A Simple Camera Calibration Method

ALDO CUMANI
Computer Vision Lab, Systems Engineering Department
Istituto Elettrotecnico Nazionale Galileo Ferraris
Str. delle Cacce, 91
ITALY

Abstract: This work discusses a calibration method based on a simple and inexpensive setup. With respect to a similar method recently proposed by Zhang, our approach allows a substantial decoupling of the distortion model from the rest of the camera model, which means that distortion estimation can make a better use of the available image data. Preliminary experimental tests show promising results.

Key-Words: Camera calibration, Lens distortion, Absolute conic

1 Introduction

Current camera calibration techniques roughly divide into methods based on known high-precision (and high cost) laboratory artefacts [3, 10], and methods which try to directly exploit geometric scene constraints via self-calibration [1, 8, 9].

The method proposed in this paper lies somewhere in between the two categories stated above, since it does make use of a calibration artefact, that can however be realised in an inexpensive way by printing a pattern of squares with a laser printer and sticking the printed sheet to a reasonably planar surface (e.g. a cheap frame mount). Our method is based on well known properties of the Image of the Absolute Conic (IAC) [7, 8], and has some similarity with one recently proposed by Zhang [12]. The main difference lies in using a different distortion model, which allows to almost decouple distortion estimation from pinhole estimation, while making a more efficient use of image data.

2 Pinhole calibration

The image formation process, as described by the standard pin-hole model, is sketched in Fig. 1. If \( \mathbf{X} = [X_1, X_2, X_3, X_4]^T \) is a world point and \( \mathbf{x} = [x_1, x_2, x_3]^T \) its image, then

\[
\mathbf{x} = \mathbf{P} \mathbf{X} \quad \text{with} \quad \mathbf{P} = \mathbf{A}[\mathbf{R} \mid \mathbf{t}] \tag{1}
\]

where the factoring of the projection matrix \( \mathbf{P} \) into a rototranslation \( \mathbf{R}, \mathbf{t} \) and an intrinsic matrix

\[
\mathbf{A} = \begin{bmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}
\]

only holds for Euclidean world/image plane frames. Assuming \( \mathbf{A} = 1 \) yields normalised image coordinates, while a non-unit \( \mathbf{A} \) can model the physical image scaling on the camera sensor, as well as the subsequent sampling. Note that Eq. (1) is only valid in the absence of lens distortion (Sec. 3).

We assume all observed 3D points on a same plane \( \pi \), which without loss of generality can be taken as the \( z \) plane \( X_3 = 0 \). Therefore \( \mathbf{x}_p = [X_1, X_2, X_4]^T \) are projective (actually, Euclidean) coordinates on that plane, and Eq. (1) becomes

\[
\mathbf{x} = [\mathbf{p}_1 \mathbf{p}_2 \mathbf{p}_4] \mathbf{x}_p = \mathbf{H} \mathbf{x}_p \tag{3}
\]

where the homography \( \mathbf{H} = [\mathbf{p}_1 \mathbf{p}_2 \mathbf{p}_4] = [\mathbf{h}_1 \mathbf{h}_2 \mathbf{h}_3] \) between the object and image planes can be determined from at least four correspondences between known plane points and their images.

In projective 3-space, the absolute conic \( \Omega_\infty \) [7] is a conic of purely imaginary points lying on the plane at infinity \( \pi_\infty \). Under Eq. (1), \( \Omega_\infty \) projects into the Image of the Absolute Conic (IAC) \( \omega \) which...
is itself a conic on the image plane. It is well known that \( \omega \) is linked to the intrinsic matrix \( A \) by
\[
\omega^{-1} = AA^T
\]  
(4)
For any \( \pi, \Omega \infty \) passes through its circular points, so \( \omega \) passes through their images \( h_1 \pm ih_2 \). For given \( H \) this yields the pair of linear constraints on \( \omega \)
\[
h_1^T \omega h_1 - h_2^T \omega h_2 = 0
\]  
(5)
\[
h_1^T \omega h_2 = 0
\]  
(6)
Since \( \omega \) has 5 degrees of freedom, if we have \( n \geq 3 \) independent homographies, derived from different images of the same planar object, we can stack up the above pairs of constraints obtaining an overdetermined \( 2n \times 6 \) homogeneous system
\[
D \omega = 0
\]  
(7)
where \( \omega = [\omega_{11}, \omega_{12}, \omega_{22}, \omega_{13}, \omega_{23}, \omega_{33}]^T \); this system can be solved e.g. by eigendecomposition of \( D^TD \), finally yielding \( A \) via Eq. (4).

The pinhole calibration procedure can therefore be sketched as follows:
- grab \( n \geq 3 \) images of a known planar pattern from different positions and orientations;
- for each image, estimate \( H \) from Eq. (3);
- build Eq. (7) using the first two columns of \( H \) from each image, and solve it;
- get \( A \) from (4) by Cholesky factorisation.

### 3 Distortion calibration

The previous discussion ignores geometric lens distortion, which is seldom negligible, especially with wide-angle lenses. Photogrammetric experience [2] shows that the most part of lens distortion can be modelled as a nonlinear radial scaling of the image around a distortion center \( u_d = [u_d, v_d]^T \). Such a scaling can in turn be approximated by a truncated Taylor series:
\[
u' - u_d = (1 + k_1 \rho^2 + k_2 \rho^4 + ...) (u - u_d)
\]  
(8)
where \( u = [u, v]^T \) are the image coordinates of the observed feature, \( u' = [u', v']^T \) the so-called undistorted coordinates (i.e. the ones that would result from an ideal distortionless lens), and \( \rho^2 = (u-u_d)^2 + (v-v_d)^2 \). Note that some authors reverse the role of distorted and undistorted coordinates in (8); while the two models are both series approximations, Eq. (8) is better suited to computations in actual usage, as we typically observe distorted coordinates \( u \) and want to get the undistorted ones \( u' \).

However, Eq. (8) implicitly assumes equal scaling in the \( u \) and \( v \) directions, so it only applies to normalised coordinates; it is generally not valid if using frame grabber coordinates. To correct for this, the distortion model we actually use is
\[
u' - u_d = (1 + k_1 f_u^2) (u - u_d)
\]  
(9)
where \( p_r \) is the pixel ratio \( f_u / f_v \). In principle, we should also take into account the skew factor \( s \), but since the latter is usually quite small with respect to the focal lengths, its contribution can be safely neglected.

For a given camera setup, the distortion parameters \( k_1, u_d, v_d \) (and possibly \( p_r \)) can be estimated e.g. by grabbing an image with several straight lines, and fitting the model (9) by nonlinear optimisation. In summary, the distortion calibration procedure can be sketched as follows:
- grab an image with several straight lines, and extract these lines as sets of image points;
- for given \( k_1, u_d, v_d \), transform the points according to Eq. (9), fit a least squares line to each set of transformed points, and compute the sum of fitting residuals;
- estimate \( k_1, u_d, v_d \) by finding a minimum of the so defined fitting error.

### 4 Implementation details

A calibration target that meets the requirements of the previous sections may consist of an array of \( N \times N \) identical, equispaced black squares. Such a pattern can be easily realized by printing it on paper with a laser printer and sticking the paper on a planar surface, as proposed in [12]. Note that, for what concerns the internal camera parameters, the actual pattern size is irrelevant. The plane coordinates of the square corners can then be assumed integers in the range \( 0.2N - 1 \), though non-unit size or spacing ratios can be easily incorporated into the algorithm.

#### 4.1 Feature extraction

The first step consists in extracting from the image the contours of the squares. This is done by a second-order differential algorithm [6, 4] yielding contour lines as lists of image points to sub-pixel precision. Point lists are ordered so that the lighter (white) side is consistently on the left of the contour line, which eases the identification of square
contours, as closed or almost closed contours surrounding a dark region.

The contours are then segmented by estimating local curvature at each point, so identifying square sides as runs of low-curvature points. The intersections of straight lines fitted to the latter yield a preliminary estimate of the square corners (see Fig. 2).

The convex hull of square corners allows to identify the four corners of the entire pattern as the most “spiky” vertices on the hull. The homography mapping those four corners to the points \( (0,0), (0,2N-1), (2N-1,2N-1), (2N-1,0) \) is then computed; assuming that the distortion is not too severe, this allows to index the imaged square corners by transforming their image coordinates via Eq. (3) and rounding the result.

4.2 Distortion estimation

Labelling corners also identifies a set \( G \) of \( 4N \) groups \( g_i, i = 1..4N \), of contour line segments, each group containing images of aligned target points. The distortion estimation then proceeds as follows. First, a function \( J \) of the distortion parameters \( k_1, u_d, v_d \) is defined as the sum, over the \( 4N \) groups, of the squared distances of points in each group from the corresponding least-squares line, after transforming point coordinates by Eq. (9). For-
back at each iteration the $p_r$ value given by the ratio $f_u/f_v$ estimated at the second step. This iterative process has been found to converge to a stable solution within 3 to 5 iterations.

5 Experimental results

In this section we provide some results from applying the above method to real cameras. Since our method bears several similarities to the one proposed by Zhang in [12], we also compare our results to those of Zhang’s method, using the implementation provided by the same author on the Internet.

5.1 Test 1

In this test, the camera (a Sony XC55 with a 6 mm lens) was mounted on the end effector of a lightweight 6 DOF industrial manipulator, a Samsung AW1 recently installed at our Laboratory. Non-interlaced images of 640×480 8-bit pixels were acquired via a Matrox Meteor II board.

Figure 3: Images used for Test 1.

Fig. 3 shows five images of the target used for calibration. Table I shows the results of both our method and Zhang’s, when applied to the 16 possible combinations of at least three images from this set (only averages and r.m.s. deviations reported, for reasons of space). This table shows quite similar results for the pinhole parameters $f_u, f_v, s, u_0$ and $v_0$. For what concerns distortion, the comparison is less easy, due to the use of different models. However, it can be observed that our method yields a $k_1$ value stable to about 1%, while the corresponding $k_1$ for Zhang’s method has a variability of about 5%. This is likely due to the fact that our method uses all the available contour points in each image, for estimating the distortion model, while the former only uses the $4N^2$ feature points (the square corners). Also, the value of $k_2$ exhibits a rather large variability, which could be probably explained by overparametrisation (i.e., $k_2$ is not actually needed to model the distortion, and the estimated values simply accumulate the effects of noise).

Anyway, the only decisive test about the goodness of a calibration consists in using it to perform actual measurements. To this extent, two images of a white-painted wooden box were taken from different positions, as shown in Fig. 4. These two images were processed by extracting of contour lines, transforming contour points to normalised image coordinates using one of the available sets of calibration parameters and segmenting into rectilinear strokes, and finally finding box corners as intersections of concurrent segments.

Figure 4: Two images of a box, used for checking the accuracy of the results from Test 1.

This procedure yielded seven pairs of corresponding points in the two images, from which the fundamental matrix $F$ for the image pair was computed by the well-known 7-point method [11]. Since in our case $F = E$, the essential matrix (because we used calibrated image coordinates), from $E = [\mathbf{t}\mathbf{s}]R$ the rotation $R$ of the camera between the two grabbing positions was finally determined, allowing then to compute the angle $\alpha_{eg}$ between the camera’s optical axes at the two positions. The results are reported in Table II, columns 2 and 3.

It would be tempting to compare the latter to the angle between the robot’s end-effector Z-axis at the two positions, for which a ground truth value of $25.798^\circ$ was known. This comparison is however improper, since there is no guarantee that the camera’s optic axis be aligned with the robot Z axis; indeed, the latter was estimated independently by determining the focus of expansion of the image when the robot wrist is moved, without rotation, along its Z axis, and its position was found to be at about $(352, 209)$, i.e. significantly separated from the optic center. From the latter, for a given set of calibration parameters the direction of the robot’s Z axis in
This method

<table>
<thead>
<tr>
<th></th>
<th>$u_d$</th>
<th>$v_d$</th>
<th>$k_1$</th>
<th>$f_u$</th>
<th>$s$</th>
<th>$f_v$</th>
<th>$u_0$</th>
<th>$v_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>337.67</td>
<td>266.51</td>
<td>3.09e-7</td>
<td>819.92</td>
<td>0.24</td>
<td>823.06</td>
<td>333.46</td>
<td>263.67</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>5.38</td>
<td>4.91</td>
<td>3.36e-9</td>
<td>2.47</td>
<td>1.03</td>
<td>2.15</td>
<td>4.50</td>
<td>3.15</td>
</tr>
</tbody>
</table>

Zhang’s method

<table>
<thead>
<tr>
<th></th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$f_u$</th>
<th>$s$</th>
<th>$f_v$</th>
<th>$u_0$</th>
<th>$v_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>-0.163995</td>
<td>-0.062648</td>
<td>817.88</td>
<td>-0.03</td>
<td>820.26</td>
<td>331.61</td>
<td>261.03</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.008791</td>
<td>0.082329</td>
<td>2.09</td>
<td>0.26</td>
<td>2.30</td>
<td>1.16</td>
<td>2.95</td>
</tr>
</tbody>
</table>

Table I: Results of Test 1.

the camera reference can be computed at both positions, yielding the values of $\alpha^Z$ shown in columns 4 and 5 of Table II. It can be noticed from these figures that our method, in spite of a slightly larger variability, yields a quite unbiased estimate of the above angle.

5.2 Test 2

This section reports some results from the application of our method to a consumer-grade imaging system, namely an old Panasonic NVG1 VHS-C camcorder, connected to a Matrox Marvel G200 grabber installed on a Pentium II/400. The camera was mounted on a tripod, in order to allow usage of the full (704 x 576) image resolution - with a handheld camera, the unavoidable motion jitter would have forced to use only one field of each frame, so halving the vertical resolution.

Figure 5: Images used for Test 2.

Fig. 5 shows the images used for calibration, and Tables III and IV report the results of applying both our method and Zhang’s to all combinations of at least 3 images from the available set.

Although the size of the test sample is too small to draw definite conclusions, in this case our method seems to perform slightly better than Zhang’s, which is likely due to the fact that the used calibration target has much less feature points than the one of Test 1 (64 squares vs. 256). In particular, our method yields a quite stable estimate of the pixel ratio.

6 Concluding remarks

The availability of a simple, cheap, and reasonably accurate calibration method is highly desirable for any Computer Vision application aiming at making quantitative measurements on the observed scene. The method presented in this work goes in that direction, by allowing a good calibration with a simple and inexpensive setup. Moreover, the procedure presented in this work has been implemented as a complete, easy to use package - from digitised images to calibration parameters - which is surely a not marginal issue for the end user.

Acknowledgments - The author wishes to thank his colleagues of the IEN Computer Vision Laboratory for their continuing support. This work was partly supported by the Italian Space Agency (ASI) under grant no. I/R/46/00.

References:


---

### Table II: Comparison of measurements for Test 1.

<table>
<thead>
<tr>
<th>images</th>
<th>$u_d$</th>
<th>$v_d$</th>
<th>$k_1$</th>
<th>$f_u$</th>
<th>$s$</th>
<th>$f_v$</th>
<th>$u_0$</th>
<th>$v_0$</th>
<th>$p_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>402.15</td>
<td>352.75</td>
<td>2.82e-7</td>
<td>923.90</td>
<td>1005.66</td>
<td>-2.26</td>
<td>394.31</td>
<td>325.50</td>
<td>0.9187</td>
</tr>
<tr>
<td>1234</td>
<td>395.23</td>
<td>350.50</td>
<td>2.96e-7</td>
<td>923.22</td>
<td>1005.01</td>
<td>-0.63</td>
<td>391.72</td>
<td>324.06</td>
<td>0.9186</td>
</tr>
<tr>
<td>124</td>
<td>397.12</td>
<td>351.81</td>
<td>2.92e-7</td>
<td>925.98</td>
<td>1008.63</td>
<td>0.13</td>
<td>390.46</td>
<td>325.19</td>
<td>0.9181</td>
</tr>
<tr>
<td>134</td>
<td>395.34</td>
<td>352.25</td>
<td>2.94e-7</td>
<td>921.74</td>
<td>1003.98</td>
<td>-0.00</td>
<td>390.88</td>
<td>323.52</td>
<td>0.9181</td>
</tr>
<tr>
<td>234</td>
<td>389.19</td>
<td>342.58</td>
<td>3.25e-7</td>
<td>920.29</td>
<td>1001.66</td>
<td>0.28</td>
<td>390.37</td>
<td>322.06</td>
<td>0.9188</td>
</tr>
<tr>
<td>mean</td>
<td>395.80</td>
<td>349.98</td>
<td>2.98e-7</td>
<td>923.02</td>
<td>1004.99</td>
<td>-0.50</td>
<td>391.55</td>
<td>324.07</td>
<td>0.9184</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>4.16</td>
<td>3.77</td>
<td>1.43e-8</td>
<td>1.93</td>
<td>2.27</td>
<td>0.93</td>
<td>1.46</td>
<td>1.23</td>
<td>0.0003</td>
</tr>
</tbody>
</table>

### Table III: Results of Test 2.

<table>
<thead>
<tr>
<th>images</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$f_u$</th>
<th>$s$</th>
<th>$f_v$</th>
<th>$u_0$</th>
<th>$v_0$</th>
<th>$p_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>-0.227129</td>
<td>-0.081330</td>
<td>917.68</td>
<td>-2.92</td>
<td>994.66</td>
<td>378.32</td>
<td>310.00</td>
<td>0.9226</td>
</tr>
<tr>
<td>1234</td>
<td>-0.231462</td>
<td>-0.139634</td>
<td>922.36</td>
<td>0.26</td>
<td>1001.57</td>
<td>380.52</td>
<td>307.53</td>
<td>0.9209</td>
</tr>
<tr>
<td>124</td>
<td>-0.211595</td>
<td>-0.256400</td>
<td>927.20</td>
<td>1.62</td>
<td>1008.20</td>
<td>374.67</td>
<td>312.61</td>
<td>0.9197</td>
</tr>
<tr>
<td>134</td>
<td>-0.254935</td>
<td>-0.061727</td>
<td>931.79</td>
<td>1.50</td>
<td>1009.35</td>
<td>379.08</td>
<td>323.48</td>
<td>0.9232</td>
</tr>
<tr>
<td>234</td>
<td>-0.252753</td>
<td>-0.089722</td>
<td>911.20</td>
<td>0.39</td>
<td>991.25</td>
<td>384.48</td>
<td>302.10</td>
<td>0.9192</td>
</tr>
<tr>
<td>mean</td>
<td>-0.235575</td>
<td>-0.125763</td>
<td>922.05</td>
<td>0.17</td>
<td>1001.00</td>
<td>379.41</td>
<td>311.15</td>
<td>0.9211</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.016329</td>
<td>0.070189</td>
<td>7.19</td>
<td>1.64</td>
<td>7.17</td>
<td>3.18</td>
<td>7.08</td>
<td>0.0015</td>
</tr>
</tbody>
</table>

### Table IV: Results of Test 2, using Zhang’s method.