$ and the Z axis is the normal vector of the plane. This gives rise to the following simplified relation between the world coordinates and image coordinates:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = K \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{T} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

where $\boldsymbol{r}_i = R(:, i)$. The matrix $H = K(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{T}) \in \mathbb{R}^{3\times 3}$ is the homography from the checkerboard to the image plane:

$$\lambda \boldsymbol{x}' = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Using the standard trick of multiplying both sides by $\widehat{\boldsymbol{x}'}$ to eliminate unknown depth gives us

$$\widehat{\boldsymbol{x}'} H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \boldsymbol{0}$$

[2] See the Matlab calibration toolbox from Jean-Yves Bouguet.

From this we can solve for H using the 4 point algorithm. The first two columns are $[\boldsymbol{h}_1, \boldsymbol{h}_2] \sim K[\boldsymbol{r}_1, \boldsymbol{r}_2]$ or $K^{-1}[\boldsymbol{h}_1, \boldsymbol{h}_2] \sim [\boldsymbol{r}_1, \boldsymbol{r}_2]$. Since $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are orthogonal (b/c $R \in SO(3)$), their inner product with each other must be zero,

$$\boldsymbol{h}_1^\top K^{-\top} K^{-1} \boldsymbol{h}_2 = 0$$

and their inner product with themselves must be one,

$$\boldsymbol{h}_1^\top K^{-\top} K^{-1} \boldsymbol{h}_1 = \boldsymbol{h}_2 K^{-\top} K^{-1} \boldsymbol{h}_2 = 1$$

These equations are quadratic in the entries of K (b/c K is multiplied by K^T). But, if we neglect the structure of $S = (KK^\top)^{-1}$, we can solve for S linearly and factor it to get K.

It is often safe to assume that the cameras have zero skew, i.e., $s_\theta = 0$. When this is true, we have the additional constraint that so $\boldsymbol{e}_1^\top S \boldsymbol{e}_2 = 0$, where $\boldsymbol{e}_1 = (1, 0, 0)^\top$ and $\boldsymbol{e}_2 = (0, 1, 0)^\top$. (Note that these $\boldsymbol{e}_i$'s are not epipoles; this is MaSKS's convention for unit vectors.)