Camera Calibration using a Color-Depth Camera:
Points and Lines Based DLT including Radial Distortion

Manuel Silva, Ricardo Ferreira, José Gaspar

Institute for Systems and Robotics, Instituto Superior Técnico / UTL, Lisbon, Portugal

manuel.silva.thesis@gmail.com, ricardo@isr.utl.pt, jag@isr.utl.pt

Abstract— In this paper we approach the problem of in-situ camera calibration using an auxiliary mobile color-depth camera. The calibration is based on image lines and the Direct Linear Transformation (DLT). Calibration includes intrinsic and extrinsic parameters, and radial distortion. Traditionally, camera calibration methods based on DLT combine 2D and 3D points. Using 2D image lines allows adding simple image processing methods for fine tuning the calibration data (lines). Experimental comparison of points and lines based calibration shows that the automatic fine tuning of the lines is beneficial for the calibration process in noisy conditions. Experiments on both synthetic and real setups show that despite the low computational complexity of the DLT, the proposed calibration methodology yields promising accurate results.

I. INTRODUCTION

Scheduling calibration to occur just after installing a network of cameras has the advantage of allowing the choice of zoom and focus in-situ, according to the scenarios at hand. On the other hand, this in-situ calibration usually turns un-practical the conventional calibration tools. Imaging a known pattern is required in conventional calibration methodologies such as the ones proposed by Tsai [1], Heikkilä [2], Zhang [3] and Bouguet [4]. Precise calibration demands that the known pattern is imaged covering most of the imaging area, which implies that the pattern has to be impractically large if the camera is mounted in 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 cameras of a network of cameras. In other words, are not designed to provide a global coordinate system for all cameras.

Creating a global coordinate system for a set of cameras having non-overlapping fields of view (FOV) has been approached in various works [5], [6]. In [5], prior knowledge of the dynamics of a mobile target, tracked by the cameras, has been shown to compensate for the lack of overlap between the camera fields of view. In [6] a team of mobile robots is used to provide a global coordinate system to a network of fixed cameras. One of the robots carries a total station for measuring precisely the angle and distance of the other robots which are also tracked by the fixed cameras. In [5]

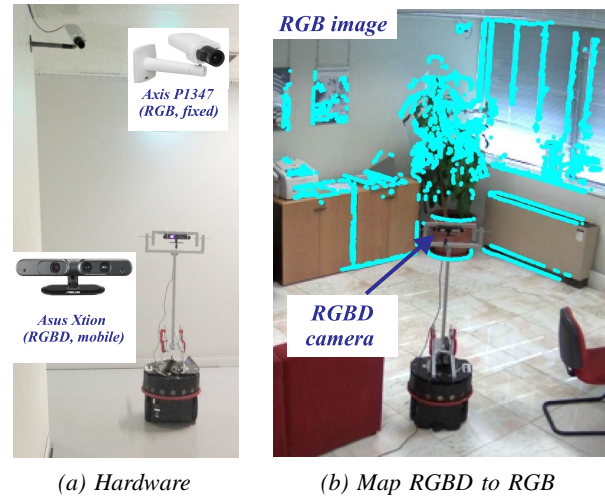*(a) Hardware*   *(b) Map RGBD to RGB*

Fig. 1. A calibrated color-depth camera, ASUS X-Tion (RGBD) allows calibrating a color camera (RGB), Axis P1347 HD (a). After calibration, the RGBD data, e.g. edge points and their 3D positions, can be mapped over the RGB image (b).

and [6] there are not required calibration patterns, but the robots have to be imaged by the network of cameras. Noting that mobile robots equipped with ranging and imaging sensors effectively allow for Simultaneously Localizing and Mapping the environment (SLAM), it is natural to generalize camera calibration to rely on SLAM done by the mobile robots, instead of just using the robots as targets to track. This is advantageous for example to cover fields of view encompassing areas that cannot be traversed by the robots. In qualitative terms, this redefines the camera calibration from a *see me* paradigm, in which a known pattern or robot has to be observed, to a *see what I see* paradigm, where a robot maps the scene and provides that information for camera calibration.

Laser Range Finders (LRF) combined with SLAM proved to reliably provide scene information (3D clouds of points) of large areas [7], [8]. As proposed in [9], the 3D maps can therefore be used to calibrate a camera by selecting a region of interest on the map and adjusting an initial guess to the projection matrix. This adjustment is done by minimizing the re-projection error of the 3D points on the camera FOV.

Recently, Color-Depth Cameras, also known as RGBD cameras, have become an interesting low cost alternative to LRF [10]. A set of 3D points is simply acquired by

back-projecting 2D points from the RGBD image plane. Features can be detected in a network camera image and then matched with the RGBD image points. This defines a set of 2D-to-3D points correspondences which can be used to estimate the camera projection matrix using the Direct Linear Transformation (DLT) [11], [12].

In this paper we introduce an in-situ calibration methodology, based in the DLT, that allows estimating the camera projection matrix and radial distortion using image lines and 3D lines represented by 3D points. The 3D data is acquired by a mobile robot equipped with a calibrated color-depth camera. The mobile robot/camera provides a global coordinate system to the fixed cameras or, in other words, the fixed cameras are fused into a global map.

The paper is organized as follows. Sec.II introduces the camera projection model, including radial distortion, and briefly describes DLT based on point correspondences. Sec.III introduces the proposed calibration methodology, *DLT-Lines*, based on image lines. An experimental noise analysis is also included. Calibration experiments on simulated and real setups are presented in Sec.IV. Finally, conclusions are drawn and future work is discussed in Sec.V.

II. CAMERA MODEL AND *DLT-Points*

A. Pin-hole Camera Model

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 $m = [u\ v\ 1]^T$:

$$m \doteq P\ M = K\ [R \quad t]M \tag{1}$$

where $\doteq$ denotes equal up to a scale factor, P is a 3×4 projection matrix, K is a 3×3 upper triangular matrix containing the intrinsic parameters of the camera, R is a 3×3 rotation matrix representing the orientation of the camera and t is a 3×1 vector representing the position of the camera [13]. The rotation, R and translation, t are defined with respect to a fixed absolute (world) coordinate frame. Having estimated the camera projection matrix, the intrinsic and extrinsic parameters can be estimated by decomposing P [14].

B. Radial Distortion

As noted by Fitzgibbon [15], true lens distortion curves are typically very complex to represent, implying the use of high-order models or lookup tables to model camera radial distortion effect with high precision. On the other hand, considering typical computer vision applications, accuracies of the order of a pixel are all that is required, and an approximation to the cameras' true distortion functions perform as well as the preciser ones.

Fitzgibbon proposed the so called *Division Model* where an undistorted image point, $\hat{m}_u = [u_u\ v_u]^T$ is computed from a radially distorted image point $\hat{m}_d = [u_d\ v_d]^T$. More precisely, $\hat{m}_u = \hat{m}_d/(1 + \lambda \|\hat{m}_d\|^2)$, where λ represents the

radial distortion parameter. The *Division Model* can also be conveniently written in homogeneous coordinates:

$$\begin{bmatrix} u_u \\ v_u \\ 1 \end{bmatrix} \doteq \begin{bmatrix} u_d \\ v_d \\ 1 + \lambda \|\hat{m}_d\|^2 \end{bmatrix}. \tag{2}$$

Note that an undistorted point, $m_u = [u_u\ v_u\ 1]^T$ is a simple function of a distorted point, $m_d = [u_d\ v_d\ 1]^T$:

$$m_u \doteq m_d + \lambda e_d \tag{3}$$

where $e_d = [0\ 0\ \|\hat{m}_d\|]^T$. 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 $\hat{c}_o = [c_u\ c_v]^T$.

C. DLT-Points

The Direct Linear Transformation (DLT), developed by Aziz and Karara [11], [14], allows estimating the camera projection matrix, P, by solving a linear system on the matrix entries based on a set of 3D points, $\{M_i : M_i = [X_i\ Y_i\ Z_i\ 1]^T\}$ and the 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 Eq.1, $m_i \times m_i = m_i \times (P\ M_i)$, results in zero in the left hand side of the equation 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 Kronecker product [16], $\otimes$, allows one to obtain an equation factorizing the data and variables to estimate:

$$(M_i^T \otimes [m_i]_\times)\ vec(P) = 0 \tag{4}$$

where $vec(P)$ denotes the vectorization of the matrix P, formed by stacking the columns into a single column vector. Each pair (M_i, m_i) allows writing Eq.4 once, and thus provides a set of three equations in the entries of $vec(P)$, but only two of them linearly independent. In order to estimate P one has to have at least six pairs of 3D-to-2D corresponding points.[1] Pre-normalization of the input data is crucial on implementing this algorithm as noted by Hartley in [18]. Hartley suggested that the appropriate transformation is to translate 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.

The Fitzgibbon's division model allows a simple 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 Eq.3, in the *DLT-Points* factorized equation (Eq.4) results in:

$$\left(M_i^T \otimes [m_{id} + \lambda e_{id}]_\times\right)\ vec(P) = 0 \tag{5}$$

which can be rewritten as $(A_{i1} + \lambda A_{i2})vec(P) = 0$, where $A_{i1} = M_i^T \otimes [m_{id}]_\times$ and $A_{i2} = M_i^T \otimes [e_{id}]_\times$. Considering

---

[1]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 $\|A\ vec(P)\|^2$ s.t. $\|vec(P)\| = 1$ [17].

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 [15], left-multiplying the stacked matrices by A_1^T results in a Polynomial Eigenvalue Problem (PEP), $(A_1^T A_1 + \lambda A_1^T A_2)vec(P) = 0$, which can be solved for example in Matlab using the `polyeig` function. Its solution gives simultaneously the projection matrix, $vec(P)$ 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 [15], 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 overcome the approximation.

III. CALIBRATION BASED ON IMAGE LINES

In this section we introduce the calibration methodology *DLT-Lines* including the estimation of radial distortion. As indicated by the name, we consider lines identified on the image of the camera to obtain its parameters. Contrarily to 3D lines, which are normally represented using Plucker coordinates [14], 2D lines have simple representations as cross products of image points in homogeneous coordinates. In the following we explore this representation to build the calibration methodology. The use of lines, as opposed to using isolated image points, brings an advantage. Image processing can be used for fine tuning the location of the lines in the image and therefore automatically improving the calibration data input.

As in *DLT-Points* two cases are considered, namely (i) non-existent radial distortion and (ii) significant radial distortion. In the case where the radial distortion is considered, it is modeled using Fitzgibbon's division model.

A. DLT-Lines

Given a 3D line L_i, its projection on the camera image plane, l_i can be represented by the cross product of two image points in projective coordinates, $l_i = m_{1i} \times m_{2i}$. Any point m_{ki} lying in the line l_i implies that $l_i^T\ m_{ki} = 0$. Applying the multiplication by l_i^T on both sides of Eq.1, i.e., $l_i^T\ m_{ki} = l_i^T\ P\ M_{ki}$, leads to:

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

where M_{ki} is a 3D point in projective coordinates lying in L_i. As in the case of *DLT-Points*, using the Kronecker product one obtains a form factorizing the vectorized projection matrix:

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

Each pair of 3D point and its corresponding image line, (M_{ki}, l_i), allows writing Eq.7 once, and thus provides one linear constraint in the entries of $vec(P)$. In order to estimate P one needs at least 12 pairs (M_{ki}, l_i). [2] Considering $N \geq$
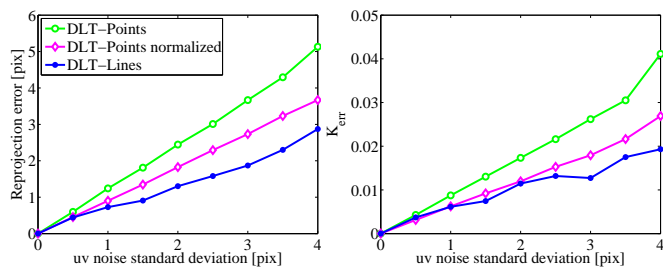

Fig. 2. Comparing *DLT-Points* and *DLT-Lines* using a VRML setup. Points chosen are extremes of line segments (left plot). Gaussian noise added in the points (image) data indicated in the horizontal axis. K_{err}, right plot, denotes horizontal focal length relative error[4].

---

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, more precisely the minimizer of $\|B\ vec(P)\|^2$ s.t. $\|vec(P)\| = 1$, is the right singular vector corresponding to the least singular value of B.

From Eq.7 and Eq.4 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 $vec(P)$, allowing to 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 indicates, for example, many-3D-points to one-2D-line correspondence. Any point $M_{ki} \in L_i$ forms a linear constraint with l_i (Eq.7). This property of *DLT-Lines* allows to apply additional image processing tools that add robustness to the extraction of calibration data. In particular, *DLT-Lines* involves finding lines both in the RGB and RGBD images. These 2D-lines, l_i can be fine tuned to better match edge points, i.e. local gradient information:

$$l_i^* = \arg_{l_i} \max \sum_k \|\nabla I(m_k)\|, \quad m_k \in (l_i \cap \mathcal{R}) \qquad (8)$$

where I is a RGB converted to gray level image, ∇ denotes image gradient and $\mathcal{R}$ is a region of interest containing a straight-line segment plus some tolerance (e.g. ± 10 pixel around the segment extremes). 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 [14].

Figure 2 compares *DLT-Points* with *DLT-Lines* in a synthetic (VRML [19]) setup. The calibration data is based on a number of image points indicating extremes of line segments corresponding to edge lines. The 3D data is inferred from

---

[2]Alternatively, one can state that nondegenerate six 3D lines configuration and their corresponding six image lines are enough to estimate P.

[4]The decomposition of estimated projection matrix, detailed in [14], 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 $K(3,3) = K_e(3,3) = 1$.

the 2D data (back-projection of image points intersecting scene facets of the VRML model). The image points are disturbed by adding Gaussian noise. The locations of the image lines are fine tuned based on Eq.8. The plots show the mean reprojection error and the mean of the relative error of the estimated horizontal focal length, considering 100 calibration experiments for each noise level (standard deviation, horizontal axis). The two plots are similar, showing that the common cost function used in nonlinear calibration, the mean reprojection error, is effectively a good indicator of the accuracy of the calibration. As noted by Hartley [18], data normalization (magenta) improves the conditioning of the *DLT-Points* problem (green). The fine tuning of the lines (blue) brings some additional decrease of the reprojection error.

B. DLT-Lines *with Radial Distortion*

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

$$l_{12} = \begin{bmatrix} u_{1d} \\ v_{1d} \\ 1 + \lambda s_1^2 \end{bmatrix} \times \begin{bmatrix} u_{2d} \\ v_{2d} \\ 1 + \lambda s_2^2 \end{bmatrix} = \hat{l}_{12} + \lambda e_{12} \quad (9)$$

where s_i is the norm of distorted image point i, $s_i^2 = u_{id}^2 + v_{id}^2$, the distorted line is denoted as $\hat{l}_{12} = [u_{1d} \ v_{1d} \ 1]^T \times [u_{2d} \ v_{2d} \ 1]^T$ and there is a distortion correction term $e_{12} = [v_{1d}s_2^2 - v_{2d}s_1^2, \ u_{2d}s_1^2 - u_{1d}s_2^2, \ 0]^T$. Applying Eq.9 into the point-to-line constraint, Eq.7, one has:

$$\left(M_{k12}^T \otimes (\hat{l}_{12} + \lambda e_{12})^T \right) vec(P) = 0 \quad (10)$$

which can be rewritten as:

$$\left(B_{ki1} + \lambda B_{ki2} \right) vec(P) = 0 \quad (11)$$

where $B_{ki1} = M_{k12}^T \otimes \hat{l}_{12}^T$, $B_{ki2} = M_{k12}^T \otimes e_{12}^T$ and M_{k12} denotes the kth 3D point projecting to the distorted line l_{12}. 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}. Using once more Fitzgibbon's suggestion [15], left-multiplying the stacked matrices by B_1^T results in a Polynomial Eigenvalue Problem (PEP), $(B_1^T B_1 + \lambda B_1^T B_2) \ vec(P) = 0$, which can be solved for example in Matlab using the `polyeig` function. Its solution gives simultaneously the projection matrix, $vec(P)$, the radial distortion parameter λ, and $P' = T^{-1}P$, where T is defined in sec II.C. In a similar way as explained before, both DLT methods applied to the radial distorted camera, can be combined to estimate P and λ.

In order to organize and summarize the aspects already described, we outline now the complete *DLT-Lines* calibration methodology (see Fig. 3). As input one has 2D lines in a RGBD image acquired by a calibrated camera[5], and 2D lines in a RGB image acquired by the camera to calibrate. Each 2D line of the RGB image is described by a number of points.

[5]Alternatively, one can have simply 3D points describing 3D lines (two points per line).
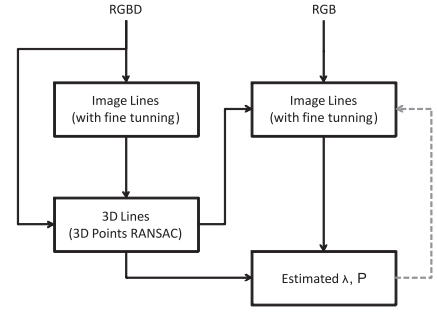

Fig. 3. Camera calibration methodology, using Color-Depth (RGBD) camera and *DLT-Lines*.

Usually one needs more than two points per line in order to identify the radial distortion. Step 1 of the methodology consists of estimating λ, P and finally P' (using equations 9 till 11). Step 2 consists of a local fine tuning of the lines in the RGB image. The image lines are composed by a number of parts (as described by the original number of 2D points). For each of the parts of a line, an optimization process is run as described by Eq.8. Steps 1 and 2 are repeated a number of times, using as an approximation of the principal point the values extracted from the estimated P', in order to overcome approximation of the principal point done in the initialization. The process stops when the fine tuning of the lines does not change significantly the lines.

IV. EXPERIMENTS

In this section the proposed calibration methodology is tested in one simulated and one real setups. The simulated setup consists of a VRML world containing fixed RGB cameras and one mobile RGBD camera. A surveillance camera network and a mobile robot equipped with a Asus X-tion (RGBD) camera, are used for the acquisition of the real data.

A. Virtual Reality Setup

A virtual reality world representing an indoor environment, containing one mobile and various fixed cameras, has been built in VRML in order to create a calibration setup with ground truth and thus allowing for a quantitative assessment of the precision of *DLT-Lines*. The VRML was programmed using the Virtual Reality toolbox of Matlab. The depth of the RGBD camera is computed using the projection matrix and the set of facets composing the scenario.

The first step of the experiment involves identifying corresponding lines both in the RGBD and RGB images. Then, a line fitting algorithm fine tunes the location of each line, as described in Sec.III-A (see Eq.8). Since the RGBD camera gives the depth for each image pixel (using the depth map), one can obtain a 3D line from each 2D line, considering that the intrinsic and extrinsic parameters of the camera are known. Figure 4 shows the input data, lines in both RGBD and RGB images (Fig.4(d) and (e)), and results of *DLT-Lines* calibration in the VRML setup ((Fig.4(g)).

Given the calibration data, i.e. N 3D line points, M_l, and their projections on the RGB image plane, m_l, the

(a) Setup   *(b) RGBD data*   *(c) RGBD depth map*

(d) RGBD data   *(e) RGB data*   *(f) Qualitative verification*   *(g) Result and ground truth*
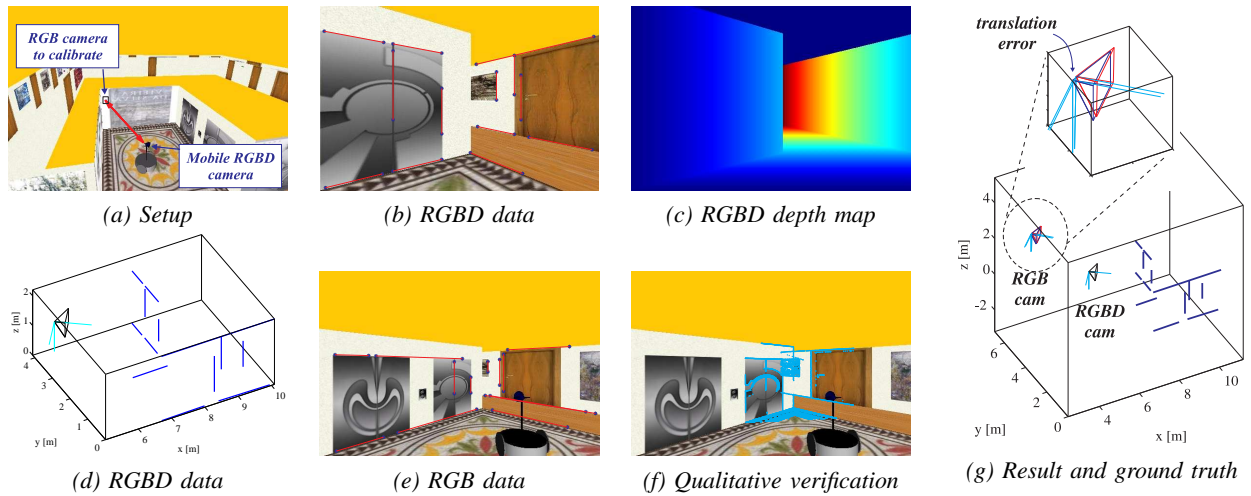
Fig. 4. Camera calibration in a VRML setup (a). Each line defined in the RGBD image (b,c) leads to a 3D line in the world/RGBD coordinate system (d). The RGBD lines and the corresponding lines in the RGB image (e), form the required input data for *DLT-Lines* calibration. After the projection matrix of the RGB camera has been estimated, any RGBD 3D info, such as edge points and their 3D locations, can be mapped to the RGB image (f). Decomposing the estimated projection matrix as $K[R\ t]$ [14], provides a graphical comparison with the ground truth, blue and red cameras, respectively in (g).

reprojection error was found to have the value $Err = \sum(m_l - \hat{m}_l)^2/N = 0.4707[pix^2]$, where $\hat{m}_l$ is the estimated projection of M_l. This subpixel error is qualitatively verified to be small in Fig.4(f), where edge points found in the RGBD image were back-projected to 3D points using the depth information and projected to the RGB camera using the estimated projection matrix. In particular one verifies that the edge points of the RGBD image transported to the RGB image are consistent with the edges of the RGB image.

Selecting the (horizontal) focal length as a representative of the intrinsic parameters, the difference between estimated and real values was found to be $K_{err} = 4.9 \times 10^{-5}$, where K_{err} is defined as in Fig.2. The estimation error in the camera orientation can be assessed using a distance between the real and the estimated rotation matrices [6]. The distance between rotation matrices was found to be $R_{err} = 0.01[rad]$. The camera position can be evaluated comparing the real distance between RGBD and RGB cameras, $d = 3.2454[m]$, and the estimated one, $d_e = 3.2546[m]$. The difference between these distances is $d_{err} = 0.0092[m]$, meaning a relative error of approximately 3×10^{-3}. Considering that the experiment involved using rendered images, and thus introduced significant pixelization error in the calibration data, the absolute and relative errors are effectively small.

B. Real Setup

In this experiment, the objective is to calibrate an Axis P1347 high definition surveillance camera (RGB), with radial distortion, installed on a waiting room. A ASUS X-Tion (RGBD) camera, mounted on mobile platform, is used to capture 3D scene information. The RGBD camera is assumed to be calibrated.

Figures 5(b) and (f) show the lines identified on the RGBD and RGB cameras. The noise in the depth map, Fig. 5(c),

[6]The distance between two rotation matrices, R_1 and R_2, can be calculated using the norm of vector $v_d = logm(R_1^T R_2)$, where $logm$ denotes the matrix logarithm. The difference is given as $R_{err} = 1/2\|v_d\|$.

implies noise in the 3D lines which can be attenuated using RANSAC (see Fig. 5(d)). The data in Figs.5(e) and (f), allow applying *DLT-Lines* and obtaining the results shown in Fig.5(g). As in the simulated setup, a qualitative assessment of the precision of the calibration can be made by transporting edges from the RGBD image to the RGB image (see Fig.1(b), where the RGB image Fig. 5(f) has been cropped to show just the area covered by the RGBD camera). The qualitative good conformance of transported edges and RGB edges asserts a qualitatively precise calibration.

In order to obtain quantitative assessment, some more tests have been conducted. In particular J. Y. Bouguet's calibration toolbox [4] was used to estimate the intrinsic parameters of the RGB camera. The difference between the (horizontal) focal length obtained using Bouguet's toolbox and the one extracted from the estimated projection matrix using *DLT-Lines*, was found to be $K_{err} = 0.05$ (relative error defined as described in Fig.2). The real distance from RGBD to RGB was $d = 3.55[m]$, measured with a tape, while the estimated using *DLT-Lines* was found to be $d_e = 3.53[m]$. The small relative errors assert that *DLT-Lines* can provide accurate results.

V. CONCLUSIONS AND FUTURE WORK

This paper introduces *DLT-Lines*, a camera calibration methodology based in the DLT. While in the ancestor methodology, *DLT-Points*, the input data is formed by paired 3D-to-2D points, in *DLT-Lines* the input data is formed by a set of image lines and the corresponding 3D lines described by 3D points. Similarly to *DLT-Points*, *DLT-Lines* has been extended to consider radial distortion, using the division model proposed by Fitzgibbon [15].

One advantage of *DLT-Lines* is that one or more 3D points can be combined with two or more 2D points defining an image line, instead of exact 1-to-1 correspondences as required by *DLT-Points*. This is interesting as it allows

(a) Setup   *(b) RGBD data*   *(c) RGBD depth map*   *(d) RANSAC on one 3D line*

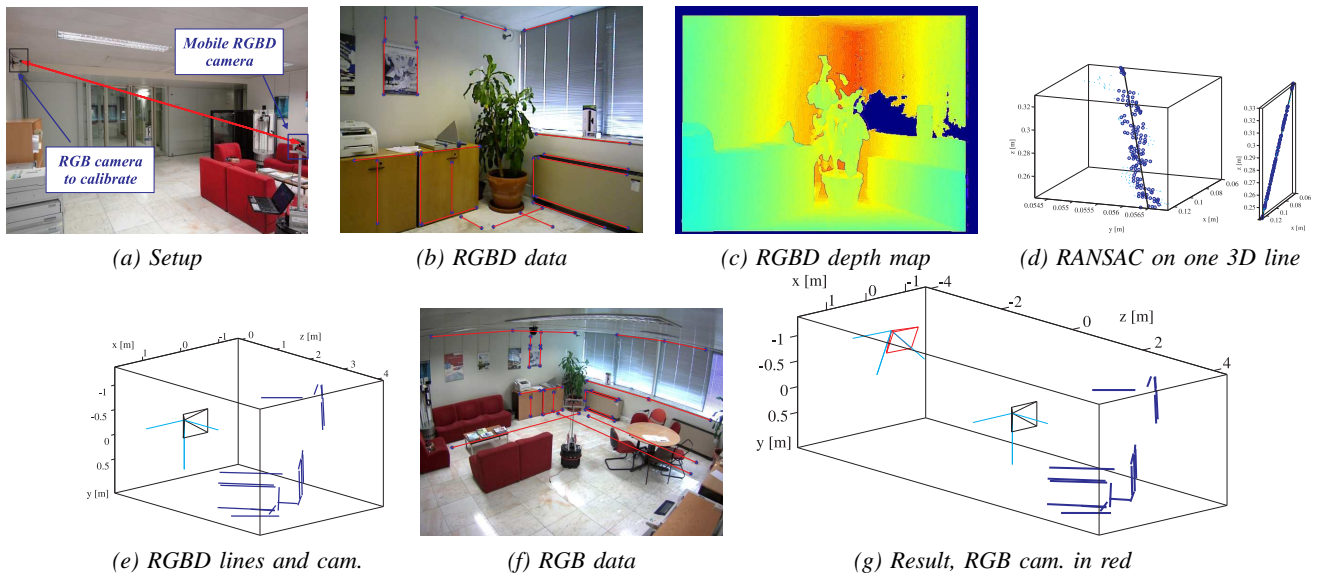*(e) RGBD lines and cam.*   *(f) RGB data*   *(g) Result, RGB cam. in red*

Fig. 5. Calibration of a surveillance camera, Axis P1347 (RGB), using a mobile robot equipped with a color-depth camera, Asus X-Tion (RGBD) (a). Lines in the RGBD image (b,c) define 3D lines (d,e). Each line formed directly from the depth map (cyan dots) is filtered using RANSAC (blue and black dots), as shown in (d), where the left/right plot has different/equal scales in the axis. RGBD (e) and RGB lines (f), form the input dataset for *DLT-Lines*. Decomposing the estimated projection matrix as $K[R\ t]$, provides the camera position and orientation on the world coordinate system (g).

applying automatic filtering techniques such as RANSAC on 3D points defining a line, or fine tuning 2D lines to come closer to local maxims (*roof ridges*) of module image gradient. Simulation involving Gaussian noise on the input data showed that *DLT-Lines* can outperform *DLT-Points* in terms of reprojection error, because of the additional (automatic) filtering.

Experiments on a camera network simulated using VRML allowed assessing *DLT-Lines* performance, in terms of estimated camera parameters, by comparing the results of the calibration of the cameras with the ground truth. *DLT-Lines* proved to be accurate, estimating the camera intrinsic and extrinsic parameters with precision. *DLT-Lines* was tested also in a real indoors setup, involving the calibration of a real high resolution camera using a low budget color-depth camera to acquire 3D information. In this case the estimated intrinsic parameters were compared with Bouguet's calibration. Was verified that the results were similar (order of 10^{-2} relative error), and therefore confirmed the effectiveness of the proposed methodology.

In terms of future work, Steele and Jaynes [20] have shown that is possible to improve the numerical accuracy of the polynomial eigenvalue problem as introduced by Fitzgibbon [15], which can be beneficial also for *DLT-Lines*.

REFERENCES

[1] R. Tsai, "A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras," *IEEE J. Robot. Automat.*, vol. 3, no. 4, pp. 323–344, Aug 1987.

[2] J. Heikkilä and O. Silvén, "A four-step camera calibration procedure with implicit image correction." in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition.*, 1997, pp. 1106–1112.

[3] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pat. Anal. Machine Intel.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[4] J. Bouguet, "Camera calibration toolbox for matlab," http://www.vision.caltech.edu/bouguetj.

[5] A. Rahimi, B. Dunagan, and T. Darrell, "Simultaneous calibration and tracking with a network of non-overlapping sensors," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition.*, 2004, pp. 187–194.

[6] T. Yokoya, T. Hasegawa, and R. Kurazume, "Calibration of distributed vision network in unified coordinate system by mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 1412–1417.

[7] V. Ila, J. Andrade-Cetto, R. Valencia, and A. Sanfeliu, "Vision-based loop closing for delayed state robot mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, Nov. 2007, pp. 3892–3897.

[8] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[9] A. Ortega, B. Dias, E. Teniente, A. Bernardino, J. Gaspar, and J. Andrade-Cetto, "Calibrating an outdoor distributed camera network using laser range finder data," in *Proc. IROS 2009*, 2009, pp. 303–308.

[10] F. Endres, J. Hess, N. Engelhard, J. Sturm, and W. Burgard, "Openslam," http://openslam.org/rgbdslam.html.

[11] A. Aziz and H. Karara, "Direct linear transformation into object space coordinates in close-range photogrametry," in *Proc. of the Symposium on Close-Range Photogrammetry.*, 1971, pp. 1–18.

[12] M. Hansard, R. Horaud, M. Amat, and S. Lee, "Projective alignment of range and parallax data," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition.*, 2011, pp. 3089–3096.

[13] O. Faugeras, *Three-Dimensional Computer Vision - A Geometric Viewpoint. Artificial intelligence.* M.I.T. Press, 1993.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[15] A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition.*, vol. 1, 2001, pp. 125–132.

[16] H. Lütkepohl, *Handbook of Matrices.* Wiley and Sons, 1996.

[17] R. Hartley, "Minimizing algebraic error in geometric estimation problems," in *Proc. of the 6th Int. Conf. on Computer Vision*, 1998, pp. 469–476.

[18] ——, "In defense of the eight-point algorithm," in *IEEE Trans. Pattern Anal. Machine Intelligence*, vol. 19, 1997, pp. 580–593.

[19] W. Consortium, "Virtual reality modeling language," http://www.w3.org/MarkUp/VRML/.

[20] R. Steele and C. Jaynes, "Overconstrained linear estimation of radial distortion and multi-view geometry," in *Computer Vision – ECCV 2006.* Springer, 2006, pp. 253–264.