

Uncertainty Analysis of the DLT-Lines Calibration Algorithm for Cameras with Radial Distortion

Ricardo Galego¹, Agustin Ortega², Ricardo Ferreira¹, Alexandre Bernardino¹,
Juan Andrade-Cetto², José Gaspar¹

Abstract

3D metric data of environmental structures is nowadays present in many information sources (maps, GIS) and can be easily acquired with modern depth sensing technology (RGBD, laser). This wealth of information can be readily used for single view calibration of 2D cameras with radial distortion, provided that image structures can be matched with the 3D data.

In this paper we present an analysis of the level of accuracy that can be obtained when such calibration is performed with the 2D-3D *DLT-Lines* algorithm. The analysis propagates uncertainty in the detection of features at the image level to camera pose, and from there to 3D reconstruction. The analytic error propagation expressions are derived using first order uncertainty models, and are validated with Monte Carlo simulations in a virtual indoor environment. The method is general and can be applied to other calibration methods, as long as explicit or implicit expressions can be derived for the transformation from image coordinates to 3D reconstruction.

We present results with real data for two applications: i) the 3D reconstruction of an outdoors building for which 3D information is given by a map, observed by a mobile phone camera; and ii) the uncertainty in the localization at the floor plane of points observed by a robot equipped with an RGBD camera navigating in a typical indoor environment.

Keywords: *DLT-Lines*, camera calibration, error propagation, uncertainty

¹R. Galego, R. Ferreira, A. Bernardino and J. Gaspar are with the Institute for Systems and Robotics at Instituto Superior Técnico, University of Lisbon, Portugal, {jag}@isr.ist.utl.pt.

²A. Ortega and J. Andrade-Cetto are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain, {cetto}@iri.upc.edu.

1. Introduction

Camera calibration is required in a wide range of applications. Geo-localization of detected events, measurement of people's height or vehicle's speed, associating events between cameras, are examples of tasks which can only be tackled if cameras are calibrated with respect to the coordinate system of a known scenario. Whereas calibrating camera translation can, in many cases, be performed by a simple measurement, calibrating rotation and intrinsic parameters are more difficult tasks. In many cases the selection of the viewing direction, zoom and focus is only done while mounting the camera. Calibration in-situ is thus convenient for many applications.

In-situ calibration with conventional calibration tools requires the use of calibration rigs or patterns which may be impractical in many cases. For instance the methods of Tsai [?], Heikkilä [?], Zhang [?], Kannala [?] and Bouguet [?] demand that the known pattern covers most of the imaged area, which requires an impractically large calibration pattern if the camera is mounted at a high position, far from the floor level. In addition, conventional calibration methodologies are mostly focused in the intrinsic parameters, and thus do not provide distances (rigid pose transformations, extrinsic parameters) among the various units within a camera network. In other words, they are not designed to provide a global coordinate system for all cameras.

In this work we consider a calibration methodology that instead of using a calibration pattern, uses the scene data imaged by a camera, mounted in place, and auxiliary data encoding some 3D structure about the imaged area (Fig. 1). In some cases the auxiliary data can be simply a map or an aerial image. In-situ calibration based on world data brings two benefits, (i) there is no need for calibration patterns, and (ii) the world data specifies a coordinate system which can be used by multiple cameras.

The most common type of data used for camera calibration is the correspondence between 3D scene points and their 2D image projections [? ?]. Considering pin-hole camera models, one finds that 3D points and their 2D projections are related by simple linear constraints in homogeneous coordinates that can be combined in the so-called *Direct Linear Transformation (DLT)* [? ?]. Particularly in urban scenarios, straight lines are omnipresent. This motivates the use of lines for calibration. In fact, using lines instead of points is advantageous specially because simple image processing tools for local line fitting can be used to reduce the

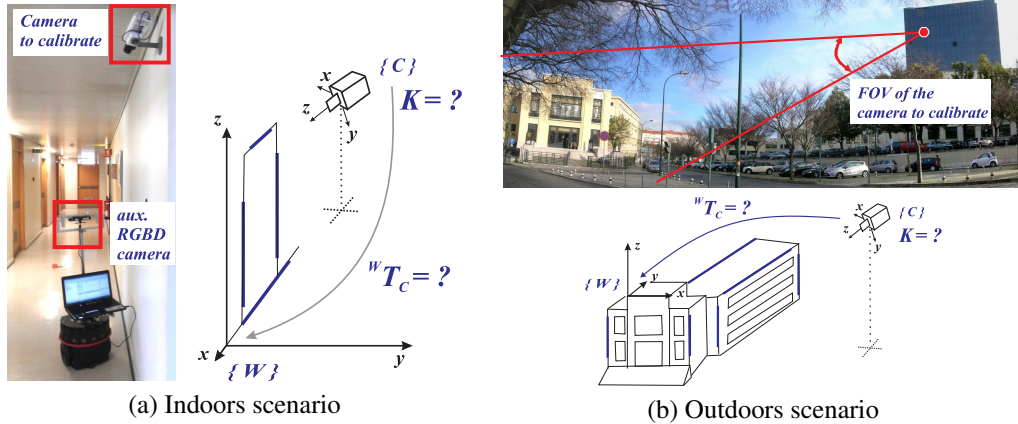


Figure 1: Indoors and outdoors camera calibration. (a) Calibration of a fixed camera using data acquired by an auxiliary mobile RGB-D camera. (b) Calibration of a fixed camera using the 3D structure of imaged buildings. The matching of 3D data (thick lines) with image data allows calibrating the fixed cameras $\{C\}$, and in particular estimating their poses ${}^W T_C$, and intrinsic parameters K .

matching error between 2D and 3D structures. The identification of corresponding lines in multi-modal data is less sensitive to errors than establishing point correspondences [? ?]. Interestingly, writing linear constraints for calibration from line correspondences, i.e. estimating the projection matrix, is as simple as with points [?]. Once the projection matrix is computed, the camera pose and intrinsic parameters are readily obtained by factorization [?]. The quality of the estimated parameters depends on a series of factors: structure of the observed environment, selected data matches, model parameterization and optimization criteria [?].

The ability to assess the uncertainty in the estimates of a calibration method is important not only to infer errors on 3D reconstruction but also as a means to validate and improve the calibration process. One way to do this is to propagate the uncertainty in the measurement process (variance in pixel coordinates of the feature points or lines used for calibration) to uncertainty values on the calibration parameters, either intrinsic, extrinsic, or both; and from these uncertainties, to that of the computed 3D reconstruction when such calibration values are used.

This paper presents such detailed noise propagation analysis for the case of the *DLT-Lines* camera calibration process. We derive expressions to compute estimates for the the covariance of the camera intrinsic parameters as well as estimates for the covariance of the camera pose. To do so, we compute a first order linearization of the calibration method around the obtained solution and use it to

perform first order error propagation [? ?]. We validate the obtained uncertainty with synthetic data based on Monte-Carlo simulations, and with real data from a non-overlapped camera network.

The uncertainty analysis is derived first for the *DLT-Lines* formulation presented in [?] and further extended to introduce novel cases of interest, namely the combination of *DLT-Lines* with square pixel constraints, and with radial distortion. Finally, the analysis is also extended for the case in which the scene can be considered to obey a floor plane constraint, typical of aerial imagery.

2. Related work

Cameras can be installed outdoors or indoors [? ?], and can be used to cover overlapped or non-overlapped shared views [? ?]. Overlapping fields of view usually allow matching scene features among different cameras and therefore provide extrinsic information for their calibration. Conversely, non-overlapped fields of view require alternative sources of extrinsic information.

State of the art techniques for camera calibration require the use of non-planar [?] or planar [?] patterns, usually made of points, lines or checkerboards [?], conics [?], or even ArTag markers [?]. Unfortunately, for large outdoor camera networks, calibration patterns of reasonable sizes often project on images with very small resolution, mainly because cameras are located at a considerable height with respect to the floor. In addition, a pattern-based independent calibration of each camera would require a secondary process to relate all camera coordinate systems to a global reference frame, but establishing this relation with small to null overlapping fields of view is nearly impossible. For planar scenarios, a DLT [? ? ?] suffices to estimate image to plane homographies [?]. In practice however, the planar scenario assumption is too restrictive as nonparallel locally planar surfaces, such as ramps and plazas, often occur in real urban environments.

An interesting in situ calibration methodology was proposed by Svoboda et al. [?]. This technique requires to use a bright moving spot to calibrate the camera network, simplifying the process with respect to the standard approaches. The technique assumes overlapping fields of view to estimate the epipolar geometry, to extract homographies, estimate depth, and finally compute the overall calibration of the camera network. However the method is not suited for outdoor scenarios or when the fields of view of the cameras do not overlap. Another alternative is to place the led light on a moving robot and to have a secondary robot equipped with a laser sensor track the first one, relating their position estimates to the camera network [?]. Yet another system that relies on tracking a moving object to

estimate the extrinsic parameters is [?], which assumes a constant velocity model for the target. Tracking a moving target each time the system needs recalibration might be prohibitive. The estimation of the camera location purely by analyzing cast shadows is also mathematically possible, but with very low position accuracy in practice [?], and if one is interested only in the topology of the network configuration and not in a metric calibration, multi-target tracking of people could also be an alternative [?]. In contrast to these approaches, we opt for a system that does not rely explicitly on a moving pattern or shadow to calibrate the network, and that produces accurate metric calibration, even in outdoor scenarios.

Analogously to use moving targets, one may have moving cameras, e.g. cameras encompassing controlled orientation changes (pan and tilt). Agapito et al. [?] use the properties of the image of the absolute conic to auto-calibrate such moving cameras. The computed intrinsic parameters, and the alignment of landmarks to image features allows to estimate also the extrinsic parameters [?]. However, camera networks are often composed of fixed cameras and therefore require scene information to compensate the lack of self-motion. Caprile and Torre [?] use the properties of vanishing points in two steps. First they estimate the focal length and the principal point of a camera, while in a second step they estimate the transformation between two cameras, with corresponding vanishing points. Another way to calibrate the camera network is to use a vertical vanishing point and the knowledge of a line in a plane orthogonal to the vertical direction on each camera image [?]. In [?] fundamental matrices are used to compute the relative camera positions. They utilize a vertical vanishing-point and an infinity homography. A common world coordinate system is defined to eliminate the overlapping constrain.

Our work is in this vein of fixed cameras which are calibrated with information commonly available in urban scenarios. Our calibration scenarios consist of networks of fixed monocular cameras which are traversed by an auxiliary LIDAR or color-depth (RGB-D) mobile camera (see Fig. 1). The auxiliary camera provides 3D data of the scenario which is sufficient for the calibration and thus overcomes the need of additional calibration artifacts. In addition, the mobile camera is assumed to have an estimate of its own localization, and therefore provide a global (single) coordinate system to the fixed cameras. As there are many well known methodologies for self localizing the mobile camera, as e.g. *Simultaneous Localization and Map Building (SLAM)*, we focus our work on the aspect of calibrating the fixed camera with respect to the mobile camera. Our recent work on camera calibration based in point clouds [?], is extended here with a more comprehensive Gauss-Newton formulation. This formulation allows the easy introduction of

additional constraints and algorithm analysis.

An analysis of camera calibration uncertainty is needed to evaluate the quality of metric reconstructions that can be inferred from the images in the camera network [?]. In a wide variety of cases, uncertainty propagation can be analyzed using first order techniques [?]. In [?] for instance, a first order error analysis is used to propagate the calibration uncertainty to the motion parameters. Ansar and Daniilidis [?] study the sensitivity of the camera pose estimation while using points or lines with corresponding 3D coordinates. In Daniilidis and Ernst [?] uncertainty propagation techniques are used to develop an unbiased estimator. Salvi et. al. [?] tested the accuracy of several classical calibrations methodologies. Sun and Cooperstock [?] show empirically the effects in camera camera calibration, of data size and input noise. In [?], the analysis is made via parameter correlation, and in [?] we introduce a calibration uncertainty analysis for cameras without radial distortion. In this paper we propose a first order estimation analysis of the full extrinsic calibration of cameras that have radial distortion.

3. Camera calibration

The pin-hole camera model maps the 3D projective space to the 2D projective plane. Using homogeneous coordinates, a scene point, $M = [X \ Y \ Z \ 1]^T$ is imaged as a point $m = [u \ v \ 1]^T$,

$$m \doteq P M = K [R \ t] M, \quad (1)$$

where \doteq denotes equal up to a scale factor and P is a 3×4 projection matrix. The 3×3 upper triangular matrix K contains the intrinsic parameters. The rotation matrix R and translation vector t form a rigid transformation from world to camera coordinates. In other words, t denotes the location of the origin of the world frame in camera coordinates and R describes the world frame also in camera coordinates.

To model radial distortion, we use the *division model*, proposed by Fitzgibbon [?], where an undistorted image point, $\hat{m}_u = [u_u \ v_u]^T$ is computed from a radially distorted image point $\hat{m}_d = [u_d \ v_d]^T$ by $\hat{m}_u = \hat{m}_d / (1 + \lambda \|\hat{m}_d\|^2)$, and λ represents the radial distortion parameter. The coordinates of \hat{m}_u and \hat{m}_d are expressed in a 2D coordinate system having the origin coincident with the image principal point $\hat{c}_o = [c_u \ c_v]^T$. The *division model* can also be conveniently written in homogeneous coordinates with

$$\begin{bmatrix} u_u \\ v_u \\ 1 \end{bmatrix} \doteq \begin{bmatrix} u_d \\ v_d \\ 1 + \lambda \|\hat{m}_d\|^2 \end{bmatrix} \Leftrightarrow m_u \doteq m_d + \lambda e_d. \quad (2)$$

3.1. DLT-Points

The Direct Linear Transformation (DLT), developed by Aziz and Karara [?], allows estimating the camera projection matrix in (1), P , by solving a linear system on the matrix entries using a set of 3D points, $\{M_i : M_i = [X_i \ Y_i \ Z_i \ 1]^T\}$ and their corresponding 2D image points $\{m_i : m_i = [u_i \ v_i \ 1]^T\}$. Applying a cross product by m_i to both sides of the equation, $m_i \times m_i = m_i \times (P M_i)$, becomes zero in the left hand side and thus $[m_i]_{\times} P M_i = 0$, where $[m_i]_{\times}$ represents the linear cross product operation as a skew-symmetric matrix of m_i . The properties of the Kronecker product [?], \otimes , allow one to obtain an equation factorizing the data and variables to estimate

$$(M_i^T \otimes [m_i]_{\times}) \text{vec}(P) = 0, \quad (3)$$

where $\text{vec}(P)$ denotes the vectorization of the matrix P , formed by stacking the columns into a single column vector.

Each pair (M_i, m_i) provides a set of three equations in the entries of $\text{vec}(P)$ but only two of them are linearly independent. Therefore, one needs at least six pairs of 3D-to-2D corresponding points to estimate the projection matrix. Having $N \geq 6$ pairs of 3D-to-2D correspondences, in a nondegenerate configuration, allows forming a matrix A , $3N \times 12$, by stacking N matrices $M_i^T \otimes [m_i]_{\times}$. The singular vector corresponding to the smallest singular value of A is an estimate of projection matrix (vectorized), minimizing the error $\|A \text{vec}(P)\|^2$ s.t. $\|\text{vec}(P)\| = 1$ [?].

Note that pre-normalization of the input data is crucial on implementing this algorithm [?]. Hartley suggested that the appropriate pre-normalization method consists in translating all data points (3D and 2D points) so that their centroids are at the origin. Further the data should be scaled so that the average distance, of data points to the origin, is equal to $\sqrt{2}$ for image points and $\sqrt{3}$ for 3D points.

Fitzgibbon's division model provides an extension of the *DLT-Points* calibration methodology to deal with the estimation of the camera projection matrix, P , directly from radially distorted image data. Substituting the right hand side of (2), in the *DLT-Points* factorized equation (3) results in

$$(M_i^T \otimes [m_{id} + \lambda e_{id}]_{\times}) \text{vec}(P) = 0 \quad (4)$$

which can be rewritten as $(A_{i1} + \lambda A_{i2}) \text{vec}(P) = 0$, where $A_{i1} = M_i^T \otimes [m_{id}]_{\times}$ and $A_{i2} = M_i^T \otimes [e_{id}]_{\times}$. Considering N pairs (M_i, m_i) one forms two $3N \times 12$ matrices, A_1 and A_2 , by stacking matrices A_{i1} and A_{i2} . As suggested by Fitzgibbon [?], left-multiplying the stacked matrices by A_1^T results in a polynomial eigenvalue

problem $(A_1^T A_1 + \lambda A_1^T A_2) \text{vec}(P) = 0^3$. Its solution gives simultaneously the projection matrix, $\text{vec}(P)$ and the radial distortion parameter, λ . Noting that the distortion model involves representing points around the principal point, which we assume to be approximately equal to the image center [?], the estimated projection matrix is finally obtained with $P' = T^{-1}P$, where T is a 3×3 matrix defining the translation of the image coordinate reference to the principal point. Having estimated P' , one has an estimate of the principal point, which can be used to iterate the calibration procedure and therefore improve the approximation.

However, in some scenarios it is impossible to calibrate a camera using *DLT-Points*. An example is shown in Fig. 4 where we do not have direct 2D-3D correspondences between Fig. 4 (b) and (c). In such scenarios one can use *DLT-Lines*, as will be discussed in the next section.

3.2. DLT-Lines

The use of lines for camera calibration, as opposed to isolated image points, allows for fine tuning the location of the lines in the image through simple line fitting algorithms.

Given a 3D line L_i , its projection on the camera image plane, l_i can be represented by an implicit equation, $f(m) = am_x + bm_y + c = 0$, where m is a point on the line. The parameters of this equation, $l = [a \ b \ c]^T$, can be defined by the cross product of two normalized points, m_1 and m_2 ,

$$l = m_1 \times m_2 = \begin{bmatrix} m_{1y} - m_{2y} \\ m_{2x} - m_{1x} \\ m_{1x}m_{2y} - m_{1y}m_{2x} \end{bmatrix}. \quad (5)$$

Note that a and b form a sub-vector that is perpendicular to the line, $n_l = [a \ b]^T$. The distance from point to line is $d = l^T m / \|n_l\|$, and one can normalize the line distance with $\|n_l\| = 1$.

Camera calibration based in *DLT-Lines* involves matching image lines with 3D lines or, more precisely, 3D points. As in *DLT-Points* two cases are considered, namely (i) no radial distortion and (ii) significant radial distortion. In the case where the radial distortion is considered, it is modeled using Fitzgibbon's division model. The case of radial distortion is detailed in the next section. First we detail the simpler case of no radial distortion.

³which can be solved in Matlab using the `polyeig` function

Any point m_{ki} lying on a line l_i satisfies $l_i^T m_{ki} = 0$. Multiplying by l_i^T on both sides of (1), i.e., $l_i^T m_{ki} = l_i^T P M_{ki}$, leads to:

$$l_i^T P M_{ki} = 0 \quad (6)$$

where M_{ki} is a 3D point in projective coordinates lying in L_i the 3D line imaged as l_i . As in the case of *DLT-Points*, using the Kronecker product one obtains a form where the elements of the projection matrix are factorized:

$$(M_{ki}^T \otimes l_i^T) \text{vec}(P) = 0. \quad (7)$$

Each pair of 3D point and its corresponding image line, (M_{ki}, l_i) , provides a single linear constraint in the entries of $\text{vec}(P)$. In order to estimate P one needs at least 12 pairs (M_{ki}, l_i) ⁴. Considering $N \geq 12$ pairs (M_{ki}, l_i) , one forms a matrix B , $N \times 12$, by stacking the N matrices $M_{ki}^T \otimes l_i^T$.

The least squares solution

$$p^* = \arg_p \min \|Bp\|^2 \quad \text{s.t.} \quad \|p\| = 1, \quad (8)$$

where $p = \text{vec}(P)$, is the right singular vector corresponding to the least singular value of B .

From (7) one has that in a noiseless case $Bp = 0$ with $\|p\| \neq 0$ and therefore $\max(\text{rank}(B)) = 11$. As already discussed, to obtain the maximal rank in B requires at least twelve pairs (M_{ki}, l_i) . However, if one has some calibration parameters known a priori then the rank of B may be lesser than the maximum and the amount of required 3D information diminishes.

Some more properties of interest can be stated for proposed calibration methodologies. From (7) and (3) it is possible to conclude that *DLT-Points* can be incorporated on *DLT-Lines*, by concatenation of matrices A and B , respectively. Both matrices represent equations on entries of p , allowing two paired (M_i, m_i) points, be combined with (M_{ki}, l_i) to estimate the projection matrix P .

Comparing both DLT methods, it is important to note that while in *DLT-Points* one has to provide one 3D-point to one 2D-point correspondences, in *DLT-Lines* one 2D-line, l_i is an image of a 3D-line, L_i and thus associates many-3D-points to one-2D-line. Any point $M_{ki} \in L_i$ forms a linear constraint with l_i (7). This property of *DLT-Lines* allows to apply additional image processing tools that add

⁴Alternatively, one can state that nondegenerate six 3D lines configuration and their corresponding six image lines, i.e. six 4-tuples $(M_{i1}, M_{i2}, m_{i1}, m_{i2})$, are enough to estimate P .

robustness to the extraction of calibration data. In particular, *DLT-Lines* can be used for fine tuning finding lines both in the RGB and RGBD images.

In addition, any line defined in the RGBD image indicates 3D points (from the depth data) that are expected to form a line in 3D. The points forming the 3D line have noise, e.g. due to the finite depth resolution, which is important to filter using a RANSAC procedure [?].

3.3. DLT-Lines with square pixels

One specific case of interest consists in knowing the pixels aspect ratio. Assuming the case of square pixels, common in modern cameras, allows decreasing the required completeness of data forming B , accepting for example $\text{rank}(B) = 10$. Using SVD factorization, let $B = U\Sigma V^T$ and $V = [v_1 \dots v_{11} v_{12}]$ be the singular vectors of B . In the case of $\text{rank}(B) = 10$ the calibration solution is a linear combination of the last two singular vectors, v_{11} and v_{12} . The square pixels constraint allows then writing the calibration problem as a 1D non-linear optimization problem:

$$\begin{aligned} p^* &= w^* v_{11} + \sqrt{1 - (w^*)^2} v_{12} \\ w^* &= \arg_w \min \|K(1, 1) - K(2, 2)\| \end{aligned} \quad (9)$$

where $w^* \in [0, 1]$ and the intrinsic parameters matrix, K , is computed through QR decomposition of the vector p^* reshaped to a 3×4 matrix. Note that by construction $\|p^*\| = 1$, since v_{11} and v_{12} are orthogonal and have unit norm. The interval $w^* \in [-1, 0]$ is not considered since $-p^*$ and p^* lead to the same solution.

3.4. DLT-Lines with radial distortion

Using (2), which describes the relationship between distorted and undistorted image points, a line l_{12} can be defined as the cross product of two undistorted points:

$$l_{12} = \begin{bmatrix} u_{1d} \\ v_{1d} \\ 1 + \lambda \|\hat{m}_{1d}\|^2 \end{bmatrix} \times \begin{bmatrix} u_{2d} \\ v_{2d} \\ 1 + \lambda \|\hat{m}_{2d}\|^2 \end{bmatrix} = \hat{l}_{12} + \lambda e_{12} \quad (10)$$

where the line can be divided in two terms, one without the radial distortion parameter, and a distortion correction term $e_{12} = [v_{1d} \|\hat{m}_{2d}\|^2 - v_{2d} \|\hat{m}_{1d}\|^2, u_{2d} \|\hat{m}_{1d}\|^2 - u_{1d} \|\hat{m}_{2d}\|^2, 0]^T$. Applying (10) into the point-to-line constraint (7), one has:

$$(M_k^T \otimes (\hat{l}_{12} + \lambda e_{12})^T) \text{vec}(P) = 0 \quad (11)$$

where M_k denotes the k^{th} 3D point projecting to the distorted line l_{12} . The equation can be rewritten as:

$$(B_{ki1} + \lambda B_{ki2}) \text{vec}(P) = 0 \quad (12)$$

where $B_{ki1} = M_k^T \otimes \hat{l}_{12}^T$, $B_{ki2} = M_k^T \otimes e_{12}^T$.

Considering $N \geq 12$ pairs (M_{ki}, \hat{l}_i) , where $N = k_{max} i_{max}$, one forms two $N \times 12$ matrices, B_1 and B_2 , by stacking matrices B_{ki1} and B_{ki2} .

Let us define the cost function:

$$f(m, M) = \|(B_1 + \lambda B_2)p\|^2 \quad (13)$$

The minimization problem to find $p = \text{vec}(P)$ can be written as:

$$\begin{aligned} (p^*, \lambda^*) &= \arg_{p, \lambda} \min f(m, M; p, \lambda) \\ \text{s.t. } p^T p - 1 &= 0. \end{aligned} \quad (14)$$

Applying a change of variable $\lambda p = q$, one obtains the Lagrangian:

$$\begin{aligned} \mathcal{L} &= p^T B_1^T B_1 p + p^T (B_1^T B_2 + B_2^T B_1) q + q^T B_2^T B_2 q + \\ &+ \sigma (p^T p - 1) + v^T (q - \lambda p) \end{aligned} \quad (15)$$

where σ, v are KKT multipliers, therefore yielding the KKT conditions of the problem which can be written as the system of equations:

$$\begin{aligned} G(m, M, p; v, \sigma) &= 0_{38 \times 1} \\ &\Updownarrow \\ \left\{ \begin{array}{l} 2B_1^T B_1 p + (B_1^T B_2 + B_2^T B_1) q + 2\sigma p - \lambda v = 0_{12 \times 1} \\ (B_1^T B_2 + B_2^T B_1) p + 2B_2^T B_2 q + v = 0_{12 \times 1} \\ q - \lambda p = 0_{12 \times 1} \\ -v^T p = 0_{1 \times 1} \\ p^T p - 1 = 0_{1 \times 1} \end{array} \right. \end{aligned} \quad (16)$$

Given the KKT conditions we can use Gauss-Newton to estimate p . More precisely we solve the system by iterating the solution through the linear system:

$$J_G \theta = -G \quad (17)$$

where θ denotes the Newton step which needs to be solved for, and will be used to update p, v and σ . The Jacobian J_G assumes the form:

$$J_G = \begin{bmatrix} 2B_1^T B_1 + 2I\sigma & B_1^T B_2 + B_2^T B_1 & -v & 2p & -I\lambda \\ B_1^T B_2 + B_2^T B_1 & 2B_2^T B_2 & 0 & 0 & I \\ -v^T & 0^T & 0 & 0 & -p^T \\ 2p^T & 0^T & 0 & 0 & 0^T \\ -I\lambda & I & -p & 0 & \mathbf{0} \end{bmatrix}. \quad (18)$$

To initialize the iterative method, we use Fitzgibbon's suggestion [?]: left-multiplying the stacked matrices by B_1^T results, once more, in a polynomial eigenvalue problem $(B_1^T B_1 + \lambda B_1^T B_2) \text{vec}(P) = 0$, which was solved in Matlab using the `polyeig` function. Its solution gives simultaneously the projection matrix, $\text{vec}(P)$, the radial distortion parameter λ , and $P' = T^{-1}P$, where T is defined in sec 3.1. In a similar way as explained before, both DLT methods applied to the radial distorted camera, can be combined to estimate P and λ . Note that the solution will only be an approximation since the minimization functions are not the same.

Having estimated the projection matrix, P , the camera intrinsic and extrinsic parameters can be obtained using QR-decomposition [?].

3.5. Summary and back-projection application

In the previous sections we described *DLT-Points* and *DLT-Lines*, which involved solving for the intrinsic and extrinsic parameters as least squares problems. In the more general case, in which we estimate also the radial distortion, we proposed a Gauss-Newton iterative solution. Many other interesting and relevant cases can be found within these frameworks. For instance, when K is known, its inverse can be applied to the 2D data and, in the case of *DLT-Points*, the calibration problem is converted to solving the well known *Perspective-n-Point* (PnP) problem [?]. Another simple case is the one where t is given and therefore *DLT-Points* and *DLT-Lines* are reduced to the estimation of an homography, KR , by subtracting the camera center to all the 3D data. Other interesting cases arise by specifying constraints in R or t , or in K , such as square pixels and principal point coincident with the image center. These cases can be dealt with Gauss-Newton solutions as proposed for *DLT-Lines* with radial distortion.

The simple data-based constraints, created by DLT like formulations, are useful in many applications. Here we start describing the application central to this paper, namely the analysis of uncertainty propagation in calibration. In particular

we detail the effect of calibration uncertainty into the reconstruction of ground points, i.e. points back-projected to the ground plane. First we introduce back-projection and in subsequent sections we study its uncertainty propagation.

Considering a point m observed in an image, we know it is related to a 3D point on the mobile camera coordinate system, ${}^M M$, through $m \doteq P {}^M M$. Using the transformation from the world coordinate frame to the mobile camera, ${}^M T_W$, we can rewrite it to consider points in world coordinates $m \doteq P {}^M T_W {}^W M$. Using the plane constraint of floor points ${}^W M_z = 0$ we can further write it as $m \doteq P {}^M T_W E^T E {}^W M$ where E is an elimination matrix removing the third row of ${}^W M$. Inverting the system we have the projection of the point on the floor in homogeneous coordinates

$$E {}^W M \doteq (P {}^M T_W E^T)^{-1} m. \quad (19)$$

4. Error propagation

In this section we derive first order error propagation formulas, from 2D and 3D data to the calibration parameters, while using the *DLT-Lines* calibration process. Section 4.1 shows that line fitting helps reducing 2D noise when lines are defined by their end points. However, despite of the fitting, residual 2D noise remains present due to the discrete nature of pixels, the discrete nature of pixel values and noise in the pixel values.

Uncertainty in the 3D data (points) is due to a wider variety of reasons. Assuming that 3D data is acquired by a sensor such as a LIDAR or an RGBD camera, then error in the estimated pose of the sensor or error in the intrinsic parameters of the sensor or the discretization of the sensor inputs, are all sources of uncertainty. For example, an offset in the real location of a sensor implies also an offset in the position of the calibrated camera. Error in the intrinsic parameters of the 3D sensor may induce artificial zooming in the calibrated camera.

Having indicated a number of common sources of uncertainty, we detail now the uncertainty propagation analysis through the *DLT-Lines* calibration methodology. In a first step we derive the expression propagating error in the calibration data to error in the estimated lines parameters ①. In a second step, ②, we derive error propagation between lines parameters and the 3D data to the camera projection matrix. In a third step, ③, we derive error propagation from the matrix entries to the camera projection center, orientation and intrinsic parameters. The whole process can be summarized as shown in the next diagram:

$$\begin{array}{ccccccc}
m_i, M_i & \xrightarrow{\textcircled{1}} & l_i, M_i & \xrightarrow{\textcircled{2}} & P & \xrightarrow{\textcircled{3}} & K, R, t \quad M_W \\
\Sigma_{m_i}, \Sigma_{M_i} & \rightarrow & \Sigma_{l_i}, \Sigma_{M_i} & \rightarrow & \Sigma_P & \rightarrow & \Sigma_K, \Sigma_R, \Sigma_t \quad \Sigma_{M_W}
\end{array}$$

While the last section discussed how to compute the solution to ①, ② and ③, this section will propagate the covariance information through these same transformations. The general rule used is that given a function $y = F(x)$ and assuming F is differentiable near a particular x , a first order covariance propagation Σ_y is obtained from the input noise covariance as $\Sigma_y = J_F \Sigma_x J_F^T$.

Transformation ① is explicitly available so the Jacobian is readily computable. Since the covariance of the line measurements is obtained from two image points, it can be computed as:

$$\Sigma_{l_i} = J_{l_i} \Sigma_{m_{1i}, m_{2i}} J_{l_i}^T \quad (20)$$

where J_{l_i} is the Jacobian of either (5) or (10). The covariance matrix of the imaged line-segment end-points, $\Sigma_{m_{1i}, m_{2i}}$, is usually set a-priori according to typical user-clicking errors. Alternatively, in case one uses a line fitting process (sec. 4.1), the edge points near the fitted line inform about the fitting error, Σ_{l_i} , which is transformed to error at the line-segment end-points.

The same approach cannot be used for ② and ③ where no closed form expression is known, as they are computed using numeric algorithms. Nonetheless, since an implicit characterization of these transformations exists, the Jacobian is computable using the implicit function theorem [?]. Given a system of equations $G(x, y) = 0$ defining implicitly an unknown function $y = F(x)$ satisfying the theorem conditions, the Jacobian of F can be computed from the derivatives of G as $J_F = -[D_y G]^{-1} D_x G$ where $D_x G$ and $D_y G$ are the partial derivatives of G with respect to x and y respectively. The implicit function theorem requires the matrix $D_y G$ to be invertible, a condition we assume to be true in this paper, as otherwise the input data consists of a singular case where a calibration solution is not possible. The rest of this section will make this propagation explicit for the two cases (with and without radial distortion).

4.1. Image lines uncertainty

DLT-Lines allows filtering the input data by using line fitting methodologies. This is beneficial as the uncertainty before ① can be attenuated and, consequently, obtaining more precise and accurate calibrations. In this section we detail the image line fitting methodology we use and study its effect on noisy data.

Each extracted 2D-line l_i can be the result of fine tuning a line \hat{l} to better match edge points in an image. The user defines \hat{l} by clicking the two ends of the line.

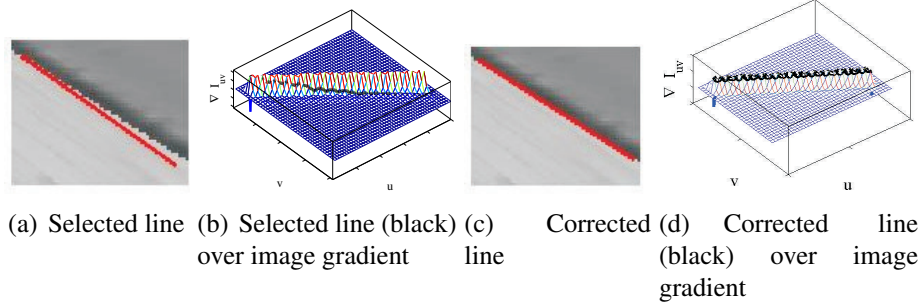


Figure 2: Fine tuning the line location. (a) Input data, (b) input data over image gradient, (c) and (d) result after line optimization.

Line \hat{l} allows defining gradient values, ∇I in a direction perpendicular to \hat{l} . Line l_i is therefore refined to better match the image appearance by an optimization process, that maximizes the sum of image gradients and passes through a constant texture:

$$l_i^* = \arg_{l_i} \max \sum_k \|\nabla I(m_k)\| - \beta \sigma_{I(m_k)}, \quad m_k \in (l_i \cap \mathcal{R}) \quad (21)$$

where I is the RGB image converted to gray scale, ∇ denotes the image gradient in the direction orthogonal to \hat{l} , $\sigma_{I(m_k)}$ is the local image variance, β is a constant gain, and \mathcal{R} is a region of interest containing the straight-line segment plus some tolerance (e.g. ± 10 pixel) around the segment extremes. This refinement procedure is illustrated in Fig. 2. Frame (a) shows the initialization of the algorithm. In this case, the algorithm was initialized with a significant error (see red line). After the optimization procedure one can see that the line was correctly estimated, frame (c). Frames (b) and (d) show that image gradient values at line points (black line) are maximized by the procedure.

After the optimization procedure, one obtains line end-points, m_1 and m_2 , whose error is propagated to the line as $\hat{l} = l + \varepsilon$, where ε is the estimation error. The error ε has only significant impact on the line parameters if m_1 and m_2 are close to each other. If the points are far away, e.g. $\|m_{1x} - m_{2x}\| \gg \|\varepsilon_x\|$, the pixel error can be almost ignored, $\hat{l} = l / \|\hat{n}_l\| + \varepsilon / \|\hat{n}_l\|$, since $\|\hat{n}_l\| \gg \|\varepsilon\| \Rightarrow \varepsilon / \|\hat{n}_l\| \approx 0$.

In order to study the estimation error of the line parameters we run a Monte Carlo test with a simple line fitting algorithm. The test consists in finding a line in a image, given two initial end points. Figure 3(a-right) shows the typical image used. The test encompasses several different lines, i.e. with different angles, sizes, and sub-pixel translations. Figure 3(a-left) shows a subset of the line orientation

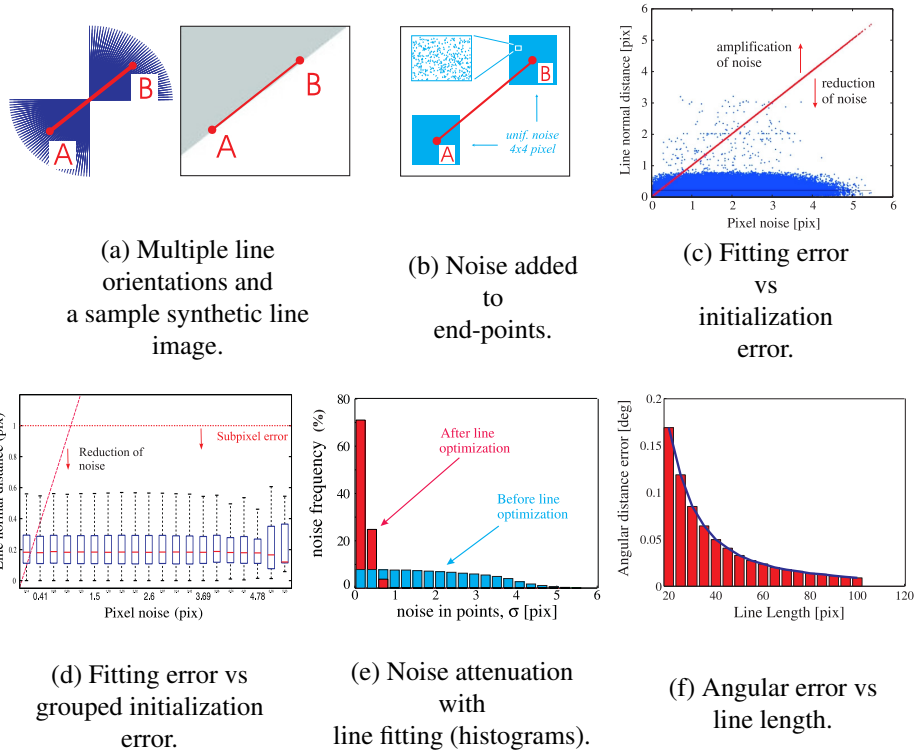


Figure 3: Line fitting Monte Carlo uncertainty analysis. (a) Range of line directions in the test set used for the Monte Carlo analysis. Sample input image built to represent one ground truth line (AB in red) from the test set. (b) Histogram of the normal output error, when a uniform noise is used as input. (c) Plot of the line fitting error according to initialization error. (d) Histogram of the line fitting error according to initialization error. (e) Example of the noise used in the initialization process, the noise was a uniform distribution. (f) Evolution of the angular error, $\arccos \hat{n}_l^T n_l$, estimation when is used different line lengths.

and sizes used. In the Monte Carlo simulation we initialized every line fitting with random uniform noise, between -4 and 4 pixels in each of the u and v directions (see Fig. 3(b)).

The blue dots in Fig. 3(c) show initialization and fitting errors. The initialization error (horizontal axis) is the distance from the initialization line end-points to the ground truth line, which has an approximately uniform distribution between 0 and $4\sqrt{2}$ pixel, caused by the noise shown in Fig. 3(b). The fitting error (vertical axis) is the distance from the estimated line end-points to the ground truth line and has a less than 1 pixel mean error. The red line has unitary slope and thus indicates noise reduction or amplification for blue points under or over the red line. One observes that in general fitting reduces noise except when the initialization noise is already subpixel. Figure 3(d) is a box plot indicating median, 25th and 75th percentiles, which is obtained from (c) by splitting the horizontal axis in a number of regions. This plot shows that fitting error is approximately constant in the 75th percentiles of subpixel range, and therefore indicates an algorithm that upon convergence is independent on the initialization error.

Figure 3(e) shows initialization and fitting errors as histograms. The initialization error has an approximately uniform distribution, while the fitting error has an histogram shifted to subpixel error as desired. Figure 3 (f) shows, as expected, that the angular distance ($\arccos \hat{n}_l^T n_l$) between the estimated line and the real line improves with the line width as shown in [? ?].

4.2. DLT-Lines *error propagation without radial distortion*

This section explores the sensitivity of the process ②, namely how error propagates from $\Sigma_{l_i}, \Sigma_{M_i}$ to Σ_P when it is solved with a least squares (7).

The optimization problem in (8) can be seen as function that accepts lines (l_i) and points (M_i) and outputs the least squares estimate of the projection matrix P :

$$P = f(l_1, \dots, l_N, M_1, \dots, M_N). \quad (22)$$

To compute the covariance propagation we need the Jacobian of f , i.e. J_f . Although f does not have an explicit solution (it is the result of an optimization problem), an implicit system can be written from the KKT conditions of the constrained least squares problem in (8).

$$G(l, M; p, \gamma) = \begin{cases} 2B_{(l,M)}p + 2\gamma p & = 0, \\ p^T p - 1 & = 0. \end{cases} \quad (23)$$

The implicit function theorem can then be used to provide

$$J_f = \left[- (D_{\{p,\gamma\}} G)^{-1} D_{\{l_i, M_i\}} G \right]_{12 \times 6N}$$

where the brackets denote taking only the first 12 rows of the result since the last row is the row equivalent to the Lagrange multiplier γ .

4.3. Error propagation in the presence of radial distortion

Suppose we have used sets \hat{m} and \hat{M} to estimate p using the Gauss-Newton algorithm. Although $p = f(l, e, M)$ does not have an explicit formulation again, we can use the implicit function theorem to compute the Jacobian of the method:

$$J_f = \left[- (D_{\{p,\sigma,v\}} G)^{-1} D_{\{l,e,M\}} G \right]_{12 \times 6N},$$

where G is the KKT system in (16), and whose derivatives are given in (18). Uncertainty can thus be propagated as

$$\Sigma_p = J_f \Sigma_{l,e,M} J_f^T \quad (24)$$

4.4. Error propagation of P towards camera pose

In [?] it is shown how the projection matrix can be decomposed by the internal and external calibration parameters. Here we propagate the uncertainty from Σ_P to $\Sigma_{K,R,t}$, ③.

We start by looking at (1), noticing that we can divide P in two blocks, $P_{3 \times 3} = KR$ and $p_4 = Kt$. Once again an explicit solution does not exist, however the separation of P into two blocks defines an implicit system if we take into account the upper triangular structure of K and that R is a rotation matrix:

$$G(P, K, R, t) = \begin{cases} P_{3 \times 3} = KR \\ p_4 = Kt \\ R^T R = I \\ K_{21} = 0 \\ K_{31} = 0 \\ K_{32} = 0, \end{cases}$$

The derivatives of this implicit system are:

$$J_G = \begin{bmatrix} -I & 0 & I \otimes K & R^T \otimes I & 0 \\ 0 & -I & 0 & t^T \otimes I & K \\ 0 & 0 & (I_9 + C_{3 \times 3})(I \otimes R^T) & 0 & 0 \\ 0 & 0 & \mathbf{1}_2^T & 0 & 0 \\ 0 & 0 & \mathbf{1}_3^T & 0 & 0 \\ 0 & 0 & \mathbf{1}_6^T & 0 & 0 \end{bmatrix}, \quad (25)$$

where $C_{3 \times 3}$ is a 9×9 circulant matrix (i.e., it's the matrix that makes $\text{vec}(R^T) = C_{3 \times 3} \text{vec}(R)$ true for any 3×3 matrix R), and $\mathbf{1}_i^T$ is a vector 9×1 of zeros with 1 in position i . Note that the constraints $R^T R = I$ are redundant (symmetric), hence we manually remove the rows of J_G corresponding to the redundant equations. Once again we can use the implicit function theorem to find $J_f = -[D_{K,R,T}G]^{-1}D_P G$.

The estimation of Σ_{KRT} can be obtained by:

$$\Sigma_{K,R,T} = J_F \Sigma_P J_F^T. \quad (26)$$

These equations are valid for both calibration methodologies, with and without radial distortion.

4.5. Error propagation of P towards reconstructed floor points

Propagating uncertainty from a camera point m to the world floor takes into account three sources of uncertainty: the camera point m itself, the camera position represented by the projection matrix P and the mobile camera position represented by ${}^M T_W$. (19) provides an explicit equation but due to the inverse function involved, its derivatives are not trivial to compute. We do so in two steps, first interpreting (19) as an equality and then applying the normalization. The derivative of the equality with respect to m , P , and ${}^M T_W$ is

$$D = [Q \quad -m^T Q^T E \quad {}^M T_W^T \otimes Q \quad -m^T Q E \otimes Q P]$$

where, $Q = (P {}^M T_W E^T)^{-1}$. After applying the projective normalization, the complete Jacobian of (19) is

$$J = \begin{bmatrix} 1/z & 0 & -x/z^2 \\ 0 & 1/z & -y/z^2 \end{bmatrix} D$$

where $[x \ y \ z]^T$ is the projective result of (19). Again, the error propagation equation is

$$\Sigma_{x,y} = J \Sigma_{m,P,{}^M T_W} J^T. \quad (27)$$

5. Experimental results

In order to validate the proposed methods and uncertainty analysis we conduct some experiments in a virtual indoor environment, for which one has available precise and accurate ground truth. We show a simple real degenerate case which can be solved using *DLT-Lines*, with a single constrain. In addition, we apply the proposed uncertainty analysis to a real setup based on an indoor scene encompassing a wall and a couple of doors, which has been reconstructed in 3D using color-depth (RGB-D) data, thus providing directly the required 3D information for the *DLT-Lines* calibration methodology.

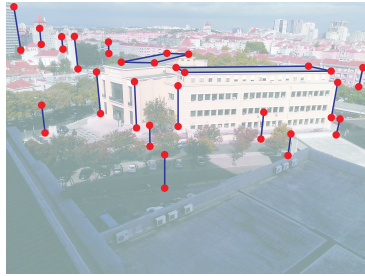
5.1. *DLT-Lines calibration experiment*

In this first experiment we illustrate the *DLT-Lines* calibration methodology when available data does not fully constrain the solution (see sec. 3.3 (9)).

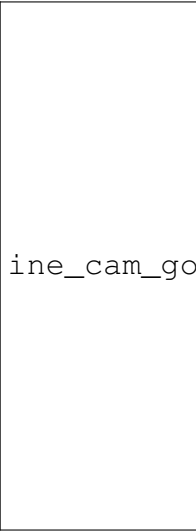
The camera to calibrate is a cellphone camera using a single image of the streets and buildings from a window in the 5th floor of a tower (see Fig. 4(a)). The terrain is uneven and therefore the zero height was selected to be defined by the roof of the large building imaged in front of the camera. The 3D calibration information is only partially available as the 3D lines are pointed in a 2D aerial (orthophoto) view and therefore lack a depth coordinate, Z (see Fig. 4(b)). Vertical lines are defined by two points $Z = 0$ and $Z = 1$. Horizontal lines in the planar roofs of the building are also defined by choosing $Z = 0$. Other horizontal lines, not at the roof height, have an unknown Z .

Using only vertical and horizontal lines at the roof level, $Z = 0$, results in rank deficient problem, more precisely, $rank(B) = 10$ in (7). The solution based on the two singular vectors corresponding to the least singular values, $p = w_{11}v_{11} + w_{12}v_{12}$, has an ambiguity between the camera height and the vertical focal length. The null space is a set of camera configurations where the camera is fixed at X and Y coordinates while Z varies. As the Z gets higher the camera is rotated downwards and the vertical focal length is augmented so that the imaging does not change.

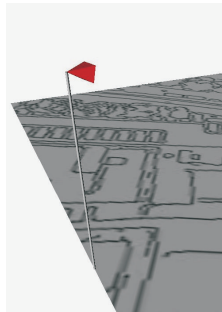
In order to constrain the solution, we use the square pixels constraint proposed in the form of a 1D optimization problem in (9). The solution obtained, displayed graphically in Fig. 4(c), allows retrieving the camera pose (localization and rotation) and its intrinsic parameters. The estimated localization of the camera matches well the information of the aerial view (see in Fig. 4(b) the blue cross on the facade of the tower).



(a) Image taken by the camera to calibrate and image lines used for the calibration.



(b) Aerial view (Google maps) and horizontal (blue) and vertical lines (red dots) used for calibration.



(c) Estimated pose of the camera shown by the pyramid. Estimated facade heights are used to refine the ground plane.

Figure 4: Calibration and reconstruction experiment. Intrinsic and extrinsic calibration of a camera based on imaged lines (a) and 3D data inferred from an aerial image (b). The reconstruction of facade heights complements information not available in the aerial view (c).

In addition, the known scale of the aerial (orthophoto) image allows converting the 3D calibration data and the projection matrix into meter units. Consequently, it is possible also to infer in meter units some distances in the scenario, such as heights of buildings. The estimated height of the building observed in the middle of the image is 11.0 meters which can be compared with the height read in blueprints, 10.4 meters, yielding an experimental error of just 6% justified by data noise.

5.2. Synthetic experiments

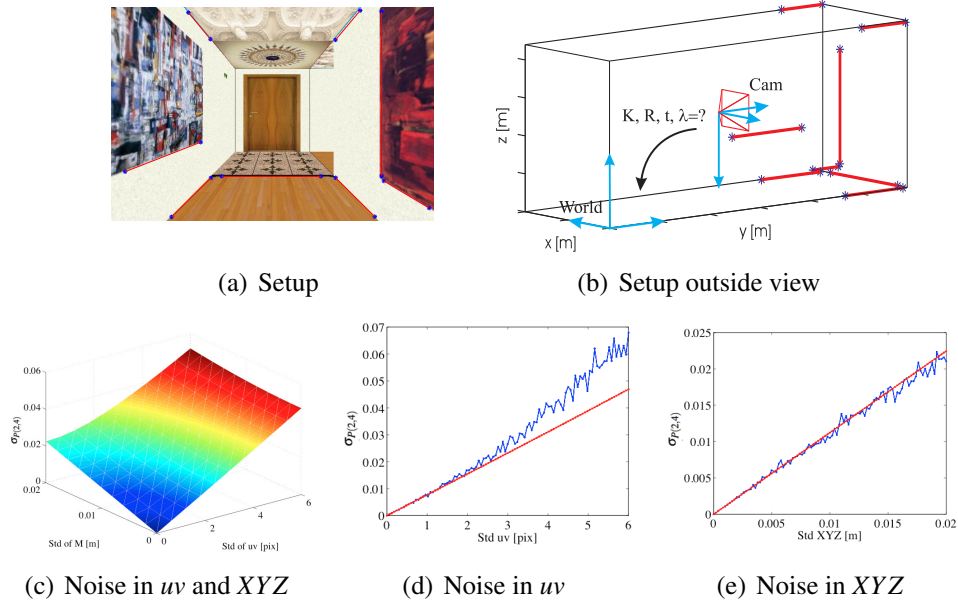


Figure 5: Single camera setup. (a) Ground truth 3D information is known for the image lines shown. (c) Standard deviation of the estimation error of p_{11} , entry (2,4) of P , given noise in uv and XYZ . (d) Standard deviation of calibration error propagated from noise in uv ranging from 0 to 6 pixel, red line; compared against Monte Carlo uncertainty analysis, blue line. (e) Standard deviation of calibration error propagated from noise in XYZ ranging 0 to 0.02 meters, red line; compared against Monte Carlo uncertainty, blue line.

⁴The decomposition of estimated projection matrix, detailed in [?], allows factorizing the intrinsic and extrinsic parameters as $P = K[R t]$, and therefore comparing them with the ground truth. The horizontal focal length relative error is defined as $K_{err} = (K(1,1) - K_e(1,1))/K(1,1)$, where K is the VRML camera true intrinsic parameters matrix, K_e is the estimated one and $K(3,3) = K_e(3,3) = 1$.

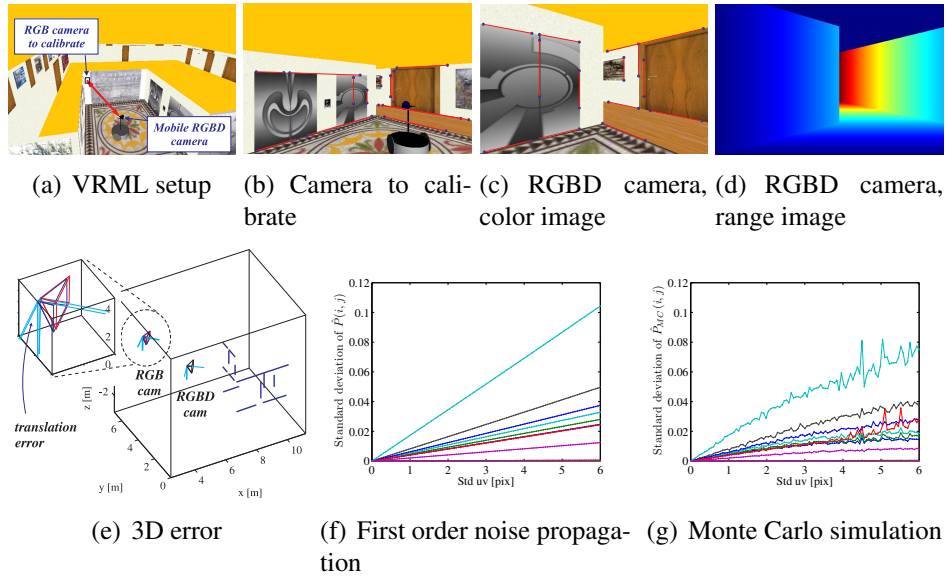


Figure 6: Analysis of calibration uncertainty in a two cameras setup. (a) VRML setup. (b) RGB image. (c) RGBD intensity image. (d) RGBD range image. Each line defined in the RGBD image corresponds to a line in the RGB image, and leads to a 3D line in the world/RGBD coordinate system. (e) 3D lines form the required input data for *DLT-Lines* calibration. (f) Relation between the error in the RGB image coordinates and the projection matrix parameters. (g) Monte Carlo simulations of the same relation between image error standard deviation and the standard deviation of the projection matrix elements.

In this section the variance of the entries of the projection matrix, Σ_P , predicted using the proposed uncertainty analysis is compared with a Monte Carlo based uncertainty analysis. The experiments are done using synthetic data (VRML) which allows us to define with high accuracy a camera location and a set of fixed image points and the corresponding set of 3D points. We consider various levels of white Gaussian noise in image points, 3D points, or both. Experiments are conducted in three different scenarios in order to show that the proposed uncertainty propagation methodology effectively takes into account the structure of the scenario.

The first setup is based on a single RGBD camera. The setup can be seen in Fig. 5(a), which corresponds to a typical 'L' shaped corridor, the overall setup can be seen in Fig. 5(b). Camera calibration ground truth is known and is used to assess the validity of the noise propagation estimation method and the accuracy of the calibration. Figure 5(c) shows the theoretical value for $\sigma_{P(2,4)}$ in the presence of noise simultaneously in both the image and the range values. The plot shows the correlated effects between the image and range noise values.

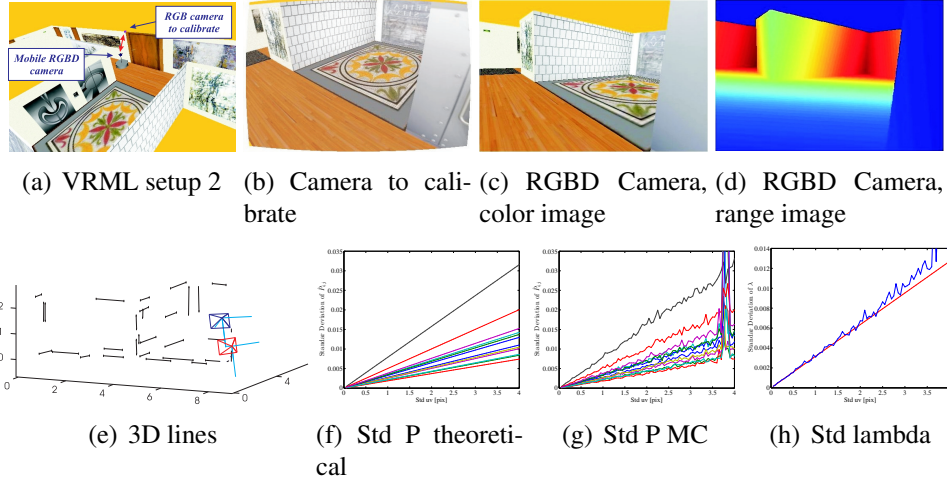


Figure 7: Calibration uncertainty analysis for a camera with radial distortion. (a) VRML setup 2. (b) RGB image, with radial distortion. (c) RGBD intensity image. (d) RGBD range image. Each line defined in the RGBD image corresponds to a line in the RGB image, and leads to a 3D line in the world/RGBD coordinate system. (e) 3D lines form the required input data for *DLT-Lines* calibration with radial distortion. (f) Relation between the error in the RGB image coordinates and the projection matrix parameters in presence of radial distortion. (g) Monte Carlo simulations of the same relation between image error standard deviation and the standard deviation of the projection matrix elements. (h) Monte Carlo simulations and theoretical relation between image error standard deviation and the standard deviation of the radial distortion parameter.

Monte Carlo simulations were also run for this setup. Plots (d) and (e) show both the analytic and estimated value of σ for $P(2,4)$ as a function of variations in image and depth noise.

Plot (d) shows once more that the first order approximation is only valid up to around $\sigma_m = 3$ pixels. The theoretical prediction is nevertheless accurate for lower levels of noise showing that the proposed uncertainty analysis takes correctly into account the scene structure (l_i, M_i) . Noise in range data exhibits less non-linear effects, as shown in plot (e).

The second synthetic setup is formed by two cameras, namely a mobile color-depth (RGBD) camera which collects 3D data and a fixed RGB camera (Fig 6(a)), in this setup the cameras have different optical centers and orientations. Fig. 6(b) shows a synthetic image acquired by the RGB camera, while Figs. 6(c) and 6(d) show the synthetic intensity and range images acquired by the RGBD camera.

In this setup, we analyze what happens when just the RGB image has noise.

In other words, the noise in 3D points is set null ($\sigma_M = 0$)⁵. Uncertainty analysis was done using both the proposed propagation methodology and Monte Carlo simulations. Monte Carlo was configured to do 300 runs for each level of noise. The standard deviation of the noise in the 2D points varies from 0 to 6 pixels ($\sigma_m = 0 : .06 : 6$ pixels). Having all the runs, the variance of every entry of P , i.e. $\Sigma_{P(ij)}$ for $i = 1..3$ and $j = 1..4$, has been estimated.

The linear propagation of the standard deviation of each of the entries of P , computed with the proposed methodology, is shown in Fig. 6(e). As expected some entries of P are more robust to noise than others. Figure 6(f) shows the Monte Carlo simulation results for each level of noise, again for all the entries of P . Plots (f) and (g) indicate that the analytical values obtained using the linear propagation analysis match those of Monte Carlo results for values of σ_m lower than approximately 3 pixels. Nonlinearities have more incidence for large image noise, making our first order approximation unreliable. Nonetheless, pixel value noises in ranges below 3 pixels are acceptable for most imaging sensors.

The final synthetic experiment was done with two cameras, as the previews one, a fixed RGB camera and a mobile color-depth (RGBD) camera which collects 3D data. See Fig 7(a). However in this case the RGB camera has radial distortion, Fig 7(b). Figure 7(c) and 7(d) show respectively the RGBD intensity and 3D range data. Figure 7(e) shows a set of 3D lines given to the calibration algorithm, in red is the RGBD camera position and orientation, while in blue is the camera with radial distortion to calibrate. The proposed uncertainty analysis was once again compared to Monte Carlo simulations. In this case the standard deviation of noise in 2D varies from 0 to 4 pixels ($\sigma_m = 0 : .05 : 4$ pixels). For each level of noise we did 400 runs. The theoretical prediction of Σ_P is plotted in Fig. 7(f), while Fig. 7(g) shows the Monte Carlo results. We can observe that the theoretical results are in accordance with the Monte Carlo simulations. Figure 7(h) shows the theoretical and the Monte Carlo Σ_λ . Once again both results are similar empirically confirming the correctness of our approach.

5.3. Error propagation real dataset

In this section we present the results of our algorithm in a real data set acquired in a typical indoor scene, with several doors and a corridor, see Fig. 8(b). This experiment was done with a ASUS X-Tion (RGBD camera) mounted in a mobile

⁵To improve readability, variance is written using upper case, Σ , and standard deviation is written using lower case, σ .

robot, Fig. 8(a), and Axis P1347 (typical RGB surveillance camera). The RGB image used for calibration is shown in Fig. 8(e), and has resolution of 1920 by 2560 pixels. Twenty lines were marked in the calibration process. This camera has significant radial distortion (see for example the first door in the left). Figure 8(c) shows the depth image of the RGBD sensor and Fig. 8(b) shows its intensity image, where the same set of twenty lines that can be seen in the RGB images are overlaid. The intensity and depth images have the same resolution, 480 by 640 pixels. 3751 3D points along the 20 lines were automatically sampled.

The calibration was done using *DLT-Lines* with radial distortion. Figure 8(f) shows the original image without radial distortion. Figure 8(d) shows the theoretical evolution of σ_t in the presence of white Gaussian noise in the RGB camera. The plot shows the prediction of σ_t for different levels of σ_m , namely from 0 to 3 pixels. Figure 8(g) shows the result of the calibration algorithm, namely the estimated pose and the position uncertainty of the RGB camera. The uncertainty, represented as an ellipsoid, was estimated assuming a white Gaussian noise with $\sigma_m = 1.5$ pixels in the RGB while the 3D noise was set to zero, $\sigma_M = 0$ meters. Figure 8(h) shows uncertainty at the floor plane, $Z = 0$ meters, for a number of image points given the estimated calibration uncertainty. As expected the uncertainty gets higher as the selected points get farther from the camera. This is mainly explained by foreshortening. Due to the perspective view of the camera, as we get closer to the vanishing line of the ground plane the 3D distance between each pixel approaches infinity. As a result the uncertainty tends towards infinity.

6. Conclusions

In this work we presented uncertainty analysis for two in-situ camera calibration methodologies, namely *DLT-Lines* and *DLT-Lines* with radial distortion. We provide analytic solutions to uncertainty propagation from errors on the input data to the estimated parameters and empirically verify the validity of the analysis with Monte Carlo simulations. From the uncertainty in the projection matrix we derive (i) the uncertainty in intrinsic and extrinsic parameters, including radial distortion, and (ii) the uncertainty of 3D points in the ground plane.

We tested our methodologies both in a virtual environment with ground truth and with real cameras. In particular we demonstrated uncertainty propagation to the camera location and to points in the ground plane.

In the future work we will focus in finding structures that allow lower variances in the camera parameters. We also will focus in finding points and lines that increase the accuracy of the calibration.

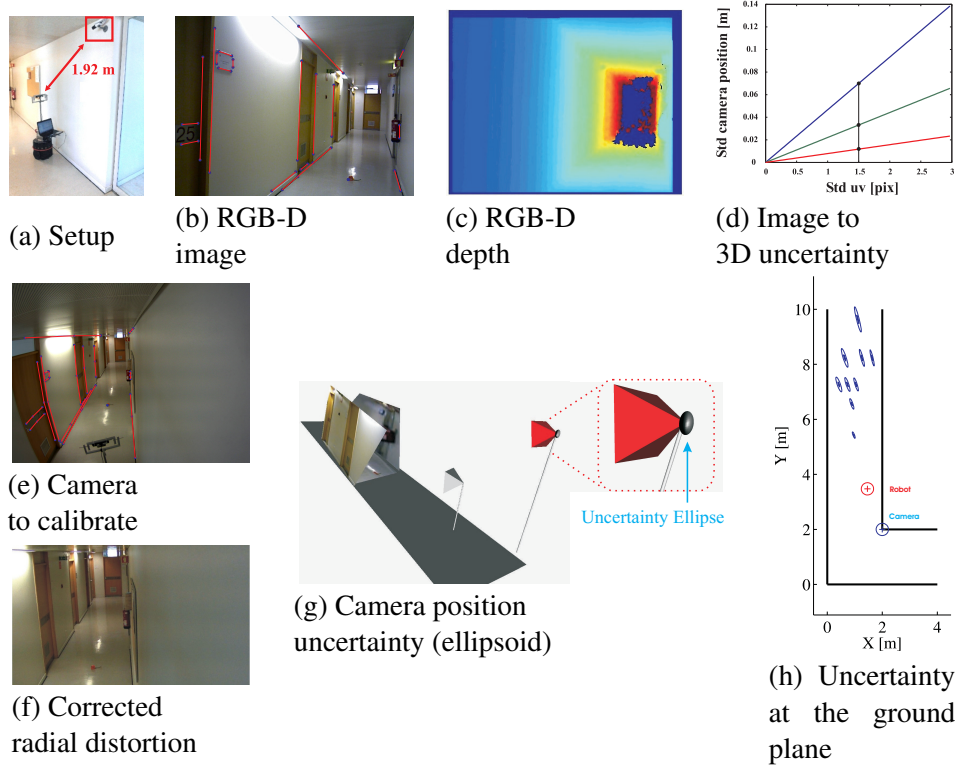


Figure 8: Analysis of camera calibration uncertainty. (a) Real indoor setup. (b) RGBD intensity image. (c) RGBD range image. Each line defined in the RGBD image corresponds to a line in the RGB image, and leads to a 3D line in the world/RGBD coordinate system. (d) Evolution of σ_r with different noise levels of σ_m . At black is marked $\sigma_m = 1.5$. Blue, green, red and are X, Y, Z coordinates respectively. (e) RGB image without radial distortion correction, in red lines used to calibrate the camera. (f) The RGB images with radial distortion correction. (g) 3D reconstruction of the scene, white the robot camera, blue the estimated camera pose and orientation. Ellipsoid representing the position uncertainty when in presence of $\sigma_m = 1.5$ can be seen in the zoomed part. (h) Blue print of the corridor, the red with a plus sign is the robot position and in blue is the camera position. The blue ellipse are the floor uncertainty given an image point. The inner ellipse represents 1σ , while the outer ellipse represents 3σ , meaning that with 99.7% the image point is projected at the given floor location.

Acknowledgments

This work has been partially supported by the FCT project PEst-OE / EEI / LA0009 / 2013, by the MINECO project DPI2011-27510, and by the EU Projects POET-ICON++ EU-FP7-ICT-288382 and Cargo-ANTs FP7-SST-605598. The authors thank Manuel Silva for the valuable initial datasets.