



Published in Image Processing On Line on 2014–11–20.  
Submitted on 2013–11–01, accepted on 2014–10–20.  
ISSN 2105–1232 © 2014 IPOL & the authors CC–BY–NC–SA  
This article is available online with supplementary materials,  
software, datasets and online demo at  
<https://doi.org/10.5201/ipol.2014.106>

# Automatic Lens Distortion Correction Using One-Parameter Division Models

Miguel Alemán-Flores<sup>1</sup>, Luis Alvarez<sup>1</sup>, Luis Gomez<sup>2</sup>, Daniel Santana-Cedrés<sup>1</sup>

<sup>1</sup> CTIM (Centro de Tecnologías de la Imagen),  
Departamento de Informática y Sistemas  
Universidad de Las Palmas de Gran Canaria, Spain  
([maleman,lalvarez,dsantana}@ctim.es](mailto:{maleman,lalvarez,dsantana}@ctim.es))

<sup>2</sup> CTIM (Centro de Tecnologías de la Imagen),  
Departamento de Ingeniería Electrónica y Automática  
Universidad de Las Palmas de Gran Canaria, Spain  
([lgomez@ctim.es](mailto:lgomez@ctim.es))

*Communicated by* Rafael Grompone von Gioi

*Demo edited by* Agustín Salgado

## Abstract

We present a method to automatically correct the radial distortion caused by wide-angle lenses using the distorted lines generated by the projection of 3D straight lines onto the image. Lens distortion is estimated by the division model using one parameter, which allows to state the problem into the Hough transform scheme by adding a distortion parameter to better extract straight lines from the image. This paper describes an algorithm which applies this technique, providing all the details of the design of an improved Hough transform. We perform experiments using calibration patterns and real scenes showing a strong distortion to illustrate the performance of the proposed method.

## Source Code

The source code, the code documentation, and the online demo are accessible at the IPOL web page of this article<sup>1</sup>. In this page, an implementation is available for download. Compilation and usage instructions are included in the README.txt file of the archive.

**Keywords:** lens distortion; wide-angle lens; Hough transform; line detection

<sup>1</sup><https://doi.org/10.5201/ipol.2014.106>

# 1 Introduction

Due to radial and decentering distortion, the image of a photographed line will not be a straight line once projected onto the image plane. The correction of such distortions (especially the radial distortion, which is considered to be the most relevant one) becomes a major issue in camera calibration. Several methods have been proposed to correct radial distortion for lenses causing *moderate* distortion, such as the radial model by Brown [5], the rational model by Claus et al. [8] or the division model by Lenz [15].

Distortion correction is also closely related to professional photography fields such as close-range photography or wide-angle photography using wide-angle lenses. The use of a wide-angle lens makes it possible to acquire a large area of the surrounding space (up to 180 degrees) in a single photograph. Other Computer Vision tasks also use wide-angle lenses for purposes such as surveillance, real-time tracking, or simply for aesthetic purposes.

Radial distortion is usually modeled by means of the radial model introduced by Brown [5]

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

where  $(x, y)$  is the original (distorted) point,  $(\hat{x}, \hat{y})$  is the corrected (undistorted) point,  $(x_c, y_c)$  is the center of the camera distortion model,  $L(r)$  is the function which defines the shape of the distortion model and  $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ . According to the choice of function  $L(r)$ , there exist two widely accepted types of lens distortion models: the polynomial model and the division model (see a recent review on distortion models for wide-angle lenses in Hughes et al. [13]).

The polynomial used in the simple radial distortion model is

$$L(r) = 1 + k_1 r^2 + k_2 r^4 + \dots, \quad (2)$$

where the set  $\mathbf{k} = (k_1, \dots, k_{N_k})^T$  contains the distortion parameters estimated from image measurements, usually by means of non-linear optimization techniques (see Alvarez et al. [2]).

The division model was initially proposed by Lenz [15], but has received special attention after the more recent research by Fitzgibbon [12]. It is formulated as

$$L(r) = \frac{1}{1 + k_1 r^2 + k_2 r^4 + \dots}. \quad (3)$$

The main advantage of the division model is the requirement of fewer terms than the polynomial model for the case of severe distortion. Therefore, the division model seems to be more adequate for wide-angle lenses. Additionally, when using only one distortion parameter, its inversion is simpler, since it requires finding the roots of a second degree polynomial instead of a third degree polynomial. In fact, a single parameter version of the division model is normally used.

Most algorithms estimate the radial distortion parameters by considering that 3D lines in the image must be projected onto 2D straight lines and minimizing the distortion error. This residue is given by the sum of the squares of the distances from the points to the lines (Devernay et al. [10]).

The design of a correction method is completed by deciding how to apply it. There are basically two alternatives: those methods which need human supervision and those which are *automatic*. Human-supervised methods rely on *manually* identifying some known straight lines within the scene (Alvarez et al. [3], Brown [5], Wang et al. [17]). These user-dependent methods are very robust and independent of the camera parameters, and do not require a calibration pattern. However, due to human supervision, they are difficult to apply when dealing with large images.

In Bukhari et al. [6], an automatic radial estimation method working on a single image without user intervention is discussed. This method does not require any special calibration pattern. The

method applies the one-parameter division model to estimate the distortion from a set of automatically detected non-overlapping circular arcs within the image.

In Cucchiara et al. [9], the authors propose an automatic method for radial lens distortion correction using an adapted Hough transform including the radial distortion parameter to automatically detect straight lines within the image. The radial model with one parameter is used and an exhaustive search is performed testing each distortion parameter in a discretized set  $(k_{min}, k_{max})$  and then selecting the one that provides the best fit (i.e. maximizes the *straightness* of the candidate lines). The method can work automatically, but better results are obtained with a semi-automatic version, which includes the manual selection of a ROI (region of interest) containing some straight lines.

Alemán-Flores et al. [1] adapt the Hough transform to automatically correct radial lens distortion for wide-angle lenses using the one-parameter division model. This method uses the information provided by all edge points, which allows a better identification of straight lines when applying the extended Hough transform. The distortion parameter which is obtained is finally optimized with a simple gradient-based method. The method is applied to the whole image and, therefore, there is no need to select a ROI as recommended in Cucchiara et al. [9].

A fully automatic distortion correction method, which also embeds the radial distortion parameter into the Hough transform to better detect straight lines, is presented by Lee et al. [14]. It is applied to endoscopic images captured with a wide-angle *zoomable* lens. In this case, the division model with one parameter is used to deal with the strong distortion caused by the wide-angle lens, and the method is adapted to include the effects related to the variation of the focal length due to zooming operations. The method is intended for real time applications once mapped to a GPU computing platform. The distortion parameter is estimated by optimizing the Hough entropy in the image gradient space.

In this work, we propose an unsupervised method which makes use of the one-parameter division model to correct the radial distortion caused by a wide-angle lens just using a single image. First, we automatically detect the distorted lines within the image by adapting the Hough transform to our problem. By embedding the radial distortion parameter into the Hough parametric space, the longest arcs (*distorted lines*) within the image are extracted. From the improved Hough transform, we obtain a collection of distorted lines and an initial approximation for the distortion parameter  $k_1$ . Then, a numerical optimization scheme is applied to optimize this value by minimizing the Euclidean distance from the corrected line points to the set of straight lines. Table 1, summarizes the main stages of the proposed algorithm.

Stage	Description
1.	Edge detection using Canny method. in: Input image. out: Edge points coordinates and orientation.
2.	Initial estimation of the lens distortion parameter using improved Hough transform. in: Edge points coordinates and orientation. out: Estimation of the distortion parameter and the most voted lines.
3.	Distortion parameter optimization. in: Estimation of the distortion parameter and the most voted lines. out: Optimized lens distortion parameter.
4.	Image distortion correction. in: Input image and optimized lens distortion model. out: Distortion-free output image obtained using the estimated model.

Table 1: Summary of the algorithm for lens distortion correction.

This work is based on a previous paper, [1], where we presented a theoretical description of the

proposed method. In this paper we present the algorithm and the implementation of the method. We also provide additional details of the methodology discussed in [1] regarding the voting approach in the Hough space. New experiments with real images using a Tokina 11-16mm Ultra-wide Angle Zoom Lens have also been included.

The organization of this paper is as follows: First, we present, in separate sections, the details of the different stages of the algorithm shown in Table 1. Next, in Section 7, we show some experimental results on real images. Finally, some conclusions are discussed.

## 2 Edge Detection Using the Canny Method

To implement the Canny edge detector we use the standard approach proposed in [7]. Next we present some technical details about our implementation.

**Gaussian convolution:** We use the recursive filter approximation of Gaussian convolution proposed in [4]. We use this approach because it is very fast and provides a good approximation of the Gaussian convolution.

**Image gradient:** To compute the image gradient we use the following  $3 \times 3$  convolution masks

$$\frac{\partial I}{\partial x} = 0.5 \begin{bmatrix} -\frac{2-\sqrt{2}}{2} & 0 & \frac{2-\sqrt{2}}{2} \\ -(\sqrt{2}-1) & 0 & \sqrt{2}-1 \\ -\frac{2-\sqrt{2}}{2} & 0 & \frac{2-\sqrt{2}}{2} \end{bmatrix} \quad \text{and} \quad \frac{\partial I}{\partial y} = 0.5 \begin{bmatrix} -\frac{2-\sqrt{2}}{2} & -(\sqrt{2}-1) & -\frac{2-\sqrt{2}}{2} \\ 0 & 0 & 0 \\ \frac{2-\sqrt{2}}{2} & \sqrt{2}-1 & \frac{2-\sqrt{2}}{2} \end{bmatrix}.$$

Among the different  $3 \times 3$  masks we can use to discretize the gradient, we have chosen these ones because they satisfy that the norm of the gradient estimated using these masks is invariant under 45 degree rotations.

**Setting Canny thresholds:** To apply the Canny method, we need to fix two thresholds (low and high). Both thresholds are represented as percentiles (the thresholds below which a given percentage of the norms of the gradient fall).

**Non-maximum suppression:** Once the Canny high threshold is fixed, we compute an initial estimation of edge points using the gradient norm and we remove points which are not local maxima in the direction of the gradient. With this aim, we compare the value of the gradient norm of each pixel with the norm of its neighbors in the gradient direction. If this value is greater than the value of its neighbors and above the high threshold, the pixel is considered as an edge pixel. However, if the value fulfills the condition of the gradient direction but it is lower than the high threshold, it will be considered in the subsequent hysteresis step.

**Hysteresis implementation:** Hysteresis implementation is performed using the low and high Canny thresholds. Starting from the initial collection of edge points obtained using the high threshold and after the non-maximum suppression, we use a recursive algorithm to get new edge points using the low threshold. A point where the norm of the gradient is between the lower and higher thresholds is considered to be an actual edge pixel if it has a neighbor edge pixel, and then its neighborhood is recursively explored.

Finally, the detected edges are stored in a structure with their positions and orientations. This information will be necessary for the next stage, where the improved Hough transform is used.

In Table 2 we present the main parameters of the Canny edge detector. For each parameter we present the default value which is used in all the experiments presented in this paper and we also point out if the parameter is included in the IPOL demo user interface.

Canny function		
Parameters	Default Value	Interface
Standard deviation of the Gaussian	2.0	No
Low threshold for the Canny method	0.7	No
High threshold for the Canny method	0.8	Yes

Table 2: Parameters of the Canny function.

### 3 Estimation of a One-Parameter Division Model Using the Improved Hough Transform

In what follows, we will focus on the one-parameter division model. Using equations (1) and (3), we obtain

$$\hat{r} = \frac{r}{1 + k_1 r^2}, \quad (4)$$

where  $\hat{r}$  is the distance from the center of distortion to the point after applying the model. To simplify the discussion, we will consider that the center of distortion  $(x_c, y_c)$  is the center of the image, which is usually a good approximation.

To correct the radial distortion, the distortion parameter  $k_1$  must be estimated from the available information, that is, from the line primitives. Line primitives are searched for in the edge image which is computed using the Canny edge detector. Then, lines (*straight* lines) can be extracted from the set of edges by applying the Hough transform, which searches for the most reliable candidates within a certain two-dimensional space corresponding to the orientation and the distance to the origin of the candidate lines. Each edge point votes for those lines which are more likely to contain it. The lines which receive the highest scores are considered to be the most reliable ones and, consequently, taken into account in order to estimate the distortion.

Note that the *classical* Hough transform does not consider the influence of the distortion in the alignment of the edge points and, for that reason, straight lines are split into different segments. This directly translates into a loss of information because some lines may not be properly recognized as lines and, therefore, the distortion correction must be performed without using the maximum number of candidate straight lines. In this paper, we extend the usual two-dimensional Hough space by adding a new dimension, namely, the radial distortion parameter.

For practical reasons, instead of considering the distortion parameter value itself in the new Hough space, we use a normalized distortion parameter  $p$ , given by the percentage of correction of the distance to the center of distortion

$$p = (\tilde{r}_{max} - r_{max})/r_{max}, \quad (5)$$

where  $r_{max}$  is the distance from the center of distortion to the farthest point in the original image, and  $\tilde{r}_{max}$  is the same distance, but after applying the distortion model. The above simple normalization allows giving a geometrical meaning to the parameter  $p$ . Another advantage of using  $p$  as a new parameter in the Hough space is that it is independent of the image resolution. The relation between parameter  $p$  and  $k_1$  is straightforward and is given by the following expression

$$k_1 = \frac{-p}{(1 + p)r_{max}^2}. \quad (6)$$

To define the range of the normalized distortion parameter  $p$  in the Hough space, we first have to take into account that transformation (4) should be bijective in the interval  $[0, r_{max}]$ . In particular,

the derivative of the transformation in (4) with respect to  $r$  should be positive, so that

$$\frac{(1 + k_1 r^2) - 2k_1 r^2}{(1 + k_1 r^2)^2} > 0 \quad \Rightarrow \quad 1 - k_1 r^2 > 0 \quad \Rightarrow \quad k_1 < \frac{1}{r_{max}^2}. \quad (7)$$

Now, replacing  $k_1$  by  $p$  (using (6)), a straightforward computation yields that

$$p > -0.5 \quad (8)$$

In practice, to include the distortion parameter in the Hough space we choose an interval  $[p_{min}, p_{max}]$  (with  $p_{min} > -0.5$ ) and we take

$$p \in P = \left\{ p_{min} + n \cdot \delta_p : n = 0, 1, \dots, \frac{p_{max} - p_{min}}{\delta_p} \right\}, \quad (9)$$

where  $\delta_p$  is the discretization step for the distortion parameter. We point out that a line in this extended Hough space is defined by a triplet  $(d, \alpha, p)$ , where  $d$  and  $\alpha$  represent, respectively, the distance and line orientation parameters in the Hough space.

In the voting step, for each value of  $p \in P$ , we first correct the edge point coordinates and orientation using the lens distortion model associated to  $p$ . Next, each edge point votes for a line only if the edge point orientation is coherent with the line orientation. To do that, we impose that the angular difference between the edge orientation and the line orientation must be smaller than a certain value  $\delta_\alpha$ . This value is a parameter of the method and, in the numerical experiments we have performed, we have chosen  $\delta_\alpha = 2$  degrees. Moreover, the vote of a point for a line depends on how close they are. It is given by  $v = 1/(1 + d_p)$ , where  $d_p$  is the distance from the point to the line.

Once the voting step is over, the best distortion parameter is computed by maximizing the following criterion

$$\max_{p \in P} \left\{ \sum_{j=1}^N \text{votes}(d_{Hough,j}^p, \alpha_{Hough,j}^p, p) \right\},$$

where  $\{(d_{Hough,j}^p, \alpha_{Hough,j}^p, p)\}_{j=1}^N$  represent the  $N$  most voted lines in the Hough space for the normalized lens distortion parameter  $p \in P$ . Values  $d_{Hough,j}^p$  and  $\alpha_{Hough,j}^p$  are the distance and line orientation parameters of line  $j$  in the Hough space. To compute the  $N$  most voted lines, we look for the highest local extrema in the Hough voting matrix. In order to avoid duplicated lines due to small variations in the parameters, we fix a neighborhood of the local extrema in the Hough space and we reject those lines within the neighborhood of an already selected line. With this purpose, we set two parameters  $min\_dif_o$  and  $min\_dif_d$  to determine the minimum difference in the orientation and distance to the origin between two lines. When the orientation of a line differs less than  $min\_dif_o$  from an already selected line, and their distances to the origin differ less than  $min\_dif_d$ , the line is rejected.

We denote by  $p_0$  the value  $p \in P$  where the maximum of the above voting criterion is reached. Figure 1 illustrates how the maximum of the voting score varies within the Hough space according to the percentage of correction determined by the normalized distortion parameter  $p$ .

Associated to the best value of  $p$ , and according to the above criterion, we obtain a collection of  $N$  lines. We associate to each line  $j$  the edge points  $\{\bar{x}_{ji} = (x_{ji}, y_{ji})\}_{i=1, \dots, N(j)}$  that fulfill two conditions: The difference between the line orientation and the orientation of the corrected edge point  $(\hat{x}_{ji}^p, \hat{y}_{ji}^p)$  must be lower than a threshold  $\delta_\alpha$ . Besides, the distance from the corrected point to the line must be lower than a threshold  $max_d$ . Both thresholds are parameters of the algorithm. We remark that  $(\hat{x}_{ji}^p, \hat{y}_{ji}^p)$  are obtained using Equation (1) with the model given by  $p$ . A summary of the modified Hough transform is described in Algorithm 1. In Table 3, we present the main parameters of this

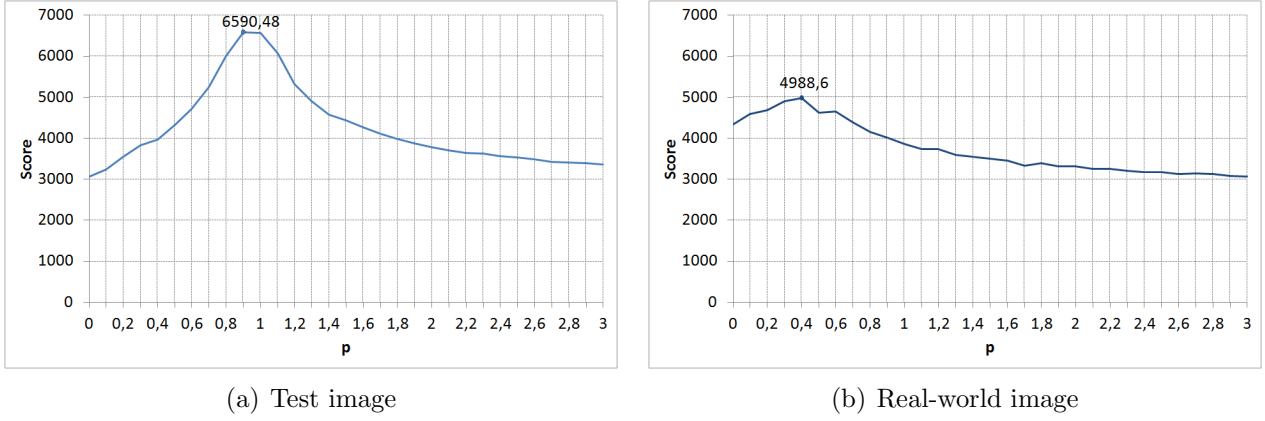


Figure 1: Values of the maximum in the voting space with respect to the percentage of correction using the modified Hough transform and a division lens distortion model for (a) the calibration pattern in Figure 2 and (b) the building in Figure 3.

algorithm. For each parameter, we indicate the default value which is used in all the experiments presented in this paper and we also point out whether the parameter is included in the IPOL demo user interface or not.

## 4 Distortion Parameter Optimization

In this stage, we refine the value of  $p_0$  to obtain a more accurate approximation of the distortion parameter. First, we simplify the collection of lines obtained from the modified Hough transform by removing the lines with a number of associated points  $N(j)$  smaller than a certain parameter  $min\_length$  (we use 5 points in our experiments). We point out that lines with a small number of points do not provide relevant information (in comparison with large lines). Moreover the larger the number of points of a line, the higher the probability of being an actual line and not a spurious line. On the other hand, the initial parameter  $N$ , which represents the maximum number of lines, becomes irrelevant as far as it is greater than the number of lines with more than  $min\_length$  associated points. Therefore, we finally obtain a number of lines  $Nl$  which is usually lower than the maximum number of lines  $N$ . In the experiments we present in this paper, we have selected  $N = 30$ . The value of  $Nl$  in each experiment depends on the number of visible lines in the image.

For a given  $p \in R$  and a given collection of line points  $\{\bar{x}_{ji} = (x_{ji}, y_{ji})\}_{i=1, \dots, N(j)}$  we can optimize the equation of the associated straight line by minimizing

$$D(\alpha, d) = \sum_{i=1}^{N(j)} (\cos(\alpha)\hat{x}_{ji}^p + \sin(\alpha)\hat{y}_{ji}^p + d)^2, \quad (10)$$

where  $\{(\hat{x}_{ji}^p, \hat{y}_{ji}^p)\}_{i=1, \dots, N(j)}$  are the corrected points using the model associated to  $p$ . This well known minimization problem has a simple closed-form solution (see for instance [11] for more details). We denote by  $(\alpha_j^p, d_j^p)$  the pair where the above function reaches its minimum.

The optimization of the lens distortion model is performed by minimizing the average of the square of the distance from the corrected primitive points to their associated lines. Consequently, using the notation introduced above, this optimization consists in minimizing the following error

---

**Algorithm 1:** Line detection using the improved Hough transform.

---

**input** : Image edge point coordinates  $\{(x, y)\}$ , image edge point orientations  $\{\alpha(x, y)\}$  and maximum number of expected lines  $N$

**output:** An estimation of the normalized distortion parameter  $p_0$  and the most voted lines  
**begin**

```

     $max_{Hough} = -1;$ 
    for  $p = p_{min}$  to  $p_{max}$  do
         $k_1 = \frac{-p}{r_{max}^2(1+p)};$ 
        for  $(x, y) \in Image\ edge\ pixels$  do
             $r = \sqrt{(x - x_c)^2 + (y - y_c)^2};$ 
             $\hat{x} = x_c + \left(\frac{1}{1+k_1r^2}\right)(x - x_c);$ 
             $\hat{y} = y_c + \left(\frac{1}{1+k_1r^2}\right)(y - y_c);$ 
            // Next we update the edge point orientation according to lens
            // distortion correction.
             $\alpha_{new} = Update\_orientation(x, y, \alpha(x, y), k_1);$ 
            for  $\beta \in [\alpha_{new} - \delta_\alpha, \alpha_{new} + \delta_\alpha]$  do
                 $d = Floor(-(\cos(\beta)\hat{x} + \sin(\beta)\hat{y}));$ 
                for  $dist \in [d - 2, d + 2]$  do
                     $d_l = |\cos(\beta)\hat{x} + \sin(\beta)\hat{y} + d|;$ 
                     $votes(dist, \beta, p) = votes(dist, \beta, p) + \frac{1}{1+d_l};$ 
                end
            end
        end
        // For the  $N$  most voted lines given by  $\{(d_{Hough,j}^p, \alpha_{Hough,j}^p, p)\}_{j=1}^N$  we compute
         $sum = \sum_{j=1}^N votes(d_{Hough,j}^p, \alpha_{Hough,j}^p, p);$ 
        if  $(sum > max_{Hough})$  then
             $p_0 = p;$ 
             $max_{Hough} = sum;$ 
            // We update the  $N$  most voted lines to  $\{(d_{Hough,j}^p, \alpha_{Hough,j}^p, p)\}_{j=1}^N$ 
        end
    end
end

```

---



<b>Improved Hough function</b>		
Parameters	Default Value	Interface
Set of detected edges and their orientations	None	No
Maximum distance allowed between points and associated lines (pixels)	3.0	Yes
Maximum difference of the point orientation angle and the line angle (degrees)	2.0	Yes
Maximum number of lines to detect	30	No
Angle resolution for the angle parameter in the Hough space (degrees)	0.1	No
Distance resolution for the distance parameter in the Hough space (pixels)	1.0	No
Minimum value for the normalized distortion parameter	0.0	Yes
Maximum value for the normalized distortion parameter	3.0	Yes
Distortion parameter discretization step in our improved Hough space	0.1	No
Minimum length of the lines (pixels)	5	No
Minimum difference for the orientation (degrees)	2	No
Minimum difference for the distance to the origin (pixels)	20	No

Table 3: Parameters of the improved Hough function.

function

$$E(p) = \frac{\sum_{j=1}^{Nl} \sum_{i=1}^{N(j)} (\cos(\alpha_j^p) \hat{x}_{ji}^p + \sin(\alpha_j^p) \hat{y}_{ji}^p + d_j^p)^2}{\sum_{j=1}^{Nl} N(j)}. \quad (11)$$

The initial value for  $p$  is  $p_0$ , obtained by the modified Hough transform. In Algorithm 2, we describe the minimization algorithm. We use a modified Newton-Raphson method where we include a damping parameter  $\gamma$  to ensure that the error decreases across the iterations. When  $\gamma$  is close to 0, the method is close to the Newton-Raphson method and, when  $\gamma$  is large, the method is close to the usual gradient descent method. If the error for a candidate solution is lower than the previous one, we accept the new candidate solution and we divide  $\gamma$  by 10 in order to be closer to the Newton-Raphson method. Otherwise, we reject the candidate solution, we multiply  $\gamma$  by 10 to get closer to the gradient descent method and we recompute the solution. Usually, the value of  $\gamma$  is higher when we are far from the solution and decreases when we approach it. The idea of using a damping parameter in optimization algorithms was introduced by Levenberg in 1944 [16] in the context of the well-known Levenberg-Marquardt optimization algorithm.

## 5 Image Distortion Correction

In stage 4 of the algorithm we create a distortion-free image using the estimated lens distortion model. To do that, we need to compute the inverse of the lens distortion transformation. Using Equation (4) we obtain that

$$\hat{r} + k_1 \cdot \hat{r} \cdot r^2 - r = 0, \quad (12)$$

which is a 2-degree polynomial in  $r$ . We point out that we only consider those lens distortion models satisfying condition (8), which defines a one-to-one transformation in the original image.

---

**Algorithm 2:** Algorithm to optimize  $p_0$  by minimizing  $E(p)$

---

**input** : The initial value of the normalized distortion parameter, discretized step for the computation of the derivatives and tolerance for convergence

**output:** The optimized lens distortion parameter

$p_{opt} = p_0;$

$\gamma = 1.0;$

$p = p_0 + TOL + 1;$

**while** ( $|p - p_{opt}| > TOL$ ) **do**

    //  $E$  is defined in Equation (11)

$E'(p_{opt}) = \frac{E(p_{opt}+h) - E(p_{opt}-h)}{2h};$

$E''(p_{opt}) = \frac{E(p_{opt}+h) - 2E(p_{opt}) + E(p_{opt}-h)}{h^2};$

$p_{new} = p_{opt} - \frac{E'(p_{opt})}{E''(p_{opt}) + \gamma};$

**while** ( $E(p_{new}) > E(p_{opt})$ ) **do**

$\gamma = \gamma \times 10;$

$p_{new} = p_{opt} - \frac{E'(p_{opt})}{E''(p_{opt}) + \gamma};$

**end**

$\gamma = \gamma / 10;$

$p = p_{opt};$

$p_{opt} = p_{new};$

**end**

**return**  $p_{opt};$

---

In particular, for any  $\hat{r} \in (0, \hat{r}_{max}]$  there is a single root  $r$  of the above polynomial in the interval  $(0, r_{max}]$ . Such root must be the smallest positive root of the above polynomial, which is given by

$$r = \frac{1 - \sqrt{1 - 4k_1(\hat{r})^2}}{2 \cdot k_1 \cdot \hat{r}}. \quad (13)$$

Note that if  $\hat{r} = 0$  then  $r = 0$ . To speed up the algorithm, instead of estimating the above inverse transformation for each pixel, we first build a vector, *InverseVector*, where we store the result of the expression in (13) in the range  $[0, \hat{r}_{max}]$ . In Algorithm 3 we describe the complete algorithm to compute the distortion-free image.

## 6 Complexity Analysis

In this section we present the complexity analysis of the described algorithms. First, we start by defining some variables for the complexity description:

- $N_{pixels}$ : Number of image pixels.
- $N_{edges}$ : Number of edge pixels.
- $N_{lines}$ : Number of lines to extract from the Hough space.
- $S_{distortion}$  : Size of the discretized interval of distortion parameters used in the Hough space.
- $S_{Hough}^{matrix}$ : Size of the Hough voting matrix.

---

**Algorithm 3:** Free distortion image computation.

---

```

input : Input image and estimated lens distortion model
output: Distortion free output image
begin
    // We compute the InverseVector in  $[0, \hat{r}_{max}]$  (as explained in the text).
    for  $(\hat{x}, \hat{y}) \in \text{Output image pixels}$  do
         $\hat{r} = \sqrt{(\hat{x} - x_c)^2 + (\hat{y} - y_c)^2}$ ;
         $index = \text{Floor}(\hat{r})$ ;
         $weight = \hat{r} - index$ ;
         $r = (1 - weight) \cdot \text{InverseVector}[index] + weight \cdot \text{InverseVector}[index + 1]$ ;
         $x = x_c + r \frac{\hat{x} - x_c}{\hat{r}}$ ;
         $y = y_c + r \frac{\hat{y} - y_c}{\hat{r}}$ ;
         $\text{OutputImage}(\hat{x}, \hat{y}) = \text{InputImage}(x, y)$ ;
        // In the above expression we use bilinear interpolation to estimate
         $\text{InputImage}(x, y)$ 
    end
end

```

---

- $S_{Hough}^{voting}$ : Size of the neighborhood used for each point to vote in the Hough score matrix (this neighborhood includes the lines passing near the edge point and with an orientation similar to the edge point orientation)

**Canny edge detector:** In the Canny edge detection algorithm, the complexity is determined by the algorithm we use to implement the Gaussian convolution. As we use the fast recursive algorithm proposed in [4], as an approximation of the Gaussian convolution, which has a linear complexity with respect to the image size, we can conclude that the complexity of the Canny edge detector we use is

$$O(N_{pixels}).$$

**Improved Hough transform:** The traditional Hough transform has a complexity related to the number of provided edge points and the dimension of the voting matrix. In our case, we divide the complexity analysis in the following stages:

1. We consider the complexity of the voting process. The computational cost of this stage is based on the number of detected edge points provided to our improved Hough Transform by the Canny edge detector. Moreover, we have to take into account the dimension of the voting matrix, determined by the number of values considered in the discretized intervals for the lens distortion parameter, the distance and the angle variation for each voting edge point. In this way, we can write the complexity of this section as

$$O(N_{edges} \times S_{distortion} \times S_{Hough}^{voting}).$$

2. As well as in the above description, in the selection of the maximum, we consider the size of the Hough voting matrix. However, as we provide a limit of lines to be considered, we need to include this number in the complexity estimation. Therefore, the cost of the selection of the maximum will be

$$O(N_{lines} \times S_{Hough}^{matrix}).$$

3. The complexity of the association of the edge points with the lines depends on the number of edge points and the number of lines to compute

$$O(N_{lines} \times N_{edges}).$$

**Distortion parameter optimization:** The modified Newton-Raphson method proposed for distortion parameter optimization converges in a few number of iterations for all the experiments performed. Therefore, time complexity comes from the evaluation of the function to optimize (11). In our case such complexity is given by

$$O(N_{edges}).$$

**Image distortion correction:** For the correction of the distortion, we have to go through the image pixels and correct the lens distortion, so the complexity is

$$O(N_{pixels}).$$

Therefore, the total complexity of the method is

$$O(N_{pixels}) + O(N_{pixels}) \approx O(N_{pixels}).$$

Although it is difficult to provide accurate estimates for the variables or for the relationship between them, in practice we have that  $N_{lines} \ll N_{edges} \ll N_{pixels} \approx S_{Hough}^{matrix}$ . Therefore, taking into account the above expressions the total complexity of the method is

$$O(N_{pixels}).$$

and then the time complexity of the proposed method is linear with respect to the number of pixels.

## 7 Experimental Results

We have tested our model in some images showing wide-angle lens distortion. The images have been acquired with a Nikon D90 camera and a Tokina 11-16mm Ultra-wide Angle Zoom Lens. Figure 2 shows the results for a calibration pattern and Figure 3 presents the results for an image of a building. We have represented each line using a different color to identify them. In both images, from the detected arcs or distorted lines, the distortion is estimated and the images are corrected. As observed, for the calibration pattern all the lines corresponding to the edges of the squares are correctly identified (some other straight lines which do not correspond to the squares are also extracted). For the image of the building, a set of significant lines have also been extracted. For both images, the proposed automatic correction works well. Numerical results are summarized in Table 4, where we show that the optimization method we apply effectively reduces the value of the energy function.

More results corresponding to real-world images are illustrated in Figure 4. These results show the robustness and accuracy of our method. Some of them are difficult to correct due to severe fish-eye distortion. We observe that, even when there are only a few visible lines, the results are satisfactory. All the results have been achieved with the same collection of parameters that are shown in tables 2 and 3.

Images	Nlines	Npoints	$p_0$	$E(p_0)$	$p_{opt}$	$E(p_{opt})$
pattern	29	9101	0.9	0.86598	0.957329	0.62965
building	27	6096	0.4	0.88286	0.383621	0.82366

Table 4: Numerical results.

## 8 Conclusions

In this paper we present an algorithm to automatically correct wide-angle lens distortion. We use a one-parameter division model and an improved Hough space where the distortion parameter has been added to be able to find distorted lines in the image. We optimize the resulting distortion parameter by minimizing the distance from the lines to their associated points. We have presented the details of the algorithms of the different stages, as well as some experiments which show the ability of the method to correct wide-angle lens distortion.

The introduction of the distortion parameter into the Hough space allows considering the lines and their points as they appear in the image, i.e. as distorted lines which are not straight. Consequently, it is possible to associate points and segments which, otherwise, would not be connected by the line detection process. Furthermore, the optimization process refines the distortion parameter in order to obtain a better adjustment between the edge points and the distorted straight lines. This makes our proposal much more robust. Not only does it identify longer segments of the lines, but it is also able to cope with significant distortions. In addition, the fact that the method works properly for a large variety of images, with different conditions and distortions, and without the need to find *ad hoc* values for the parameters, emphasizes its robustness. The quantitative results (number of line points which are detected and distance from these points to their corresponding lines) as well as the qualitative ones (undistorted images) support the effectiveness of this technique. In the future, we plan to extend the present work to the estimation of two-parameter lens distortion models.

## Acknowledgments

This work has been partially supported by the MICINN project reference MTM2010-17615 (Ministry of Science and Innovation, Spain).

## Image Credits

All images by the authors (license CC-BY-SA).

## References

- [1] M. ALEMÁN-FLORES, L. ALVAREZ, L. GOMEZ, AND D. SANTANA-CEDRÉS, *Wide-angle lens distortion correction using division models*, in Lecture Notes in Computer Science. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 2013, pp. 415–422. [http://dx.doi.org/10.1007/978-3-642-41822-8\\_52](http://dx.doi.org/10.1007/978-3-642-41822-8_52).
- [2] L. ALVAREZ, L. GOMEZ, AND J.R. SENDRA, *Algebraic Lens Distortion Model Estimation*, Image Processing On Line, 1 (2010). <http://dx.doi.org/10.5201/ipol.2010.ags-alde>.

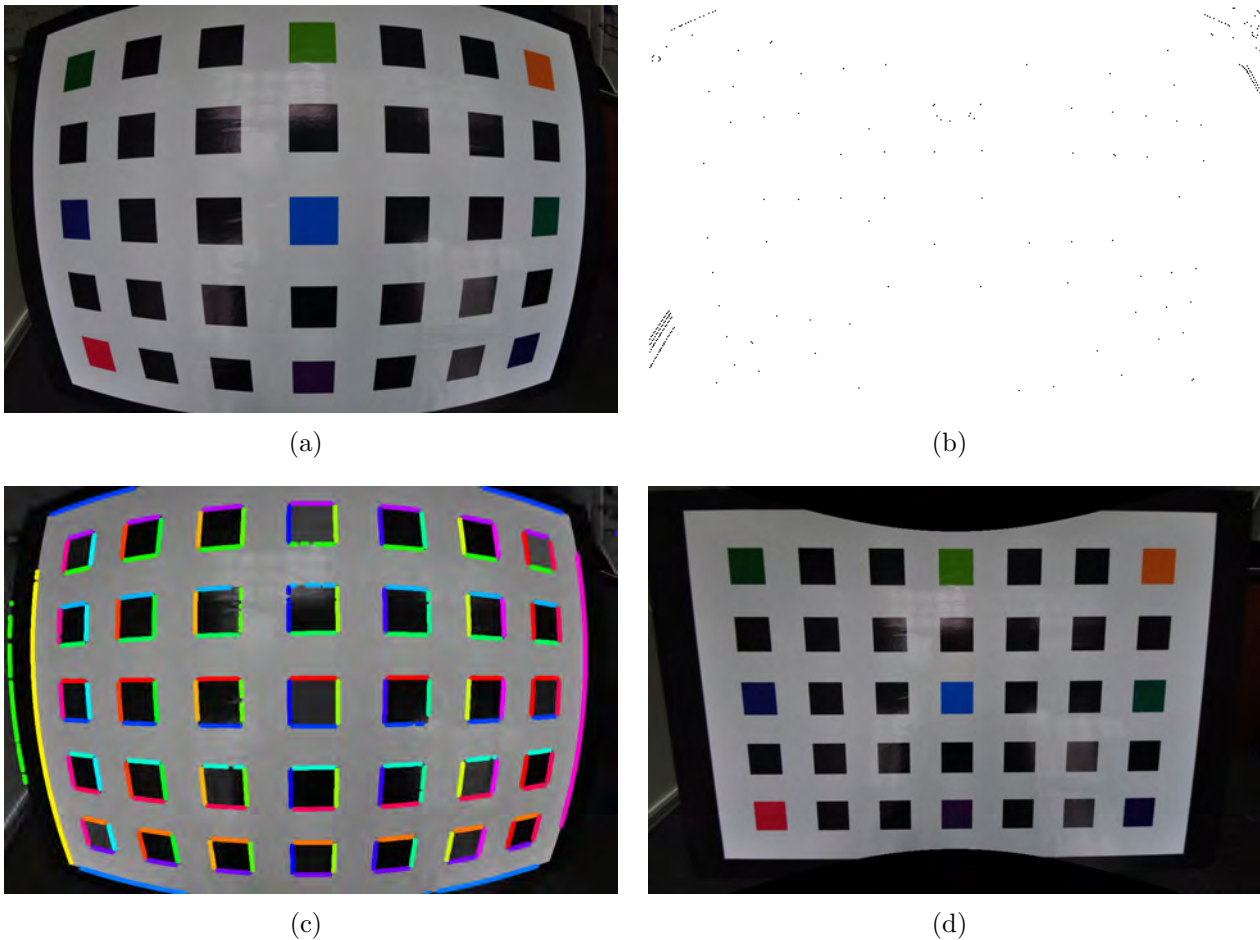


Figure 2: Lens distortion correction for an image of a calibration pattern: (a) original image, (b) detected edges using the Canny method, (c) lines detected using the proposed method, and (d) undistorted image using the proposed method.

- [3] L. ALVAREZ, L. GOMEZ, AND R. SENDRA, *Accurate depth dependent lens distortion models: an application to planar view scenarios*, Journal of Mathematical Imaging and Vision, 39 (2011), pp. 75–85. <http://dx.doi.org/10.1007/s10851-010-0226-2>.
- [4] L. ALVAREZ AND L. MAZORRA, *Signal and Image Restoration using Shock Filters and Anisotropic Diffusion*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 590–605. <http://dx.doi.org/10.1137/0731032>.
- [5] D.C. BROWN, *Close-range camera calibration*, Photogrammetric Engineering, 37 (1971), pp. 855–866.
- [6] F. BUKHARI AND M. DAILEY, *Automatic radial distortion estimation from a single image*, Journal of Mathematical Imaging and Vision, 45 (2012), pp. 31–45. <http://dx.doi.org/10.1007/s10851-012-0342-2>.
- [7] J. CANNY, *A computational approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 (1986), pp. 679–698. <http://dx.doi.org/10.1109/TPAMI.1986.4767851>.

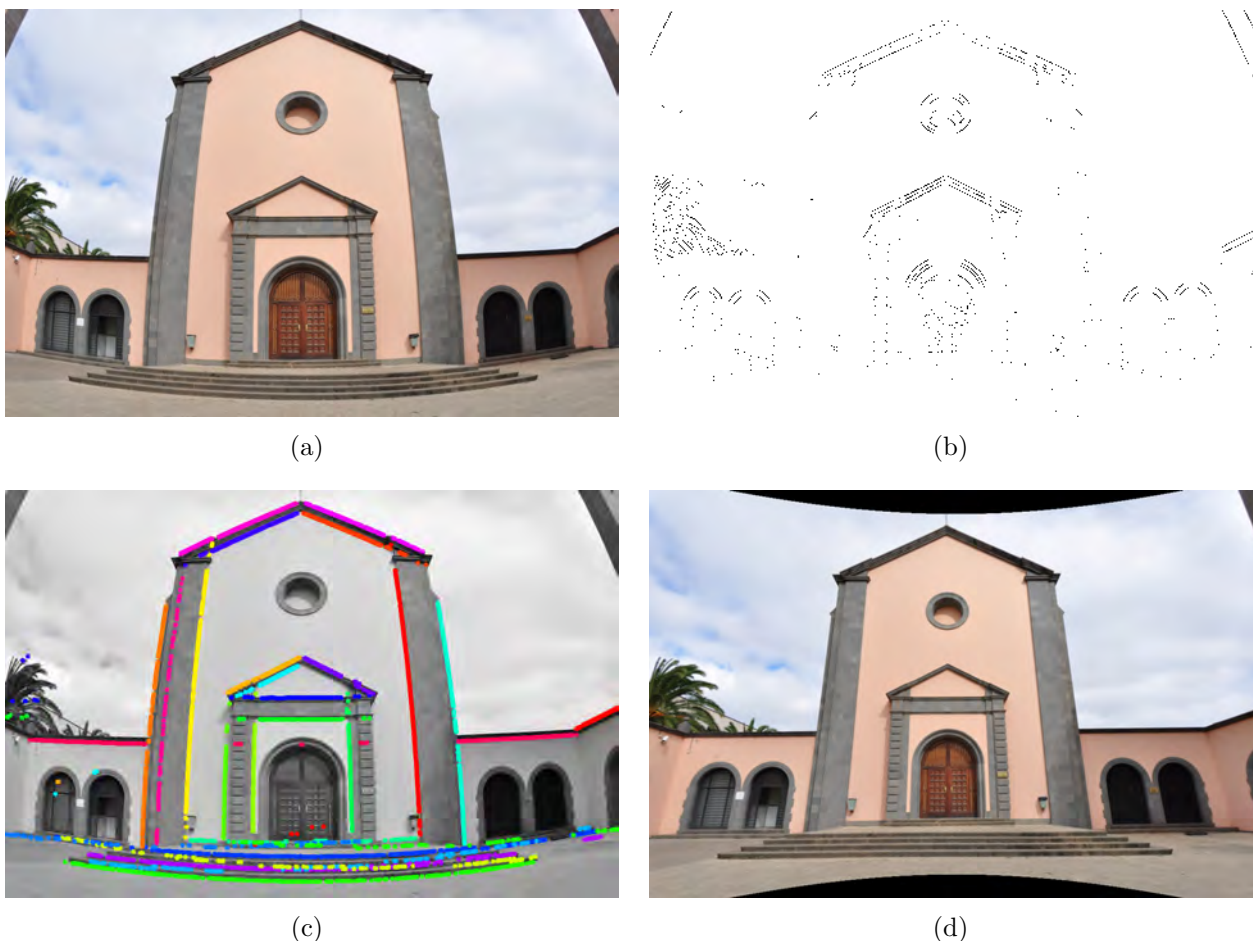


Figure 3: Lens distortion correction for an image of a building: (a) original image, (b) detected edges using the Canny method, (c) lines detected using the proposed method, and (d) undistorted image using the proposed method.

- [8] D. CLAUS AND A.W. FITZGIBBON, *A rational function lens distortion model for general cameras*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2005, pp. 213–219. <http://dx.doi.org/10.1109/CVPR.2005.43>.
- [9] R. CUCCHIARA, C. GRANA, A. PRATI, AND R. VEZZANI, *A Hough transform-based method for radial lens distortion correction.*, in Proceedings of 12th International Conference on Image Analysis and Processing., 2003, pp. 182–187. <http://dx.doi.org/10.1109/ICIAP.2003.1234047>.
- [10] F. DEVERNAY AND O. FAUGERAS, *Straight lines have to be straight*, Machine Vision and Applications, 13 (2001), pp. 14–24. <http://dx.doi.org/10.1007/PL00013269>.
- [11] O. FAUGERAS, *Three-dimensional computer vision*, MIT Press, 1993.
- [12] A. W. FITZGIBBON, *Simultaneous linear estimation of multiple view geometry and lens distortion*, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, (2001), pp. 125–132. <http://dx.doi.org/10.1109/CVPR.2001.990465>.
- [13] C. HUGHES, M. GLAVIN, E. JONES, AND P. DENNY, *Review of geometric distortion compensation in fish-eye cameras*, in Signals and Systems Conference, 2008, pp. 162–167.

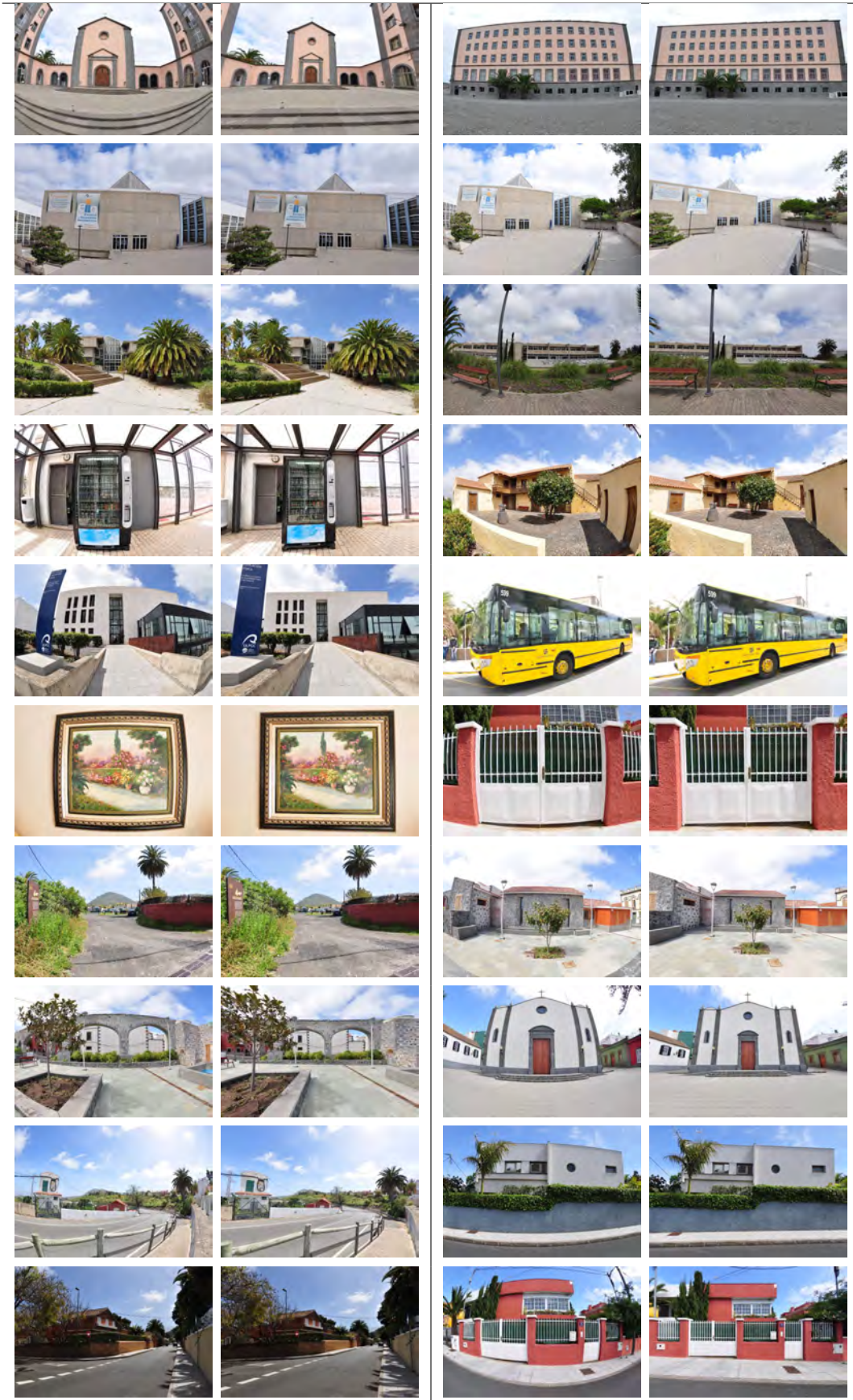


Figure 4: Some examples of image distortion correction using the proposed method. All the results have been obtained using a single collection of parameters presented in tables 2 and 3.



- [14] T-Y. LEE, T-S. CHANG, C-H. WEI, S-H. LAI, K-C. LIU, AND H-S. WU, *Automatic distortion correction of endoscopic images captured with wide-angle zoom lens*, IEEE Transactions on Biomedical Engineering, 60 (2013), pp. 2603–2613. <http://dx.doi.org/10.1109/TBME.2013.2261816>.
- [15] R. LENZ, *Linsenfehlerkorrigierte Eichung von Halbleiterkameras mit Standardobjektiven für hochgenaue 3D - Messungen in Echtzeit*, in Mustererkennung, vol. 149 of Informatik-Fachberichte, 1987, pp. 212–216. [http://dx.doi.org/10.1007/978-3-662-22205-8\\_55](http://dx.doi.org/10.1007/978-3-662-22205-8_55).
- [16] K. LEVENBERG, *A method for the solution of certain non-linear problems in least-squares*, Quarterly of Applied Mathematics, 2 (1944), pp. 164–168.
- [17] A. WANG, T. QIU, AND L. SHAO, *A simple method to radial distortion correction with centre of distortion estimation*, Journal of Mathematical Imaging and Vision, 35 (2009), pp. 165–172. <http://dx.doi.org/10.1007/s10851-009-0162-1>.