

Parameter-free Lens Distortion Calibration of Central Cameras

Filippo Bergamasco, Luca Cosmo, Andrea Gasparetto, Andrea Albarelli and Andrea Torsello
Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari - Venice, Italy

{filippo.bergamasco, luca.cosmo, andrea.gasparetto, andrea.albarelli, andrea.torsello}@unive.it

Abstract

At the core of many Computer Vision applications stands the need to define a mathematical model describing the imaging process. To this end, the pinhole model with radial distortion is probably the most commonly used, as it balances low complexity with a precision that is sufficient for most applications. On the other hand, unconstrained non-parametric models, despite being originally proposed to handle specialty cameras, have been shown to outperform the pinhole model, even with the simpler setups. Still, notwithstanding the higher accuracy, the inability of describing the imaging model by simple linear projective operators severely limits the use of standard algorithms with unconstrained models. In this paper we propose a parameter-free camera model where each imaging ray is constrained to a common optical center, forcing the camera to be central. Such model can be easily calibrated with a practical procedure which provides a convenient undistortion map that can be used to obtain a virtual pinhole camera. The proposed method can also be used to calibrate a stereo rig with a displacement map that simultaneously provides stereo rectification and corrects lens distortion.

1. Introduction

The pinhole camera model is by far the most widely used image formation model. The reasons behind its huge success among researchers and practitioners is due to several factors. First, its simplicity. In fact, the model is fully determined by its optical center and principal point. Given this basic information, the whole imaging process can be modeled as a direct projection of the scene onto the image plane. Second, the availability of mathematical tools. Its plain formulation allows to easily apply a wide spectrum of powerful and well understood mathematical tools, ranging from epipolar geometry to projective invariance of conics [6] and straight lines [10]. Finally, its wide range of applicability. Specifically, provided that the field of view of the camera fully intersects the image plane, a proper lens distortion model can be applied to approximate an ideal pinhole. Generally speaking, the role of the distortion model

is to describe how the incoming light rays diverge from the expected central direction once they enter the lens assembly of a real camera. In practice, since for any given lens configuration, the geometry that relates the optics to the image sensor is fixed, the distortion model is often defined just as a displacement function to be applied over the image plane. Moreover, due to physical considerations, many distortion models are radial; that is, they are defined as a displacement function that only depends on the distance from the principal point. This is the case with the seminal model proposed by Tsai [29], where the distortion function is modeled as a polynomial radial translation with two non-zero coefficients respectively for the second and fourth degree terms. Such model, albeit simpler than previous proposals [7, 8], obtained a large success, mainly because it was able to offer a good approximation of the imaging process and an easy calibration procedure was available since its introduction. For this reason, it has been embraced and extended several times in the last three decades. Zhang, when introducing his well-known calibration procedure [31], adopts the same model with slight modifications. More recently, Claus and Fitgibbon [9] proposed a rational function as a replacement for the original polynomial term. This latter approach is currently one of the most successful, probably because of its inclusion in the OpenCV library. Other recent approaches include variations on the number and type of parameters [18, 30, 11], the enforcement of projective invariants for parameter estimation [10, 5, 2], simultaneous calibration of multiple cameras [13, 12], and extensions designed to work with highly distorted cameras [24, 26]. Indeed, it is when addressing the need to calibrate fisheye and catadioptric cameras that the need to diverge from the pinhole model is felt the most, as any pinhole-based model, regardless of its level of sophistication, is geometrically unable to properly describe cameras exhibiting a view angle wider than 180 degrees. To overcome such limitation, several alternative parametric models have been proposed. Some of them try to modify the captured image in order to follow the original pinhole behaviour [16]. Others go through a totally different path by introducing novel image

formation models [17, 27]. Also catadioptric [1, 19] and light-field [3] systems have been widely covered in the literature, with a large selection of calibration methods. Given the high variability of non-pinhole cameras, it has been proven to be very difficult to define a parametric distortion model able to accommodate the diverse behaviour of physical lenses. This hindrance has been addressed by introducing general camera models based on unconstrained rays (usually called raxels) [14, 25] as well as partially non-parametric distortion models such as [28] and [15]. For former approach, a remarkable recent contribution is given in [20], but the calibration is limited to a small set of 3D rays that have to be interpolated to comprise all the pixels in the image space. The non-parametric distortion approach of [15] is similar in spirit to our work but it is not directly comparable since the distortion is still modeled as a radial function depending on the distance from a point on the image plane. Similarly to [23], we also adopt a structured-light target strategy to obtain a dense non-parametric image plane undistortion. However, [23] only uses a single shot of the calibration target so it cannot increase the accuracy by averaging several target views nor be used to calibrate multiple cameras at once.

Traditionally, literature has deemed the unconstrained models and related calibration procedures a last resort to adopt only when traditional approaches fail for geometrical or methodological issues. This is due to the fact that their flexibility comes at the price of a higher calibration complexity and (sometimes) cruder approximations. Recent research shows that a fully unconstrained imaging model can be applied effectively to real-world pinhole [4] cameras obtaining better accuracy and without needing a complex calibration procedure. However, the advantages in terms of precision are partially offset by the inability to use the wide mathematical toolset devised for the pinhole model. This is unfortunate, since when dealing with standard cameras with no large distortions, the central model is still reasonable.

With this study, we try to fill the gap between traditional pinhole calibration techniques and unconstrained models. Namely, we propose a model where the only raxels constraint it to cross a common projection center. Under this assumption, after performing a proper calibration, it is easy to define a non-parametric displacement map over the image plane that can be used to move back and forth from the unconstrained model to a virtual pinhole camera. This, in turn, allows to exploit the full arsenal of tools designed to work with pinhole cameras. To this end, the contribution of this paper is threefold. First, we introduce an effective calibration procedure for the proposed semi-constrained model. Second, we define an optimal approach to create a virtual pinhole camera from the calibrated model. Finally, we show how to naturally extend the method to the calibration and rectification of a stereo pair.

2. Novelty and Application Scenario

Differently from many other non-parametric methods, our goal is not to cope with non-conventional imaging geometries. Rather, we are indeed dealing with central cameras, that is, imaging devices where rays are supposed to converge to a common optical center. This is a domain that has been traditionally addressed by pinhole models augmented with a radial distortion function. These models, and the related calibration methods, are so widely used that it is quite natural to wonder if it actually makes any sense to adopt more complex solutions. This question is partially answered by the continued interest in enhanced central distortion models [22], which is a statement of the active state of the research. An additional hint to the open nature of this problem, is also offered by recent research efforts toward the adoption of general distortion models [21], that in some cases can be estimated even with no direct recovery camera pose and pinhole parameters [23]. With respect to this latter method, the general goal could seem very similar to the technique we are introducing. In fact the authors propose an image-based undistortion estimation that can be performed on a single image without the need for an explicit assessment of intrinsic and extrinsic camera matrices. However, we would like to stress an important aspect: image-based methods are not actually devoid of projective parameters as the estimated distortion is affected by an homography H , mixing intrinsic and extrinsic parameters. Actually, the method proposed in [23] obtains an estimate for H assuming the distortion to be zero around the image center, which is indeed a quite strong assumption. Unfortunately, this makes it difficult to effectively combine more than one target view. This is the reason because only one pose is used. Conversely, we explicitly separate the pose-invariant camera rays from the pose-dependent target orientation, thus offering a geometrically sound constraint among multiple exposures. Such constraints become even tighter when simultaneously accounting for multiple cameras, which is enabled by design with our method. To this end, our approach is well suited in all the scenarios where the high accuracy of general models is sought, still the notion of an image plane is needed. This is the case for most setups where epipolar geometry is exploited, including (for instance) structured-light scanning, stereo reconstruction and multiple camera tracking. It should be also noted that these kind of applications are actually the ones that benefit the most from the ability of our method to easily compute a unified rectification and undistortion map that can span multiple cameras.

Finally, the experimental assessment of the actual impact of the model is also an useful contribution of this paper. Specifically, we show that our hybrid approach yields an enhanced performance with respect to the parametric distortion model, while retaining the aforementioned advantages related to the availability of an actual image plane.

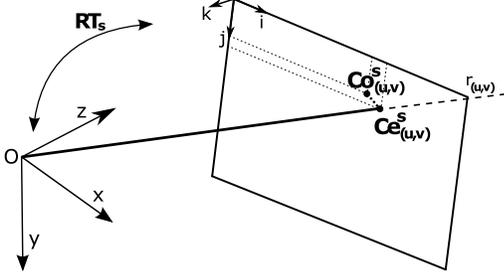


Figure 1. Schema of the optimized camera model involving the optical center o , the ray direction, the observed and expected code and a target pose.

3. Unconstrained Distortion Calibration

To estimate a dense non-parametric lens distortion we start by recovering the 3D light rays associated to each image pixel. Specifically, we formalize the light path entering the lens and hitting the sensor at discrete pixel coordinate (u, v) with the straight line $r_{(u,v)} = (o, d_{(u,v)})$ passing through the point o and oriented along the direction $d_{(u,v)}$. The common point o constrains our model to be central while no restriction is enforced on the directions $d_{(u,v)}$. Also, the uniform-spaced grid of the CCD provides an ordering on the rays spatial topology.

Since the model implies 3 degrees of freedom for each pixel, plus additional 3 for the optical center o , standard point-to-point calibration approaches like [31, 29, 9] cannot provide enough data to recover a valid solution. We solve this by adopting the dense structured-light target proposed in [4]. Specifically, we use an high-resolution LCD screen displaying phase shift patterns to uniquely localize each target point seen by the camera.

This has a strong advantage with respect to common calibration targets composed by a discrete set of features: unlike methods based on corner or ellipse localization, we can reason in terms of discrete camera coordinates. Indeed, for each camera pixel (u, v) a precise sub-pixel localization of the 2D target-space coordinate of the observed point can be recovered from the phase unwrapping process. In addition, the coding theory approach is robust against possible visual artifacts that may appear on target surface like illumination gradients, specular highlights and shadows. Finally, we take advantage of the precise manufacturing of modern LCD displays to get an affordable target far more accurate than the ones created through ink printing process.

To estimate the optical center and the direction of each ray, the calibration target is exposed to the camera in different positions and orientations. We denote with \mathbf{RT}_s the 3×4 matrix describing the roto-translation of the target with respect to the camera in the pose s , assuming the target reference frame located in the upper left monitor pixel with $\vec{i}, \vec{j}, \vec{k}$ versors being respectively the monitor columns, rows

and normal. For each pose, let $\mathbf{Co}_{(u,v)}^s \in \mathbb{R}^2$ be the code observed by the ray $r_{(u,v)}$ in shot s .

Since the LCD geometry and the displayed codes are known, we can take advantage of a pose \mathbf{RT}_s to map each code to a 3D point in the camera space, viceversa. For instance, the intersection between a ray $r_{(u,v)}$ and the target plane defined by a pose \mathbf{RT}_s yields to the expected code $\mathbf{Ce}(r_{(u,v)}, \mathbf{RT}_s) \in \mathbb{R}^2$ that the ray should have observed (Fig. 1).

3.1. Single Camera calibration

Following [4], we recover the geometry of the rays entering the camera as the generalized least-squares minimization problem:

$$\operatorname{argmin}_{r_{(u,v)}, \mathbf{RT}_s} \sum_{u,v,s} (\varepsilon_{(u,v)}^s)^T (\Sigma_{(u,v)}^s)^{-1} \varepsilon_{(u,v)}^s \quad (1)$$

where $\varepsilon_{(u,v)}^s = \mathbf{Co}_{(u,v)}^s - \mathbf{Ce}(r_{(u,v)}, \mathbf{RT}_s)$ are the residuals on the target plane between the observed and expected codes and $(\Sigma_{(u,v)}^s)^{-1}$ is the error covariance matrix for the given ray-pose combination that accounts for errors heteroscedasticity in the image plane.

In our setting, we aim to simultaneously minimize the optical center o , the direction $d_{(u,v)}$ of all rays and the pose \mathbf{RT}_s for each exposure of the target. Similarly to [4], we can also take advantage of the conditional independence of the parameters to implement an alternating optimization scheme that seeks optimal o and $d_{(u,v)}$ assuming last estimation of \mathbf{RT}_s fixed, and vice-versa. While our optimization involves less parameters, the optimization itself is more complex since the common optical center introduces a coupling between the rays which cannot be estimated independently anymore. As a consequence, the rays optimization step simultaneously estimates the optical center o and the ray directions $d_{(u,v)}$ given all the poses. For the pose estimation step we adopt the same ICP-based optimization introduced in [4]. The former step is discussed in detail in section 3.1.1 while, for the latter, we refer the reader to the original paper.

To start the alternating optimization we need a good initial approximation for the involved parameters. To this end, we gather a set of 3D-2D point correspondences assuming a discrete grid of target points similar to what can be commonly obtained with a chessboard. Then, we use *calibrateCamera* function provided by OpenCV to obtain target poses for each exposure and the direction of each ray. Note that starting condition is not a critical aspect since most baseline calibration methods are more than adequate to provide a reliable initial configuration, especially when dealing with cameras that can be assumed to be almost central (differently, the method proposed in [4] should be preferred). In addition, the iterative method is based on a least-square minimization which is guaranteed to converge to a (possibly local) minimum.

3.1.1 Optical Center and Rays Direction Optimization

In the o and $d_{(u,v)}$ optimization step we consider target poses constant. Let

$$\mathbf{x}_{(u,v)}^s = RT_s \begin{pmatrix} \mathbf{Co}_{(u,v)}^s \\ 0 \\ 1 \end{pmatrix}$$

be the 3D coordinates of the observed code $\mathbf{Co}_{(u,v)}^s$ transformed through the pose RT_s . As shown in [4], the generalized least squares formulation with respect to the target coordinates corresponds to a linear least squares with the distance of each ray and its associated 3D point $\mathbf{x}_{(u,v)}^s$. We can formulate the estimation of the optical center o as:

$$\operatorname{argmin}_o \sum_{u,v} \min_{d_{(u,v)}} \sum_s \|(h_{(u,v)}^s)^T (I - d_{(u,v)} d_{(u,v)}^T)\|^2 \quad (2)$$

where $h_{(u,v)}^s = (\mathbf{x}_{(u,v)}^s - o)$, and the internal minimization apt to find the best $d_{(u,v)}$ minimizing the sum of squared distances between $r_{(u,v)}$ and all the $\mathbf{x}_{(u,v)}^s$. We start by re-writing the squared norm in (2) as $(h_{(u,v)}^s)^T (I - d_{(u,v)} d_{(u,v)}^T) h_{(u,v)}^s$ to obtain

$$\operatorname{argmin}_o \sum_{u,v} \sum_s \|h_{(u,v)}^s\|^2 - \max_{d_{(u,v)}} \sum_s (d_{(u,v)}^T h_{(u,v)}^s)^2 \quad (3)$$

Let $\bar{x}_{(u,v)}$ be the centroid of the point cloud generated by the intersections of the ray $r_{(u,v)}$ and the target for each observed pose. Also, let $\bar{h}_{(u,v)} = (\bar{x}_{(u,v)} - o)$ be the distance vector between o with such centroid. By expressing $h_{(u,v)}^s$ as the summation of the two components:

$$h_{(u,v)}^s = (\mathbf{x}_{(u,v)}^s - \bar{x}_{(u,v)}) + \bar{h}_{(u,v)}$$

and expanding the formulation in (3) we obtain:

$$\operatorname{argmin}_o \sum_{u,v} N_{(u,v)} (tr(\mathbf{S}_{(u,v)}) + \|\bar{h}_{(u,v)}\|^2) - \quad (4)$$

$$- \max_{d_{(u,v)}} d_{(u,v)}^T (N_{(u,v)} \mathbf{S}_{(u,v)} + N_{(u,v)} \bar{h}_{(u,v)} \bar{h}_{(u,v)}^T) d_{(u,v)} \quad (5)$$

where $\mathbf{S}_{(u,v)}$ and $N_{(u,v)}$ are respectively the covariance matrix and the size of the point cloud generated by $r_{(u,v)}$.

Since we start our optimization with a configuration close to the optimum, we expect that the distance between each ray and its expected code is as small as few target pixels. This implies that the spatial extent of each point cloud is order of magnitude smaller than the distance $\|\bar{h}_{(u,v)}\|_2$. Under this assumption, an approximate maximizer for (5) is given by

$$d_{(u,v)} = \frac{\bar{h}_{(u,v)}}{\|\bar{h}_{(u,v)}\|^2} \quad (6)$$

By substituting (6) into (4) and (5), after some simplifications, we obtain the following alternative formulation

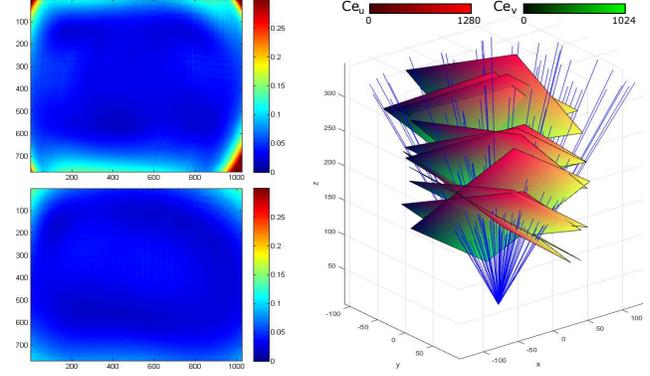


Figure 2. Left: RMSe between observed and expected codes for each $r_{(u,v)}$ at the first (top) and last (bottom) iteration. Right: Rays and calibration target poses recovered by the optimization. Only a subset of the camera rays are plotted for visualization purposes.

$$\operatorname{argmax}_o \sum_{u,v} N_{(u,v)} \frac{\bar{h}_{(u,v)}^T \mathbf{S}_{(u,v)} \bar{h}_{(u,v)}}{\|\bar{h}_{(u,v)}\|^2} \quad (7)$$

Problem (7) cannot be solved in a closed form. To provide a good approximate solution we compute the derivative with respect to o :

$$\frac{\partial}{\partial o} \sum_{u,v} N_{(u,v)} \frac{\bar{h}_{(u,v)}^T \mathbf{S}_{(u,v)} \bar{h}_{(u,v)}}{\|\bar{h}_{(u,v)}\|^2} = \sum_{u,v} 2N_{(u,v)} K_{(u,v)} \bar{h}_{(u,v)} \quad (8)$$

$$K_{(u,v)} = \frac{\left(-\mathbf{S}_{(u,v)} \|\bar{h}_{(u,v)}\|^2 + \mathbf{I} (\bar{h}_{(u,v)}^T \mathbf{S}_{(u,v)} \bar{h}_{(u,v)}) \right)}{\|\bar{h}_{(u,v)}\|^4} \quad (9)$$

If $K_{(u,v)}$ is known, o can be obtained by setting to zero Equation (8) and solving the resulting linear system:

$$\sum_{u,v} 2N_{(u,v)} K_{(u,v)} o = \sum_{u,v} 2N_{(u,v)} K_{(u,v)} \bar{x}_{(u,v)} \quad (10)$$

Since $K_{(u,v)}$ is itself a function of o , the maximization problem (7) is tackled iteratively by computing $K_{(u,v)}$ with the estimate of o at iteration $t-1$ and then solving (10) to obtain a new estimate at iteration t and repeating this process until $\|o^{(t)} - o^{(t-1)}\| < \epsilon$. When the optical center is found, the direction of each ray is computed with equation (6). A qualitative result of the effect of the optimization process is shown in Fig. 2.

3.1.2 From ray bundle to virtual pinhole camera

After the optimization of rays, optical center and poses we obtain a detailed model describing the light path entering the camera. Next, we need to choose a convenient image plane that define the intrinsic parameters of a new virtual pinhole camera, along with a non-parametric dense undistortion function to be applied to the acquired images.

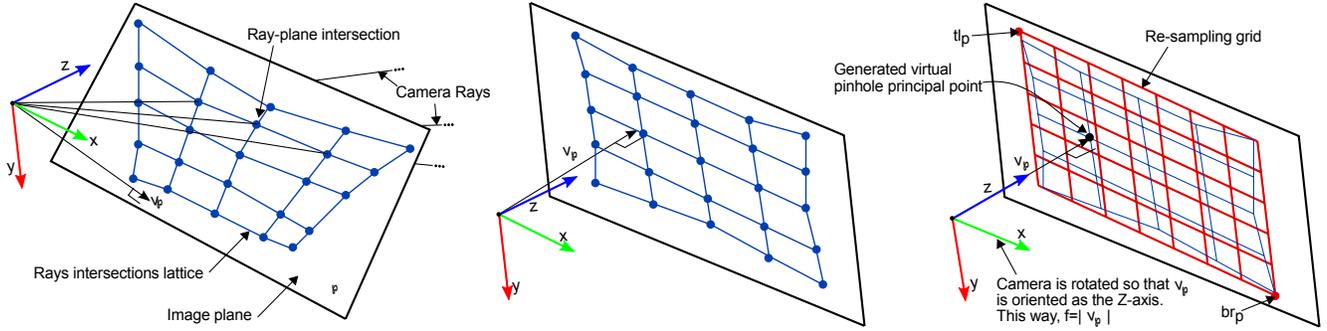


Figure 3. Left: the intersection of the camera rays with an image plane φ generates a projective distorted lattice. Center: An optimal plane is estimated to let the lattice be as regular as possible. Right: Camera reference system is rotated so that the image plane is orthogonal to the optical axis. Then, the lattice is re-sampled on an uniform grid to compute the undistortion function.

As a preliminary step, all rays are translated so that their unique intersection point o lies at the origin. After that, we define a plane φ as the locus of points x satisfying $\langle x - v_\varphi, n_\varphi \rangle = 0$, with $n_\varphi = \frac{v_\varphi}{\|v_\varphi\|}$. As soon as an image plane φ is chosen, it generates a virtual camera with the z-axis oriented as the vector n_φ and with a focal length $f = \|v_\varphi\|$. When choosing any φ intersecting the ray bundle, all the intersection points inherit the lattice topology from the camera sensor (Fig. 3, Left). However, the lattice projective distortion and size are affected by the orientation and distance of φ , respectively. Through an optimization process we first find the best image plane to minimize the lattice projective distortion. Then, we generate the image undistortion function by re-sampling the lattice in a uniform grid. The grid position and size will define the projection of the optical center on the image plane (i.e. the pinhole parameters c_x and c_y) and the undistorted image size (see Fig. 3 for a complete description of the process).

3.1.3 Image Plane Estimation

Choosing a good plane φ is crucial to ensure the lattice be as regular as possible and increase the quality of the subsequent re-sampling and interpolation processes. The optimal image plane is computed in two steps. First, we estimate the plane orientation n_φ to minimize the variance of the squared distance between each plane-ray intersection point and its neighbours. This ensure the lattice to be as regularly shaped as possible (Fig. 3, Center). After that, the scaling factor $\|v_\varphi\|$ (i.e. the plane distance from the origin) is computed so that the average distance between all the points is equal to 1.

Let $I_d \subset \mathbb{R}^2$ be the set of (u, v) indices of the rays. Let $U(i \in I_d) = U(u, v) = \{(u-1, v), (u+1, v), (u, v-1), (u, v+1)\}$ the function defining the set of four neighbours of a ray indexed by i . The squared distance between the 3D intersections generated by rays r_i and $r_{j \in I_d}$ with a plane φ lying at unitary distance from the origin is given by:

$$D_{i,j}^2 = \left\| \frac{d_i}{n_\varphi^T d_i} - \frac{d_j}{n_\varphi^T d_j} \right\|^2 \quad (11)$$

Consequently, the variance of the squared distances $D_{i,j}^2$ between each ray and its neighbours is given by the function

$$f_D = \sum_i \sum_{j \in U(i)} \left(D_{i,j}^2 \right)^2 - \left(\sum_i \sum_{j \in U(i)} D_{i,j}^2 \right)^2 \quad (12)$$

We cast the plane orientation problem as

$$\begin{aligned} \operatorname{argmin}_{n_\varphi} \quad & f_D \\ \text{such that} \quad & \|n_\varphi\| = 1 \end{aligned} \quad (13)$$

solved via geodesic steepest descent. We start with an initial estimate of $n_\varphi^{(0)} = (0 \ 0 \ 1)^T$. For each iteration t , we update the estimate of $n_\varphi^{(t)}$ enforcing the constraint of $\|n_\varphi\| = 1$ by rotating $n_\varphi^{(t)}$ around the rotation axis $\Psi = \nabla f_D^{(t-1)} \times n_\varphi^{(t-1)}$ for an angle $\theta = \lambda \min(\|\Psi\|, \epsilon)$. The constant λ affects the speed of the gradient descent while ϵ gives an upper bound on the amount of rotation to avoid instabilities.

To perform effectively the optimization, ∇f_D can be analytically computed as follows:

$$\begin{aligned} \nabla f_D = & \sum_i \sum_{j \in U(i)} 2D_{i,j}^2 \frac{\partial}{\partial n_\varphi} D_{i,j}^2 - \\ & - \left(\sum_i \sum_{j \in U(i)} D_{i,j}^2 \right) \left(\sum_i \sum_{j \in U(i)} \frac{\partial}{\partial n_\varphi} D_{i,j}^2 \right) \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial}{\partial n_\varphi} D_{i,j}^2 = & \frac{2}{(n_\varphi^T d_i)^2} \left(\frac{d_i^T d_j}{n_\varphi^T d_j} - \frac{d_i^T d_i}{n_\varphi^T d_i} \right) d_i + \\ & + \frac{2}{(n_\varphi^T d_j)^2} \left(\frac{d_j^T d_i}{n_\varphi^T d_i} - \frac{d_j^T d_j}{n_\varphi^T d_j} \right) d_j \end{aligned} \quad (15)$$

Once n_φ has been recovered, we set $f = \|v_\varphi\| = \frac{1}{2N} \sum_i \sum_{j \in U(i)} D_{i,j}$ where N is the number of lattice edges.

3.1.4 Generating the Undistortion Map

Once an optimal plane has been found, we setup an interpolation procedure to re-sample the points on a regular grid. Let $p_{(u,v)}$ be the intersection of $r_{(u,v)}$ with the optimized plane φ . First, all the points $p_{(u,v)}$ are rotated around the origin so that the vector v_φ coincides with the z-axis (Fig. 3, Right). After discarding the third component of all the points (all equal to 1 after the rotation), we compute the integral coordinates of the top-left $tl_p \in \mathbb{Z}^2$ and bottom right $br_p \in \mathbb{Z}^2$ corners of the containing bounding-box. At this point, we can provide the intrinsic matrix of the new virtual pinhole camera as:

$$K = \left(\begin{array}{cc|c} \|v_\varphi\| & 0 & -tl_p \\ 0 & \|v_\varphi\| & \\ \hline 0 & 0 & 1 \end{array} \right) \quad (16)$$

The undistorted image associated to the camera described by K corresponds to a unit-spaced grid inside the area of the bounding box. This leads to the construction of a dense displacement function $U_d : B \rightarrow \mathbb{R}^2$ that maps the coordinates of the output undistorted image $B \subset \mathbb{N}^2$ to sub-pixel coordinates of the input image¹.

To produce the displacement map U_d , we generate the quadrilaterals $q_1 \dots q_n$ formed when considering the 4-neighbours connectivity of the points $p_{(u,v)}$ with the topology induced by the rays lattice. For each quadrilateral q_i , we compute the homography H_i transforming the inner space bounded by its four vertices into the square defined by the CCD location of the four rays associated to each vertex. Then, the displacement map U_d can be obtained by:

$$U_d(u', v') = H_{Q(u', v')}(u', v', 1)^T \quad (17)$$

where $Q(u', v')$ is the function that returns the index of the quadrilateral containing the point $(u' \ v')^T$, if exists.

3.1.5 Filtering data outliers

Apart for being central, our model gives no constraint on the relative position of the rays. As a consequence, erroneous data caused by failures in the phase decoding process may lead to outliers in the ray estimation. Since no regularization is involved, we included a data filtering step at each alternation of rays-poses optimization. Specifically, we define the function $E(u, v)_s : I_d \rightarrow \mathbb{R}$ as the point-line distance between the ray $r_{(u,v)}$ and the point $x_{(u,v)}^s$. We then filter the observed codes $\mathbf{Co}_{(u,v)}^s$ by considering the median of the error function E in a squared neighbourhood of each point (u, v) . If $E(u, v)_s$ is greater than κ times the median, $\mathbf{Co}_{(u,v)}^s$ is marked as invalid and not used any more in the subsequent iterations. Rays with less than 5 observed codes

¹The obtained undistorted image has the same size of the bounding box

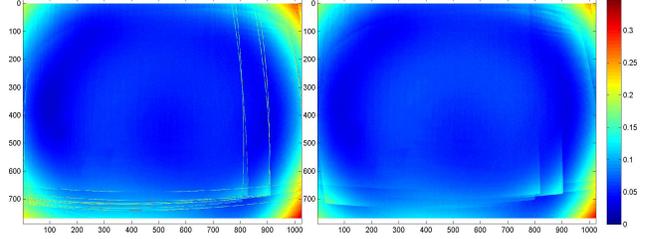


Figure 4. The effect of codes filtering step displayed by accumulating the value $E(u, v)$ among all the poses s . Left: original data may contain clear outliers near the boundary of the target. Right: after the filter almost all the spikes are no more visible. Note that remaining artefacts are not topological outliers (as for 3.1.5) and are probably due to the target non-planarity at the corners.

are completely removed from the optimization. A qualitative example of the output of the filtering process is shown in Fig. 4. Even if we filter erroneous observed codes, it may happen to obtain some quadrilaterals q_i for which the topological order of the vertices is not coherent with the order of the relative rays. Since this would lead to a non-injective displacement map U_d , such rays are marked as outlier and subsequently replaced by a linear interpolation performed over all the unmarked neighbours.

3.2. Dealing with Stereo Cameras

Since each ray acts independently with respect to the others, our approach can be easily extended to simultaneously calibrate and rectify a stereo rig. The pose optimization step remains exactly the same with the foresight to merge the two bundle of rays associated to each camera. Conversely, the optical centre and rays direction optimization can be performed independently on the two sets operating the same instance of target poses.

As a starting configuration for the subsequent optimization we performed the intrinsic and extrinsic calibration of the camera rig using the function provided by OpenCV library. Then, we are creating a single virtual imaging device by adopting the reference frame of the first camera for both. At the end of the optimization, we obtain an estimate of the two optical centres o_1 and o_2 and the directions of the rays in the two bundles. From this point, we roto-translate the rays to let o_1 coincide with the origin and the epipole $e = (o_2 - o_1)$ being oriented along the x-axis.

3.2.1 Rectification and Undistortion Map

If we constrain the image plane optimization so that n_φ remains orthogonal to e , the estimated plane would have the property to keep all the epipolar lines for the left and right cameras being parallel. To achieve this, we slightly modify the optimization discussed in section 3.1.3 by fixing the rotation axis $\Psi = \frac{e}{\|e\|}$ and the rotation angle to $\theta = \lambda \min(\langle \nabla f_D^{(t-1)}, \Psi \times n_\varphi^{(t-1)} \rangle, \epsilon)$.

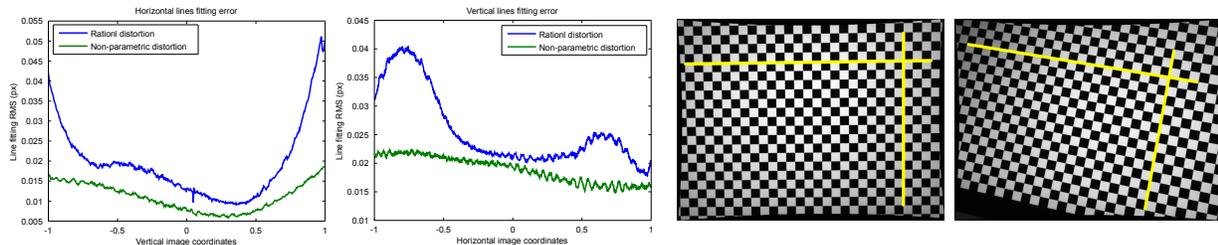


Figure 5. Left plots show a comparison of our non-parametric distortion model compared with Fitzgibbon’s rational model. On the right, two examples of an undistorted images generated with our method. Note that black artifacts on the edges are due to the lack of clipping after undistortion (yellow guides have been superimposed in order to help to better appreciate rectification).

After image plane optimization, two sets of points are generated by the intersection of the two ray bundles with the plane. The set of points generated by the right camera is translated in the opposite direction of the x -axis by the length of the baseline $T = \|e\|$ to let the right optical center coincide with the left one. Subsequently, two different bounding boxes are generated with the two sets of points. The height of the two boxes (i.e. the vertical coordinates of the top-left and bottom-right corners) are forced to be equal so that the epipolar lines are coincident with the rows of the two images. To keep the largest available common area between the two images, the left edge of the merged bounding box is taken from the bounding box of the right point set. Symmetrically, the right edge is taken from the left point set. Note that, this way, the intrinsic matrices of the two cameras are exactly the same.

Finally, we compute the re-projection matrix

$$Q = \left(\begin{array}{ccc|c} 1 & 0 & 0 & -tl_p \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & \frac{\|v_\phi\|}{\|v_\phi\|} \\ 0 & 0 & 1/T & 0 \end{array} \right) \quad (18)$$

so that, given any dimensional image point (u', v') and its associated disparity d , it can be projected into four-dimensional projective space with

$$(x \ y \ z \ w)^T = Q (u' \ v' \ d \ 1)^T \quad (19)$$

4. Experimental Section

In order to assess the performance of the proposed approach, we compared it against the unconstrained model [4] and the rational distortion model proposed by Claus and Fitzgibbon² [9] in both single camera and stereo setups³.

Our test setup included two PointGrey Flea3 1Mp grayscale cameras with approximately 60° field of view, fastened to a common rigid bar with a disparity of about 5cm. The calibration target was a 380×304 mm commercial LCD monitor with a resolution of 1280×1024 pixels.

²The most accurate method available from the OpenCV library

³Source code and datasets available at: *anonymized URL*

The cameras have been calibrated using a set of 20 shots and tested over a set composed of 40 different shots of the same active target. These shots have been acquired both with a single camera and with the complete camera pair, taking care to cover as much as possible of the respective fields of view. The target has been acquired at random poses with a distance from the cameras ranging from roughly 100 to 300 mm and a rotation with respect to the optical axis ranging from 0 to about $\frac{\pi}{4}$ radians. (Fig. 2, Right) Using the same data sets, we performed three different calibrations, using respectively the fully unconstrained model (Unconstrained), the non-parametric distortion proposed in this paper (Non-Parametric) and the rational distortion (Rational). Average convergence time for both Unconstrained and Non-Parametric calibration was about 10 minutes, while Rational always required less than 2 minutes.

Finally, we compared the performance of these methods by means of two experiments, assessing respectively the ability of providing a strictly projective virtual camera and to perform an accurate triangulation in the 3D space.

4.1. Image Undistortion

With this experiment we are testing the quality of the undistortion, that is how well the virtual pinhole camera obtained with the different methods approximate an ideal projective geometry. To this end, we exploited the projective invariance of straight lines. Specifically, for each horizontal and vertical scanline of the undistorted camera we collect the observed codes and we fit a straight line on them. Since we can assume the screen pixels to be regular and equally spaced, better pinhole approximation should exhibit a lower RMS error to the fitted line. Since a virtual pinhole cannot be produced with the fully unconstrained model, in Fig. 5 we plotted only the results for the Non-Parametric and the Rational model. While both methods are affected by some error, it is clear that the approximation given by the Non-Parametric approach yields less distorted lines. Furthermore, the structured nature of the RMS error obtained with the Rational model strongly suggests a systematic error due to the inability of the model to properly fit the data.

Note that, while our approach yields a much smaller

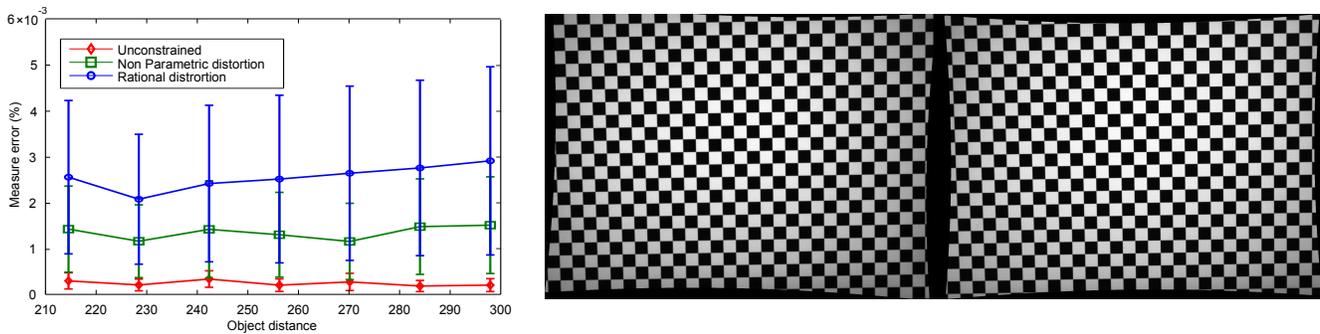


Figure 6. In the left-most plot we show the average relative error between the expected and measured distance of two triangulated points. In the right we show an example of a stereo-rectified image pair.

undistortion error than the Rational model, there is still a strong spatial coherency in the error. We think the non-centrality of the camera to be the predominant error source, since the RMS consistently grows towards the image boundaries. We suggest that the non-planarity could also contribute, being the borders of the target usually located near the border of the captured image.

4.2. 3D Measurement

In our second experiment we investigate the calibration quality of the camera pair. Using the rectified image pair, we triangulate the 3D position of two random screen pixels observed by both cameras. We repeated the experiment for several shots with different positions of the target screen. In Fig. 6 we plotted the average relative error between the expected and measured distance of two triangulated points with respect to the distance of the target from the camera pair. In this case, the Unconstrained model shows the best performance as it produces a lower error at any distance and more repeatable measures. The Non-Parametric model exhibits a slightly higher error. This proves that the additional constraint hinders a perfect calibration. Still it is noticeably more reliable than the Rational model, thus it can be



Figure 7. Reconstructed range-map triangulated with OpenCV calibration and rectification (Left) and our proposed method (Right).

deemed as a reasonable alternative to the totally free model when high accuracy is needed, but it is not desirable to lose the advantages of the pinhole model. Finally, in Fig. 7 we give a qualitative example of a reconstructed range-map after stereo rectification provided by the OpenCV *stereoRectify* function and our calibration pipeline. The better alignment of epipolar lines with image rows gives a more precise and dense reconstruction, especially on grazing surfaces.

5. Conclusions

In this paper, we proposed a new calibration technique to model a central camera with a distortion described as a non-parametric dense displacement map on the input image. This approach combines the simplicity of a central camera model, enabling the usage of powerful projective geometry tools, while sharing the ability of unconstrained models to accommodate subtle lens characteristics that cannot be generalized by parametric functions (A synthetic example is shown in Fig. 8). Moreover, the independence of each ray entering the camera can be exploited to calibrate a stereo setup with minor modifications on the process itself. By means of a couple of specially crafted experiments, we show that this hybrid approach works better than standard radial models and can be used to enhance the accuracy of quasi-central cameras without the hindrances associated with non-parametric approaches. In this rather common scenario, our method can be used as a drop-in replacement providing a significant boost without any added drawback.

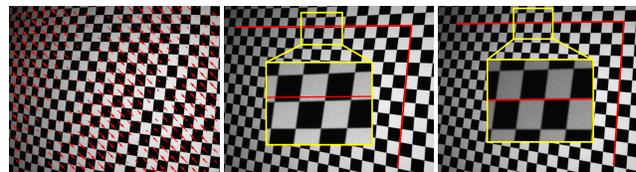


Figure 8. A small synthetic sinusoidal pixel displacement has been further applied to the acquired images (left). As expected, the rational model is unable to handle this distortion (center), which can be corrected with our non-parametric distortion model (right).

References

- [1] D. G. Aliaga. Accurate catadioptric calibration for real-time pose estimation of room-size environments. In *International Conference of Computer Vision*, pages 127–134, 2001.
- [2] F. Bergamasco, A. Albarelli, L. Cosmo, E. Rodola, and A. Torsello. An accurate and robust artificial marker based on cyclic codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(12):2359–2373, 2016. cited By 3.
- [3] F. Bergamasco, A. Albarelli, L. Cosmo, A. Torsello, E. Rodolà, and D. Cremers. Adopting an unconstrained ray model in light-field cameras for 3d shape reconstruction. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:3003–3012, 2015. cited By 4.
- [4] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello. Can a fully unconstrained imaging model be applied effectively to central cameras? In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1391–1398, June 2013.
- [5] F. Bergamasco, A. Albarelli, and A. Torsello. Image-space marker detection and recognition using projective invariants. *Proceedings - 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2011*, pages 381–388, 2011. cited By 6.
- [6] F. Bergamasco, L. Cosmo, A. Albarelli, and A. Torsello. Camera calibration from coplanar circles. *Proceedings - International Conference on Pattern Recognition*, pages 2137–2142, 2014. cited By 5.
- [7] D. C. Brown. Decentering Distortion of Lenses. *Photometric Engineering*, 32(3):444–462, 1966.
- [8] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [9] D. Claus and A. W. Fitzgibbon. A rational function lens distortion model for general cameras. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 213–219, 2005.
- [10] F. Devernay and O. Faugeras. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *Mach. Vision Appl.*, 13(1):14–24, Aug. 2001.
- [11] O. Faugeras, L. Quan, and P. Sturm. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. *Patt. Anal. and Machine Intelligence, IEEE Transactions on*, 22(10):1179–1185, Oct 2000.
- [12] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–125. IEEE, 2001.
- [13] Y. Furukawa and J. Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009.
- [14] M. D. Grossberg and S. K. Nayar. A general imaging model and a method for finding its parameters. In *International Conference of Computer Vision*, pages 108–115, 2001.
- [15] R. Hartley and S. B. Kang. Parameter-free radial distortion correction with centre of distortion estimation. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1834–1841 Vol. 2, Oct 2005.
- [16] C. Hughes, P. Denny, E. Jones, and M. Glavin. Accuracy of fish-eye lens models. *Appl. Opt.*, 49(17):3338–3347, Jun 2010.
- [17] J. Kannala and S. S. Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1335–1340, Aug. 2006.
- [18] L. Ma, Y. Hen, and K. L. Moore. Flexible camera calibration using a new analytical radial undistortion formula with application to mobile robot localization. In *IEEE International Symposium on Intelligent Control*, pages 799–804, 2003.
- [19] O. Morel, R. Seulin, and D. Fofi. Catadioptric camera calibration by polarization imaging. In *Pattern Recognition and Image Analysis*, volume 4478 of *Lecture Notes in Computer Science*, pages 396–403. Springer, 2007.
- [20] S. Ramalingam and P. Sturm. A unifying theory for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016.
- [21] C. Ricolfe-Viala and A.-J. Sánchez-Salmerón. Robust metric calibration of non-linear camera lens distortion. *Pattern Recognition*, 43(4):1688 – 1699, 2010.
- [22] C. Ricolfe-Viala and A.-J. Sánchez-Salmerón. Using the camera pin-hole model restrictions to calibrate the lens distortion model. *Optics and Laser Technology*, 43(6):996 – 1005, 2011.
- [23] R. Sagawa, M. Takatsuji, T. Echigo, and Y. Yagi. Calibration of lens distortion by structured-light scanning. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 832–837, Aug 2005.
- [24] S. Shah and J. Aggarwal. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation*. *Pattern Recognition*, 29(11):1775 – 1788, 1996.
- [25] P. Sturm and S. Ramalingam. A generic calibration concept: Theory and algorithms. Tech.Report 5058, INRIA, dec 2003.
- [26] R. Swaminathan and S. K. Nayar. Nonmetric calibration of wide-angle lenses and polycameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1172–1178, Oct. 2000.
- [27] K.-H. Tan, H. Hua, and N. Ahuja. Multiview panoramic cameras using mirror pyramids. *Patt. Anal. and Machine Intelligence, IEEE Transactions on*, 26(7):941–946, July 2004.
- [28] J.-P. Tardif, P. Sturm, and S. Roy. Self-calibration of a general radially symmetric distortion model. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, volume 3954 of *Lecture Notes in Computer Science*, pages 186–199. Springer Berlin Heidelberg, 2006.
- [29] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.
- [30] J. Wang, F. Shi, J. Zhang, and Y. Liu. A new calibration model of camera lens distortion. *Pattern Recognition*, 41(2):607 – 615, 2008.
- [31] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, Nov. 2000.