Correction of Geometric Lens Distortion Through Image Warping*

Luca Lucchese
School of Electrical Engr. and Computer Science
Oregon State University
Corvallis, OR 97331
luca@ece.orst.edu

Sanjit K. Mitra
Dept. of Electrical and Computer Engr.
University of California
Santa Barbara, CA 93106
mitra@ece.mitra.edu

Abstract

This paper presents a new technique for estimating and correcting the geometric distortion produced by common off-the-shelf lenses. The taks of recovering the three most important parameters modeling the lens distortion of a digital camera is formulated as a problem of image warping where the distorted image of a planar calibration plate taken by the camera has to be registered onto the virtual undistorted image of the same target that would be produced by an ideal pinhole digital camera in a convenient position with respect to it. The Levenberg-Marquardt algorithm is chosen for solving the associated minimization problem in which the cost function is the squared $\mathcal{L}_2$ norm of the difference between the intensities of the reference and distorted image. In this paper, the algorithm is applied to a practical case of lens distortion correction; the experimental results reported here confirm the effectiveness of the image warping approach.

1 Introduction

In Computer Vision, *camera calibration* refers to the problem of recovering the *external* and *internal geometry* of an optical acquisition device in order to have a complete description of its image formation process and, therefore, to be able to make accurate 3-D measurements from 2-D imagery [1, 2, 3]. The external geometry of a camera is defined as the 3-D motion which relates the 3-D reference frame attached to the camera, the *camera coordinate system*, to a given 3-D *world coordinate system*. The internal geometry refers instead to the parameters of the mathematical model describing the geometric aberrations produced by the lens system of the camera. The displacements due to this kind of distortion are usually modeled with appropriate polynomials in the coordinates of the 2-D frame indexing the camera image plane [4].

Many calibration algorithms estimate the external and internal camera geometry at the same time by establishing relationships between 3-D fiducial markers and their projections on the image plane [1, 2]. However, the correction for lens distortion can be tackled and solved as a problem *per se*, without recovering the external geometry of the camera. Usually, calibration targets containing straight lines [5, 6, 7, 8], grids, or regular patterns [9] are used to estimate the internal geometry since straightness or spatial regularity constitute convenient prior information that can be exploited. In this paper, we resort to image warping [10] as a means for retrieving the internal geometry of an unknown digital camera. The idea of using image warping was originally proposed by Collins and Tsin for calibrating active camera systems [11]; their method is based on a dense optical flow approach. In our method instead, the distorted image of a regular calibration pattern – a black-and-white checkerboard – is registered on a virtual reference image generated by an ideal pinhole camera devoid of any lens and perspective warp. The experimental results presented in this paper confirm the effectiveness of our algorithm

This paper has five sections. Section 2 introduces the mathematical model used for correction of lens distortion. Section 3 presents the image warping algorithm. Section 4 reports some experimental results. Section 5 draws the conclusions.

2 Mathematical Model for Correction of Geometric Lens Distortion

2.1 Internal Geometry of the Camera

Let us consider the picture of Fig. 1 (a); it represents the image $f_d(x, y)$, $(x, y) \in \mathbb{R}^2$, of a planar calibration plate with a black-and-white regular checkerboard pattern. [1] Owing to the geometric distortion introduced by the optical system of the camera, lines which are straight on the actual calibration

*This work was supported in part by a University of California MICRO grant with matching supports from Conexant Corp., Lucent Technologies, and Philips Research Labs, and in part by Microsoft Corp.

[1] The image of Fig. 1 (a), downloaded from [12], belongs to a set of 25 images of the same calibration checkerboard taken from various viewpoints.

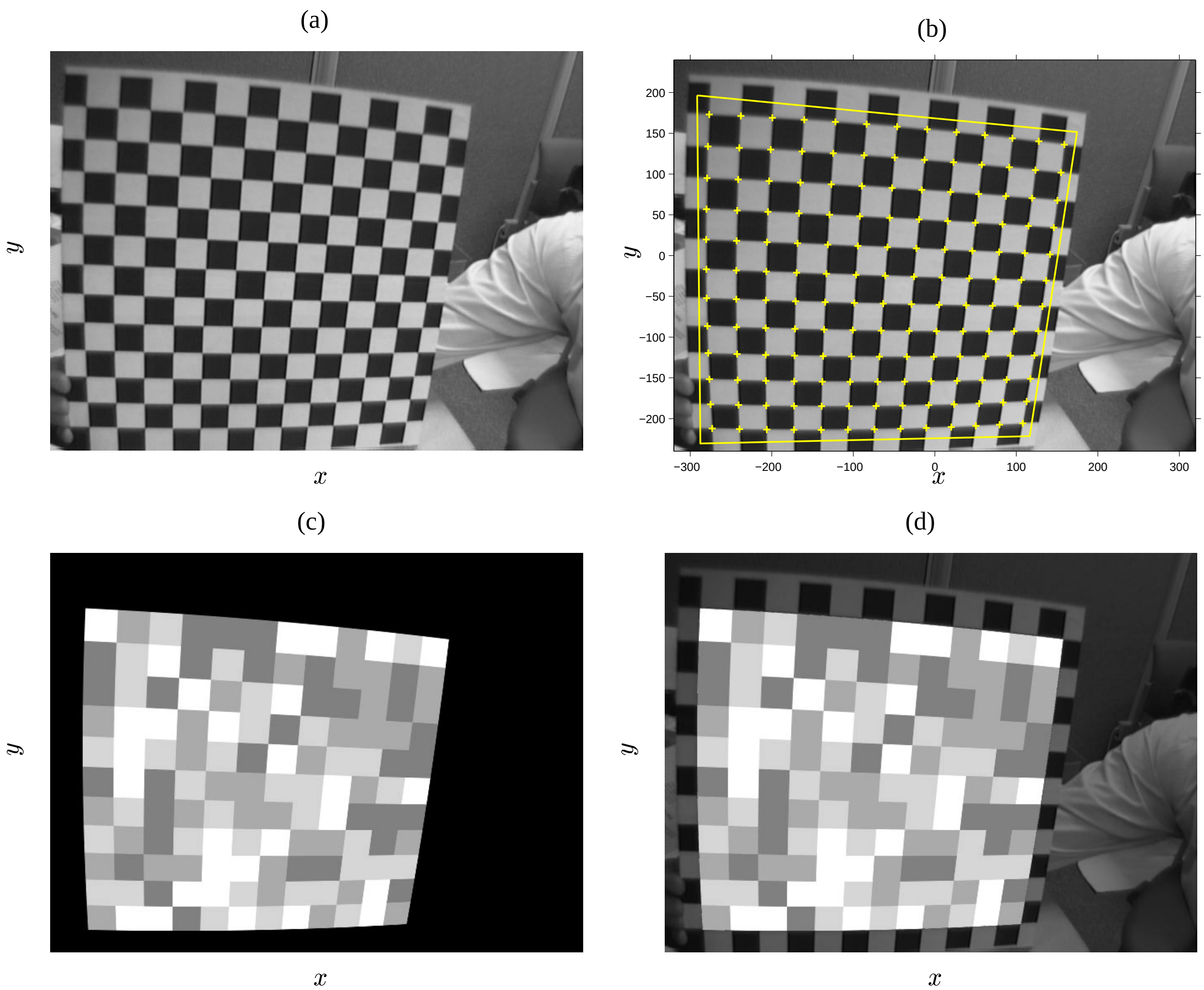

Figure 1. (a) Image $f_d(x, y)$ of a planar calibration plate with a checkerboard pattern; (b) Selection of the region of interest in $f_d(x, y)$ and extraction of the X-junctions; (c) Synthetic random pattern $g_d(x, y)$ built from the X-junctions extracted from $f_d(x, y)$; (d) Image $g_d(x, y)$ overlaid to $f_d(x, y)$.

plate appear curved in $f_d(x, y)$. The orthogonal Cartesian frame xy defines the reference system of the camera image plane.

By denoting with $\boldsymbol{x}_d \doteq (x_d, y_d)$ any generic (distorted) point of $f_d(x, y)$, the correct location $\boldsymbol{x}_d \doteq (x_u, y_u)$ that would be measured if the camera were devoid of geometric lens distortion is given by

$$\begin{cases} x_u = \psi_1(x_d, y_d; \boldsymbol{\zeta}_\ell) = x_d + \delta_1(x_d, y_d; \boldsymbol{\zeta}_\ell), \\ y_u = \psi_2(x_d, y_d; \boldsymbol{\zeta}_\ell) = y_d + \delta_2(x_d, y_d; \boldsymbol{\zeta}_\ell), \end{cases} \tag{1}$$

where $\delta_1(x_d, y_d; \boldsymbol{\zeta}_\ell)$ and $\delta_2(x_d, y_d; \boldsymbol{\zeta}_\ell)$ are the amounts of correction along the x-axis and y-axis, respectively. According to Eqs. (1), the two corrective offsets are functions of the distorted location $\boldsymbol{x}_d$ as well as of the parameters $\boldsymbol{\zeta}_\ell$ describing the *internal* (or *intrinsic*) *geometry* of the camera [1, 2, 3]. Brown advanced an accurate mathematical model for $\delta_1(x_d, y_d; \boldsymbol{\zeta}_\ell)$ and $\delta_2(x_d, y_d; \boldsymbol{\zeta}_\ell)$ which depends on three sets of parameters [1, 4]: i) the coordinates (x_p, y_p) of the principal point (where the optical axis of the camera intersects the image plane); ii) the polynomial coefficients κ_i of the radial component of distortion; and iii) the polynomial coefficients p_i of the decentering component of distortion. In many practical cases, a very good description of the geometric lens aberration can be obtained by keeping only the three parameters which account for most of the distortion, namely the coordinates of the principal point and the first radial distortion coefficient κ_1. The two offsets in Eqs. (1) thus read

$$\begin{cases} \delta_1(x_d, y_d; \boldsymbol{\zeta}_\ell) = \kappa_1(x_d - x_p)r^2, \\ \delta_2(x_d, y_d; \boldsymbol{\zeta}_\ell) = \kappa_1(y_d - y_p)r^2, \end{cases} \tag{2}$$

where $r^2 \doteq (x_d - x_p)^2 + (y_d - y_p)^2$ and the internal geometry of the camera is described by the vector $\boldsymbol{\zeta}_\ell \doteq [x_p \; y_p \; \kappa_1]^T \in \mathbb{R}^3$.

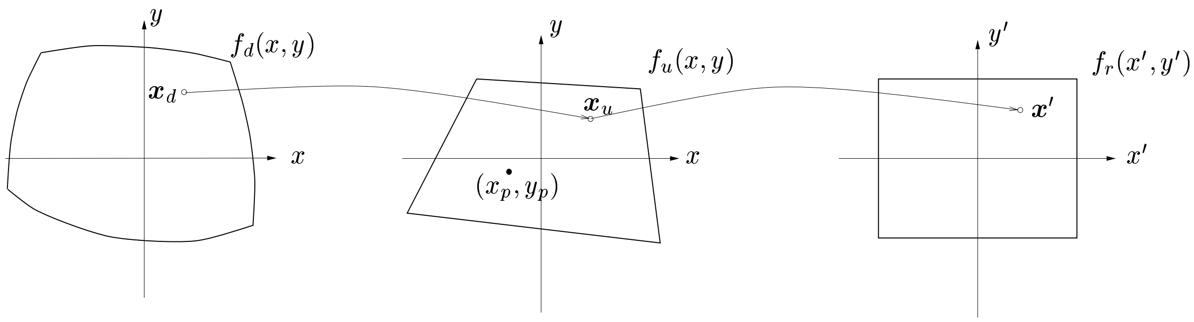

Figure 2. The two geometric transformations relating image $f_d(x, y)$ to the virtual reference image $f_r(x', y')$; $f_u(x, y)$ represents the version of $f_d(x, y)$ after compensation for lens distortion; (x_p, y_p) are the coordinates of the principal point of camera.

2.2 2-D Projective Transformation

If the parameters $\boldsymbol{\zeta}_\ell$ in Eqs. (2) were known, it would be possible to compensate the image $f_d(x, y)$ for lens distortion through Eqs. (1) and (2) and transform it into a new 'undistorted' image $f_u(x, y)$; an ideal digital pinhole camera with the same focal length would yield such an image. Let us suppose now that we can move this virtual pinhole camera into a *centered fronto-parallel configuration* [2] with respect to the calibration plate, the planar checkerboard. This simply means that the axes of its image plane are parallel to the sides of the checkerboard and that its optical axis pierces the plate at the center. Let $f_r(x', y')$, $(x', y') \in \mathbb{R}^2$ denote the image of the calibration pattern seen from this position, $x'y'$ being the reference frame of the virtual pinhole camera image plane. Based on the planarity of the calibration plate, it is straightforward to show that the two images $f_u(x, y)$ and $f_r(x', y')$ relate through a 2-D *projective transformation* or *projectivity* [3] defined by $f_r(x', y') = f_u(x, y)$, where

$$\begin{cases} x' = \dfrac{a_{11}(x - x_p) + a_{12}(y - y_p) + b_1}{c_1(x - x_p) + c_2(y - y_p) + 1}, \\ y' = \dfrac{a_{21}(x - x_p) + a_{22}(y - y_p) + b_2}{c_1(x - x_p) + c_2(y - y_p) + 1}, \end{cases} \tag{3}$$

with $a_{ij}, b_i, c_i \in \mathbb{R}$, $i, j = 1, 2$. Eqs. (3) can be simplified and rewritten as

$$\begin{cases} x' = \varphi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\mathcal{P}) \doteq \dfrac{\alpha_{11}x + \alpha_{12}y + \beta_1}{\gamma_1 x + \gamma_2 y + 1}, \\ y' = \varphi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\mathcal{P}) \doteq \dfrac{\alpha_{21}x + \alpha_{22}y + \beta_2}{\gamma_1 x + \gamma_2 y + 1}, \end{cases} \tag{4}$$

where $\alpha_{ij} \doteq a_{ij}/d$, $\beta_i \doteq (b_i - a_{i1}x_p - a_{i2}y_p)/d$, and $\gamma_i \doteq c_i/d$, with $d \doteq 1 - c_1 x_p - c_2 y_p$, $i, j = 1, 2$, $\boldsymbol{x} \doteq [x\ y]^T$, and $\boldsymbol{\zeta}_\mathcal{P} \doteq [\alpha_{11}\ \alpha_{21}\ \alpha_{12}\ \alpha_{22}\ \beta_1\ \beta_2\ \gamma_1\ \gamma_2]^T \in \mathbb{R}^8$. By combining the two geometric transformations of Eqs. (1) and (2) and Eqs. (4), the two images $f_d(x, y)$ and $f_r(x', y')$ relate through $f_r(x', y') = f_d(x, y)$, where

$$\begin{cases} x' = \varphi_1\left(\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right) = \\ \quad = \dfrac{\alpha_{11}\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + \alpha_{12}\psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + \beta_1}{\gamma_1\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + \gamma_2\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + 1}, \\ y' = \varphi_2\left(\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right) \\ \quad = \dfrac{\alpha_{21}\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + \alpha_{22}\psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + \beta_2}{\gamma_1\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + \gamma_2\psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell) + 1}, \end{cases} \tag{5}$$

The two geometric transformations are schematically represented in Fig. 2.

3 Image Warping

Based on Eqs. (5), the estimation of the internal geometry of the camera can be set forth as a problem of image warping [10] where the image $f_r(x', y')$ has to be registered over $f_d(x, y)$ by minimizing their distance expressed by the squared $\mathcal{L}_2$ norm, *i.e.*,

$$\begin{aligned} &\min_{\boldsymbol{\zeta}} \chi^2(\boldsymbol{\zeta}) \doteq \\ &\min_{\boldsymbol{\zeta}} \Big\| f_r\Big(\varphi_1\left(\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right), \\ &\quad \varphi_2\left(\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right)\Big) - f_d(\boldsymbol{x})\Big\|^2_{\mathcal{L}_2} = \\ &\min_{\boldsymbol{\zeta}} \int_{\mathbb{R}^2} \Big(f_r\Big(\varphi_1\left(\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right), \\ &\quad \varphi_2\left(\psi_1(\boldsymbol{x}; \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}; \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right)\Big) - f_d(\boldsymbol{x})\Big)^2 d\boldsymbol{x}, \end{aligned} \tag{6}$$

where $\boldsymbol{\zeta} \doteq [\boldsymbol{\zeta}_\ell^T\ \boldsymbol{\zeta}_\mathcal{P}^T]^T \in \mathbb{R}^{11}$ denotes the overall parameter vector. The eight coefficients of the 2-D homography in Eqs. (5) are obtained as a byproduct of the solution to the problem in Eq. (6).

For convenience, the function to minimize in Eq. (6) can be rewritten as

$$\begin{aligned} &\chi^2(\boldsymbol{\zeta}) \doteq \sum_{n \in \mathbb{P}} \left(\Delta(n, \boldsymbol{\zeta})\right)^2 \doteq \\ &\sum_{n \in \mathbb{P}} \Big(f_r\Big(\varphi_1\left(\psi_1(\boldsymbol{x}(n); \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}(n); \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right), \\ &\quad \varphi_2\left(\psi_1(\boldsymbol{x}(n); \boldsymbol{\zeta}_\ell), \psi_2(\boldsymbol{x}(n); \boldsymbol{\zeta}_\ell); \boldsymbol{\zeta}_\mathcal{P}\right)\Big) - f_d(\boldsymbol{x}(n))\Big)^2, \end{aligned} \tag{7}$$

where the sum numerically approximates the integral in Eq. (6) and $\mathbb{P}$ indicates the set of pixels of the digital versions of the two images. The problem of Eq. (6) can be solved with the *Levenberg-Marquardt algorithm* [2] which iteratively refines the estimate of the extremal point by solving the *normal equations*

$$\delta\boldsymbol{\zeta} = -\left(\boldsymbol{\mathcal{H}}_{\boldsymbol{\zeta}}\chi^2 + \xi \boldsymbol{I}_{11}\right)^{-1} \nabla_{\boldsymbol{\zeta}}\chi^2, \tag{8}$$

where $\boldsymbol{\mathcal{H}}_{\boldsymbol{\zeta}}\chi^2 \in \mathbb{R}^{11\times 11}$ and $\nabla_{\boldsymbol{\zeta}}\chi^2 \in \mathbb{R}^{11}$ respectively denote the Hessian matrix and the gradient of the cost function $\chi^2(\boldsymbol{\zeta})$ with respect to the parameter vector $\boldsymbol{\zeta}$, $\boldsymbol{I}_{11} \in \mathbb{R}^{11\times 11}$ being the identity matrix, and ξ is an appropriate iteration-dependent stabilization parameter; if $\hat{\boldsymbol{\zeta}}^{(k)}$ is the estimate of the extremal point at the k-th iteration, its update is given by $\hat{\boldsymbol{\zeta}}^{(k+1)} = \hat{\boldsymbol{\zeta}}^{(k)} + \delta\boldsymbol{\zeta}$.

For compactness of notation, let us denote the matrix $\boldsymbol{\mathcal{H}}_{\boldsymbol{\zeta}}\chi^2$ and the vector $\nabla_{\boldsymbol{\zeta}}\chi^2$, respectively, as $\boldsymbol{\mathcal{H}}_{\boldsymbol{\zeta}}\chi^2 = \left[\frac{\partial^2\chi^2(\boldsymbol{\zeta})}{\partial\zeta_i\partial\zeta_j}\right]$ and $\nabla_{\boldsymbol{\zeta}}\chi^2 = \left[\frac{\partial\chi^2(\boldsymbol{\zeta})}{\partial\zeta_j}\right]$, $i,j = 1,\ldots,11$. The first partial derivatives of χ^2 are

$$\frac{\partial\chi^2(\boldsymbol{\zeta})}{\partial\zeta_j} = 2\sum_{n\in\mathbb{P}} \Delta(n,\boldsymbol{\zeta})\frac{\partial f_r}{\partial\zeta_j}. \tag{9}$$

The derivatives of f_r with respect to the projective transformation parameters $\boldsymbol{\zeta}_{\mathcal{P}}$ can be expressed as

$$\frac{\partial f_r}{\partial\zeta_j} = \nabla_{\boldsymbol{\varphi}}^T f_r \begin{bmatrix} \frac{\partial\varphi_1}{\partial\zeta_j} \\ \frac{\partial\varphi_2}{\partial\zeta_j} \end{bmatrix}, \quad j = 4,\ldots,11, \tag{10}$$

where $\nabla_{\boldsymbol{\varphi}} f_r \doteq \left[\frac{\partial f_r}{\partial\varphi_1}\ \frac{\partial f_r}{\partial\varphi_2}\right]^T$. The nonzero partial derivatives of φ_1 and φ_2 with respect to the projective transformation parameters $\boldsymbol{\zeta}_{\mathcal{P}}$ are:

$$\begin{aligned}
&\frac{\partial\varphi_1}{\partial\alpha_{11}} = \frac{\partial\varphi_2}{\partial\alpha_{21}} = \frac{\psi_1}{D}, \quad \frac{\partial\varphi_1}{\partial\alpha_{12}} = \frac{\partial\varphi_2}{\partial\alpha_{22}} = \frac{\psi_2}{D},\\
&\frac{\partial\varphi_1}{\partial\beta_1} = \frac{\partial\varphi_2}{\partial\beta_2} = \frac{1}{D}, \quad \frac{\partial\varphi_1}{\partial\gamma_1} = -\varphi_1\frac{\psi_1}{D},\\
&\frac{\partial\varphi_1}{\partial\gamma_2} = -\varphi_1\frac{\psi_2}{D}, \quad \frac{\partial\varphi_2}{\partial\gamma_1} = -\varphi_2\frac{\psi_1}{D}, \quad \frac{\partial\varphi_2}{\partial\gamma_2} = -\varphi_2\frac{\psi_2}{D},
\end{aligned} \tag{11}$$

where $D \doteq \gamma_1\psi_1 + \gamma_2\psi_2 + 1$. The derivatives of f_r with respect to the lens distortion parameters $\boldsymbol{\zeta}_\ell$ can be expressed as

$$\frac{\partial f_r}{\partial\zeta_j} = \frac{1}{D}\nabla_{\boldsymbol{\varphi}}^T f_u \left[\boldsymbol{A} - \boldsymbol{\varphi}\,\boldsymbol{c}^T\right] \begin{bmatrix} \frac{\partial\psi_1}{\partial\zeta_j} \\ \frac{\partial\psi_2}{\partial\zeta_j} \end{bmatrix}, \quad j = 1,2,3, \tag{12}$$

where $\boldsymbol{A} \doteq \left[\begin{smallmatrix}\alpha_{11} & \alpha_{12}\\ \alpha_{21} & \alpha_{22}\end{smallmatrix}\right] \in \mathbb{R}^{2\times 2}$, $\boldsymbol{\varphi} \doteq [\varphi_1\ \varphi_2]^T \in \mathbb{R}^2$, and $\boldsymbol{c} \doteq [\gamma_1\ \gamma_2]^T \in \mathbb{R}^2$. The partial derivatives of ψ_1 and ψ_2 with respect to the three lens distortion parameters $\boldsymbol{\zeta}_\ell$ are

$$\begin{aligned}
&\frac{\partial\psi_1}{\partial x_p} = -\kappa_1\left(r^2 + (x-x_p)^2\right),\\
&\frac{\partial\psi_2}{\partial x_p} = \frac{\partial\psi_1}{\partial y_p} = -2\kappa_1(x-x_p)(y-y_p),\\
&\frac{\partial\psi_2}{\partial y_p} = -\kappa_1\left(r^2 + 2(y-y_p)^2\right),\\
&\frac{\partial\psi_1}{\partial\kappa_1} = r^2(x-x_p), \quad \frac{\partial\psi_2}{\partial\kappa_1} = r^2(y-y_p),
\end{aligned} \tag{13}$$

By collecting the eleven partial derivatives of f_r with respect to $\boldsymbol{\zeta}$ into the gradient vector

$$\nabla_{\boldsymbol{\zeta}} f_r \doteq \begin{bmatrix} \frac{\partial f_r}{\partial\zeta_1} \\ \vdots \\ \frac{\partial f_r}{\partial\zeta_3} \\ \hline \frac{\partial f_r}{\partial\zeta_4} \\ \vdots \\ \frac{\partial f_r}{\partial\zeta_{11}} \end{bmatrix} = \begin{bmatrix} \frac{1}{D}\nabla_{\boldsymbol{\varphi}}^T f_r \left[\boldsymbol{A} - \boldsymbol{\varphi}\,\boldsymbol{c}^T\right] \begin{bmatrix} \frac{\partial\psi_1}{\partial\zeta_1} \\ \frac{\partial\psi_2}{\partial\zeta_1}\end{bmatrix} \\ \vdots \\ \frac{1}{D}\nabla_{\boldsymbol{\varphi}}^T f_r \left[\boldsymbol{A} - \boldsymbol{\varphi}\,\boldsymbol{c}^T\right] \begin{bmatrix} \frac{\partial\psi_1}{\partial\zeta_3} \\ \frac{\partial\psi_2}{\partial\zeta_3}\end{bmatrix} \\ \hline \nabla_{\boldsymbol{\varphi}}^T f_r \begin{bmatrix} \frac{\partial\varphi_1}{\partial\zeta_4} \\ \frac{\partial\varphi_2}{\partial\zeta_4}\end{bmatrix} \\ \vdots \\ \nabla_{\boldsymbol{\varphi}}^T f_r \begin{bmatrix} \frac{\partial\varphi_1}{\partial\zeta_{11}} \\ \frac{\partial\varphi_2}{\partial\zeta_{11}}\end{bmatrix} \end{bmatrix}, \tag{14}$$

the gradient of the objective function can finally be written, through Eq. (10), as

$$\nabla_{\boldsymbol{\zeta}}\chi^2 = 2\sum_{n\in\mathbb{P}} \Delta(n,\boldsymbol{\zeta})\nabla_{\boldsymbol{\zeta}} f_r. \tag{15}$$

The second partial derivatives of $\chi^2(\boldsymbol{\zeta})$ are

$$\frac{\partial^2\chi^2(\boldsymbol{\zeta})}{\partial\zeta_i\partial\zeta_j} = 2\sum_{n\in\mathbb{P}}\left(\frac{\partial f_r}{\partial\zeta_i}\frac{\partial f_r}{\partial\zeta_j} + \Delta(n,\boldsymbol{\zeta})\frac{\partial^2 f_r}{\partial\zeta_i\partial\zeta_j}\right), \tag{16}$$

$i,j = 1,\ldots,11$. Usually, the Hessian $\boldsymbol{\mathcal{H}}_{\boldsymbol{\zeta}}\chi^2$ is computed by disregarding in Eq. (16) the term containing the second partial derivatives of f_r which, as the algorithm progressively converges to the optimal solution, becomes negligible in comparison to the term involving only first partial derivatives of f_r [2]. Therefore, the Hessian of the objective function can be approximated as

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{\zeta}}\chi^2 = 2\sum_{n\in\mathbb{P}} \nabla_{\boldsymbol{\zeta}} f_r \nabla_{\boldsymbol{\zeta}}^T f_r. \tag{17}$$

In order to apply our image warping algorithm, the image $f_d(x,y)$ of Fig. 1 (a) has to be segmented first. A region of

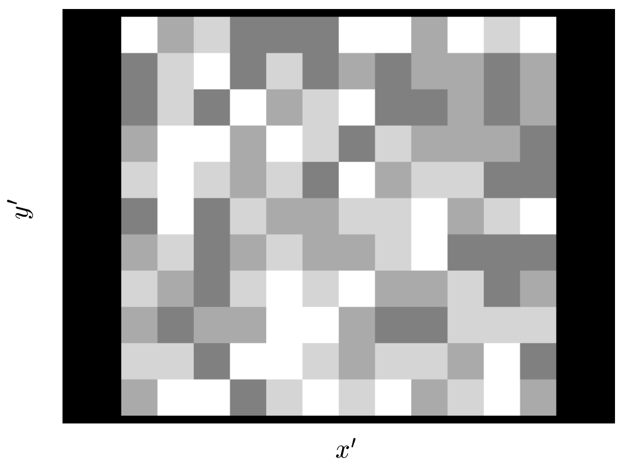

Figure 3. Virtual reference image $g_r(x', y')$.

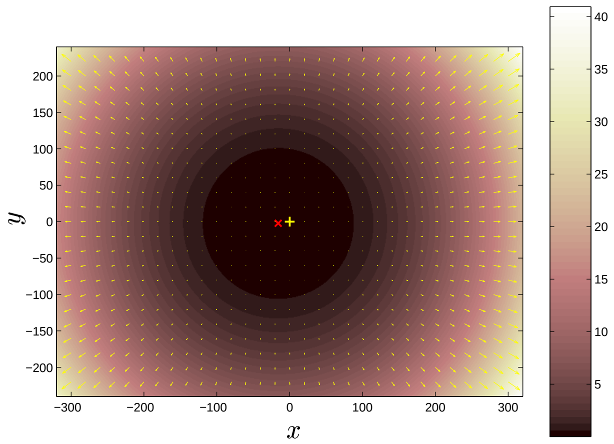

Figure 4. Compensation for lens distortion: the intensity is proportional to the corrective offset $\sqrt{\delta_1^2 + \delta_2^2}$ (in pixels) as indicated by the colorbar.

interest (ROI) is defined by drawing with a graphical user interface (GUI) we have implemented the four lines shown in Fig. 1 (b). The algorithm of [13] is then applied within the ROI to determine the subpixel locations of the X-junctions (or corners) of the checkerboard pattern in $f_d(x, y)$; the X-junctions found with this method are marked with yellow '+' symbols in Fig. 1 (b). The grid associated with these features is then filled with randomly generated gray levels to generate the textured image $g_d(x, y)$ shown in Fig. 1 (c); Fig. 1 (d) shows the superposition of this image to the actual image $f_d(x, y)$. The same gray levels are used to generate the virtual image seen by the ideal pinhole camera in the centered fronto-parallel configuration with respect to the calibration plate; the corresponding image $g_r(x', y')$ is shown in Fig. 3. Of course, the two images $g_d(x, y)$ and $g_r(x', y')$ relate as $f_d(x, y)$ and $f_r(x', y')$, *i.e.*, through Eqs. (5).

4 Experimental Results

The image warping algorithm of Section 3 can be directly applied to $g_d(x, y)$ and $g_r(x', y')$. However, if these images are large in size, like in our case, being 480×680 pixels, the solution of Eq. (6) may be very rather lengthy. It is therefore advisable to isotropically shrink the two images according to $g_d(\rho x, \rho y)$ and $g_r(\rho x', \rho y')$, with $0 < \rho < 1$; in our implementation, we have set $\rho = 0.5$ so that the two shrunk images are one fourth the area of the original images. The new parameter vector becomes $\boldsymbol{\zeta}_\rho = \left[(x_p/\rho) \;\; (y_p/\rho) \;\; (\rho^2\kappa_1) \;\; \alpha_{11} \;\; \alpha_{21} \;\; \alpha_{12} \;\; \alpha_{22} \;\; (\beta_1/\rho) \;\; (\beta_2/\rho) \;\; (\rho\gamma_1) \;\; (\rho\gamma_2)\right]^T \in \mathbb{R}^{11}$.

The solution of the minimization problem of Eq. (6) requires an appropriate choice of the initial values of the parameters $\boldsymbol{\zeta}$ in order to guarantee the global convergence of the normal equations of Eq. (8) to the optimal solution of the cost function $\chi^2(\boldsymbol{\zeta})$. By setting $x_p = 0$, $y_p = 0$, and $\kappa_1 = 0$ in Eqs. (5), the parameters $\boldsymbol{\zeta}_{\mathcal{P}}$ can be found in a closed form with a linear least-squares algorithm since Eqs. (5) are linear with respect to these parameters [2, 3] and the locations of the X-junctions of $f_d(x, y)$ are known together with the corresponding points of $f_r(x', y')$. Let $\hat{\alpha}_{ij}$, $\hat{\beta}_i$, and $\hat{\gamma}_i$, $i, j = 1, 2$, be the estimates of the coefficients of $\boldsymbol{\zeta}_{\mathcal{P}}$ obtained with this method. The starting point for the Levenberg-Marquardt algorithm is therefore set as $\boldsymbol{\zeta}_o = [0 \; 0 \; 0 \; \hat{\alpha}_{11} \; \hat{\alpha}_{21} \; \hat{\alpha}_{12} \; \hat{\alpha}_{22} \; \hat{\beta}_1 \; \hat{\beta}_2 \; \hat{\gamma}_1 \; \hat{\gamma}_2]^T$. Usually, this initial condition turns out to be very close to the optimal solution whence only a few iterations are necessary for the convergence of the algorithm; in the example of this paper, convergence was achieved in a dozen iterations.

The second column of Table 1 displays the parameter vector $\boldsymbol{\zeta}$ obtained from the internal and external parameters estimated with the *Matlab* Calibration Toolboox of [12]; the third column shows instead the results returned by out image warping algorithm. It should be noticed that the two sets of parameters are very close. Figure 4 provides a graphical rendition of the overall corrective offset $\sqrt{\delta_1^2 + \delta_2^2}$, where δ_1 and δ_2 are the two displacements of Eqs. (2) for the estimated values of x_p, y_p, and κ_1; the yellow '+' symbol denotes the physical center of the image plane whereas the red '×' symbol indicates the estimated principal point. The legend of Fig. 4 clearly shows that, near the four corners, the necessary corrections are of about forty pixels. Figure 5 (a) displays the original image $f_d(x, y)$ after compensation for lens distortion with the values of x_p, y_p, and κ_1 in the third column of Table 1; Figure 5 (b) shows the *rectification* of $f_d(x, y)$, *i.e.*, the compensation of this image for both lens distortion and perspective warp according to the estimates of the third column of Table 1.

A final note on the use of the randomly textured pattern introduced in Section 3. The algorithm was also tested by filling the grid of Fig. 1 (b) with a regular black-and-white

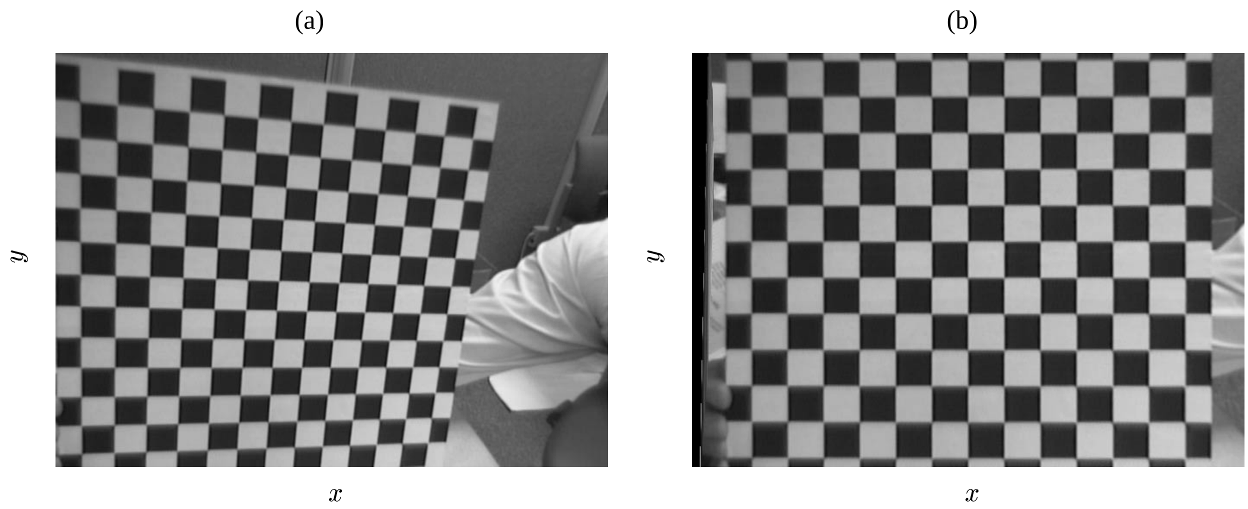

Figure 5. (a) Image $f_d(x, y)$ after compensation for lens distortion; (b) Image $f_d(x, y)$ after rectification.

checkerboard pattern like the one of the original image, but the randomly textured pattern, unlike the other, guaranteed the convergence to the optimal solution. This can be explained by observing that the normal equations of Eq. (8) used by the Levenberg-Marquardt algorithm are based on image gradients and patterns like that shown in Fig. 1 (c) result in a gradient across the image more diverse than the gradient produced by a regular black-and-white checkerboard; different image portions are then associated with different gradient values and, therefore, more spatial clues are available for steering the warping of the reference image $f_r(x', y')$ onto $f_d(x, y)$.

5 Conclusions

In this article, we have advanced a new method for estimating the geometric aberration produced by the lens system of a digital camera. Our algorithm estimates its internal geometry by using image warping. The image of a planar calibration plate is first conveniently processed in order to extract a pattern of interest; this image is then warped onto a virtual image of the same pattern that would be generated by an ideal pinhole camera devoid of lens and perspective warp. The experiment presented in the paper shows the effectiveness of this technique.

ζ	Calibration Toolbox	image warping
x_p	-16.48	-15.72
y_p	-2.90	-2.26
κ_1	$5.71\,10^{-7}$	$5.71\,10^{-7}$
α_{11}	1.18	1.18
α_{21}	0.03	0.03
α_{12}	-0.09	-0.09
α_{22}	1.22	1.22
β_1	69.91	69.82
β_2	53.57	53.46
γ_1	$-3.30\,10^{-4}$	$-3.32\,10^{-4}$
γ_2	$3.79\,10^{-4}$	$3.79\,10^{-4}$

Table 1. Estimates of the parameter vector ζ obtained with the *Matlab* Calibration Toolbox of [12] (second column) and with our technique (third column).

References

[1] R. Jain, R. Kasturi, and B.G. Shunck, *Machine Vision*, McGraw-Hill, Inc., New York, 1995.

[2] D.A. Forsyth and J. Ponce, *Computer Vision – A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 2002.

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

[4] D.C. Brown, "Close Range Camera Calibration," *Photogrammetric Engineering*, Vol. 37, No. 8, pp. 885-866, Aug. 1971.

[5] B. Prescott and G.F. McLean, "Line-Based Correction of Radial Lens Distortion," *Graphical Models and Image Proc.*, Vol. 59, No. 1, pp. 39-47, Jan. 1997.

[6] R. Swaminathan and S.K. Nayar, "Nonmetric Calibration of Wide-Angle Lenses and Polycameras," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1172-1178, Oct. 2000.

[7] F. Devernay and O. Faugeras, "Straight Lines Have To Be Straight," *Machine Vision and Applications*, Srpinger-Verlag, Germany, Vol. 13, No. 1, pp. 14-24, 2001

[8] M. Ahmed and A. Farag, "Non-Metric Calibration of Camera Lens Distortion," *Proc. of 2001 Int'l Conf. on Image Processing (ICIP'01)*, Thessaloniki, Greece, Oct. 7-10 2001, Vol. 2, pp. 157-160.

[9] J. Baltes, "Camera Calibration Using Rectangular Textures," *Proc. of Int'l Workshop RobVis 2001*, Berlin Germany, 2001, pp. 245-51.

[10] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, Los Alamitos, California, 1992.

[11] R.T. Collins and Y. Tsin, "Calibration of an Outdoor Active Camera System," *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition 1999*, Vol. 1 , pp. 528-534.

[12] J.-Y. Bouguet, *Camera Calibration Toolbox for Matlab*, `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

[13] L. Lucchese and S.K. Mitra, "Using Saddle Points for Subpixel Feature Detection in Camera Calibration Targets," *Proc. of the 2002 Asia Pacific Conference on Circuits and Systems*, Singapore, Dec. 16-18, 2002.