Computer Vision

CPS 296.1 Supplementary Lecture Notes

Carlo Tomasi – Duke University

Fall 2007

3. Camera Calibration

[Last Modified on October 15, 2007]

Warning. These notes are being written as the course progresses. As errors are discovered, old notes will be corrected and the new version will be posted online.

3 Camera Calibration

3.1 Introduction

A significant fraction of computer vision work involves reconstructing the geometry of visible surfaces in the world from their images. Reconstruction attempts to reverse image formation, so it is important to know the values of the parameters of the projection model for the specific cameras with which the images were taken. *Interior camera calibration* is an experimental[1] procedure for determining these values. The projection model has been developed in the Section on image formation, and its parameters include focal length, image position of the principal point, pixel size, and distortion coefficients.

The origin of the coordinate reference system for the camera model is most conveniently placed at the center of projection, which is obviously a very special point in the geometry of image formation. However, when two or more cameras are simultaneously present, it is not possible[2] to place them all with their centers of projection at the same point, so one must also consider the situation in which some of the cameras have their centers of projection at points other than the origin.

Even when a single camera moves over time it is often useful to describe both world and camera positions in a single reference frame, so that the coordinates of the camera's center of projection vary with time.

In these cases, one then needs to determine the position and orientation of a camera in a separate system of reference, typically in the form of a translation and a rotation of one reference frame with respect to the other. This is done through an experimental procedure called *exterior camera calibration*.

In both types of camera calibration procedures, interior and exterior, the input is a number of images of one or more *calibration objects* designed and built for the purpose, and whose dimensions and characteristics are known. A numerical optimization procedure then fits the appropriate parameters in the known parametric relationship between world coordinates (the known coordinates of points on the calibration objects) and image coordinates (the coordinates of the corresponding points measured in the images).

This optimization problem is typically non-convex, so an approximate solution is often required as input as well. As an alternative, an approximate, convex problem is solved first to determine the initial solution. The final, numerical, local optimization stage is called *bundle adjustment*, because it refines (adjusts) the parameters that describe the stars (bundles) of projection rays that intersect at the various centers of projection.

A classical case of camera calibration is the calibration of a pair of cameras used for stereoscopic vision. In this case, two internal calibrations and one external calibration are needed. This case needs separate attention because both reference frames are tied to cameras. The use of the calibration parameters in stereo is described in a later Section.

The next two Sections summarize the image formation model for perspective projection and the

[1] "Experimental" here means that experiments, that is, measurements are involved, and not that the procedure is still under experimental development.

[2] Except for special devices with so-called *confocal optics*.

geometry of reference system transformations. The Sections thereafter discuss interior calibration, exterior calibration, stereo calibration, practical issues in calibration, and some numerical aspects of bundle adjustment.

3.2 Interior Parameters

The essential equations of the interior camera model are now summarized from the Section on image formation. For convenience, the ideal pinhole camera is assumed to have a focal distance of one. This is equivalent to saying that the *ideal coordinates* (x_c, y_c) in the image are measured in units of focal distance. In addition, the camera reference frame is now redefined to have its z_c axis point *away* from the scene, rather than towards it. This is done in order to make the camera reference system right-handed, a very convenient feature when dealing with transformations of coordinates. Under perspective projection, the world point with camera-system coordinates (X_c, Y_c, Z_c) then projects to the image point with camera-system coordinates

$$x_c = -\frac{X_c}{Z_c} \quad \text{and} \quad y_c = -\frac{Y_c}{Z_c} \tag{1}$$

where the minus signs come from the new convention about the z_c axis. Lens distortion then transforms these coordinates to *distorted coordinates*

$$x_d = x_c d(r) \quad \text{and} \quad y_d = y_c d(r) \tag{2}$$

where

$$r = \sqrt{x_c^2 + y_c^2}$$

is the distance of the ideal image point from the principal point, and the *distortion function* $d(r)$ is approximated by the following fourth order polynomial in r:

$$d(r) = 1 + k_2 r^2 + k_4 r^4 \ . \tag{3}$$

Finally, if x_0 and y_0 are the coordinates in pixels of the principal point of the image in the ideal, pinhole image reference system (x, y), then an image point with distorted coordinates $(x_d, y_d, 1)$ (in units of focal distance) in the camera reference frame has image coordinates (in pixels)

$$x = s_x x_d + x_0 \quad \text{and} \quad y = s_y y_d + y_0 \tag{4}$$

where s_x and s_y are scaling constants expressed in pixels per focal distance. In other words, s_x is the effective[3] focal distance expressed in pixel widths, and s_y is the effective focal distance expressed in pixel heights. The aspect ratio of a pixel is $w/h = s_y/s_x$ (please pay attention here: a greater value of s_x corresponds to a smaller pixel width).

[3] "Effective" here refers to the fact that these focal distances account for lens distortion.

Practical Aspects: Other calibration parameters. Many camera models also include a *skew* factor to account for a horizontal skew between rows. Skew is typically zero with today's solid-state cameras. In calibration methods that output skew, this parameter is often clamped to zero.

A sixth-degree distortion term and a radial distortion factor are part of many camera models as well. These distortions are almost always negligible, and the greater numerical instability added by the introduction of the corresponding parameters typically outweighs any benefits deriving from their use.

3.3 Exterior Parameters

The Cartesian coordinates of points in the world are expressed with reference to a *world reference system* whose *origin* is an arbitrary but well defined point[4] O. The Z axis has unit vector[5] $\mathbf{e}_z = [0, 0, 1]$ and points *up*, that is, in the direction opposite to gravity. The X and Y axes are horizontal (that is, orthogonal to gravity), mutually orthogonal, and have unit vectors $\mathbf{e}_x = [1, 0, 0]$, and $\mathbf{e}_y = [0, 1, 0]$, respectively. The positive directions of the X and Y axes are chosen so that the triple XYZ is right-handed. Other than these constraints, the directions of the X and Y axes are arbitrary.

The position and orientation of each camera is identified by its own *camera reference system*. The origin of this system is the camera's center of projection. Its coordinates in the world are collected in a column vector $\mathbf{t} = [t_x, t_y, t_z]^T$. The Z_c axis of a camera's reference system points along the optical axis of the camera and away from the scene. The X_c and Y_c axes are respectively parallel to the rows and columns of the camera's imaging sensor array. The positive direction of the Y_c axis is to the right when the image is viewed upside-up, and the positive direction of the Y axis is upwards in the image. This choice makes the camera reference system right-handed.

The three unit vectors along the X_c, Y_c, Z_c axes of a camera reference frame are denoted by $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ (possibly with subscripts to distinguish among different cameras). Their coordinates (obviously in the world reference frame) are conveniently collected in a 3×3 rotation matrix

$$R = \begin{bmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{bmatrix} .$$

The transformation between coordinates $\mathbf{X} = (X, Y, Z)$ of a point in the world reference frame and the coordinates $\mathbf{X}_c = (X_c, Y_c, Y_c)$ of the same point in the camera reference frame are (see Section on 3D Linear Algebra in the Math Supplement)

$$\mathbf{X}_c = R(\mathbf{X} - \mathbf{t})$$

and

$$\mathbf{X} = R^T \mathbf{X}_c + \mathbf{t} .$$

[4] It is often convenient to pick O as the corner of a room or a table, or a point with known GPS coordinates.

[5] Vectors along reference axes are written as row vectors. Vectors that denote points in the world are written as column vectors.

3.4 Mathematical Structure of the Calibration Problem

In camera calibration, images are taken of a known target. The coordinates of a set of points on the target, together with the coordinates of the corresponding points in the images, are the input to calibration. The output is a set of camera parameters. The relationship between all these quantities is known as the image formation model, and was developed in earlier Sections.

Here we temporarily abstract away the precise meaning of each variable in order to place more emphasis of the general structure of the calibration problem. To this end, collect all the world coordinates into a vector $\mathbf{X}$, and all the image coordinates into a vector $\mathbf{x}$. If n points are identified on the calibration object, then $\mathbf{X}$ has $3n$ entries. With m images taken of the object, the n points give rise to $2mn$ entries in $\mathbf{x}$. The number k of parameters to be determined is small in the simpler cases: six for internal calibration, and six more for external. Let $\mathbf{p}$ be the vector that collects these parameters. We can perform internal calibration first and external thereafter, so we have two calibration problems where k is six. Alternatively, we can think of a single calibration procedure, so that $k = 12$ instead. Other types of calibration procedures, described in later Sections, have more parameters because multiple camera positions are considered.

In any case, the equations of image formation describe a function that depends on the parameters $\mathbf{p}$ and yields image points $\mathbf{x}$ as a function of world points $\mathbf{X}$. For a single point with world coordinates $\mathbf{X}^{(j)}$ and coordinates $\mathbf{x}_i^j$ in image i,

$$\mathbf{x}_i^j = \mathbf{f}_i(\mathbf{X}^{(j)}; \mathbf{p})$$

where $\mathbf{f}_i : \mathbb{R}^3 \times \mathbb{R}^k \to \mathbb{R}^2$. For the vectors $\mathbf{X}$ and $\mathbf{x}$, we can write

$$\mathbf{x} = \mathbf{f}(\mathbf{X}; \mathbf{p}) \ . \tag{5}$$

where $\mathbf{f} : \mathbb{R}^{3n} \times \mathbb{R}^k \to \mathbb{R}^{2mn}$ is obtained by stacking n replicas of $\mathbf{f}_1, \ldots, \mathbf{f}_m$.

Calibration then assumes $\mathbf{X}$ and $\mathbf{x}$ to be given and solves equation (5) for $\mathbf{p}$. So, in spite of the variable names, the unknown is $\mathbf{p}$.

The solution of equation (5) could be approached in the literal sense: find a vector $\mathbf{p}$ such that the equation is satisfied. In reality, this equation is an idealization of reality: the function $\mathbf{f}$ is approximate (for instance, we approximated the distortion function with a fourth-degree polynomial); the coordinates $\mathbf{X}$ on the calibration object are not exact (perhaps a supposedly planar checkerboard pattern is glued on a slightly curved board instead); and the image coordinates have some errors (resulting from factors such as image discretization or quantization, lens blur, or image noise).

Errors in $\mathbf{f}$ and $\mathbf{X}$ are systematic, and can only be reduced by careful design of the functions $\mathbf{f}_i$ and by good construction of the calibration target. The errors in $\mathbf{x}$, on the other hand, can often be modeled as random, and statistical distributions can be given for them, either as a result of a separate calibration process or, as we shall see, as a byproduct of calibration itself. In that case, the problem of determining $\mathbf{p}$ is most usefully cast as one of maximum likelihood estimation: the conditional probability

$$p(\boldsymbol{\epsilon} \mid \mathbf{p}, \mathbf{X})$$

of the *vector residual*

$$\boldsymbol{\epsilon} = \mathbf{x} - \mathbf{f}(\mathbf{X}; \mathbf{p})$$

given the calibration parameters $\mathbf{p}$ is considered as a function of $\mathbf{p}$,

$$\Lambda(\mathbf{p}) = p(\boldsymbol{\epsilon} \mid \mathbf{p}, \mathbf{X}) \ ,$$

called the *likelihood function*, and parameter estimation computes

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \Lambda(\mathbf{p}) \ .$$

For lack of more detailed information, and supported by the central limit theorem, we assume that the conditional probability above is Gaussian, zero mean, and independent and with the same distribution on each scalar entry of $\mathbf{x}$.

Under these assumptions,

$$-\log \Lambda(\mathbf{p}) \propto \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \ ,$$

so that the maximum likelihood solution is equivalent to the standard least-squares solution:

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} \epsilon^2(\mathbf{p}) \quad \text{where} \quad \epsilon(\mathbf{p}) = \sqrt{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}} = \|\mathbf{x} - \mathbf{f}(\mathbf{X}; \mathbf{p})\| \ . \tag{6}$$

Since the functions f_i that describe image formation are nonlinear, so is the *scalar residual* $\epsilon^2(\mathbf{p})$. If the world coordinates $\mathbf{X}$ of the calibration object are considered to be fixed, this problem is a standard *inverse problem*. Properties of this problem and solution methods are discussed in the Section on inverse problems in the Math Supplement. That discussion shows several important points:

- As defined in equation (6), image calibration is a *consistent* estimation problem, that is one whose bias tends to zero as the number of image measurements tends to infinity.

- Calibration is solved in two steps: (i) Find a solution $\mathbf{p}_0$ to an approximate version of the problem, and then (ii) Refine that solution by a numerical, local optimization routine such as conjugate gradients.

- A χ^2 test on the residual can reveal whether a local as opposed to a global minimum is likely to have been attained.

- Study of the Jacobian of the image formation function $\mathbf{f}$ yields an understanding of the sensitivity and conditioning properties of the problem.

The next Sections revisit these properties for the specific problem of camera calibration.

3.5 Calibration of a Single Camera

In this Section, we assemble the equations from the image formation model summarized in Sections 3.2 and 3.3 into a parameter estimation problem of the form discussed in Section 3.4. We then use the results from the Section on sensitivity of inverse problems in the Math Supplement to gain insight into how to design and use a suitable calibration target.

Given n points and m images, collect the world coordinates $\mathbf{X}^{(j)}$ and image coordinates $\mathbf{x}_i^{(j)}$ of the calibration points into vector $\mathbf{X}$ and $\mathbf{x}$:

$$\begin{aligned} \mathbf{X} &= [X^{(1)}, Y^{(1)}, Z^{(1)}, \ldots, X^{(n)}, Y^{(n)}, Z^{(n)}]^T \in \mathbb{R}^{3n} \\ \mathbf{x} &= [x_1^{(1)}, y_1^{(1)}, \ldots, x_1^{(n)}, y_1^{(n)}, \ldots, x_m^{(1)}, y_m^{(1)}, \ldots, x_m^{(n)}, y_m^{(n)}]^T \in \mathbb{R}^{2mn} . \end{aligned}$$

Collect the camera parameters into a vector $\mathbf{p}$:

$$\mathbf{p} = [x_0, y_0, s_x, s_y, k_2, k_4, \mathbf{r}_1, \mathbf{t}_1, \ldots, \mathbf{r}_m, \mathbf{t}_m]^T \in \mathbb{R}^{6(m+1)} .$$

Although there is only one camera, there are multiple images, and the camera moves between images. So there is one set of internal parameters, and possibly several (m) sets of rotation and translation parameter vectors.

Let $\mathbf{f}_i$ be the function that relates image coordinates $\mathbf{x}_i^{(j)}$ of point j in image i to the world coordinates $\mathbf{X}^{(j)}$ of point j as described in Sections 3.2 and 3.3:

$$\begin{aligned} \mathbf{X}_c^{(j)} &= R_i \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \\ \mathbf{x}_{ci}^{(j)} &= -\frac{\mathbf{X}_c^{(j)}(1:2)}{\mathbf{X}_c^{(j)}(3)} \\ \mathbf{x}_i^{(j)} &= S\mathbf{x}_{ci}^{(j)} d(r(\mathbf{x}_{ci}^{(j)})) + \mathbf{x}_0 \end{aligned} \tag{7}$$

where

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \quad \text{and} \quad \mathbf{x}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

are the transformation to pixel coordinates and the image principal point, $d(\cdot)$ is the distortion function, and $r(\cdot)$ is the radial distance of an image point in camera coordinates from the principal point.

Then $\mathbf{f}_i$ relates entries number[6] $2n(i-1)+2(j-1)+1:2$ of $\mathbf{x}$ to entries $3(j-1)+1:3$ of $\mathbf{X}$ for any j, and to entries $1:6$ as well as $6+6(i-1)+1:6$ of $\mathbf{p}$. This is simple but detailed index arithmetic. A pictorial view of the situation is illustrated in Figure 3.1, which shows the so-called *fill patterns* of two separate components of the Jacobian J of the vector function

$$\mathbf{f} = [\mathbf{f}_1, \ldots, \mathbf{f}_1, \ldots, \mathbf{f}_m, \ldots, \mathbf{f}_m]^T$$

that is obtained by stacking m groups of n copies of the functions $\mathbf{f}_i$ on top of each other, in the same order as for the entries of $\mathbf{x}$.

[6]Following Matlab-like conventions, the notation $a:b$ where a and b are integers means $a, a+1, \ldots, b$.

To draw this figure, the Jacobian J of the vector function $\mathbf{f}$ for $m = 3$ cameras and $n = 10$ points is partitioned as follows:

$$J = \left[\begin{array}{c|c} J_{\mathbf{p}} & J_{\mathbf{X}} \end{array} \right]$$

so the first $6(m+1) = 24$ columns of J make up $J_{\mathbf{p}}$, the Jacobian of the image measurements with respect to the camera parameters, and the last $3n = 30$ columns make up $J_{\mathbf{X}}$, the Jacobian of the image measurements with respect to the point coordinates. Each row of the Jacobian corresponds to an image coordinate, so both Jacobian matrices have $2mn = 60$ rows.

During calibration, only $J_{\mathbf{p}}$ matters, since the world point coordinates are fixed. The point Jacobian $J_{\mathbf{X}}$, however, is useful when gaining insight into the sensitivity of calibration, because this depends on the arrangement of the calibration points in the world.

A white point in the patterns in Figure 3.1 indicates functional dependence of a particular image coordinate with respect to either a parameter or a coordinate of a world point.

Figure 3.2(b), shows the actual parameter Jacobian $J_{\mathbf{p}}$, encoded by the magnitudes of its entries, for the scenario in Figure 3.2 (a), and Figure 3.2(c) shows the pseudo-inverse $J_{\mathbf{p}}^{\dagger}$ of $J_{\mathbf{p}}$.

3.5.1 Ambiguity

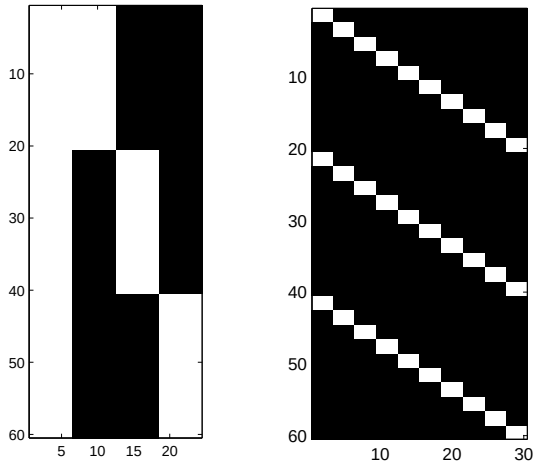
Note the high condition number of the Jacobians in Figure 3.2, which testifies to the great sensitivity of the camera calibration problem. Figure 3.3 provides a clue to one source of this problem: If the points in the world are on a plane, the condition number goes to infinity. A complete discussion of the geometric reasons for this is beyond the scope of this course (see [3] for a very thorough discussion).

However, it should be clear that a planar configuration spells trouble. Consider for instance the image of a frontal plane: a poster on a wall in front of you. For simplicity, let us assume that the lens on the camera used to take the picture has no distortion.

Say that you know the true shape of the poster, in the form of the world coordinates $\mathbf{X}$ of a set of points on the poster, relative to a reference system attached to the poster.

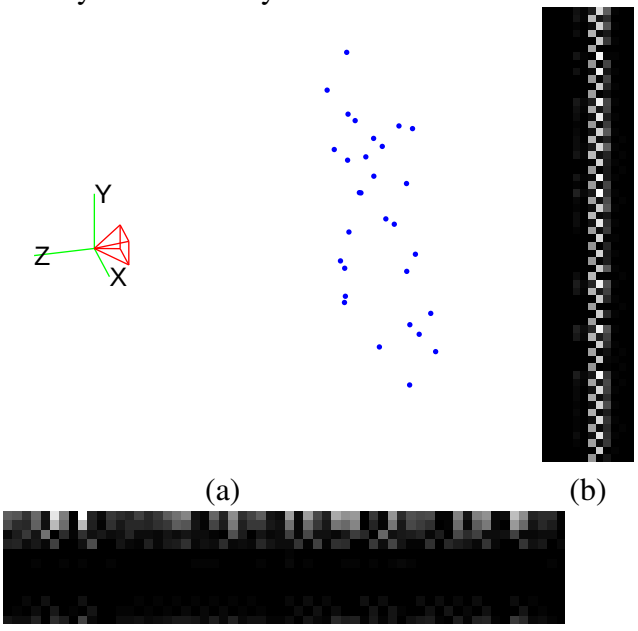
You are then told the true calibration parameters of the camera: pixel sizes s_x and s_y, principal point (x_0, y_0), translation $\mathbf{t}$ and rotation R of the camera relative to the poster's reference system. Can you change some of the camera parameters from their true values, without changing the coordinates $\mathbf{x}$ of the image points corresponding to $\mathbf{X}$? If the answer is positive, then the calibration parameters are not uniquely identified by $\mathbf{X}$ and $\mathbf{x}$: they are *ambiguous*. This ambiguity shows up as a multiplicity in the solutions to the calibration problem (6). The multiplicity could be in the form of more than one isolated solution, or of a *locus* of solutions: a curve, surface, or other manifold in the space of all possible calibration parameters $\mathbf{p}$).

A simple answer to the question posed above should come to mind rather easily: The camera can be shifted (change the translation vector $\mathbf{t}$), say, to the left. The corresponding shift of the image points to the right can be canceled by an decrease in the horizontal coordinate x_0 of the principal point. Similarly, the camera can be brought closer to the poster (decrease the z component of $\mathbf{t}$), and the corresponding increase in size can be canceled with a suitable decrease in the pixel size parameters s_x and s_y.



(a) (b)

Figure 3.1: The fill patterns of the Jacobian matrices (a) $J_{\mathbf{p}}$ and (b) $J_{\mathbf{X}}$ into which the overall Jacobian $J = [J_{\mathbf{p}} \mid J_{\mathbf{X}}]$ can be partitioned. This Jacobian is for $m = 3$ cameras and $n = 10$ points. A white point indicates a possibly nonzero entry in the Jacobian.



(a) (b)

(c)

Figure 3.2: (a) A single-camera imaging scenario. The red pyramid denotes the camera (not to scale). (b) The absolute value of the entries of the Jacobian $J_{\mathbf{p}}$ (lighter is greater) and (c) of its pseudo-inverse and $J_{\mathbf{p}}^{\dagger}$ for the scenario in (a). There is $m = 1$ camera and $n = 30$ points, so $J_{\mathbf{p}}$ is $2mn$ by $6(m + 1)$, that is, 60 by 12. The pseudo-inverse is 12 by 60. These matrices have a very high condition number, of about 10^4.

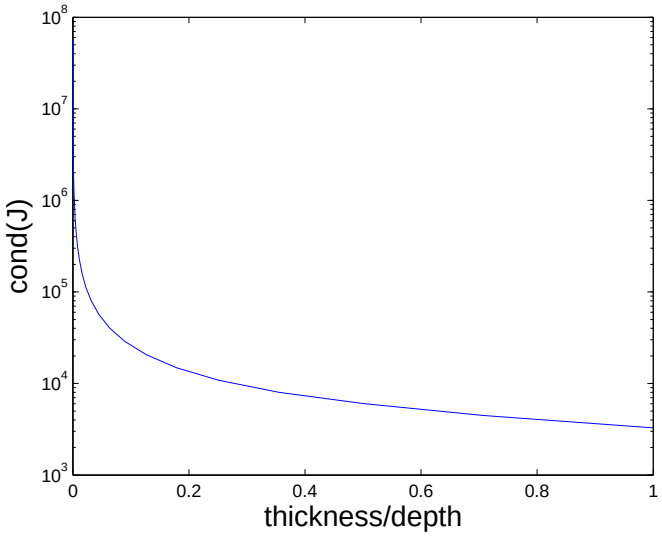


Figure 3.3: The condition number of the Jacobian matrix $J_{\mathbf{p}}$ as a function of the ratio between the thickness of the cloud of world points along the optical axis of the camera and the distance of the cloud from the camera. The condition number diverges as the thickness vanishes.

Both ambiguities would disappear if the points $\mathbf{X}$ were, say, on *two* frontal planes at different depths from the camera: shifting the camera to the left or forward would change the relative position of the image points in ways that cannot be accounted by changes to the calibration parameters. For instance, if two world points are aligned with the camera in the first view, so that one hides the other, they cannot be aligned also in a view from a different position, so the two images must be different in less trivial ways.

Ambiguities correspond to an infinite condition number. It is also possible to have "near ambiguities," corresponding to high but finite condition numbers. In the presence of image noise or imperfect arithmetic, near ambiguities can be as harmful as true ambiguities.

As a consequence, the calibration object should be designed so as to stay as far from ambiguous situations as possible. A prevalent choice in the past used to be to build the object in the form of two orthogonal planes. Two parallel planes of course would not work, unless the plane closest to the camera can be made to be transparent. Perpendicular planes make it easier to measure and specify the coordinates of points on the object.

A solution that has become more common recently is to build a single, planar calibration object, but to take several images of it, moving the object or the camera between pictures. If the set of relative positions is sufficiently diverse, good conditioning can be achieved at the cost of additional rotation and translation parameter sets, one per image. This is the situation that was formalized earlier in this Section.

3.5.2 Initialization

The calibration problem (6) implies the optimization of a likelihood or residual function that is not known to be convex.[7] A good starting solution $\mathbf{p}_0$ is then needed to initialize a numerical, local optimization algorithm.

This initial solution is typically obtained by solving an approximation to the exact problem. If lens distortion is assumed to be zero,

$$k_2 = k_4 = 0 \quad \text{so that} \quad d(r(\mathbf{x}_c)) = 1 \ \text{ for all points,}$$

and the principal point is tentatively placed in the middle of the image,

$$x_0 \approx \hat{x}_0 = (C+1)/2 \quad \text{and} \quad y_0 \approx \hat{y}_0 = (R+1)/2 \tag{8}$$

for an image with R rows and C columns, then suitable combinations of the calibration parameters appear linearly in the image formation equations. These combinations are found by solving a linear system, and algebraic manipulation finally yields the desired initial values in $\mathbf{p}_0$.

Specifically, the equations (7) can be spelled out as follows in the absence of distortion:

$$\mathbf{k}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \hat{x}_i^{(j)} = -s_x \, \mathbf{i}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \tag{9}$$

$$\mathbf{k}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \hat{y}_i^{(j)} = -s_y \, \mathbf{j}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \tag{10}$$

where

$$\hat{x}_i^{(j)} = x_i^{(j)} - \hat{x}_0 \quad \text{and} \quad \hat{y}_i^{(j)} = y_i^{(j)} - \hat{y}_0$$

and

$$\mathbf{t}_i \quad \text{and} \quad R_i = \begin{bmatrix} \mathbf{i}_i^T \\ \mathbf{j}_i^T \\ \mathbf{k}_i^T \end{bmatrix}$$

are the translation and rotation parameters for the i-th camera. Multiplication of these two equations by $\hat{y}_i^{(j)}$ and $\hat{x}_i^{(j)}$, respectively, yields two equal right-hand sides, which can be equated to each other to obtain the following equation:

$$s_x \hat{y}_i^{(j)} \mathbf{i}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) = s_y \hat{x}_i^{(j)} \mathbf{j}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) . \tag{11}$$

These can be rearranged in vector form by separating knowns from unknowns:

$$\begin{bmatrix} \hat{y}_i^{(j)} \mathbf{X}^{(j)T}, & -\hat{y}_i^{(j)}, & -\hat{x}_i^{(j)} \mathbf{X}^{(j)T}, & \hat{x}_i^{(j)} \end{bmatrix} \mathbf{v}_i = 0$$

where

$$\mathbf{v}_i = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} s_x \mathbf{i}_i \\ s_x \mathbf{t}_i^T \mathbf{i}_i \\ s_y \mathbf{j}_i \\ s_y \mathbf{t}_i^T \mathbf{j}_i \end{bmatrix} .$$

[7] It is harder, but possible, to show that the function is indeed non-convex in some situations.

These equations, one for each of the mn combinations of camera index i and point index j, can be stacked into m homogeneous linear systems, each in the eight entries of $\mathbf{v}_i$ for camera i. These systems are of size n by 8 and can be solved as shown in the Section on the Singular Value Decomposition in the Math Supplement, to give a solution $\hat{\mathbf{v}}_i$ that is proportional to $\mathbf{v}_i$:

$$\hat{\mathbf{v}}_i = a_i \begin{bmatrix} s_x \mathbf{i}_i \\ s_x \mathbf{t}_i^T \mathbf{i}_i \\ s_y \mathbf{j}_i \\ s_y \mathbf{t}_i^T \mathbf{j}_i \end{bmatrix} . \tag{12}$$

For each camera separately, we first find the sign of a_i by noting that

$$\mathbf{a}_j^T(1:4)\mathbf{v}_i(5:8) = a_i \hat{y}_i^{(j)} s_y \mathbf{j}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) = a_i s_y \hat{y}_i^{(j)} \mathbf{X}_c^{(j)}(2)$$

where $\mathbf{a}_j^T$ is the j-th row of the matrix A. The right-hand side must be positive, because $s_y > 0$ and corresponding image and world coordinates have the same sign in the camera reference system (see equations (7) and recall that $\mathbf{X}_c^{(j)}(3) < 0$). This assumes that the approximations (8) are good enough that they do not alter the sign of

$$\hat{y}_i^{(j)} = y_i^{(j)} - \hat{y}_0 \approx y_i^{(j)} - y_0 = s_y y_{ci}^{(j)} .$$

For safety, we can then determine the sign of a_i as follows:

$$s_i = \text{sign}(a_i) = \text{sign} \sum_{j=1}^{n} \mathbf{a}_j^T(1:4)\mathbf{v}_i(5:8) .$$

From the definition of $\mathbf{v}_i$ and from the fact that $\mathbf{i}_i$ and $\mathbf{j}_i$ have unit norm we then obtain immediately estimates of these vectors:

$$\hat{\mathbf{i}}_i = s_i \frac{\hat{\mathbf{v}}_i(1:3)}{\|\hat{\mathbf{v}}_i(1:3)\|} \quad \text{and} \quad \hat{\mathbf{j}}_i = s_i \frac{\hat{\mathbf{v}}_i(5:7)}{\|\hat{\mathbf{v}}_i(5:7)\|}$$

and then

$$\hat{\mathbf{k}}_i = \hat{\mathbf{i}}_i \times \hat{\mathbf{j}}_i .$$

Because of image noise, the matrix

$$\hat{R} = \begin{bmatrix} \hat{\mathbf{i}}_i \\ \hat{\mathbf{j}}_i \\ \hat{\mathbf{k}}_i \end{bmatrix}$$

is not necessarily orthogonal. It can be shown [2] that the orthogonal matrix closest to $\hat{R}$ in the least-squares sense is

$$R = UV^T \quad \text{where} \quad \hat{R} = U\Sigma V^T \text{ is the SVD of } \hat{R}.$$

The definition of $\hat{\mathbf{v}}_i$ also yields the pixel aspect ratio:

$$\alpha = \frac{s_x}{s_y} = \frac{\|\hat{\mathbf{v}}_i(1:3)\|}{\|\hat{\mathbf{v}}_i(5:7)\|} \ .$$

The remaining items to be determined are the translation vectors $\mathbf{t}_i$ and the separate pixel sizes s_x and s_y. Equations (9) and (10) can be rewritten in terms of these quantities and a_i, the unknown scalar in the definition (12) of $\hat{\mathbf{v}}_i$:

$$\begin{aligned} \mathbf{k}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \hat{x}_i^{(j)} &= -\alpha s_y \, \mathbf{i}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \\ \mathbf{k}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \hat{y}_i^{(j)} &= -s_y \, \mathbf{j}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \end{aligned}$$

and from (12)

$$\begin{aligned} \mathbf{k}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \hat{x}_i^{(j)} &= -\alpha s_y \, \mathbf{i}_i^T \mathbf{X}^{(j)} + \alpha s_y \hat{\mathbf{v}}_i(4)/a_i \\ \mathbf{k}_i^T \left(\mathbf{X}^{(j)} - \mathbf{t}_i\right) \hat{y}_i^{(j)} &= -s_y \, \mathbf{j}_i^T \mathbf{X}^{(j)} + s_y \hat{\mathbf{v}}_i(8)/a_i \ . \end{aligned}$$

Rearranging terms yields the following 2×3 system:

$$\begin{bmatrix} \hat{x}_i^{(j)} & -\alpha \mathbf{i}_i^T \mathbf{X}^{(j)} & \alpha \hat{\mathbf{v}}_i(4) \\ \hat{y}_i^{(j)} \mathbf{k}_i^T & -\mathbf{j}_i^T \mathbf{X}^{(j)} & \hat{\mathbf{v}}_i(8) \end{bmatrix} \begin{bmatrix} \tau_i \\ s_y \\ s_y/a_i \end{bmatrix} = \begin{bmatrix} \hat{x}_i^{(j)} \mathbf{k}_i^T \mathbf{X}^{(j)} \\ \hat{y}_i^{(j)} \mathbf{k}_i^T \mathbf{X}^{(j)} \end{bmatrix} \ .$$

where we defined

$$\tau_i = \mathbf{k}_i^T \mathbf{t}_i \ .$$

Stacking n such systems on top of each other yields a $2n \times 3$ linear system for each camera. These systems can be solved with the pseudo-inverse (see Section on the SVD in the Math Supplement). If $\mathbf{u}_i$ is the solution, we have

$$\tau_i = \mathbf{u}_i(1) \quad , \quad s_y = \mathbf{u}_i(2) \quad , \quad a_i = \mathbf{u}_i(2)/\mathbf{u}_i(3) \ .$$

An estimate of the vector $\mathbf{v}_i$ can now be computed by scaling through a_i (equation (12))

$$\mathbf{v}_i = \frac{1}{a_i} \hat{\mathbf{v}}_i \ .$$

From the definitions of $\mathbf{v}_i$ and τ_i, we see that

$$\begin{bmatrix} \mathbf{v}_i(4) \\ \mathbf{v}_i(8) \\ \tau_i \end{bmatrix} = \begin{bmatrix} \mathbf{i}_i^T \\ \mathbf{j}_i^T \\ \mathbf{k}_i^T \end{bmatrix} \mathbf{t}_i = R \mathbf{t}_i \ ,$$

so

$$\mathbf{t}_i = R^T \begin{bmatrix} \mathbf{v}_i(4) \\ \mathbf{v}_i(8) \\ \tau_i \end{bmatrix} \ .$$

Under ideal circumstances, all systems would yield the same vertical pixel size s_y and pixel aspect ratio α. In the presence of noise and rounding errors, a better estimate of s_y and then s_x can be obtained by averaging the estimates $\hat{s}_{yi}$ and $\hat{\alpha}_i$ obtained from each of the cameras:

$$s_y = \frac{1}{m}\sum_{i=1}^{m} \hat{s}_{yi} \quad \text{and} \quad \alpha = \frac{1}{m}\sum_{i=1}^{m} \hat{\alpha}_i \quad \text{and} \quad s_x = \alpha s_y \ .$$

This completes the computation of an initial estimate $\mathbf{p}_0$ of the calibration parameters.

3.5.3 Bundle Adjustment

An off-the-shelf conjugate gradient optimization algorithm can be used to refine the initial value $\mathbf{p}_0$ found in the previous Section. This refinement is not optional, even if lens distortion were negligible. This is because the value $\mathbf{p}_0$ was found by mathematical manipulation of the *geometric* (or reprojection) *error* considered in the original calibration problem (6). As discussed earlier, this manipulation yields an *algebraic error* whose statistics are not Gaussian, and whose minimization introduces bias in the solution in the presence of noise in the image measurements. Assuming convergence to the correct minimum (see the discussion on local convergence in the Section on inverse problems in the Math Supplement), the refinement step yields a consistent (*i.e.*, asymptotically unbiased) estimate $\hat{\mathbf{p}}$ of the parameters.

For a relatively small calibration problem with a single camera and a few images of a few dozen calibration points, conjugate gradients is usually all that is needed for a good quality solution. Two improvements are discussed here that may make a difference when image measurements are poor: schemes that handle outliers, and preconditioning methods.

A third improvement may become important as the size of the problem increases, and concerns the direct exploitation of the sparse nature of the measurement Jacobian. This type of modification, however, is very technical and complex, and is not discussed here. See [6] for a thorough treatment.

Outliers Some of the image coordinates of the calibration points may be entirely wrong, especially when these points are found automatically. Such points, whose error statistics are inconsistent with the Gaussian noise assumption for the given value of standard deviation σ, are called *outliers*. Least squares problems are sensitive to outliers, because they have a breakdown point of zero: a sufficiently wrong outlier can bias the solution by an arbitrary amount. Because of this, it is important to guard against outliers.

Outliers can be handled in different ways:

- They can be eliminated before the image coordinates are used for calibration. This is done by manual inspection, or by verifying the consistency of the coordinates with a model of the calibration target. For instance, if the target is a checkerboard pattern, points in the image must be aligned, except for the effects of lens distortion. One can then fit a quadratic polynomial to points that are supposed to be aligned, and verify the residual through a χ^2 (see the Section on inverse problems in the Math Supplement). If a large residual is found, a quadratic polynomial can be fitted to the aligned points from which one point is removed

in turn (and then reinserted). If one outlier is present, its removal will correspond to a large decrease of the residual. One may have to remove a pair of points at a time to check whether two outliers may be present.

- Outliers can be discounted by modified definitions of the residual. A typical definition replaces the square
$$\epsilon_i 2 = (x_i - f_i(\mathbf{p}))^2$$
of each component of the vector residual $\boldsymbol{\epsilon}$ with a truncated version:
$$e_i^2 = \min(\epsilon_i, \epsilon_{\max})^2$$
where ϵ_{max} is a threshold. This will increase the breakdown point of the parameter estimate.
However, the discontinuity in the first derivative of the thresholded residual may cause difficulties for man optimization methods. A smoother saturation of the residual can be achieved by modeling the error as a Gaussian with standard deviation σ, just as before, plus a uniform distribution of outliers over the entire image, or a smaller radius, resulting in a density η. This leads to a saturated residual
$$e_i^2 = -\log\left(e^{-\frac{1}{2}(\epsilon/\sigma)^2} + \eta\right) .$$

- Outliers can be eliminated during optimization. Ideally, one could use a robust estimator such as the median, rather than the mean (squared). However, the median is not a differentiable operator, so standard optimization methods cannot be used. As an alternative, one can use sampling techniques. In a nutshell, these work by performing calibration with a subset of the available measurements, and picking results that have a low residual or are consistent with results obtained from different sub-samples. A classical procedure of this type is RANSAC. [1] A simpler, method based on clustering of solutions uses measurements in a more symmetric way. [5]

Preconditioning When the condition number of the calibration problem is high, convergence can be slow or otherwise problematic. In these cases, it may be necessary (and it may be generally beneficial) to *precondition* the problem, that is, to reparameterize $\mathbf{p}$ by a new vector
$$\mathbf{q} = Q\mathbf{p}$$
where the square, invertible matrix Q is called the *preconditioner*, and is designed so as to reduce the condition number of the new problem
$$\hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \epsilon^2(\mathbf{q}) \quad \text{where} \quad \epsilon(\mathbf{q}) = \sqrt{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}} = \|\mathbf{x} - \mathbf{f}(\mathbf{X}; Q^{-1}\mathbf{q})\| . \tag{13}$$
Hartley [4] shows a preconditioning method that is based on a re-scaling of the coordinates of the measurement points, for a shape reconstruction problem that is somewhat similar to calibration, but has worse conditioning.

A more general preconditioning technique can be based on the fact that one often knows nominal values for the solution. For instance, in a camera calibration problem, the camera translation vectors $\mathbf{t}_i$ can be measured approximately with a tape measure, the camera's focal distance can be approximated by the focal length read from the camera specifications, the principal point can be placed approximately in the middle of the image, and so forth. If such a nominal parameter vector $\mathbf{p}_\nu$ can be determined, one can compute the Jacobian of $\mathbf{f}$ at this point,

$$J_\nu = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{p}^T}\right|_{\mathbf{p}=\mathbf{p}_\nu} .$$

If

$$J_\nu = U_\nu \Sigma_\nu V_\nu^T$$

is the SVD of J_ν, then one can define the preconditioner to be

$$Q = \Sigma_\nu(1:k,:) V_\nu^T .$$

Here, the Matlab-style notation $\Sigma_\nu(1:k,:)$ means "select the first k rows of Σ_ν," so Q is a $k \times k$ matrix. With this choice, the Jacobian of the preconditioned problem (13) is

$$J_{\mathbf{q}} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}^T} = \frac{\partial \mathbf{f}}{\partial \mathbf{p}^T}\frac{\partial \mathbf{p}}{\partial \mathbf{q}^T} = J\,Q^{-1} = J\,V_\nu \Sigma_\nu(1:k,:)^{-1} .$$

At the nominal point $\mathbf{p}_\nu$, the new Jacobian $J_{\mathbf{q}}$ has a condition number of one:

$$(J_{\mathbf{q}})_\nu = J_\nu\, V_\nu \Sigma_\nu(1:k,:)^{-1} = U_\nu \Sigma_\nu V_\nu^T V_\nu \Sigma_\nu(1:k,:)^{-1} = U_\nu .$$

This condition number is then hopefully small also in a sufficiently wide neighborhood of the nominal point. Once an estimate $\hat{\mathbf{q}}$ of the transformed parameter $\mathbf{q}$ has been found, the desired estimate $\hat{\mathbf{p}}$ of $\mathbf{p}$ can be computed by solving the (poorly conditioned, but linear and exact) system

$$Q\mathbf{p} = \hat{\mathbf{q}} . \tag{14}$$

Compromises between the condition number of the numerical minimization problem (13) and that of the linear system (14) can be found by replacing small singular values in $\Sigma_\nu(1:k,:)$ with a fixed, larger value $\sigma_{\min}$.

3.6 Calibration of a Stereo Pair

The calibration of a stereo pair involves finding the transformation $R, \mathbf{t}$ between the two cameras in the pair, in addition to the intrinsic parameters of each camera. The procedure is conceptually straightforward:

- Take images of the calibration object simultaneously with the two cameras in the pair, arranged in their final configuration. This yields two sets of image measurements $\mathbf{x}_1$ and $\mathbf{x}_2$.

- Calibrate the first camera in the pair using measurements $\mathbf{x}_1$ to obtain its intrinsic parameters $s_{x1}, s_{y1}, x_{01}, y_{01}, k_{21}, k_{41}$ and extrinsic parameters $R_1, \mathbf{t}_1$ (relative to the calibration object).
- Calibrate the second camera in the pair using measurements $\mathbf{x}_2$ to obtain its intrinsic parameters $s_{x2}, s_{y2}, x_{02}, y_{02}, k_{22}, k_{42}$ and extrinsic parameters $R_2, \mathbf{t}_2$ (relative to the calibration object).
- Compute the transformation between the two cameras by noticing that a point whose coordinates are $\mathbf{X}$ in the world (calibration target), $\mathbf{X}_1$ in the reference frame of camera 1, and $\mathbf{X}_2$ in the reference frame of camera 2 satisfy the following equations:

  $$\mathbf{X}_1 = R_1(\mathbf{X} - \mathbf{t}_1) \quad \text{and} \quad \mathbf{X}_2 = R_2(\mathbf{X} - \mathbf{t}_2) \ .$$

  The second equation yields

  $$\mathbf{X} = R_2^T \mathbf{X}_2 + \mathbf{t}_2$$

  and by replacing this into the first equation we obtain

  $$\mathbf{X}_1 = R_1(R_2^T \mathbf{X}_2 + \mathbf{t}_2 - \mathbf{t}_1) = R_1 R_2^T \mathbf{X}_2 + R_1(\mathbf{t}_2 - \mathbf{t}_1)$$

  and

  $$R = R_1 R_2^T \quad \text{and} \quad \mathbf{t} = R_1(\mathbf{t}_2 - \mathbf{t}_1)$$

  transforms coordinates in the second reference frame into coordinates in the first.

3.7 Removing Lens Distortion

Calibration of each camera yields the intrinsic parameters. These can be used to correct images taken with that camera in order to remove the effects of lens distortion. This will make projections of straight lines in the world straight lines in the image, and will make images equivalent geometrically to the images that a pinhole camera would take. This Section shows how to perform this correction.

Given a point $\mathbf{x}$ in the actual image, let $\mathbf{x}_u$ be the corresponding point that would be taken if the camera had zero distortion, but equal focal distance and pixel scaling factors. We leave the principal point $\mathbf{x}_{u0}$ of the ideal image unspecified for now, as it will give use useful degrees of freedom later. The removal of lens distortion computes a new image that moves each point $\mathbf{x}$ to its undistorted location $\mathbf{x}_u$. We saw in Section 3.2 that the image coordinates $\mathbf{x}$ relate to the camera coordinates in $\mathbf{x}_c$ through the following equations:

$$\mathbf{x} = S\mathbf{x}_c d(r(\mathbf{x}_c)) \ + \ \mathbf{x}_0 \tag{15}$$

where

$$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \quad \text{and} \quad \mathbf{x}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

are the transformation to pixel coordinates and the image principal point,

$$d(r) = 1 + k_2 r^2 + k_4 r^4$$

is the distortion function, and

$$r^2(\mathbf{x}_c) = x_c^2 + y_c^2$$

is the squared radial distance of an image point in camera coordinates from the principal point.

Depending on the values of the distortion coefficients, the undistorted image may be greater than the distorted one or smaller, and is not a rectangle (unless distortion is zero), but takes the form of a barrel or pincushion, as discussed in the Section on image formation.

It is convenient to scale the undistorted image to be equal in size to the distorted one, and to represent the greatest rectangle that can be inscribed in the corrected image. Figure 3.4 illustrates the reasoning for the case of pincushion distortion.

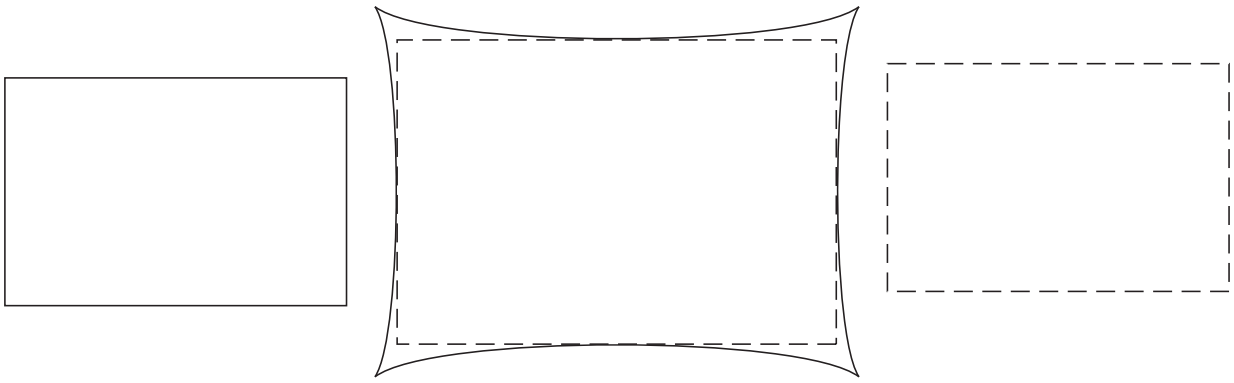


Figure 3.4: The boundary of the distorted image is a rectangle (left). If it is affected by pincushion distortion, removal of distortion scales and distorts the bounding rectangle into a pincushion-like shape (center, solid). The useful part of the image is the largest inscribed rectangle (center, dashed) in this shape, which may be larger or smaller than the original image rectangle, depending on the values of the distortion coefficients. The pixel scaling factors of the undistorted image can be chosen so that the resulting undistorted image is bound by a rectangle (right) that is equal in size and aspect ratio to that of the original, distorted image.

The greatest inscribed rectangle is tangent to the graphs of the functions that represent the top and bottom boundaries of the pincushion shape. These boundaries can be found by solving the system of equations (15) numerically for $\mathbf{x}_c$ for

$$\mathbf{x} = \begin{bmatrix} x \\ \tilde{y} \end{bmatrix}$$

where $\tilde{y}$ is fixed to 1 for the top boundary and to R for the bottom boundary. The value of x can be varied by interval search starting with the interval $(1, C)$ to find the smallest (for the top) or largest (for the bottom) value of y_c. A similar procedure, with roles of x and y axes exchanged, finds the left and right sides of the greatest inscribed rectangle.

We can then define the scaling factors of the undistorted image as follows:

$$S_u = \begin{bmatrix} s_{ux} & 0 \\ 0 & s_{uy} \end{bmatrix} = \begin{bmatrix} \frac{R-1}{t-b} & 0 \\ 0 & \frac{C-1}{r-l} \end{bmatrix} .$$

The principal point of the undistorted images is chosen so as to correspond to coordinates $x_c = 0$ and $y_c = 0$ as follows:

$$\mathbf{x}_{u0} = -S_u \begin{bmatrix} l \\ b \end{bmatrix}$$

(the entries of $\mathbf{x}_u$ are positive because l and b are negative).

This makes the undistorted image equal in size to the distorted one, and places the principal points at $\mathbf{x}_{u0}$.

Once the undistorted rectangle is determined, it can be filled with the appropriate pixel values by looping over its rows and columns. For each output (*i.e.*, undistorted) pixel position $\mathbf{x}_u = (x_u, y_u$, the following computation yields the corresponding pixel position in the distorted image:

$$\begin{aligned} \mathbf{x}_c &= S_u^{-1}(\mathbf{x}_u - \mathbf{x}_{u0}) \\ \mathbf{x} &= S\mathbf{x}_c d(r(\mathbf{x}_c)) + \mathbf{x}_0 \end{aligned}$$

(the second equation is equation (15)). The value of the input (distorted) image at $\mathbf{x}$ can be determined by bilinear interpolation (see Section on image processing) and written into pixel $\mathbf{x}_u$ of the undistorted image.

References

[1] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Proceedings of the ARPA IUS Workshop*, pages 71–88, 1980.

[2] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Md, 1996. 3rd edition.

[3] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[4] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–93, June 1997.

[5] C. Tomasi, J. Zhang, and G. Golub. A resampling method for computer vision. In *Proceedings of the Ninth International Symposium on Robotics Research, ISRR '99*, pages 89–96, Snowbird, CO, October 1999. Invited paper.

[6] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.