Automatic Correction of Perspective and Optical Distortions

Daniel Santana-Cedrés[a], Luis Gomez[b], Miguel Alemán-Flores[a], Agustín Salgado[a], Julio Esclarín[a], Luis Mazorra[a], Luis Alvarez[a,*]

[a]*CTIM, Departamento de Informática y Sistemas, Universidad de Las Palmas de G.C., Campus de Tafira, 35017 Las Palmas, Spain.*
[b]*CTIM, Departamento de Ingeniería Electrónica y Automática, Universidad de Las Palmas de G.C., Campus de Tafira, 35017 Las Palmas, Spain.*

Abstract

Perspective and optical (lens) distortions are aberrations of very different nature that can simultaneously affect an image. Perspective distortion is caused by the position of the camera, especially when it is too close to the scene. Optical distortion is a lens aberration which causes straight lines in the scene to be projected onto the image as distorted lines. Standard methods to correct perspective distortion are based on the estimation of the vanishing points, which can fail if lens distortion is significant. In this paper, we introduce a new method which addresses both problems in a single framework. First we estimate a lens distortion model by extracting a collection of distorted lines in the image. These distorted lines are afterward rectified by means of the lens distortion model and used to estimate the vanishing points. Finally, the vanishing points are used to correct the perspective distortion. We present a variety of experiments to show the reliability of the proposed method.

Keywords: lens distortion, vanishing points, perspective correction
2010 MSC: 65D18, 68U10,

1. Introduction

Lens distortion is an optical aberration which causes straight lines in the scene to be projected onto the image as distorted lines. Most commercial lenses suffer, to some extent, from lens distortion aberration and, in the case of wide-angle lenses, the distortion can be very significant. Lens distortion is usually modeled by the radial transformation $D_{x_c,y_c,L} : \Omega \to R^2$, given by

$$D_{x_c,y_c,L}(x,y) = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + L(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}, \qquad (1)$$

where (x_c, y_c) represents the center of distortion, (x, y) is a point in the image domain Ω, $(\hat{x}, \hat{y})$ is the transformed (distortion-free) point, $r = \|(x, y) - (x_c, y_c)\|$, and $L(r)$ represents the transformation produced by the distortion model. Two types of radial lens distortion models are the most frequently applied in computer vision due to their excellent trade-off between accuracy and easy calculation: the polynomial model [1] and the division model [2]. The polynomial model is formulated as

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + .. + k_n r^{2n}, \qquad (2)$$

whereas the division model can be expressed as

$$L(r) = \frac{1}{1 + k_1 r^2 + k_2 r^4 + .. + k_n r^{2n}}. \qquad (3)$$

In this paper, we use two-parameter (k_1, k_2) models, due to their simplicity and accuracy. In previous works (see [3] and [4]), we have introduced a method to estimate two-parameter models which is able to cope with a high distortion level. The method is based on the estimation of distorted lines in the image and its outcomes are:

1. A collection of distorted lines $\{l_k\}_{k=1,...,N_{lines}}$ detected in the image.
2. For each distorted line l_k, a collection of edge points $\{(x_{k,j}, y_{k,j})\}_{j=1,...,N_k}$ belonging to the distorted line.
3. A two-parameter distortion model $\mathbf{u} = (x_c, y_c, k_1, k_2)$ computed by means of line rectification.

*Corresponding author
Email address: `lalvarez@ulpgc.es` (Luis Alvarez)
URL: `http://www2.dis.ulpgc.es/~lalvarez/` (Luis Alvarez)

These outcomes will be used in the method proposed in this paper.

Perspective distortion is caused by the position of the camera with respect to the objects of interest in the scene, especially when the objects are very close to the camera. For instance, it appears quite often in architectural environments when we take pictures of buildings. Perspective distortion causes parallel lines in the scene to be projected onto the picture plane as non-parallel lines converging on a vanishing point. Standard methods to correct perspective distortion are based on the estimation of the vanishing points. The approach we use in this paper to correct perspective distortion consists in rectifying the image by applying a homography which restores the parallelism of the lines. In this paper, we propose to use the lines obtained in the estimation of the lens distortion model to compute the vanishing points by means of a voting procedure. Once the vanishing points have been estimated, a homography is computed to correct the perspective distortion. Figure 1 illustrates the different steps of the proposed method.

Summary of the contributions of the paper

1. A method to correct the lens and perspective distortions in a single framework.
2. An algorithm for the estimation of the vanishing points based on the estimation of a relatively small number of long lines in the image.
3. A homography estimation to perform image rectification based on a camera motion simulation.
4. An (online) demonstration (`www.ctim.es/demo110`) using IPOL facilities (`www.ipol.im`) where the user can test the proposed technique.

The rest of the paper is organized as follows: In section 2, we present some related works. In section 3, the proposed method is described. Section 4 presents some experiments in a variety of real images. Finally, in section 5, we present some conclusions.

2. Related works

Radial distortion models are commonly used in the literature to model lens distortion aberrations. In the seminal paper [1], the author introduces the general shape of polynomial lens distortion models. Division models were initially proposed in [5], but have received special attention after the work presented in [6], where the author proposes to use one-parameter division models in the context of fundamental matrix estimation. The main advantage of the division models is the requirement of fewer terms than the polynomial models to cope with images showing severe distortion. Therefore, division models seem to be more suitable for wide-angle lenses (see a recent review on distortion models for wide-angle lenses in [7]). In [8], the authors propose a distortion model based on rational functions. In [9] and [10], a detailed analysis of high-order models is presented, including polynomial and division models. Moreover, in the case of real images, the authors evaluate the accuracy of lens distortion models using the SIFT method to extract and match points in an image of a highly textured pattern. Lens distortion aberration is also a main issue in camera calibration. For instance, [11] presents a study about the influence of the radial distortion in the estimation of the camera intrinsic parameters, and in [12] the authors introduce some algebraic methods for the simultaneous estimation of the lens distortion model and either the essential matrix or the fundamental matrix. In [13], the authors study the uncertainty in the estimates of the DLT-Lines camera calibration process with and without considering lens distortion models.

Most of the methods to estimate lens distortion models are based on line rectification. This approach is used, for instance, in [14], [15], [16] and [17]. In the seminal paper [18], the authors present a detailed study about the geometry, constraints and algorithmic implementation of the metric rectification of planes. They provide different ways to estimate a rectification homography H depending on the available information. In particular, they show that, in order to fully determine H, we need to know the vanishing lines l_∞ and two additional independent pieces of information, such as a known angle between lines or a known length ratio. If only the locations of two vanishing points are available, H is not determined in a unique way, and different approaches have been proposed in the literature to fix H. In [19], the authors study the problem of homography estimation when the images show a significant amount of lens distortion. In [20], an algorithm for automatic tilt correction based on a rotation about the principle axis of the camera is proposed and in [22], the authors present an algorithm for upright adjustment using segment detectors.

Regarding perspective correction, most methods require the computation of vanishing points. Segment detection is one of the basic tools to estimate vanishing points. In [23, 24] some segment detection methods are proposed. In [25] the authors use line segment detection to estimate vanishing points. In [26], a robust method to estimate vanishing points based on the Helmholtz principle is used, producing a low number of false alarms. In [27] and [28], a method based on an algorithm for point alignment detection and PClines dual spaces is proposed. In [29], the authors propose a method to detect the three mutually orthogonal directions in an architectural environment. A technique for the estimation of vanishing points based on some specific distributions of parallel lines is presented in [30]. In [31], the vanishing points are estimated with a method based on expectation-maximization in the projective space. In [32], a likelihood function that characterizes the plausibility of a point as vanishing point is introduced. In [33], vanishing points are computed by means of a RANSAC algorithm based on local segment detectors. In [34], the authors use perspective correction to perform feature matching in the context of aerial images.

In [21], the authors propose a method for automatic perspective correction based on Harris edge estimation. They estimate 2 vanishing points $p^i = (x_i, y_i, z_i)$, $i = 0, 1$ and they use the following algebraic approach to compute the rectification homography H_α

$$H_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_a & l_b & l_c \end{pmatrix} \quad (4)$$

where $(l_a, l_b, l_c) = p^1 \times p^2$. We can easily show that, with this definition of H_α and for any α, $H_\alpha p^i \in \vec{l}_\infty$ for $i = 0, 1$ (that is $\left(H_\alpha p^i\right)_z = 0$). Finally α is fixed in such a way that $H_\alpha p^i$ is parallel to $(0, 1, 0)^T$ for the vanishing point $H_0 p^i$ that is closest (in an angular sense) to $(0, 1, 0)^T$. That is, α is fixed to obtain a vertical alignment of the image.

3. Proposed method

The method proposed in this paper can be divided into the following steps:

Scheme of the proposed method

1. The method proposed in [3] is applied to obtain the distorted lines $\{l_k\}_{k=1,\ldots,N_{lines}}$, their associated points $\{(x_{k,j}, y_{k,j})\}_{j=1,\ldots,N_k}$, and the lens distortion model given by the vector $\mathbf{u} = (x_c, y_c, k_1, k_2)$. Once the lens distortion model has been estimated, the distortion can be corrected using the transformation $D_{x_c,y_c,L} : \Omega \to R^2$ defined by equation (1).
2. Using the lines and the distortion model obtained in the previous step, the vanishing points are computed as the intersections of some of the distortion-free lines that have been obtained after correcting the distortion.
3. Using the estimated vanishing points, a rectification homography H is fixed by means of a camera projection equation. In order to simplify the problem, we consider 2 potential scenarios: (i) the homography is estimated using 2 vanishing points corresponding to 2 orthogonal directions in the scene, or (ii) the homography is estimated using a single vanishing point.
4. The perspective and lens distortions are simultaneously corrected by applying the transformation $H \circ D_{x_c,y_c,L} : \Omega \to R^2$ to the image.

Figure 1 illustrates the different steps of the proposed method: in 1(a) we present an input image showing strong lens and perspective distortions. In 1(b) we show the collection of distorted lines obtained using the method in [3]. In 1(c) we show the lines that are associated to each vanishing point that has been obtained, and finally, in 1(d), we show the result of the correction of lens and perspective distortions using the transformation $H \circ D_{x_c,y_c,L} : \Omega \to R^2$.

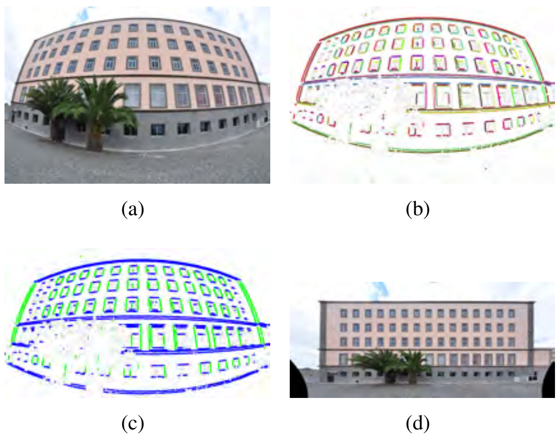

Figure 1: (a) Input image showing significant lens and perspective distortions. (b) Collection of distorted lines obtained using the method in [3]. (c) Lines associated to each vanishing point obtained by the estimation algorithm. (d) Result of the lens and perspective distortion correction using the transformation $H \circ D_{x_c,y_c,L} : \Omega \to R^2$.

Next we will explain the different steps of the proposed method in more detail.

3.1. Estimation of the distorted lines and the lens distortion model

In this stage, we use the method proposed in [3], where we introduced a technique for the automatic estimation of two-parameter radial distortion models considering polynomial as well as division models. The method first detects the edge points using the Canny edge detector. Afterward, the Hough transform enriched with a radial distortion parameter is applied to the edge points in order to extract the collection $\{l_k\}_{k=1,...,N_{lines}}$, which contains the longest distorted lines within the image and their associated edge points $\{(x_{k,j}, y_{k,j})\}_{j=1,...,N_k}$. From these lines, the first distortion parameter is estimated. The second distortion parameter is then initialized to zero, and the two-parameter model is embedded into an iterative nonlinear optimization process. This process aims at improving the estimation by minimizing a line rectification error. In order to measure the error associated to a model, we first need to determine the undistorted line corresponding to each distorted line. Afterward, we measure how far the undistorted edge points are from the corresponding undistorted line. Given a lens distortion model determined by the parameter vector $\mathbf{u}$, the undistorted line corresponding to each distorted line l_k is defined by minimizing the following distance error:

$$(\alpha_k^{\mathbf{u}}, d_k^{\mathbf{u}}) = arg\ \min_{\alpha,d} \sum_{j=1}^{N_k} \left(cos(\alpha)\hat{x}_{kj}^{\mathbf{u}} + sin(\alpha)\hat{y}_{kj}^{\mathbf{u}} + d \right)^2, \tag{5}$$

where $(\hat{x}_{kj}^{\mathbf{u}}, \hat{y}_{kj}^{\mathbf{u}})$ is obtained by the application of the lens distortion model given by $\mathbf{u}$ to the point $(x_{k,j}, y_{k,j})$. $(\alpha_k^{\mathbf{u}}, d_k^{\mathbf{u}})$ represents the orientation and distance to $(0, 0)$ for the straight line that minimizes the square distance from the corrected edge points to such line. This well-known minimization problem has a simple close-form solution (see for instance [35] for more details).

The line rectification error associated to a lens distortion model, given by the parameter vector $\mathbf{u}$, is defined by considering the collection of lines and their associated points in the following way:

$$E(\mathbf{u}) = \frac{\sum_{k=1}^{N_{lines}} \sum_{j=1}^{N_k} \left(cos(\alpha_k^{\mathbf{u}})\hat{x}_{kj}^{\mathbf{u}} + sin(\alpha_k^{\mathbf{u}})\hat{y}_{kj}^{\mathbf{u}} + d_k^{\mathbf{u}} \right)^2}{\sum_{k=1}^{N_{lines}} N_k}. \tag{6}$$

Then $\mathbf{u}$ is obtained by minimizing (6).

This optimization tries to reduce the distance from the edge points to the lines by adjusting two distortion parameters as well as the coordinates of the center of distortion. The optimization is performed using a Newton-Raphson-like algorithm including a damping parameter. If the estimation of the model is improved, some points which were not initially associated to a line could now match the line equation. For this reason, when a new lens distortion model is computed from the current primitives (the distorted lines with their associated points), the primitives are reestimated using the new distortion model in order to increase the number of points associated to them (the larger the number of line points, the more accurate the estimation of the lens distortion model). An important advantage of this method is that it can deal with high distortion levels. In Fig. 2, we summarize the different stages of the method.

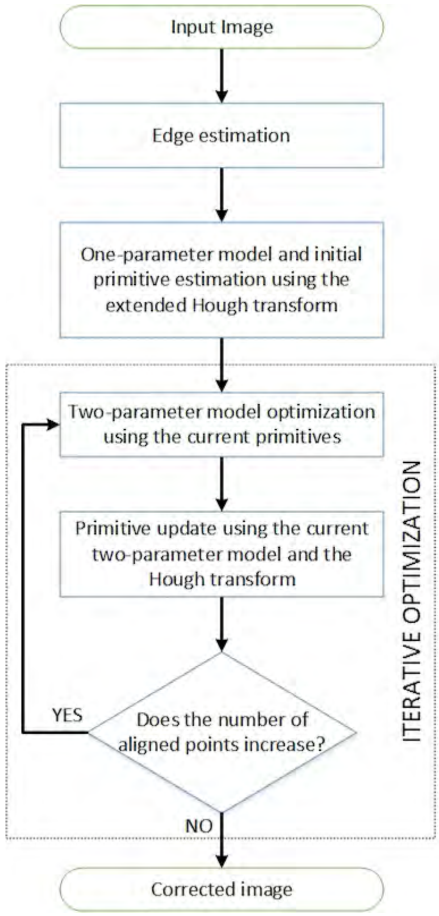

Figure 2: Flowchart of the distortion correction process. Primitives are given by the collection of detected distorted lines and their corresponding associated points. The algorithm aims at maximizing the total number of points associated to the collection of detected lines.

3.2. *Estimation of the vanishing points*

Let $\mathscr{L} = \{l_k\}_{k=1,..,N_{lines}}$ be the collection of lines obtained in the first step. The equation of each line is calculated after the correction of the lens distortion using equation (5) and can be expressed as

$$a_k x + b_k y + c_k = 0 \quad with \quad a_k^2 + b_k^2 = 1.$$

Furthermore, a collection of N_k edge points, used to obtain each line l_k, has been associated to the line.

We point out that, in this step, any technique for the estimation of vanishing points could be applied. Most of these techniques are based on local edge segment estimation, so that a large number of small segments are usually considered. However, we decided to design a new algorithm for the estimation of vanishing points that takes advantage of the structure of the collection of lines $\{l_k\}$. Indeed, $\{l_k\}$ consists of a relatively small number of long lines, and each line includes a large number of edge points. To detect the most significant vanishing point, we propose a voting procedure, in which the intersection of any pair of lines in the collection $\{l_k\}$ is considered as a candidate vanishing point. Obviously, we can implement this approach because we deal with a small number of long lines. If we dealt with a large number of small segments, it would be unfeasible to consider the intersection of any pair of segments as a candidate vanishing point. In such case, a RANSAC algorithm could be used to estimate potential candidates. However, we prefer to perform an exhaustive exploration of all potential candidates because we deal with a relatively small number of lines.

Given 2 lines l_k and l_m, their intersection in projective coordinates can be computed as the point $p^{k,m} = (p_x^{k,m}, p_y^{k,m}, p_z^{k,m})$ given by

$$p^{k,m} = (b_k c_m - b_m c_k, a_m c_k - a_k c_m, a_k b_m - a_m b_k).$$

As we work in projective coordinates, we normalize $p^{k,m}$ in such a way that $\| p^{k,m} \|_2 = 1$. A voting strategy is implemented as follows: for each point $p^{k,m}$, we define the following subset $L_{k,m}$ of the line collection $\mathscr{L}$

$$L_{k,m} = \{l_n \in \mathscr{L} : distance(l_n, p^{k,m}) < T, n \in \{1, .., N_{lines}\}\}, \tag{7}$$

where T is a threshold for the distance from the candidate vanishing point to the line. In projective coordinates, the distance from a line to a point in the infinity ($p_z^{k,m} = 0$) is not well defined. To avoid this problem, we define the distance$(l_n, p^{k,m})$ in the following way:

$$distance(l_n, p^{k,m}) = \frac{\left| a_n p_x^{k,m} + b_n p_y^{k,m} + c_n p_z^{k,m} \right|}{\left| p_z^{k,m} \right| + \epsilon}, \tag{8}$$

where $\epsilon > 0$ is introduced to avoid the division by 0 when $p_z^{k,m} = 0$. If $p_z^{k,m} \neq 0$, the above expression can be transformed into

$$distance(l_n, p^{k,m}) = \frac{\left| a_n \frac{p_x^{k,m}}{p_z^{k,m}} + b_n \frac{p_y^{k,m}}{p_z^{k,m}} + c_n \right|}{1 + \frac{\epsilon}{|p_z^{k,m}|}}, \tag{9}$$

where the numerator represents the usual Euclidean distance from the point to the line. Therefore, expression (9) tends to the usual Euclidean distance when $\epsilon \to 0^+$. Next, the following score is defined for each point $p^{k,m}$:

$$Score(p^{k,m}) = \sum_{n \in L_{k,m}} \frac{\ln(N_n)}{1 + distance(l_n, p^{k,m})}. \tag{10}$$

This measure is designed in such a way that the contribution of each line depends on the distance from the line to the point, as well as on the number of points associated to the line (the larger the line, the higher its influence on the score). The influence of the number of points N_n associated to a line is included in expression (10) by using the term $ln(N_n)$. Of course, we could use many types of expressions to include such influence, but we decided to use $ln(N_n)$ because, given the profile of function $ln(.)$, it avoids causing an extremely strong effect of N_n on the evaluation of expression (10).

Once the scores $Score(p^{k,m})$ have been estimated, the two most significant vanishing points in the image are defined as follows:

$$\begin{aligned} p^0 &= \quad \arg\max_{k,m} \; Score(k,m) \\ p^1 &= \arg\max_{k,m:|(p^{k,m},p^0)|<0.95} Score(k,m) \end{aligned}$$

where $(p^{k,m}, p^0)$ is the dot product of $p^{k,m}$ and p^0. Since the vectors are normalized, $(p^{k,m}, p^0)$ represents the cosine of the angle between both vectors. The condition $|(p^{k,m}, p^0)| < 0.95$ is introduced to avoid selecting p^1 too close to p^0.

After the estimation of the vanishing points p^0 and p^1, we recompute their coordinates taking into account all the lines that voted for these points in the voting procedure. We use an algebraic approach to recompute p^i. First we notice that

$$(a_n x + b_n y + c_n z)^2 = (x, y, z) \begin{pmatrix} a_n^2 & a_n b_n & a_n c_n \\ a_n b_n & b_n^2 & b_n c_n \\ a_n c_n & b_n c_n & c_n^2 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

Therefore, we propose to recompute p^i as the eigenvector associated to the lowest eigenvalue of the matrix

$$\sum_{n \in L_{k_i,m_i}} \ln(N_n) \begin{pmatrix} a_n^2 & a_n b_n & a_n c_n \\ a_n b_n & b_n^2 & b_n c_n \\ a_n c_n & b_n c_n & c_n^2 \end{pmatrix}, \tag{11}$$

where the influence of each line is weighted according to the number of associated points using $ln(N_n)$.

3.3. Homography estimation from the vanishing points

First we consider the case in which H is computed from 2 vanishing points $p^i = (x_i, y_i, z_i)$, $i = 0, 1$, corresponding to 2 orthogonal directions in the scene. It is known (see for instance [18]) that the information provided by 2 vanishing points is not enough to fully determine the homography H. Therefore, we introduce some plausible simplifications to compute H. Let (x_c, y_c) be the center of the image. We build a homography H satisfying

$$\begin{cases} H(0,0,1) = s_c(x_c, y_c, 1)^T \\ H(1,0,0) = s_0(x_0, y_0, z_0)^T \\ H(0,1,0) = s_1(x_1, y_1, z_1)^T \end{cases},$$

where $s_c, s_0, s_1 \neq 0$. This means that the reference system given by the center of the image and the vanishing points is transformed into the usual canonical reference system. Moreover, we will assume that the homography H corresponds to a plane-to-plane transformation obtained by a camera projection equation. We assume that the reference plane in $3D$ is the plane $Z = 0$. Therefore, for any point $(X, Y, 0, 1)$ on the reference plane, the general camera projection equation can be expressed as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & \gamma & x_c \\ 0 & 1 & y_c \\ 0 & 0 & \frac{1}{f} \end{pmatrix} \begin{pmatrix} R_{00} & R_{01} & t_x \\ R_{10} & R_{11} & t_y \\ R_{20} & R_{21} & t_z \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix},$$

where the camera intrinsic and extrinsic parameters are used. For the sake of simplicity, we assume that the pixel aspect ratio of the camera CCD is equal to 1 (square pixels). We observe that

$$\begin{pmatrix} 1 & \gamma & x_c \\ 0 & 1 & y_c \\ 0 & 0 & \frac{1}{f} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f} \end{pmatrix} \begin{pmatrix} 1 & \gamma & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} R_{00} & R_{01} & t_x \\ R_{10} & R_{11} & t_y \\ R_{20} & R_{21} & t_z \end{pmatrix} = t_z \begin{pmatrix} R_{00} & R_{01} & \frac{t_x}{t_z} \\ R_{10} & R_{11} & \frac{t_y}{t_z} \\ R_{20} & R_{21} & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{t_z} & 0 & 0 \\ 0 & \frac{1}{t_z} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Therefore, f is an isotropic scaling factor in the image plane and $1/t_z$ is an isotropic scaling factor in the reference plane. We fix both scaling factors to 1, that is, $f = 1$ and $t_z = 1$, in order to keep the same scaling factors for the reference systems in both planes. That is to say, we do not want the homography to introduce any artificial isotropic scaling factors. Consequently, the general form of the homography we deal with is

$$H = \begin{pmatrix} 1 & \gamma & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{00} & R_{01} & t_x \\ R_{10} & R_{11} & t_y \\ R_{20} & R_{21} & 1 \end{pmatrix}.$$

Next we observe that, if $H(0,0,1) = s_c(x_c, y_c, 1)^T$, then

$$H \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} t_x + \gamma t_y + x_c t_z \\ t_y + y_c t_z \\ t_z \end{pmatrix} = s_c \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix},$$

and, consequently, $t_x = t_y = 0$. Moreover, if $H(1,0,0) = s_0(x_0, y_0, z_0)^T$ and $H(0,1,0) = s_1(x_1, y_1, z_1)^T$, then

$$\begin{pmatrix} R_{00} \\ R_{10} \\ R_{20} \end{pmatrix} = s_0 \begin{pmatrix} x_0 - \gamma y_0 - z_0 (x_c - \gamma y_c) \\ y_0 - z_0 y_c \\ z_0 \end{pmatrix}$$

$$\begin{pmatrix} R_{01} \\ R_{11} \\ R_{21} \end{pmatrix} = s_1 \begin{pmatrix} x_1 - \gamma y_1 - z_1 (x_c - \gamma y_c) \\ y_1 - z_1 y_c \\ z_1 \end{pmatrix}.$$

On the other hand, as R is an orthogonal matrix, then

$$\begin{pmatrix} x_0 - \gamma y_0 - z_0 (x_c - \gamma y_c) \\ y_0 - z_0 y_c \\ z_0 \end{pmatrix}^T \begin{pmatrix} x_1 - \gamma y_1 - z_1 (x_c - \gamma y_c) \\ y_1 - z_1 y_c \\ z_1 \end{pmatrix} = 0.$$

This equation yields a two-degree polynomial in γ and, therefore, γ can be estimated as the lowest real root of such polynomial. Once γ is estimated, the homography H is fully determined.

The second option we considered consists in computing the homography from a single vanishing point p^0. In this case, in order to apply the above procedure to estimate the homography H, we compute p^1 from p^0 and (x_c, y_c) as $p^1 = (-(p^0_y - p^0_z y_c), p^0_x - p^0_z x_c, 0)$. In this way, p^1 is a point in the infinity and its orientation is orthogonal to the vector from (x_c, y_c) to p^0. This assumption is valid if, for instance, the camera motion is just a rotation about the vertical or horizontal directions. Of course, this is a strong simplification and there are many other options to define H from a single vanishing point. The results will be accurate only if the camera motion follows approximately this assumption.

3.4. Perspective and lens distortion correction

The perspective and lens distortions are simultaneously corrected by applying the transformation $H \circ D_{x_c, y_c, L} : \Omega \to R^2$. In practice, for each pixel $\mathbf{x}$ of the output image, we obtain the corresponding one in the original image by computing $D^{-1}_{x_c, y_c, L} \circ H^{-1}(\mathbf{x})$, and then determine the color of $\mathbf{x}$ by bilinear interpolation in the original image.

4. Experimental results

In Fig. 3, we illustrate the different options we have for the correction of the images: (i) correction of lens distortion without perspective correction, (ii) correction of lens and perspective distortions using a single vanishing point, and (iii) correction of lens and perspective distortions using two vanishing points.

The computational cost of the different steps to process this image (image size: 1072 × 712 pixels, detected lines: 78, detected number of points associated to the lines: 9278) on an Intel Core i7@2.67GHz with the Ubuntu 14.04 LTS operative system is:

- Estimation of the distorted lines and the lens distortion model (subsection 3.1): 5.692043 s.
- Estimation of the vanishing points (subsection 3.2): 0.008028568 s.
- Homography estimation (subsection 3.3): 0.0036781s
- Correction of perspective and lens distortions (subsection 3.4): 0.2639092 s.

Therefore, the global computational cost is around 6 seconds and most of the time is devoted to the estimation of the distorted lines and the lens distortion model. For the other images used in the experiments of this section, the computational cost is similar. We point out that we do not focus on computational cost optimization in this work and there are several options for the improvement of this aspect.

In Fig. 4 and 5, we show some experiments in a variety of real images using lens and perspective distortion correction with two vanishing points. In Fig. 4, we focus on architectural environments, and we present the results obtained by the proposed method for different pictures of buildings showing a significant optical (lens) and perspective distortion. We observe that the proposed method performs well in these images and is able to cope with images showing a high optical distortion. In Fig. 5, we present some experiments on pictures of a calibration pattern showing high perspective and optical distortions, a picture of a sport court, and a painting. The picture of the sport court has been acquired with a GoPro® Hero3 camera, whereas the rest of the pictures used in the experiments have been taken with a Tokina DX 11-16mm lens mounted in a D90 Nikon camera using different settings of the lens focal length, which provides a variety of lens distortion aberrations.

We point out that, when the estimated distorted lines lie on a plane in the 3D scene, the homography H transforms such 3D plane into the image plane. Consequently, our method is expected to work properly for points lying on such plane. However, some additional perspective distortion is expected when we apply our method to points which are far from such plane. We can observe this behavior in some of the experiments we present. In particular, the results for the picture of the sport court we present in Fig. 5 show a strong perspective distortion of the players because the homography is computed from lines lying on the floor of the sport court.

We observe that our method does not provide a metric rectification of the scene. This is due to the fact that the homography H is not fully determined by 2 vanishing points and, therefore, the estimated H cannot be used to compute metric distances in the scene. For this reason, we evaluate the results by visual inspection.

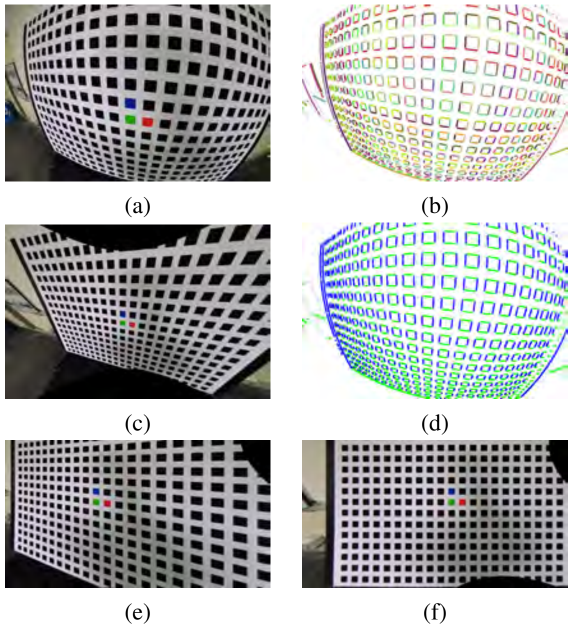

Figure 3: (a) Original image. (b) Distorted lines obtained using the method proposed in [3]. (c) Lens distortion correction without perspective correction. (d) Lines associated to the vanishing points. (e) Lens and perspective distortion correction using only 1 vanishing point (vertical alignment). (f) Lens and perspective distortion correction using 2 vanishing points.

As mentioned above, we can use any method for the estimation of vanishing points in our algorithm, but we have designed a method which exploits the characteristics of the collection of lines that we use (a small number of long lines). For comparison purposes, we have implemented a method inspired by the one proposed in [18], which is based on the analysis of the line orientation histogram. Next we summarize the main steps of this method:

Scheme of the method for the estimation of vanishing points based on the line orientation histogram

1. Step 1 : Computation of the distorted lines in the image and the lens distortion model using the method proposed in [3].

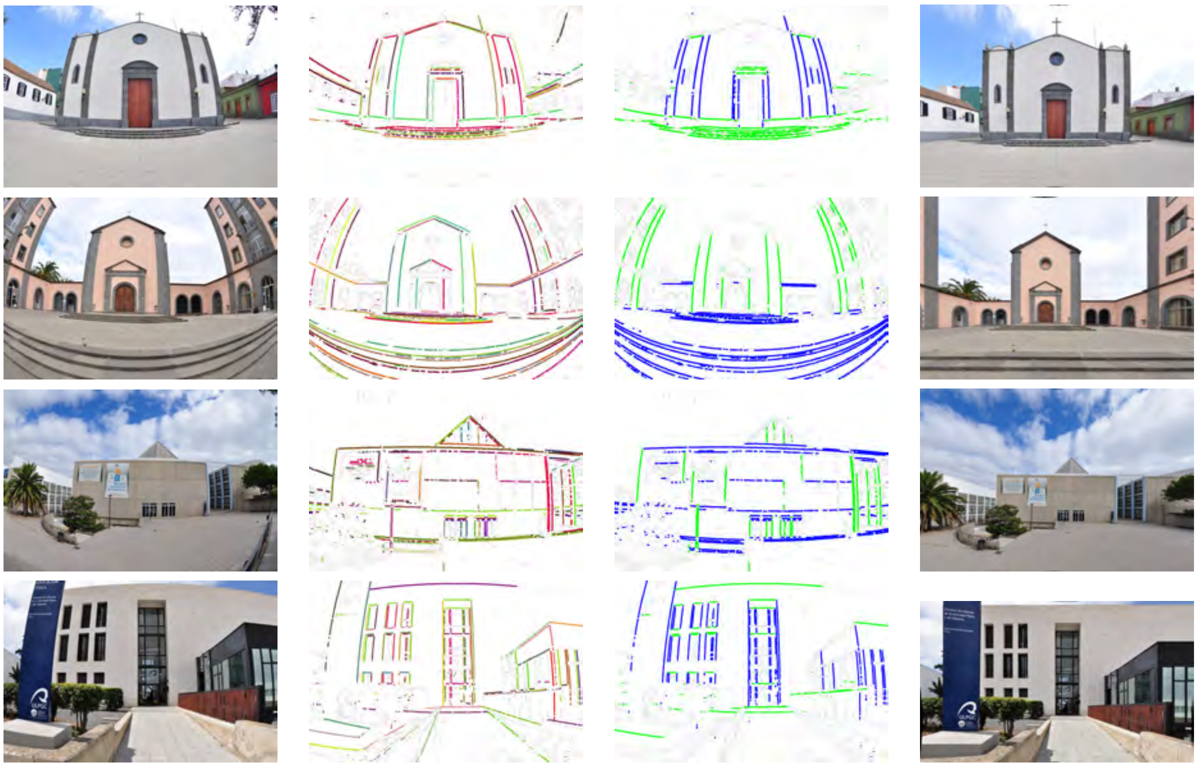

Figure 4: Illustration of the results of the proposed method for some real images in architectural environments using 2 vanishing points. For each experiment, we present the original image, the distorted lines estimated to correct the lens distortion, the lines associated to the vanishing points, and the image obtained by correcting lens and perspective distortions simultaneously. (Online figure in color).

2. Step 2 : Computation of the π periodic frequency histogram of line orientation in degrees (lines are previously corrected using the lens distortion model). The frequency is weighted by the number of points associated to each line.
3. Step 3 : Convolution with a Gaussian kernel to regularize the histogram. In the experiments we fix the standard deviation of the Gaussian to 1.
4. Step 4 : Estimation of 2 intervals in the regularized histogram corresponding to the dominant directions.
5. Step 5 : Estimation of the vanishing points as the intersections of the lines with orientations included in the intervals associated to the dominant directions.

We have experienced that this method works properly when the line orientation histogram is bimodal and the lines estimated in the scene with similar orientations are parallel. We illustrate two experiments in Fig. 6 where this method fails. For each image, we show the regularized line orientation histogram, where the extrema of the intervals associated to the dominant directions are marked using vertical lines. We also show (using different colors for each dominant direction) the lines with orientations included in such intervals, as well as the image after correcting the lens and perspective distortions using the estimated vanishing points. Comparing the results with the ones obtained for the same images in Fig. 5 using the method we propose, we observe that, for the first image, the perspective correction is inaccurate because the orientation of some lines (on the left of the calibration pattern) is similar to the orientation of the vertical lines of the calibration pattern, but they are not parallel to them. Since these lines are considered when computing the vanishing points, an inaccurate result for the estimation of the vanishing points is produced and, consequently, the perspective correction is not satisfactory. In the second image, the problem is that the line orientation histogram is not bimodal and then the number of lines associated to the intervals is not enough to properly compute the vanishing points.

When the image shows a significant lens distortion, vanishing point estimation techniques which do not take the lens distortion into account fail because the lens distortion can strongly affect the line estimation used to

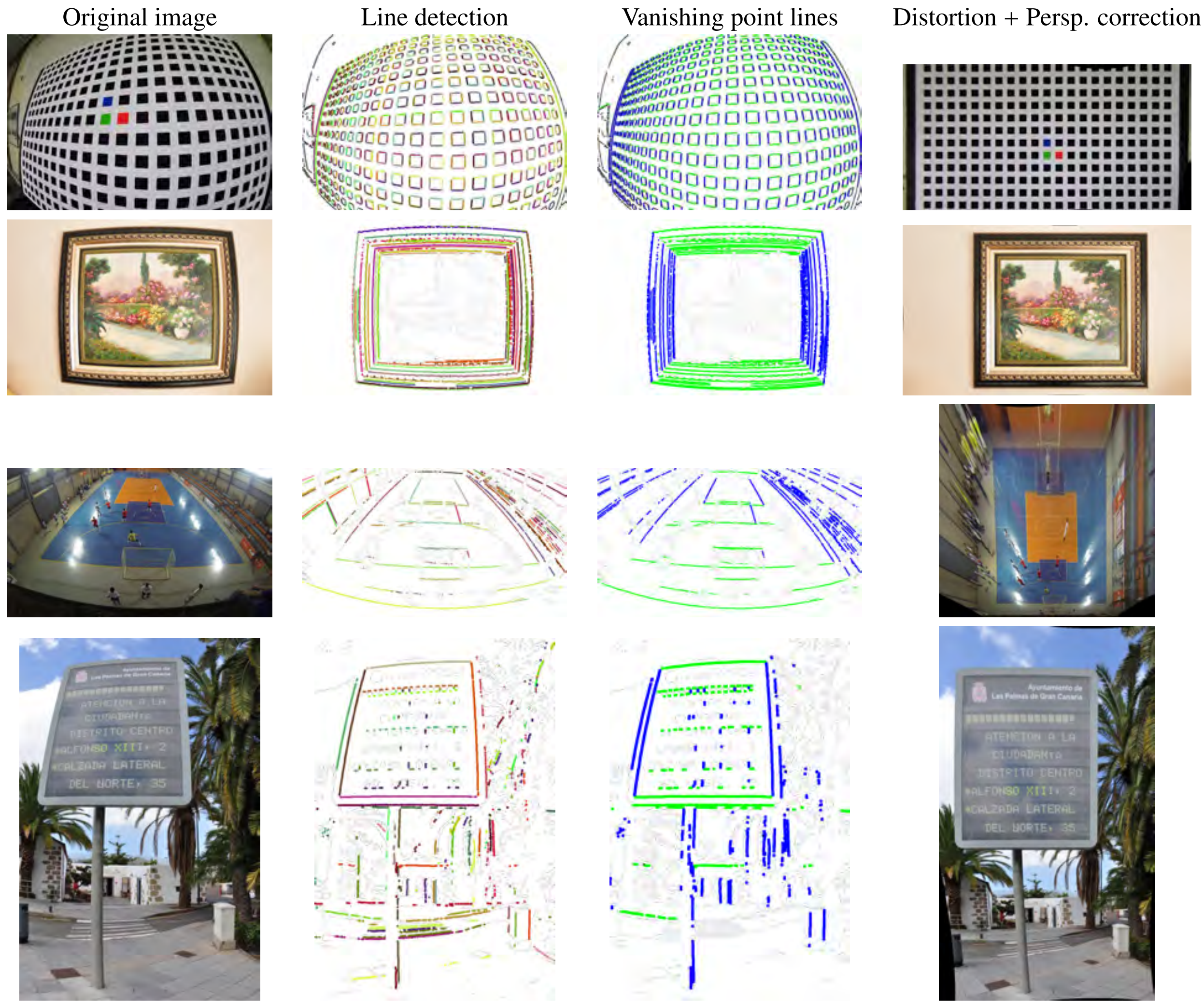

Figure 5: Illustration of the results of the proposed method for some real images using 2 vanishing points. For each experiment, we present the original image, the distorted lines estimated to correct the lens distortion, the lines associated to the vanishing points, and the image obtained by correcting lens and perspective distortions simultaneously. (Online figure in color).

compute the vanishing points. We illustrate this behavior in Fig. 7, where we compare the results obtained using the method we propose with the method introduced in [28], where the lens distortion is not taken into account. We observe that, due to the significant amount of lens distortion, the method is not able to compute the lines associated to the vanishing points in a satisfactory way. As the vanishing points are not properly computed, the perspective correction we obtain is also wrong.

In fig. 8, we present a comparison of the results obtained using the rectification homography we propose and the algebraic one proposed in [21], given by equation (4). The method proposed in [21] does not consider the lens distortion. For this reason, in order to perform a fair comparison of both methods when estimating the homographies, we replace the homography we obtain with the algebraic one obtained by [21] in the final step of our algorithm. We can observe that the algebraic method proposed in [21] produces a vertical alignment of the image, but not a horizontal alignment. On the other hand, we can observe that the method does not preserve the proportions of the objects in the image. For instance, in the image of the calibration pattern, the original squares become elongated rectangles. This kind of limitations of the algebraic method proposed in [21] supports the idea that a rectification homography estimation based on a realistic simulation of a camera motion, like the one we propose in this paper, provides

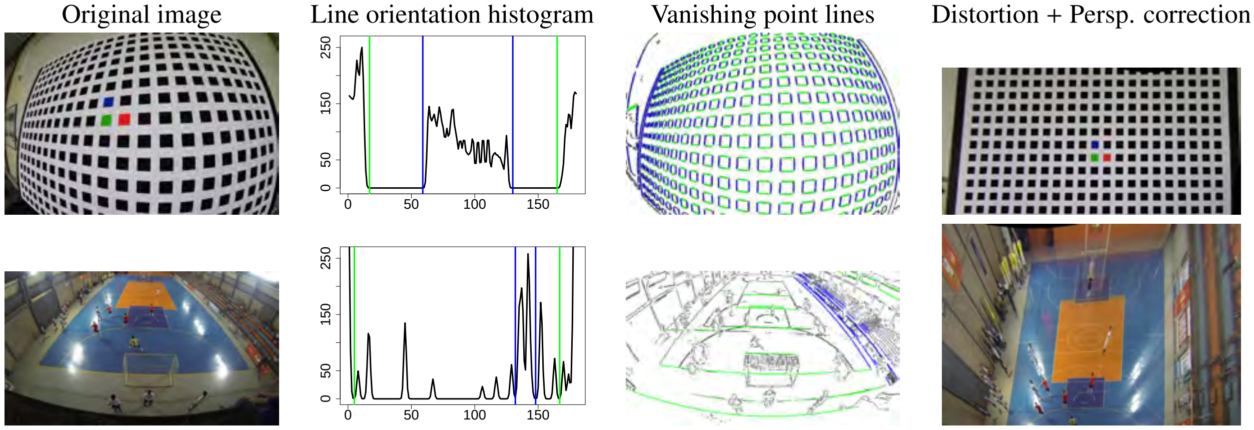

Figure 6: Illustration of the results obtained using the method based on line orientation histogram to compute the vanishing points. (Online figure in color).

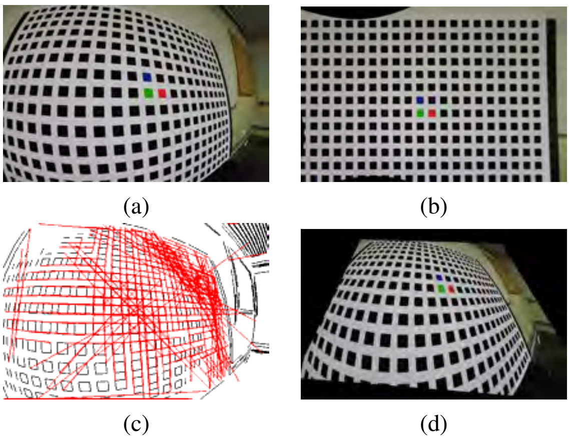

Figure 7: (a) Original image. (b) Lens and perspective distortion correction using the proposed method. (c) Lines associated to the vanishing points using the method proposed in [28]. (d) Perspective distortion correction using the vanishing points obtained using the method proposed in [28].

	p^0	p^1
$T = 1$	(0.15, 0.99, 0.00035)	(-0.99, 0.098,0.00017)
$T = 5$	(0.13, 0.99, 0.00031)	(-0.99, 0.11,0.00019)
$T = 10$	(0.12, 0.99, 0.00029)	(-0.99, 0.089,0.00016)

Table 1: Vanishing points in projective coordinates obtained for the image of Fig. 9 for different values of the threshold T for the distance between the vanishing point and the lines, which is used in the voting procedure (see expression (7)).

better results.

4.1. (Online) demonstration

Using the IPOL Journal of Image Processing On Line facilities (`www.ipol.im`), we have implemented an (online) demonstration of the proposed method that can be found at `www.ctim.es/demo110`. All the experiments shown in this paper have been performed using this (online) demonstration with the default parameters. There is a collection of parameters related to the technique introduced in [3] to estimate the collection of distorted lines and the distortion model, and we refer to such paper for a complete explanation of these parameters. In any case, as can be checked in the (online) demonstration, the default parameters work properly for most images, so that, in general, it is not necessary to adjust the parameters for each image.

In the part of the (online) demonstration algorithm concerning perspective correction, the only numerical parameter we use is the threshold T for the distance between the vanishing point and the lines, which is used in the voting procedure (see expression (7)). In Fig. 9, we illustrate the influence of such parameter using an image from Fig. 5. This image is challenging because the tree leaves in the scene make the algorithm estimate a large number of short lines that could alter the results (these lines can be observed in the results shown in Fig. 5). In Fig. 9, we observe that the larger the value of T, the larger the number of lines used to estimate each vanishing point. In any case, the method is quite robust against the choice of this parameter, and the estimated vanishing points are similar. In table 1, we show the vanishing points in projective coordinates for different values of the parameter T.

In order to increase the flexibility of the algorithm, we also allow the user to force the use of a single vanishing point in the horizontal or vertical direction. This is especially useful when we deal with images with only

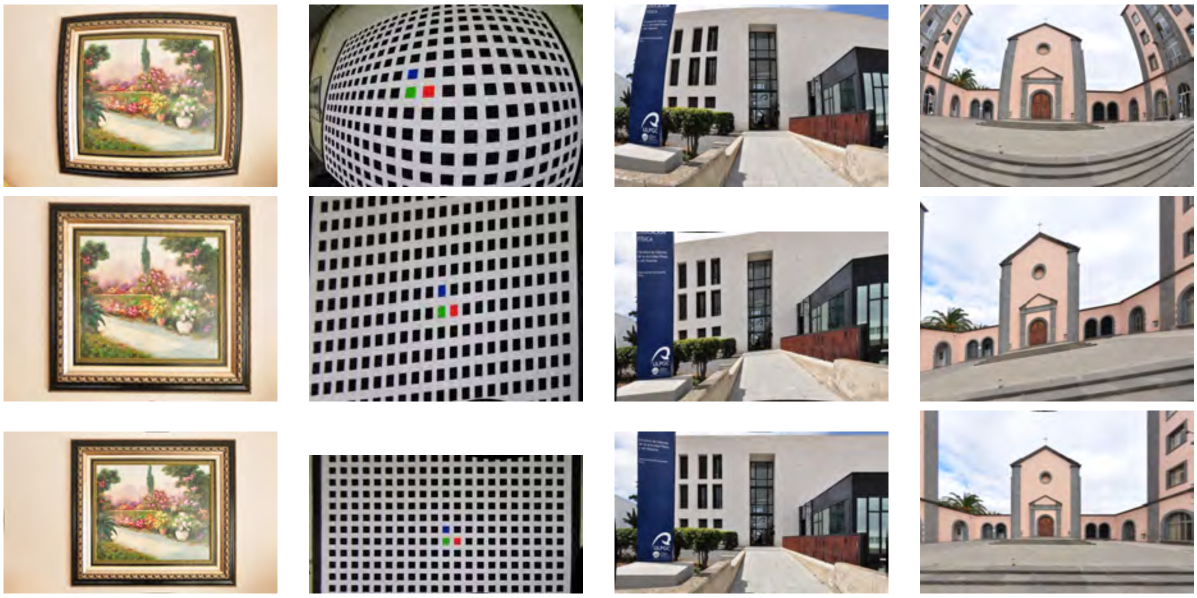

Figure 8: Comparison of the results of the proposed method for some real images using 2 vanishing points and the ones obtained using the rectification homography proposed in [21] given by equation (4). In the first line we present the original image, in the second line the results obtained using the method proposed in [21], and in the third line, the results obtained with the proposed method. (Online figure in color).

1 visible vanishing point, or when we just want to align the image in one direction (for instance, when we want to correct a vertical misalignment of the image).

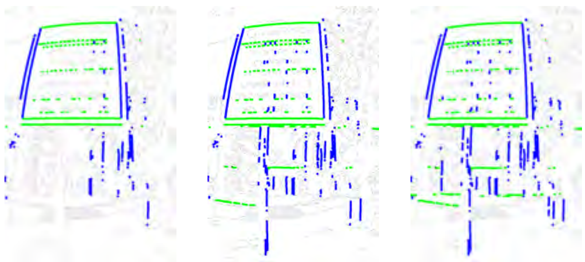

Figure 9: Illustration of the influence of threshold T for the distance between the vanishing point and the lines in expression (7). From left to right we show the lines associated to the vanishing points obtained using $T = 1, 5, 10$, respectively.

5. Conclusions

In this paper, we propose a new method for the simultaneous correction of the optical (lens) and perspective distortions. In the case of images showing a significant lens distortion, dealing with both distortions simultaneously is particularly important. If both types of distortion are not considered, the usual methods to correct perspective distortion can fail, as they are not able to cope with highly distorted lines. We first use the method introduced in [3] to compute the distorted lines and the lens distortion model. Then we use the collection of distorted lines which were obtained, which consists of a small number of long lines, to apply a voting procedure for the estimation of the vanishing points. We show that this method to estimate the vanishing points outperforms the method proposed in [18], based on line orientation histogram. Afterward, we estimate a homography H to correct the perspective distortion using the general camera projection equation.

In order to fully determine H from 2 vanishing points, we remove the isotropic scaling factors in the homography estimation. We also extend the method to the case of using a single vanishing point. Finally, we simultaneously correct the lens and perspective distortions by applying a single transformation given by the composition of the lens distortion model and the homography H. We show the performance of the proposed method in a variety of experiments. We have built an (online) demonstration (`www.ctim.es/demo110`) where all the presented experiments can be reproduced.

Acknowledgment

We thank Dr. José Lezama for providing us with the vanishing points obtained with the method introduced in [28] that we use in the experiment illustrated in Fig. 7.

References

[1] D. Brown, Close-range camera calibration, Photogrammetric Engineering 37 (8) (1971) 855–866.

[2] F. Bukhari, M. Dailey, Automatic radial distortion estimation from a single image, Journal of Mathematical Imaging and Vision 45 (1) (2012) 31–45.

[3] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, L. Alvarez, Invertibility and estimation of two-parameter polynomial and division lens distortion models, SIAM Journal on Imaging Sciences 8 (3) (2015) 1574–1606.

[4] D. Santana-Cedrés, L. Gomez, M. Alemán-Flores, A. Salgado, J. Esclarín, L. Mazorra, L. Alvarez, An iterative optimization algorithm for lens distortion correction using two-parameter models (companion paper), IPOL Image Processing On Line.

[5] R. Lenz, Linsenfehlerkorrigierte Eichung von Halbleiterkameras mit Standardobjektiven für hochgenaue 3D - Messungen in Echtzeit, in: E. Paulus (Ed.), Mustererkennung 1987, Vol. 149 of Informatik-Fachberichte, Springer Berlin Heidelberg, 1987, pp. 212–216.

[6] A. W. Fitzgibbon, Simultaneous linear estimation of multiple view geometry and lens distortion, In: Proc. IEEE International Conference on Computer Vision and Pattern Recognition (2001) 125–132.

[7] C. Hughes, M. Glavin, E. Jones, P. Denny, Review of geometric distortion compensation in fish-eye cameras, in: Signals and Systems Conference, 208. (ISSC 2008). IET Irish, Galway, Ireland, 2008, pp. 162–167.

[8] D. Claus, A. W. Fitzgibbon, A rational function lens distortion model for general cameras, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 213–219.

[9] Z. Tang, R. Grompone, P. Monasse, J. M. Morel, Self-consistency and universality of camera distortion models, In hal-00739516, version 1 (2012) 1–8.

[10] R. Grompone, P. Monasse, J. M. Morel, Z. Tang, Towards high-precision lens distortion correction, In: Proc. of the 17th IEEE International Conference on Image Processing (ICP) (2010) 4237–4240.

[11] B. Tordoff, D. W. Murray, The impact of radial distortion on the self-calibration of rotating cameras, Computer Vision and Image Understanding 96 (1) (2004) 17 – 34.

[12] Z. Kukelova, M. Byrod, K. Josephson, T. Pajdla, K. Astrom, Fast and robust numerical solutions to minimal problems for cameras with radial distortion, Computer Vision and Image Understanding 114 (2) (2010) 234 – 244.

[13] R. Galego, A. Ortega, R. Ferreira, A. Bernardino, J. Andrade-Cetto, J. Gaspar, Uncertainty analysis of the DLT-lines calibration algorithm for cameras with radial distortion, Computer Vision and Image Understanding 140 (2015) 115 – 126.

[14] F. Devernay, O. Faugeras, Straight lines have to be straight, Machine Vision and Applications 13 (1) (2001) 14–24.

[15] L. Alvarez, L. Gomez, R. Sendra, An algebraic approach to lens distortion by line rectification, Journal of Mathematical Imaging and Vision 39 (1) (2008) 36–50.

[16] L. Alvarez, L. Gomez, J. R. Sendra, Algebraic Lens Distortion Model Estimation, Image Processing On Line 1 (2010) 1–10.

[17] L. Alvarez, L. Gomez, R. Sendra, Accurate depth dependent lens distortion models: An application to planar view scenarios, Journal of Mathematical Imaging and Vision 39 (1) (2011) 75–85.

[18] D. Liebowitz, A. Zisserman, Metric rectification for perspective images of planes, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '98, IEEE Computer Society, Washington, DC, USA, 1998, pp. 482–487.

[19] Z. Kukelova, J. Helle, M. Bujnak, T. Pajdla, Radial distortion homography, in: Computer Vision and Pattern Recognition, CVPR 2015. IEEE Computer Society Conference on, Vol. 1, IEEE, 2015, pp. 639–647.

[20] A. C. Gallagher, Using vanishing points to correct camera rotation in images, in: Proceedings of the 2nd Canadian Conference on Computer and Robot Vision, CRV '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 460–467.

[21] K. Chaudhury, S. DiVerdi, S. Ioffe, Auto-rectification of user photos, in: Image Processing (ICIP), 2014 IEEE International Conference on, 2014.

[22] H. Lee, E. Shechtman, J. Wang, S. Lee, Automatic upright adjustment of photographs with robust camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (5) (2014) 833–844.

[23] Z. Xu, B.-S. Shin, R. Klette, A statistical method for line segment detection, Computer Vision and Image Understanding 138 (2015) 61 – 73.

[24] R. Grompone, J. Jakubowicz, J. Morel, G. Randall, LSD: a line segment detector, IPOL : Image Processing On Line 2 (2012) 35–55.

[25] F. A. Andaló, G. Taubin, S. Goldenstein, Efficient height measurements in single images based on the detection of vanishing points, Computer Vision and Image Understanding 138 (2015) 51 – 60.

[26] A. Almansa, A. Desolneux, S. Vamech, Vanishing point detection without any a priori information, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (4) (2003) 502–507.

[27] J. Lezama, R. Gioi, G. Randall, J.-M. Morel, Finding vanishing points via point alignments in image primal and dual domains, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 509–515.

[28] J. Lezama, G. Randall, R. G. von Gioi, Vanishing point detection in urban scenes using point alignments, IPOL Image Processing On Line (submitted) %urlhttp://www.ipol.im/pub/pre/148/.

[29] C. Rother, A new approach to vanishing point detection in architectural environments, Image and Vision Computing 20 (9) (2002) 647–655.

[30] F. Schaffalitzky, A. Zisserman, Planar grouping for automatic detection of vanishing lines and points, Image and Vision Computing 18 (2000) 647–658.

[31] M. Nieto, L. Salgado, Simultaneous estimation of vanishing points and their converging lines using the EM algorithm, Pattern Recogn. Lett. 32 (14) (2011) 1691–1700.

[32] H. Baltzakis, P. Trahanias, The VPLF method for vanishing point computation, Image and Vision Computing 19 (6) (2001) 393 – 400.

[33] L. I. Hai-Feng, L. I. U. Jing-Tai, Optimal vanishing point estimation with performance evaluation, Acta Automatica Sinica 38 (2) (2012) 213 – 220.

[34] H. Kume, T. Sato, N. Yokoya, Bundle adjustment using aerial images with two-stage geometric verification, Computer Vision and Image Understanding 138 (C) (2015) 74–84.

[35] O. Faugeras, Three-dimensional computer vision, MIT Press, 1993.