Algebraic Lens Distortion Model Estimation

Luis Alvarez, Luis Gomez and J. Rafael Sendra

published • 2010-07-28
reference • Luis Alvarez, Luis Gomez and J. Rafael Sendra, *Algebraic Lens Distortion Model Estimation*, Image Processing On Line, 2010.
DOI : http://dx.doi.org/10.5201/ipol.2010.ags-alde

- Luis Alvarez lalvarez@dis.ulpgc.es, Universidad de Las Palmas de Gran Canaria
- Luis Gomez lgomez@diea.ulpgc.es, Universidad de Las Palmas de Gran Canaria
- J. R. Sendra Rafael.Sendra@uah.es, Universidad de Alcalá

Communicated by Pascal Monasse monasse@imagine.enpc.fr, IMAGINE, ENPC ParisTech
Edited by Nicolas Limare nicolas.limare@cmla.ens-cachan.fr, CMLA, ENS Cachan

Overview

A very important property of the usual pinhole model for camera projection is that 3D lines in the scene are projected to 2D lines. Unfortunately, wide-angle lenses (specially low-cost lenses) may introduce a strong barrel distortion, which makes the usual pinhole model fail. Lens distortion models try to correct such distortion. In reference 1., we propose an algebraic approach to the estimation of the lens distortion parameters based on the rectification of lines in the image.

Using the proposed method, the lens distortion parameters are obtained by minimizing a 4 total-degree polynomial in several variables. We perform numerical experiments using calibration patterns and real scenes to show the performance of the proposed method.

References

1. Luis Alvarez, Luis Gomez and Rafael Sendra, An Algebraic Approach to Lens Distortion by Line Rectification,
   Journal of Mathematical Imaging and Vision, vol. 35, nº 1, pp. 36 - 50, September 2009. DOI:10.1007/s10851-009-0153-2.
2. Luis Alvarez, Luis Gomez and Rafael Sendra, Lens distortion Model Software Application (Windows) (accessed July 16, 2010).
3. F. Winkler, *Polynomial Algorithms in Computer Algebra*. Wien New York: Springer-Verlag, 1996. ISBN:3211827595.

The Distortion Model

The basic standard model for barrel and pincushion distortion compensation is a radial distortion model given by the following expression:

$$\begin{pmatrix} x^* - x_c \\ y^* - y_c \end{pmatrix} = L(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}$$

where (x, y) are the original point coordinates (distorted), (x^*, y^*) are the corrected (undistorted) point coordinates, (x_c, y_c) is the center of the camera distortion model, usually the center of the image (in fact, in this paper we will always take the center of the image as distortion center), the distance for any image point to the center of the image is given by

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

and *L(r)* is the function defining the shape of the distortion model, which is usually approximated by a Taylor expansion as

$$L(r) = k_0 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots\ldots,$$

where the set

$$\mathbf{k} = (k_0, k_1, ..., k_{N_k})^T$$

gathers the distortion parameters. The complexity of the model is given by the number of terms of the Taylor expansion we use to approximate *L(r)*.

We use the general approach to determine *L(r)* by imposing the requirement that the projection of 3D lines has to be 2D straight lines in the image. This approach has been succesfully used in previous works (see reference 1.) to minimize the objective error functions which are expressed in terms of line equations or distance functions.

In this paper we propose a new fast technique to obtain the distortion parameter model using a new lens distortion measure error. The main advantage of our formulation is that it yields a general 4-degree polynomial in the distortion parameters **k** that can be minimized using powerful techniques of computer algebra in one step for the 3 parameter case

$$L(r) = k_0 + k_1 r + k_2 r^2,$$

which is, in most cases, sufficient for image distortion correction.

The lens distortion is included in the camera calibration model in the following manner: given a camera defined by a rotation matrix *R*, a focus **c**, and a *3x3* intrinsic matrix parameter *A*, the projection of a 3D point **X** in the camera is given by the following expression

$$\begin{pmatrix} x^* \\ y^* \\ 1 \end{pmatrix} = sA(R, -Rc) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix},$$

where *s* is the usual projective factor value. In the case of *L(r)* ≡ to 1, the camera model is lens distortion free and the above expression becomes the usual "pinhole" projective model.

We observe that lens distortion correction is performed in pixel image coordinates. In order that the lens distortion model be a radial function in pixels coordinates, we need to assume that the camera CCD sensor has square pixel. This square pixel size assumption is satisfied, in practice, for most modern digital cameras. On the other hand, this square size assumption could be removed if we use normalized coordinates instead of pixel coordinates, but to normalize the point coordinates we need to know the camera intrinsic parameters. Since we assume that camera intrinsic parameters are unknowns we have formulated the lens distortion correction in terms of pixel coordinates.

A New Energy Function to Measure the Distortion Error

Previous works correct the distortion by minimizing the RMS distance for all the *2D* projected points to the "a priori" known *N* straight lines (rects), rect of index *l* having N_l points. This minimization can be carried out through any optimization method (gradient-like). For these cases, the energy function is as follows:

$$D(\mathbf{k}) = \frac{1}{N} \sum_{l=1}^{N} \frac{1}{N_l} \sum_{i=1}^{N_l} \frac{\left(a_l x^*_{l,i} + b_l y^*_{l,i} + c_l\right)^2}{a_l^2 + b_l^2},$$

where ($x^*_{l,i}$, $y^*_{l,i}$) , are the undistorted points using the distortion model provided by **k**, and where $a_l\, x^* + b_l\, y^* + c_l$ is the line that minimizes *D(**k**)* for a given choice of points { ($x^*_{l,i}$,$y^*_{l,i}$) } {*i=1,....,Nl*}.

This energy function is not necessarily convex and, any optimizer will have to deal with an unknown

number of local miminizers, which, as a result, will provide a local solution, therefore, not the optimal. Since this energy function is widely applied, we have added it to our code for comparison.

Our proposal is quite different, and, as will be demonstrated through results, more efficient. We define an energy function to measure the distortion error built on statistical measure. Given the set of points, we calculate the covariance matrix *S* as

$$S^{*,l}(\mathbf{k}) = \begin{pmatrix} S^{*,l}_{xx} & S^{*,l}_{xy} \\ S^{*,l}_{xy} & S^{*,l}_{yy} \end{pmatrix} \equiv \frac{1}{N_l} \begin{pmatrix} \sum_{i=1}^{N_l}(x^*_{l,i} - \overline{x^*_{l,i}})^2 & \sum_{i=1}^{N_l}(y^*_{l,i} - \overline{y^*_{l,i}})(x^*_{l,i} - \overline{x^*_{l,i}}) \\ \sum_{i=1}^{N_l}(y^*_{l,i} - \overline{y^*_{l,i}})(x^*_{l,i} - \overline{x^*_{l,i}}) & \sum_{i=1}^{N_l}(y^*_{l,i} - \overline{y^*_{l,i}})^2 \end{pmatrix}.$$

From that, we propose a new disortion measure error as

$$E^*(\mathbf{k}) = \frac{1}{N}\sum_{l=1}^{N}\left(S^{*,l}_{xx}S^{*,l}_{yy} - \left(S^{*,l}_{xy}\right)^2\right).$$

It can be easily prooven (see reference 1.), that this new energy function is always positive and equals 0 if and only if for each rect its points are aligned. This is the functional to be minimized to correct the image distortion and the variables are the distortion parameters **k**.

Besides, if we expand this functional to account for every line and every point, this energy functional can be expressed as a 4-degree homogeneous polynomial in the variable **k** as

$$E^*(\mathbf{k}) = \frac{1}{N}\sum_{l=1}^{N}\mathbf{k}^T A^l \mathbf{k}\mathbf{k}^T B^l \mathbf{k} - \mathbf{k}^T C^l \mathbf{k}\mathbf{k}^T C^l \mathbf{k},$$

with

$$A^l_{m,n} = \frac{1}{N_l}\sum_{i=1}^{N_l}((r_{l,i})^m x_{l,i} - \overline{(r_{l,i})^m x_{l,i}})((r_{l,i})^n x_{l,i} - \overline{(r_{l,i})^n x_{l,i}}),$$
$$B^l_{m,n} = \frac{1}{N_l}\sum_{i=1}^{N_l}((r_{l,i})^m y_{l,i} - \overline{(r_{l,i})^m y_{l,i}})((r_{l,i})^n y_{l,i} - \overline{(r_{l,i})^n y_{l,i}}),$$
$$C^l_{m,n} = \frac{1}{N_l}\sum_{i=1}^{N_l}((r_{l,i})^m x_{l,i} - \overline{(r_{l,i})^m x_{l,i}})((r_{l,i})^n y_{l,i} - \overline{(r_{l,i})^n y_{l,i}}).$$

Obviously, the global minimum of the new functional corresponds to the trivial solution $\mathbf{k} = \mathbf{0}$. To avoid this problem, usually the first distortion parameter is set to one. As it is explained in the implementation section of this paper we use another approach: we fit the first distortion parameter using a zoom factor (see zoom normalization.txt) by minimizing the sum of the square distance between the distorted and undistorted points.

An Efficient Technique to Minimize the Proposed Energy Function

A detailed algebraic analysis of the new functional is described in [1], and, in this section, we briefly present the algebraic technique we apply to minimize the measure of the distortion error. For simplicity in the exposition, we present the results for polynomials with real coefficients, but it must be said that they are valid over more general polynomials rings.

As mentioned above, one needs to minimize the distortion error measure function which is a real polynomial in the variable **k**. Minimizing a polynomial in several variables can be reduced to compute the solutions of an algebraic system of equations, namely the one generated by its gradient. In our case,

$$\mathcal{S} := \left\{\frac{\partial \hat{E}(\mathbf{k})}{\partial k_i} = 0\right\}_{i=1,\ldots,N_k}$$

When the polynomial is univariate, say

$$a_0 + a_1 k_p + \ldots,$$

one just has to approximate the real roots of the univariate polynomial

$$\frac{\partial \hat{E}(k_p)}{\partial k_p}.$$

However, when more than one variable appears, the problem is not trivial. In order to approach this new situation, one can apply computer algebra techniques to prepare symbolically the algebraic system *S* before numerical methods are executed. The two-variable case can be treated by means of symbolic linear algebra techniques while the case of more than two variables requires, in general, abstract algebra techniques. In both cases, the underlining theory comes from algebraic geometry and commutative algebra. To be more precise, we first describe in detail how to approach the problem when two variables are considered, and afterward we give a brief description on how to proceed in the general case.

Let us assume that we are dealing with two variables (note that this is the case when working with two distortion parameters), and that the system *S* turns out to be

$$\mathcal{S} := \left\{ \frac{\partial \hat{E}(k_p, k_q)}{\partial k_p} = 0,\ \frac{\partial \hat{E}(k_p, k_q)}{\partial k_q} = 0 \right\}.$$

To compute the solutions of *S* we apply the so called resultant-based method. Let us shortly describe this algebraic method.

For this purpose, let $G_1(k_p,k_q)$, $G_2(k_p,k_q)$ be two bivariate polynomials with real coefficients. Choosing one variable, say k_q, as a main variable, we can rewrite $G_1(k_p,k_q)$, $G_2(k_p,k_q)$ as

$$\begin{aligned} G_1(k_p, k_q) &= a_n(k_p)k_q^n + \cdots + a_1(k_p)k_q + a_0(k_p), \\ G_2(k_p, k_q) &= b_m(k_p)k_q^m + \cdots + b_1(k_p)k_q + b_0(k_p), \end{aligned}$$

where $a_i(k_p)$, $b_i(k_p)$ are univariate polynomials with real coefficients, and $a_n(k_p)$, $b_m(k_p)$ are not identically zero, with $n > 0$, and $m > 0$.

In this situation, the **resultant** of G_1, G_2 with respect to the variable k_q (we write $\mathrm{Res}_{k_q}(G_1, G_2)$) is defined as the determinant of the $(n+m) \times (n+m)$ Sylvester matrix

$$\begin{pmatrix} a_n(k_p) & a_{n-1}(k_p) & \cdots & a_0(k_p) & 0 & \cdots & 0 \\ 0 & a_n(k_p) & a_{n-1}(k_p) & \cdots & a_0(k_p) & \cdots & 0 \\ \vdots & & \ddots & & & \ddots & \vdots \\ 0 & 0 & \cdots & a_n(k_p) & a_{n-1}(k_p) & \cdots & a_0(k_p) \\ b_m(k_p) & b_{m-1}(k_p) & \cdots & b_0(k_p) & 0 & \cdots & 0 \\ 0 & b_m(k_p) & b_{m-1}(k_p) & \cdots & b_0(k_p) & \cdots & 0 \\ \vdots & & \ddots & & & \ddots & \vdots \\ 0 & 0 & \cdots & b_m(k_p) & b_{m-1}(k_p) & \cdots & b_0(k_p) \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} m \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n \end{matrix}.$$

Note that the above **resultant** is a real univariate polynomial in only one variable, k_p, and the variable k_q has been eliminated. This is the situation when dealing with two distortion parameters, which makes this algebraic technique very appealing.

Summarizing, one can derive the following algorithm to compute the real solutions of

$$\mathcal{S} := \left\{ \frac{\partial \hat{E}(k_p, k_q)}{\partial k_p} = 0,\ \frac{\partial \hat{E}(k_p, k_q)}{\partial k_q} = 0 \right\}.$$

1. Determine

$$G_1 := \frac{\partial \hat{E}}{\partial k_p}, G_2 := \frac{\partial \hat{E}}{\partial k_q}.$$

2. Determine

$$G(k_p) := \mathrm{Res}_{k_q}(G_1, G_2)$$

and approximate the real roots of $G(k_p)$. Let $R = \{\alpha_1, \ldots, \alpha_s\}$ be the set of real roots of G.

3. For each $\alpha \in R$, approximate the common real roots of the univariate polynomials, $G_1(\alpha,k_q)$ and $G_2(\alpha,k_q)$. Let R_α be the set of these real common roots.
4. The real solutions of S are $\{(\alpha,\beta_\alpha)\,|\,\alpha \in \mathcal{R} \text{ and } \beta_\alpha \in \mathcal{R}_\alpha\}$.

In the general case, i.e. when working with $s > 2$ variables, say k_{p1}, ..., k_{ps}, the problem cannot be approached so directly by means of resultants. Nevertheless, one can apply Gröbner basis techniques or multivariate-resultants (reference 3. for further information). Certainly, Gröbner basis techniques can also be applied to the case of two variables but, in that case, we find more suitable the resultant-based method. The basic idea of Gröbner basis, as a tool for solving algebraic systems, is to provide a new algebraic system of equations equivalent to S (i.e. with the same solutions) but much simpler, and such that it has a suitable structure ("triangular") to compute the solutions. Roughly speaking, Gröbner basis can be seen as a generalization of the Gaussian elimination when the equations are not linear.

Online Demo

An on-line demo of the algorithm is available. The demo allows:

- uploading any colour image and to correcting the radial distortion by means of the proposed algebraic method;
- obtaining the set of distortion parameters and the value of the minimized energy functions associated to the method.

Algorithm

Zoom factor estimation

The distortion parameters **k** are computed setting $k_0 = 1$. In order to yield undistorted points as close as possible to the distorted ones we estimate a zoom factor to minimize the sum of the square distance between the distorted and the corrected (undistorted) points. Let s be such zoom factor, then we have:

$$\begin{pmatrix} x^* - x_c \\ y^* - y_c \end{pmatrix} = sL(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix},$$

We minimize:

$$H(s) = \sum_{l=1}^{N} \sum_{i=1}^{N_l} \left(x^*_{l,i} - x_{l,i}\right)^2 + \left(y^*_{l,i} - y_{l,i}\right)^2,$$

A straightforward computation leads to

$$H(s) = \sum_{l=1}^{N} \sum_{i=1}^{N_l} \left(s \sum_{j=0}^{N_k} k_j^n \left(r_{l,i}\right)^{j+1} - r_{l,i} \right)^2,$$

the minimum of the above function is reached at

$$s_{\min} = \frac{\sum_{l=1}^{N} \sum_{i=1}^{N_l} \sum_{j=0}^{N_k} k_j^n \left(r_{l,i}\right)^{j+2}}{\sum_{l=1}^{N} \sum_{i=1}^{N_l} \left(\sum_{j=0}^{N_k} k_j^n \left(r_{l,i}\right)^{j+1} \right)^2},$$

and finally we update the polynomial $L(r)$ (i.e. **k**) by multiplying all the coefficients by s_{min} (that is $k^n{}_j = s\, k^n{}_j \; \forall\, j$).

An interesting advantage of this approach is that the resolution of the undistorted image is similar to

the resolution of the original (distorted) image. This is a very useful property if we need to generate the undistorted image from the original distorted one.

Algorithm description

Therefore the derived algorithm for performing the numerical experiments can be structured in the following steps :

1. We compute the edges of the image using an edge detection algorithm with subpixel precision.
2. We select some collections of edge points corresponding to different 3D straight segments, that will be used to fit the distortion parameters.
3. We initialize $\mathbf{k} = (1,0,...,0)^T$.
4. We choose any pair p , $q \in Z$ ($1 \leq p, q \leq N_k$) and we optimize k_p, k_q using the proposed algebraic technique.
5. We update **k** using a zoom factor such that distorted and undistorted points are as close as possible.

Note that in the demo, the first step is not done in an automatic manner, hence, the user must select directly from the image the points supposed to be aligned.

Point coordinates normalization

It is well known that when we deal with algebraic methods it is usually better to normalize the point coordinates before computing the algebraic solution of the problem (see point coordinates normalization.txt). Following this strategy, as a first step, we normalize the edge points (x_i,y_i) using the transformation

$$x'_{l,i} = \frac{(x_{l,i} - x_c)}{A} \qquad y'_{l,i} = \frac{(y_{l,i} - y_c)}{A}$$

where A is given by

$$A = \sqrt{\frac{\sum_{l=1}^{N} \sum_{i=1}^{N_l} (x_{l,i} - x_c)^2 + (y_{l,i} - y_c)^2}{2(N_1 + N_2 + \cdots + N_N)}}$$

and we compute the distortion parameters k'_i for the normalized edge points $\{ x'_{l,i} , y'_{l,i} \}^N {}_{\{i=1\}}$.

Finally, in order to recover the distortion parameters k_i for the original edge points we have just to take into account that, following the above expression, we have

$$k_j = \frac{k'_j}{(A)^j}$$

Inversion of the radial distortion model

For some applications we need to invert the radial distortion model. For instance, to build the undistorted version of the image it is usually better to use the inverse of the radial distortion model. So we look at a radial function $G(r^*)$ such that

$$\begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} = G(r^*) \begin{pmatrix} x^* - x_c \\ y^* - y_c \end{pmatrix}$$

where

$$r^* = \sqrt{(x^* - x_c)^2 + (y^* - y_c)^2}.$$

From the above expression we obtain that

$$r = G(r^*)r^*.$$

On the other hand we have

$$\begin{pmatrix} x^* - x_c \\ y^* - y_c \end{pmatrix} = L(r) \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}$$

and therefore

$$r^* = L\left(G(r^*)r^*\right) G(r^*)r^*.$$

We conclude that $G(r^*)$ is a root of the polynomial

$$P(\lambda) = 1 - L(\lambda r^*)\lambda = 1 - \sum_{j=0}^{N_k} k_j r^{*,j} \lambda^{j+1}.$$

In order to minimize the distance between the distorted and undistorted points, we choose, among all possible real roots of $P(\lambda)$, the one nearest to 1. It can be seen in inversion of the radial distortion model.txt, an extract of the C code performing the inversion of the radial distortion model.

Examples

We present some examples to illustrate the perfomance ot the proposed method. For all the figures, the distorted original image is illustrated on the left and the undistorted (corrected) image is shown on the right image.

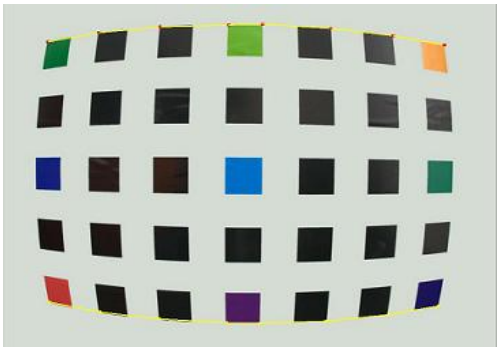
Calibration Pattern

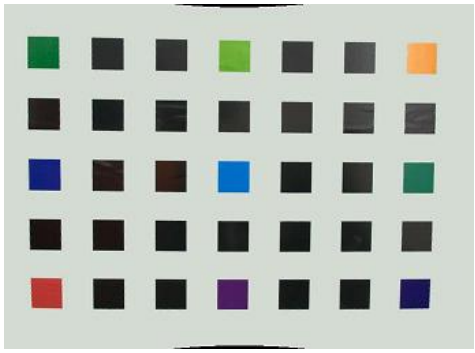
First, we start correcting the distortion for a synthetic case, that is, a calibration pattern (see figure below). The image has been obtained as follows: it has been printed out and we have taken a photo of it. The advantage of the calibration pattern is that we can easily identify the rectangles presented in the image, and automatically select the edge segments and points we will use for the estimation of the distortion model parameters.

The used primitives can be seen in the figure shown on the left as a yellow *line* joining the marked points selected by the user. Note that we know "a priori" that this yellow *line* must be straight. The primitives (selected points) for this image can be read in the file calibration pattern line primitives.txt. In this file, the number 1 (first line) accounts for the number of lines to be used by the program to correct the distortion (in this case it is only 1, but the number can be arbitrary). The following lines indicates the coordinates (pixels) of the selected points. From the result obtained, it seems that the method works well.

The black areas appearing at the top and at the bottom of the undistorted image are related to the correction procedure and can be removed just by selecting the inside region of interest. For this case, we present the numerical results after applying the algorithm. In the calibration pattern output file.txt file, the initial and final energy values and the initial (the trivial solution) and final distorted parameters can be seen. The RMS distance value is also written down to compare with. The CPU time is also added (measured on a 2.3 GHz Pentium IV machine).

distorted original image (calibration pattern) *undistorted original image (calibration pattern)*





Frame

The second result is for a colour photo showing an important barrrel distortion or pincushion distortion. For this case, the user has selected the top border of the frame as a *straight* reference line. The undistorted image once again reveals that, even selecting a simple primitive, the distortion is firmly corrected.

distorted original image (a picture) *undistorted original image (a picture)*

Building

The third example is a colour photo of a building (a church). This is an interesting case to appreciate that one can select any "a priori" known straight line, as it occurs here, where the user has selected two vertical lines, instead of the horizontal lines. The result shows the potential of the method.

distorted original image (a photo with vertical primitives)

undistorted original image

Source code

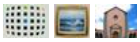
A strict ANSI C implementation is provided.

- extensively commented source code : zip, tar/gz
  The code is distributed under the GPLv3 licence.
- source code documentation : zip, tar/gz, online
- readme.txt file with a detailed explanation about how to compile and execute the program.

The code is self-contained, and the libraries for processing the images (read/write basic TIFF image procedures and calculating the lens distortion correction transformation) are also included. They have been implemented by the research group.

This is the original code used to get the numerical results appearing in reference 1. and it is the same code for the online demo.

image credits

 Luis Alvarez CC-BY