Correcting Radial Lens Distortion Using Image and Point Correspondences

Leonardo Romero and Felix Calderon

UMSNH, Morelia, Mich., 52000, Mexico
{lromero,calderon}@zeus.umich.mx

Abstract. This paper describes two new methods for lens distortion calibration using image and point correspondences. Images (or feature points) captured by a camera are undistorted and projected into a calibration pattern image. Both methods apply the Gauss–Newton–Levenberg–Marquardt non–linear optimization technique to match, in one case, the camera image and the pattern image, and in the other case, selected point correspondences from the camera image to the pattern image. One way to automatically find good point correspondences is presented. Experimental results compare the performance of both methods and show better results using point to point correspondences.

1 Introduction

Most algorithms in 3-D Computer Vision rely on the pinhole camera model because of its simplicity, whereas video optics, especially wide–angle lens, generate a lot of non–linear distortion. In some applications, for instance in stereo vision systems, this distortion can be critical.

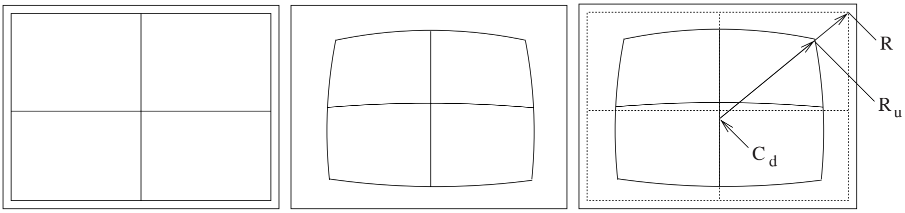
Camera calibration consists of finding the mapping between the 3-D space and the camera plane. This mapping can be separated in two different transformations: first, the relation between the origin of 3–D space and the camera coordinate system, which forms the external calibration parameters (3–D rotation and translation), and second the mapping between 3–D points in space and 2–D points on the camera plane in the camera coordinate system, which form the internal calibration parameters [1].

This paper introduces two new methods to find the internal calibration parameters of a camera, specifically those parameters related with the radial distortion due to wide–angle lens.

The first method works with two images, one from the camera and one from a calibration pattern (without distortion) and it is based on a non–linear optimization method to match both images. The search is guided by analytical derivatives with respect to a set of calibration parameters. The image from the calibration pattern can be a scanned image, an image taken by a high quality digital camera (without lens distortion), or even the binary image of the pattern (which printed becomes the pattern).

The second method works with point correspondences from the camera image to the pattern image, and apply a similar procedure to the first method to

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 204–211, 2003.



(a) Pattern in front of the camera

(b) Image captured by the camera

(c) Both images

Fig. 1. The distortion process due to lens

find the best set of parameters. The set of point correspondences are computed automatically, taking advantage of results of the first method.

The rest of this paper is organized as follows. Sections 2 and 3 describe the distortion and projective model that we are using. Sections 4 and 5 present the methods to match images and to match pairs of points, respectively. Experimental results are shown in Section 6. A brief comparison of previous calibration methods with our methods are in section 7. Finally, some conclusions are given in Section 8.

2 The Distortion Model

The distortion process is illustrated in Figure 1. Figure 1 (b) shows an image taken from the camera when the pattern shown in Figure 1 (a) is in front of the camera. Note the effect of lens, the image is distorted, specially in those parts far way from the center of the image. Figure 1 (c) shows the radial distortion in detail, supposing that the center of distortion is the point c_d with coordinates (c_x, c_y). Undistorted pixel at position R with coordinates (x, y) points to pixel R_u with coordinates (x_u, y_u).

Let I_d be the distorted image captured by the camera and I_u the undistorted image associated to I_d. The relationship between both images is modeled by:

$$I_u(\theta^d, x, y) = I_d(x_u(\theta^d, x, y), y_u(\theta^d, x, y)), \quad \theta^d = (k_1, k_2, c_x, c_y, s_x) \tag{1}$$

$$x_u = c_x + \frac{x-c_x}{s_x}(1 + k_1 r^2 + k_2 r^4),\; y_u = c_y + (y - c_y)(1 + k_1 r^2 + k_2 r^4)$$
$$r = \sqrt{(\tfrac{x-c_x}{s_x})^2 + (y - c_y)}$$

Where θ^d are internal calibration parameters of the camera. Parameters k_1 and k_2 define how strong is the radial distortion with distortion center (c_x, c_y). Parameter s_x is the aspect ratio of pixels ($s_x = 1$ means square pixels).

3 The Projection Model

Figure 1 shows and ideal case, where the plane of the pattern is parallel to the camera plane and center of the pattern coincides with the optical axis of the camera. Using homogeneous coordinates, the class of 2–D planar projective transformations between the camera plane and the pattern plane is given by [5] $[x', y', w']^t = M[x, y, w]^t$, where matrix M has eight independent parameters,

$$M = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix}$$

Plane and homogeneous coordinates are related by $(x_{p1} = x/w, y_{p1} = y/w)$ for one plane and $(x_{p2} = x'/w', y_{p2} = y'/w')$ for the other plane. Let I_p be the projection from the camera plane (with the undistorted image I_u), to the pattern plane. The new image is given by:

$$\begin{array}{ll} I_p(\theta^p, x, y) = I_d(x_p(\theta^p, x_u, y_u), y_p(\theta^p, x_u, y_u)) & \\ \theta^p = (m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7) & \\ x_p = \frac{m_0 x_u + m_1 y_u + m_2}{m_6 x_u + m_7 y_u + 1}, & y_p = \frac{m_3 x_u + m_4 y_u + m_5}{m_6 x_u + m_7 y_u + 1} \end{array} \tag{2}$$

4 The Image Registration Method

The goal is to find a set of parameters θ^d and θ^p so the projected image, I_p, match the image, I_r, of the calibration pattern put in front of the camera.

We formulate the goal of internal calibration as to find a set of parameters $\theta = (m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, k_1, k_2, c_x, c_y, s_x)$ such the sum, E_t, of square differences between pixels of I_p and I_r, is a minimum.

$$\theta = argmin\, E_t(I_p(\theta), I_r) = argmin \sum_{\forall (x,y) \in I_r} (I_p(\theta, x, y) - I_r(x, y))^2 \tag{3}$$

4.1 Non-linear Optimization

The Gauss-Newton-Levenberg-Marquard method (GNLM) [3] is a non–linear iterative technique specifically designated for minimizing functions which has the form of sum of square functions, like E_t. At each iteration, the increment of parameters, $\delta\theta$, is computed solving the following linear matrix equation:

$$\begin{array}{l} A\,\delta\theta = B \\ A = [J^t J + \lambda I],\, B = -J^t e \end{array} \tag{4}$$

If there is p pixels in images and q parameters in θ, A is a matrix of dimension $q \times q$. Matrix J, of dimension $p \times q$, is the Jacobian of e. I is the identity matrix, e is the vector of all differences of pixels between both images and has dimension $q \times 1$, so B has dimension $q \times 1$. λ is a parameter which is allowed to vary at each iteration. After a little algebra, the elements of A and B are computed using the following formulas,

$$a_{i,j} = \sum_{k=1}^{p} \frac{\partial e_k}{\partial \theta_i} \frac{\partial e_k}{\partial \theta_j}, \quad b_i = -\sum_{k=1}^{p} \frac{\partial e_k}{\partial \theta_i} e_k, \quad e_k = I_p(\theta, x_k, y_k) - I_r(x_k, y_k) \tag{5}$$

Applying the chain rule to compute the partial derivatives and considering eq. 2, we get,

$$\frac{\partial e_k}{\partial \theta_i} = \frac{\partial I_p(\theta, x = x_k, y = y_k)}{\partial \theta_i} = \frac{\partial I_p(x_p, y_p)}{\partial x_p} \frac{\partial x_p}{\partial \theta_i} + \frac{\partial I_p(x_p, y_p)}{\partial y_p} \frac{\partial y_p}{\partial \theta_i} \tag{6}$$

In order to simplify the notation, we use x_p instead of x_{pk} and y_p instead of y_{pk}. $\frac{\partial I_d(x_p,y_p)}{\partial x_p}$ and $\frac{\partial I_d(x_p,y_p)}{\partial y_p}$ are the partial derivatives of the image I_p in the x and y directions. $\frac{\partial x_p}{\partial \theta_i}$ and $\frac{\partial y_p}{\partial \theta_i}$ for $(\theta_0, \cdots \theta_7)$ can be derived from eq. 2,

$$\begin{array}{ll}
\frac{\partial x_p}{\partial m_0} = \frac{x_u}{D} & \frac{\partial y_p}{\partial m_0} = 0 \\
\frac{\partial x_p}{\partial m_1} = \frac{y_u}{D} & \frac{\partial y_p}{\partial m_1} = 0 \\
\frac{\partial x_p}{\partial m_2} = \frac{1}{D} & \frac{\partial y_p}{\partial m_2} = 0 \\
\frac{\partial x_p}{\partial m_3} = 0 & \frac{\partial y_p}{\partial m_3} = \frac{x_u}{D} \\
\frac{\partial x_p}{\partial m_4} = 0 & \frac{\partial y_p}{\partial m_4} = \frac{y_u}{D} \\
\frac{\partial x_p}{\partial m_5} = 0 & \frac{\partial y_p}{\partial m_5} = \frac{1}{D} \\
\frac{\partial x_p}{\partial m_6} = \frac{-x_u x_p}{D} & \frac{\partial y_p}{\partial m_6} = \frac{-x_u y_p}{D} \\
\frac{\partial x_p}{\partial m_7} = \frac{-y_u x_p}{D} & \frac{\partial y_p}{\partial m_7} = \frac{-y_u y_p}{D}
\end{array} \tag{7}$$

Where $D = m_6 x_u + m_7 y_u + 1$. Partial derivatives of distortion parameters are derived from eq. 1 and two more applications of the chain rule,

$$\frac{\partial x_p}{\partial \theta_i} = \frac{\partial x_p}{\partial x_u} \frac{\partial x_u}{\partial \theta_i} + \frac{\partial x_p}{\partial y_u} \frac{\partial y_u}{\partial \theta_i}, \quad \frac{\partial y_p}{\partial \theta_i} = \frac{\partial y_p}{\partial x_u} \frac{\partial x_u}{\partial \theta_i} + \frac{\partial y_p}{\partial y_u} \frac{\partial y_u}{\partial \theta_i} \tag{8}$$

$$\begin{array}{l}
\frac{\partial x_p}{\partial x_u} = (Dm_0 - (m_0 x_u + m_1 * y_u + m_2) m_6)/D^2, \ \frac{\partial x_p}{\partial y_u} = (Dm_1 - (m_0 x_u + m_1 * y_u + m_2) m_7)/D^2 \\
\frac{\partial y_p}{\partial x_u} = (Dm_3 - (m_3 x_u + m_4 * y_u + m_5) m_6)/D^2, \ \frac{\partial y_p}{\partial y_u} = (Dm_4 - (m_3 x_u + m_4 * y_u + m_5) m_7)/D^2
\end{array} \tag{9}$$

Finally, the last set of formulas presented in [6],

$$\begin{array}{l}
\frac{\partial x_u}{\partial k_1} = r^2 (x - c_x)/s_x \\
\frac{\partial y_u}{\partial k_1} = r^2 (y - c_y) \\
\frac{\partial x_u}{\partial k_2} = r^4 (x - c_x)/s_x \\
\frac{\partial y_u}{\partial k_2} = r^4 (y - c_y) \\
\frac{\partial x_u}{\partial c_x} = 1 - (1/s_x)(1 + k_1 r^2 + k_2 r^4) - 2(k_1 + 2k_2 r^2)(x - c_x)^2/(s_x^3) \\
\frac{\partial y_u}{\partial c_x} = -2(k_1 + 2k_2 r^2)(x - c_x)(y - c_y)/s_x^2) \\
\frac{\partial x_u}{\partial c_y} = -2(k_1 + 2k_2 r^2)(x - c_x)(y - c_y)/s_x \\
\frac{\partial y_u}{\partial c_y} = 1 - (1 + k_1 r^2 + k_2 r^4) - 2(y - c_y)^2 (k_1 + 2k_2 r^2) \\
\frac{\partial x_u}{\partial s_x} = -(x - c_x)(1 + k_1 r^2 + k_2 r^4)/s_x^2 - 2(k_1 + 2k_2 r^2)(x - c_x)^3/s_x^4 \\
\frac{\partial y_u}{\partial s_x} = -2(y - c_y)(k_1 + 2k_2 r^2)(x - c_x)^2/s_x^3
\end{array} \tag{10}$$

where r was defined previously in eq. 1.

4.2 The Calibration Process

The calibration process starts with one image from the camera, I_d, another image from the calibration pattern, I_r, and initial values for parameters θ. The GNLM algorithm is as follows:

1. Compute the total error, $E_t(I_p(\theta), I_r)$ (eq. 3).
2. Pick a modest value for λ, say $\lambda = 0.001$.
3. Compute the image I_p (eq. 1, 2) applying bilinear interpolation to improve the quality of the image.
4. Solve the linear system of equations (4), and calculate $E_t(I_p(\theta + \delta\theta), I_r)$.
5. if $E_t(I_p(\theta + \delta\theta), I_r) > E_t(I_p(\theta), I_r)$, increase λ by a factor of 10, and go the previous step. If λ grows very large, it means that there is no way to improve the solution θ.
6. if $E_t(I_p(\theta + \delta\theta), I_r) < E_t(I_p(\theta), I_r)$, decrease λ by a factor of 10, replace θ by $\theta + \delta\theta$, and go to the first step.

When $\lambda = 0$, the GNLM method is a Gauss–Newton method, and when λ tends to infinity, $\delta\theta$ turns to so called steepest descent direction and the size $\delta\theta$ tends to zero.

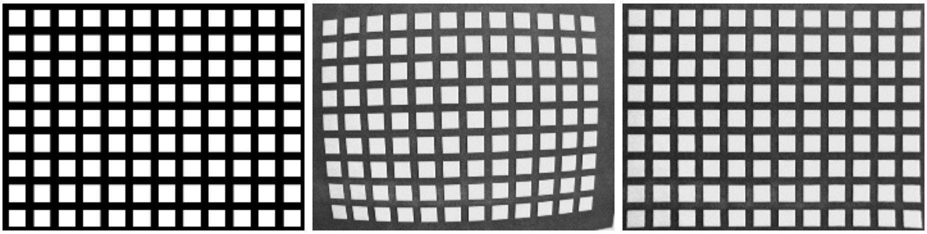
5 The Point Correspondences Method

This method tries to improve the calibration results using the approach described in previous section. When the calibration ends, the undistorted and projected image, I_p, is very similar to the pattern image, I_r. The idea is to extract points of both images associated to distinctive features. In our experiments we use corners as features because they are detected easily with subpixel precision.

The first step is to detect features in I_r and then search its correspondence in I_p. This search is limited to a small area because I_r and I_p are very similar. Let n be the number of features, (x_{rk}, y_{rk}) be the coordinates of a feature in I_r and (x_k, y_k) be its correspondence in I_p. From (x_k, y_k) and using eq. 1 and 2 we can get the coordinates (x_{pk}, y_{pk}) of the feature in the camera image (I_d). These calculations are denoted as follows, $x_{pk} = f_x^{pd}(\theta, x = x_k, y = y_k)$, $y_{pk} = f_y^{pd}(\theta, x = x_k, y = y_k)$ and $(x_{pk}, y_{pk}) = f^{pd}(\theta, x = x_k, y = y_k)$. So we have a set of pairs of points $P = \{<(x_{r1}, y_{r1}), (x_{p1}, y_{p1})>, \cdots, <(x_{rn}, y_{rn}), (x_{pn}, y_{pn})>\}$.

We formulate the goal of the calibration as to find a set of parameters θ such the sum, D_t, of square *distances* between points $f^{pd}(\theta, x_{rk}, y_{rk})$ and (x_{pk}, y_{pk}), is a minimum,

$$\begin{aligned} \theta &= argmin\, D_t(\theta, P) \\ &= argmin \sum_{k=1}^{n} (f_x^{pd}(\theta, x_{rk}, y_{rk}) - x_{pk})^2 + (f_y^{pd}(\theta, x_{rk}, y_{rk}) - y_{pk})^2 \end{aligned} \tag{11}$$



(a) Calibration Pattern (b) Image from camera (c) New image

Fig. 2. The calibration process

5.1 Non-linear Optimization

We use again the GNLM method to minimize D_t, but this time, the elements of matrix A and matrix B in eq. 4 are given by,

$$\begin{array}{ll} a_{i,j} = \sum_{k=1}^{n}(\frac{\partial x_{pk}}{\partial \theta_i}\frac{\partial x_{pk}}{\partial \theta_j} + \frac{\partial y_{pk}}{\partial \theta_i}\frac{\partial y_{pk}}{\partial \theta_j}), & b_i = -\sum_{k=1}^{n}(\frac{\partial x_{pk}}{\partial \theta_i}d_{xk} + \frac{\partial y_{pk}}{\partial \theta_i}d_{yk}) \\ d_{xk} = f_x^{pd}(\theta, x_{rk}, y_{rk}) - x_{pk}, & d_{yk} = f_y^{pd}(\theta, x_{rk}, y_{rk}) - y_{pk} \end{array} \quad (12)$$

6 Experimental Results

We test two Fire-i400 firewire industrial color camera from Unibrain with 4.00mm C-mount lens. These cameras acquire 30fps with resolution of 640 × 480 pixels.

The pattern calibration (image I_r), showed in Figure 2(a), was made using the program xfig under Linux. The image taken by the camera is shown in Figure 2(b). The corrected and projected image, using the image registration method, is shown in Figure 2(c). The GNLM process required 17 iterations and 57 seconds (using a PC Pentium IV, 1.8Ghz). We apply derivatives of Gaussians with $\sigma = 1$ pixels, initial values of $\theta^d = (0, 0, 240, 320, 1)$ and $\theta^p = (1, 0, 0, 0, 1, 0, 0, 0)$. At the end of the calibration process, the total error, E_t, between the projected image $I_p(\theta)$ and I_r (Figures 2(a) and (c)), was 14,820. This result is very good.

Corners of Figure 2(c) are easily detected applying two derivative filters. We apply a derivative of Gaussian ($\sigma = 2$ pixels) in one direction and then another in the other direction (see Figure 3). Pixels around corners have higher (or lower) values in Figure 3 (b). Corners are calculated, with subpixel precision, as the *center of mass* of pixels around the corners.

The point correspondences method required 39 iterations and less than 5 seconds. This time, E_t was 14,165, a slightly better result than with the other method. The difference is more evident from the sum of square distances, D_t. When using the image registration method we got $D_t = 518$, and $D_t = 131$ for

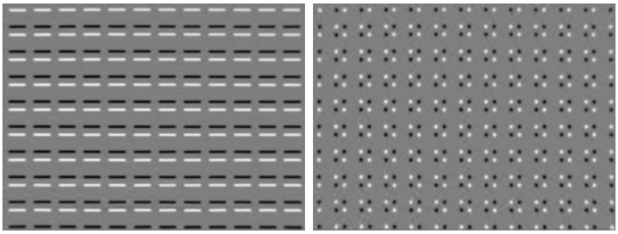


Fig. 3. Detecting corners

the point correspondences method, a significant reduction. This difference also can be observed calculating the maximum individual distance between points ($d_i = \sqrt{d_{xi}^2 + d_{yi}^2}$). Using this criteria, the image registration method got $d_i^{max} = 1.84$ pixels and the point correspondences method $d_i^{max} = 1.25$ pixels.

Finally, Figure 4 shows an application of the parameters obtained with the second method, $\theta^d = (-7.86 \times 10^{-07}, 6.43 \times 10^{-13}, 217.75, 310.68, 1.00)$. Images were expanded from $640x480$ pixels to 800×600, to see the complete expansion.

7 Related Works

There are two kinds of calibration methods. The first kind is the one that uses a calibration pattern or grid with features whose world coordinates are known. The second family of methods is those that use geometric invariants of the image features like parallel lines, spheres, circles, etc. [2].

The Methods described in this paper are in the first family of methods. The image registration method uses all points or pixels of the image as features, instead of the set of point correspondences of the second method. The correspondence with reference points are given implicitly in the pattern image (for

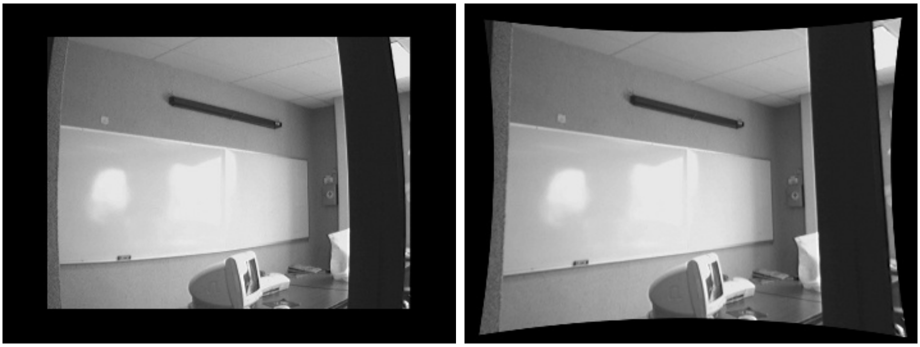


Fig. 4. Original and corrected images

the first method) or computed automatically (for the second method). Other methods require a human operator (with a lot of patience) to find such correspondences [6].

This method is an improved version of the method proposed by Tamaki et al. [6] and the differences between both approaches are:

- We take into account exact derivatives of x_p and y_p with respect to θ^p (eq. 7). Tamaki uses an approximation that is valid only when x_u and x_p (and y_u and y_p) are very similar. This approximation makes the method not very robust. Converge problems arise when parameters θ are not so closed to the right ones. Tamaki's method for the same images shown in Figure 2 gave us $E_t = 15310$, a slightly greater value than our method.
- We optimize the whole set of parameters θ using the GNLM method. Tamaki apply twice the Gauss–Newton method, one for θ^d and other for θ^p.
- We use a direct registration (from the camera image towards the pattern image), while Tamaki uses inverse registration (from the pattern image to the camera image).

8 Conclusions

We have described two calibration methods based on the Gauss–Newton–Levenberg–Marquardt non–linear optimization method using analytical derivatives. Other approaches compute numerical derivatives (e.q. [1,2,4]), so we have faster calculations and better convergence properties.

The first method is an image registration method, which is an improved version of a previous one [6]. The second method takes advantages of results from the first method to solve the correspondence problem between features of the camera image and the pattern image. Also takes advantage of detecting features (corners) with subpixel precision. This combination gives better calibration results than with the image registration method.

References

1. F. Devernay and O.D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. *SPIE*, 2567:62–72, July 1995.
2. F. Devernay and O.D. Faugeras. Straight lines have to be straight. *MVA*, 13(1):14–24, 2001.
3. W. Press, B. Flannery, S. Teukolsky, and Vetterling W. *Numerical recipes, the art of scientific computing.* Cambridge University Press, 1986.
4. G. P. Stein. Lens distortion calibration using point correspondences. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR '97)*, June 1997.
5. R. Szeliski. Video mosaic for virtual environments. *IEICE Computer Graphics and Applications*, 16(2):22–30, March 1996.
6. T. Tamaki, T. Yamamura, and N. Ohnish. A method for compensation of image distortion with image registration technique. *IEICE Trans. Inf. and Sys.*, E84-D(8):990–998, August 2001.