Citation: A. Pérez-Yus, E. Fernández-Moral, G. López-Nicolás, J. J. Guerrero, P. Rives. Extrinsic calibration of multiple RGB-D cameras from line observations. IEEE Robotics and Automation Letters, vol. 3(1), pp. 273-280, 2018.

Extrinsic calibration of multiple RGB-D cameras from line observations

Alejandro Perez-Yus[1], Eduardo Fernandez-Moral[2], Gonzalo Lopez-Nicolas[1], Jose J. Guerrero[1] and Patrick Rives[2]

***Abstract*—This paper presents a novel method to estimate the relative poses between RGB and depth cameras without the requirement of an overlapping field of view, thus providing flexibility to calibrate a variety of sensor configurations. This calibration problem is relevant to robotic applications which can benefit of using several cameras to increase the field of view. In our approach, we extract and match lines of the scene in the RGB and depth cameras, and impose geometric constraints to find the relative poses between the sensors. An analysis of the observability properties of the problem is presented. We have validated our method in both synthetic and real scenarios with different camera configurations, demonstrating that our approach achieves good accuracy and is very simple to apply, in contrast with previous methods based on trajectory matching using visual odometry or SLAM.**

***Index Terms*—Calibration and Identification, Range Sensing, Sensor Fusion, Omnidirectional Vision**

I. INTRODUCTION

In the vast majority of vision-based applications related to mobile robotics and autonomous vehicles, having a large field of view (FOV) is required or provides important advantages [1], [2], [3]. It is particularly interesting when the information of the sensor comes with both color and range data. However, most of the RGB-D cameras that dominate the market have a very narrow FOV. Thus, the idea of using several RGB-D cameras to extend the FOV is very appealing. Such system needs to be calibrated in order to fuse all the data in the same reference frame. This process is called *extrinsic calibration*, and consists in estimating the relative poses between the cameras. Current solutions for this problem are time consuming and/or may require building a specific calibration pattern. Besides, there are some additional challenges, such as trying to combine different type of cameras, or finding the calibration when the cameras have no overlap in their FOV. In this work, we propose an original method to perform extrinsic calibration of RGB-D cameras, which also works for different combinations of 3D range and image sensors, even when their FOVs do not overlap.

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.

This paper was recommended for publication by Editor Editor name upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the projects DPI2014-61792-EXP and DPI2015-65962-R (MINECO/FEDER, UE), the grant BES-2013-065834 (MINECO), and the post-doctoral fellowship program of INRIA.

[1] Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, 50018 Zaragoza, Spain. alperez@unizar.es, gonlopez@unizar.es, josechu.guerrero@unizar.es

[2] Lagadic team. INRIA Sophia Antipolis - Méditerranée. 06902 Sophia Antipolis, France. eduardo.fernandez-moral@inria.fr, patrick.rives@inria.fr

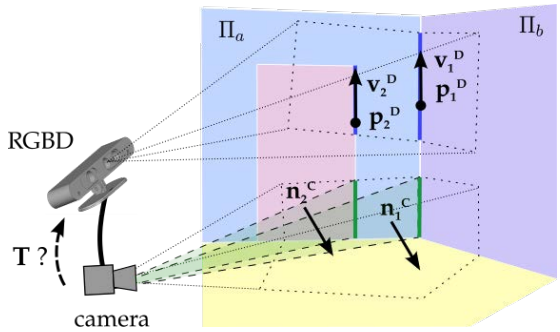

Fig. 1: Observation of lines in the scene by a pair of RGB-D and conventional cameras rigidly joined with non-overlapping field of view. Line correspondences are used to formulate geometric restrictions to compute the relative pose T between the cameras.

Several extrinsic calibration approaches for different types of camera systems have been proposed previously. A classical strategy is through the detection and matching of control points that are detected in the overlapping regions of the different cameras [4]. Line features detected by conventional cameras have also been used in a similar way to recover the essential matrices among uncalibrated cameras [5], the relative poses of calibrated cameras [6], [7], or the intrinsic and extrinsic parameters of a number of them [8], [9]. However, the overlap requirement constitutes a very strong constraint. Besides, even when some overlap exists, it is generally more complicated to match features in range images than in intensity images. With a different perspective, the use of a calibration pattern is a resource that has been employed as an *ad hoc* solution for very specific problems [10], [11]. The lack of generality of this solution for any configuration of cameras is indeed an important limitation. Also the need to create the 3D calibration pattern itself is wearisome.

A more general approach not depending on the geometric configuration of the sensors is based on ego-motion to match the camera trajectories, which are tracked independently. For that, simultaneous localization and mapping (SLAM) or visual odometry (VO) techniques are applied [12], [13], [14]. However, this kind of solution is laborious to apply, requiring robust SLAM or VO in controlled environments. Besides, they may not be able to fully observe the calibration parameters depending on the movement restrictions, e.g. the translation in the vertical axis is unobservable in the case of planar motion, so common for a wheeled robot or autonomous vehicle.

A solution for the extrinsic calibration of 3D range sensors

based on 3D plane observations was presented in [15] which works with non-overlapping sensors. This solution only requires the co-observation of planar surfaces by the different sensors and it is easy to apply, but it cannot be used with conventional cameras. Similar approaches have later been presented for other kind of sensors based on the observation of large geometric features from the scene, like [16] for a set of 2D laser range finders (LRF), or [17] which calibrates a camera and a 2D LRF. The approach presented here is inspired by this kind of solutions. In particular, it is based on the observation and matching of lines in the scene from the different sensors, which are used to formulate constraints on the relative poses of the cameras. Our method allows to calibrate any system with:

1) One sensor able to extract the parameters to completely define a line in 3D space (e.g. a depth camera).
2) One (or several) sensors able to extract the lines in the projective plane (e.g. a standard camera).

Thus, our method calibrates Color-Depth pairs $\{C,D\}$, Depth-Depth pairs $\{D_1,D_2\}$ and larger systems with N^C color cameras and N^D depth cameras $\{D_1..D_{N^D},C_1..C_{N^C}\}$ whenever $N^D \geq 1$ and $N^D+N^C \geq 2$.

Before the calibration we describe our line extraction process in color images (from both conventional and omnidirectional cameras) and from range data. In particular, from range data the lines are found as plane intersections. RGB-D sensors are a special case where we can use line extraction in the RGB image, and then use the depth to get the 3D line. This situation is illustrated in Fig. 1, where the line with sub-index 1 corresponds to the intersection of two planar surfaces (Π_a and Π_b) and thus, its 3D parameters are observable by a depth camera, while the 3D parameters of the line with sub-index 2 (which is contained in the plane Π_a) are only observable by an RGB sensor. After line extraction, we propose a robust method to find line matchings via a RANSAC approach. We have additionally included an analysis of the observability of the problem, where we discuss the minimum amount of line-matchings necessary and degenerate cases.

The main contribution of this work is a novel method for extrinsic calibration of an RGB-D multi-camera system based on line observations. Our method has important advantages with respect to other approaches in the literature: *i)* no overlapping fields of view are required among the sensors; *ii)* it can be used to calibrate different types of cameras; and *iii)* it avoids needing to build a calibration pattern. We performed experiments in simulation and with real images with different camera combinations. These experiments show the validity of our method and test the accuracy and real-world usability of the approach. We demonstrate the calibration of: an RGB-D sensor from a public dataset consisting on common indoor scenes; a fisheye camera and a depth camera; two non-overlapping RGB-D cameras; and a rig of 8 RGB-D cameras arranged in a radial configuration for omnidirectional FOV.

II. LINE EXTRACTION AND MATCHING

We use *line* to refer to the line in 3D space, and *segment* to refer to the set of collinear points found in a conventional image. Mathematically, a *line* is the set of points $\mathbf{p} \in \mathbb{R}^3$ that satisfy the following equation:

$$\mathbf{p} = \mathbf{p}_0 + \lambda \mathbf{v} = (p_{0x}, p_{0y}, p_{0z}) + \lambda\,(v_x, v_y, v_z)\,, \quad \forall \lambda \in \mathbb{R} \quad (1)$$

being $\mathbf{p}_0 \in \mathbb{R}^3$ a point in the line, and $\mathbf{v} \in \mathbb{R}^3$ the direction vector of the line (see Fig. 1). We also define the *projective plane* of the line π as the 3D plane that contains the line and the origin of the reference system (i.e. the optical center of the camera). The normal $\mathbf{n} \in \mathbb{R}^3$ of this plane (see Fig. 1), also known as the *moment vector* of the line, is the vector perpendicular to $\mathbf{p}_0$ and $\mathbf{v}$, i.e. $\mathbf{n} = \mathbf{p}_0 \times \mathbf{v}$. In the following sections we describe how we extract lines in an image depending on the type of camera used.

A. Line extraction in RGB camera

Due to the projective nature of conventional cameras we cannot compute the direction vector $\mathbf{v}$, nor any 3D point $\mathbf{p}$. Nonetheless, we can extract segments in the image to retrieve the normal vectors $\mathbf{n}$ of their projective planes. For this, the camera must be intrinsically calibrated in advance. The process of getting segments is a traditional problem in computer vision, which can be solved using widespread algorithms. In particular, our approach goes as follows:

1) Apply a Canny filter [18] to extract edges in the intensity image.
2) The edge points are grouped in *boundaries* formed by consecutive points in the image.
3) For each boundary, we apply a RANSAC procedure [19] to get the lines in 2D. The 2D lines have a direction vector in the image $\mathbf{l} = [l_x, l_y, 0]^\top$ and a set of k inlier edge points $\{\mathbf{u}_1..\mathbf{u}_k\}$ where $\mathbf{u}_i = (u_i, v_i)$.
4) The mean of the inlier points $\bar{\mathbf{u}}$ is used to compute the 3D ray $\mathbf{r} = [(\bar{u}-c_x)/f_x,\ (\bar{v}-c_y)/f_y,\ 1]^\top$ with camera's optical center (c_x, c_y) and focal length $(f_x,\ f_y)$.
5) The normal $\mathbf{n}$ is $\mathbf{n} = \mathbf{l} \times \mathbf{r}_i$

Some examples of line extraction are shown in Fig. 2. While this approach is appropriate for standard cameras, we can substitute this method to a more suitable one if we need to use a more complex type of camera. The work from Bermudez-Cameo *et al.* [20] presents a line extraction method for uncalibrated omnidirectional cameras with revolution symmetry. With this method we can calibrate a wide range of ominidirectional cameras like the fisheye shown in Fig. 3 (a).

B. Line extraction in depth camera

A depth camera permits to obtain the 3D line parameters (i.e. we can extract $\mathbf{p}$, $\mathbf{v}$ and $\mathbf{n}$ from the lines in the image). The strategy to obtain the lines may depend on the sensor. Using only depth information, the simplest way to retrieve 3D lines is by looking for 3D plane intersections. For example, in Fig. 1, the line $\{\mathbf{p}_1, \mathbf{v}_1\}$ is the intersection of the planes Π_a and Π_b (e.g. the intersection of two walls). We extract 3D planes using RANSAC for plane fitting. Some samples of real scenes with the planes extracted are shown in Fig. 2 and Fig. 3 (b). A plane Π_i has normal $\mathbf{n}_i$ and distance to the origin d_i so that all points $\mathbf{X}$ belonging to the plane satisfy

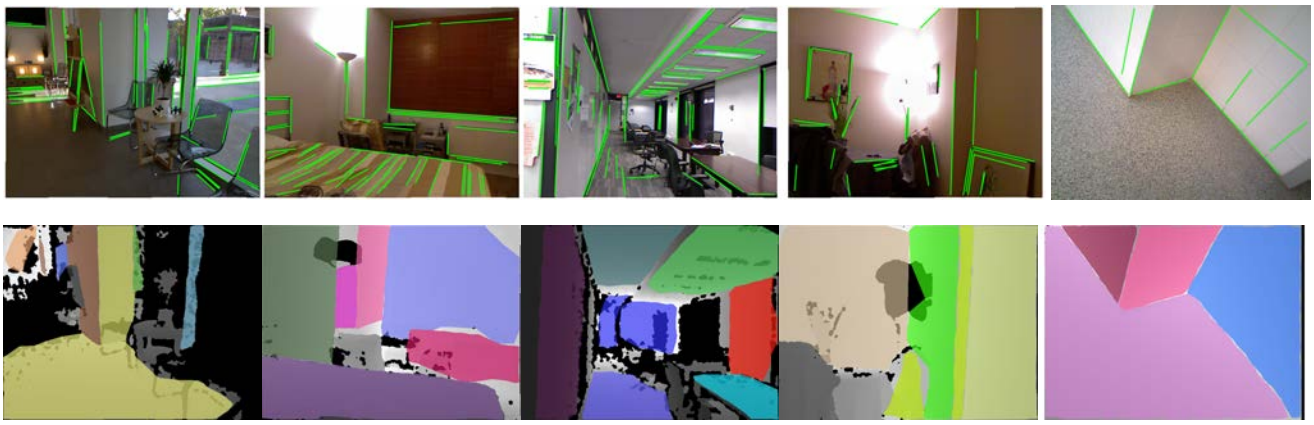

Fig. 2: Examples of common indoor scenes used for calibration, like wall-wall and wall-ceil junctions. The first row shows the RGB images with the lines extracted in green, the second row shows the corresponding depth images with the planar surfaces colored in different colors. Images (1-4) belong to the NYU2 RGB-D dataset [21].

(a) (b) (c) (d)

Fig. 3: (a) Lines extracted in a fisheye image (in green). The two relevant ones are selected in blue. (b) View from the RGB-D camera. Top: the RGB image (not used for calibration). Bottom: the planar segmentation in the depth image used to extract 3D line from plane intersections. (c) 3D planes projected to the fisheye view with the corresponding lines in blue (from RGB) and red (from depth). (d) After calibration, the projection of the 3D lines and the depth map fits the color image.

$\mathbf{n}_i \cdot \mathbf{X} + d_i = 0$. To compute the 3D line between two planes Π_a and Π_b, we get the direction $\mathbf{v}$ as the cross product of their normals, $\mathbf{v} = \mathbf{n}_a \times \mathbf{n}_b$. A 3D point of the line $\mathbf{p}_0$ is obtained as the closest point to the origin that fulfills the equations $\mathbf{n}_a \cdot \mathbf{p}_0 + d_a = 0$ and $\mathbf{n}_b \cdot \mathbf{p}_0 + d_b = 0$.

In the case of an RGB-D camera already calibrated (with per-pixel correspondence between color and depth), the easiest way to proceed is to use the segment extraction described for an RGB camera, and use the depth information to transform the segment points to 3D. RANSAC can be used to remove possible outliers in the 3D points that define the line. This approach has the advantage of being able to extract lines on planes, which is not possible only with depth data. That is the case of the line $\{\mathbf{p}_2, \mathbf{v}_2\}$ contained in the plane Π_a in Fig. 1, which can be detected from the color changes.

C. Line correspondences between cameras

After line extraction, we create a set of N_L line correspondences $\mathscr{L}_{i=1..N_L}$. In our notation, we call D the camera with depth information and C the conventional one (color or monochrome). Every correspondence $\mathscr{L}_i$ consists of a fully parametrized 3D line from D and the normal of the projective plane from C, i.e. $\mathscr{L}_i = \{\mathbf{p}_i^D, \mathbf{v}_i^D, \mathbf{n}_i^C\}$. For example, in Fig. 1 we can observe $\mathscr{L}_1$ and $\mathscr{L}_2$, and a real case with two correspondences in Fig. 3 (c).

An automatic procedure to extract line correspondences based on RANSAC is implemented as follows:

1) Extract all the lines in C and D for each image pair independently to create a broad set of correspondence candidates $\mathscr{L}$.
2) Filter $\mathscr{L}$ according to an initial estimate of the relative poses of the cameras and their uncertainty by setting angular and/or distance thresholds to remove outliers.
3) Pull a minimal set (Section IV) of three random correspondences from $\mathscr{L}$ to perform the calibration as described in Section III, and count the number of consistent correspondences in $\mathscr{L}$.
4) Repeat the previous step using RANSAC to obtain the maximal consensus (inliers).
5) The final calibration is computed from the inlier correspondences.

Note that it is easier to perform the calibration from scenes without clutter, see Fig. 3 (b), where a few lines can be robustly extracted. Cluttered scenes result in higher number

of outlier correspondences which may require a better initial approximation of the calibration to be filtered out. Nevertheless, since calibration should not be performed very often, the correpondences may also be selected with human supervision to guarantee the correctness of the calibration.

III. EXTRINSIC CALIBRATION FROM LINE OBSERVATIONS

In this section we address the problem of extrinsic calibration of a depth camera D and a color camera C. Let ${}^{C}T_D = [R|\mathbf{t}] \in \mathbb{SE}(3)$ be the relative pose between the reference frame of D with respect to C. Our goal is to find the maximum likelihood estimation (MLE) for ${}^{C}T_D$. Since rotation and translation can be decoupled, we separate the process in two stages, computing first the rotation R and then the translation $\mathbf{t}$. We consider that our line observations are affected by unbiased Gaussian noise $N(0,\sigma)$, uncorrelated between different line correspondences. Under this assumption, the MLE is equivalent to the solution of the least-squares minimization of the geometric errors of the line correspondences for the rotation and the translation.

A. Rotation estimation

From the definitions in the previous section, the direction vector of a line $\mathbf{v}$ is orthogonal to the normal vector $\mathbf{n}$. This condition holds between separate cameras C_1 and C_2 after applying the corresponding relative rotation to transform both vectors to the same reference frame. Thus, we can use the condition of orthogonality to retrieve the rotation by computing the matrix $R \in \mathbb{SO}(3)$ that satisfies:

$$\left(\mathbf{n}^{C_2}\right)^T \cdot R\mathbf{v}^{C_1} = 0 \tag{2}$$

where in our problem, $C_1 = D$ and $C_2 = C$ (we drop these super-indexes for readability). We need at least three line correspondences $\mathscr{L}_i$ to estimate the rotation R, which has three degrees of freedom. A more extended discussion about the observability of this problem is provided in section IV.

The MLE of the relative rotation is equivalent to the solution of the following non-linear least squares minimization, with the relative rotation R represented in a minimal parametrization with the exponential map from Lie algebra:

$$\arg\min_{\mu} \sum_{i=1}^{N_L} (\mathbf{n}_i^T \cdot e^{\mu} R\mathbf{v}_i)^2 \tag{3}$$

where e^{μ} is the exponential map of the increment of rotation μ on R. The vector μ has three dimensions and it is the axis-angle representation of the rotation on a manifold space of $\mathbb{SO}(3)$. We solve this non-linear least squares problem iteratively with Gauss-Newton. The increment vector μ is computed as:

$$\mu = -H^{-1} g \tag{4}$$

where H and g are the Hessian and the Gradient of the error function, computed as:

$$H = \sum_{i=1}^{N_L} J_i^T J_i \ , \qquad g = \sum_{i=1}^{N_L} J_i^T r_i \tag{5}$$

with the Jacobians and residuals given by

$$J_i = (R\mathbf{v}_i \times \mathbf{n}_i)^T \ , \qquad r_i = \mathbf{n}_i \cdot R\mathbf{v}_i \tag{6}$$

B. Translation estimation

Following a similar reasoning, the vector $\mathbf{p}$ representing any point on the 3D line is perpendicular to the normal vector $\mathbf{n}$. Therefore, the relative pose ${}^{C}T_D = [R|\mathbf{t}]$ must satisfy:

$$\left(\mathbf{n}^{C}\right)^T \cdot (R\mathbf{p}^{D} + \mathbf{t}) = 0 \tag{7}$$

Assuming that the rotation is already known, we require at least three line correspondences to find a valid solution for $\mathbf{t}$ (see section IV for special degenerate cases). The MLE of the relative translation is equivalent to the solution of the following non-linear least squares minimization:

$$\arg\min_{\mathbf{t}} \sum_{i=1}^{N_L} \left(\mathbf{n}_i^T \cdot \frac{T\mathbf{p}_i}{\|T\mathbf{p}_i\|}\right)^2 = \arg\min_{\mathbf{t}} \sum_{i=1}^{N_L} \left(\mathbf{n}_i^T \cdot \frac{R\mathbf{p}_i + \mathbf{t}}{\|R\mathbf{p}_i + \mathbf{t}\|}\right)^2 \tag{8}$$

Note that the point $\mathbf{p}_i$, after rotated and translated, must be normalized since, otherwise, points situated farther away from the origin of coordinates would have more influence in the optimization. We also solve the problem with Gauss-Newton, where the Jacobians and residuals are computed as:

$$J_i = \mathbf{n}^T \cdot \frac{I - \frac{T\mathbf{p}}{\|T\mathbf{p}\|}\left(\frac{T\mathbf{p}}{\|T\mathbf{p}\|}\right)^T}{\|T\mathbf{p}\|} \ , \qquad \mathbf{r}_i = \mathbf{n}_i^T \cdot \frac{T\mathbf{p}_i}{\|T\mathbf{p}_i\|} \tag{9}$$

C. Calibration of multiple cameras

Let us assume we have a rig of $N = N^C + N^D$ sensors, with N^C conventional cameras $\{C_1, C_2, .., C_{N^C}\}$ and N^D depth cameras $\{D_1, D_2, .., D_{N^D}\}$, if we perform pair-wise calibration for all the combinations $C_i - D_j$, the global solution will be inconsistent generally. The solution is to perform a complete non-linear optimization with all the parameters. We can set the global reference frame to one of the sensors without loss of generality, for instance C_1. Thus, we need to find the MLE for $(N-1)$ rigid transformations.

The problem is formulated as follows, for the rotation:

$$\arg\min_{\mu_2 .. \mu_N} \sum_{j=1}^{N^C} \sum_{k=1}^{N^D} \sum_{i=1}^{N_L^{jk}} \left((e^{\mu_j} R_j \mathbf{n}_{ji})^T \ \cdot \ e^{\mu_k} R_k \mathbf{v}_{ki}\right)^2 \tag{10}$$

where μ_x is the increment of rotation to R_x for each camera. N_L^{jk} stands for the number of line correspondences between cameras C_j and D_k. The Hessian and gradient, with dimensions $(3\cdot(N-1)\times 3\cdot(N-1))$ and $(3\cdot(N-1)\times 1)$ respectively, are computed following eq. 5, where the Jacobians are given by

$$\begin{aligned} J_i^{(j)} &= ((R_j \mathbf{n}_{ji}) \times (R_k \mathbf{v}_{ki}))^T \\ J_i^{(k)} &= ((R_k \mathbf{v}_{ki}) \times (R_j \mathbf{n}_{ji}))^T \end{aligned} \tag{11}$$

and the super-indexes (j) and (k) represent the three-column block corresponding to the parameters μ_j or μ_k. For the translation, the formulation of the minimization problem is:

$$\arg\min_{\mathbf{t}_2 .. \mathbf{t}_N} \sum_{j=1}^{N^C} \sum_{k=1}^{N^D} \sum_{i=1}^{N_L^{jk}} \left(\mathbf{n}_{ji}^T \cdot \frac{T_j^{-1} T_k \mathbf{p}_{ki}}{\|T_j^{-1} T_k \mathbf{p}_{ki}\|}\right)^2 \tag{12}$$

where the operation $T_j^{-1}T_k\mathbf{p}_{ki}$ transforms the point $\mathbf{p}_{ki}$ to the reference frame of C_j (note that we have employed homogeneous notation for simplicity, and the transformed points are used in its compact form afterwards). Again, instead of using the coordinates of the transformed point, we normalize to have the direction vector of such point. The resulting Jacobians are:

$$J_i^{(j)} = \mathbf{n}_{ij}^T \cdot \frac{-R_j^T - \frac{T_j^{-1}T_k\mathbf{p}_{ki}}{\|T_j^{-1}T_k\mathbf{p}_{ki}\|}\left(\frac{T_j^{-1}T_k\mathbf{p}_{ki}}{\|T_j^{-1}T_k\mathbf{p}_{ki}\|}\right)^T}{\|T_j^{-1}T_k\mathbf{p}_{ki}\|}$$
$$J_i^{(k)} = \mathbf{n}_{ij}^T \cdot \frac{R_j^T - \frac{T_j^{-1}T_k\mathbf{p}_{ki}}{\|T_j^{-1}T_k\mathbf{p}_{ki}\|}\left(\frac{T_j^{-1}T_k\mathbf{p}_{ki}}{\|T_j^{-1}T_k\mathbf{p}_{ki}\|}\right)^T}{\|T_j^{-1}T_k\mathbf{p}_{ki}\|} \quad (13)$$

IV. OBSERVABILITY

In this section we present the analysis of minimal solutions and possible degenerate cases of our calibration problem. For that, we analyse the shape of the Fisher Information Matrix (FIM) for the parameters of the maximum likelihood estimator (MLE) of the calibration presented in the previous section. The FIM coincides with the Hessian of the least squares problem resulting from the MLE, and its inverse is the covariance of the resulting calibration (which corresponds in turn to the Cramér-Rao lower bound when the MLE is given by an unbiased Gaussian distribution [22]). When the FIM is singular, the information provided is not sufficient and the MLE does not exist, therefore, the calibration problem has a solution only when the FIM has full rank.

Let us analyse first the rotation estimation problem. From the error function and its Jacobian (eqs. (3) and (6)), we have that each line correspondence imposes a new constraint between a pair of sensors. Thus, we need at least 3 measurements to compute the relative rotation. These constraints must be linearly independent, which is the case when the direction vectors $\mathbf{v}^D$ of the 3D lines as seen by the depth camera are not all parallel (i.e. two of the three can be parallel). Notice that two parallel lines in 3D are not necessarily parallel in the image, as they may intersect in a vanishing point.

Regarding the estimation of the translation, assuming that the rotation is known, each line correspondence imposes a new constraint between the pair of sensors. In order to get a full rank FIM, the Jacobians of the 3 constraints (9) must be linearly independent. For that, at least 2 normal vectors $\mathbf{n}^C$ must be linearly independent, which is true for any pair of different lines in the projected image (even for parallel lines in the image). This condition is trivially fulfilled for any three constraints for which the rotation's FIM has full rank. Also, the lines used for calibrating the translation cannot intersect all in the same 3D point (e.g. a trihedron), because the translation along the projection ray which contains the optical center of the camera and the line's intersection would not be observable. Hence, the observation of the three non-parallel lines which do not intersect in the same point provides enough information to localize a conventional camera with respect to a depth camera.

The required line correspondences may be observed in several views or in a single one (e.g. the rightmost image pair of Fig. 2). Note that the 3 direction vectors do not have to form an orthogonal base, as it was required in other works [17]. For a rig with N sensors, the number of line correspondences will depend on the type of sensors of the system. Again, by analysing the FIM of both rotation and translation estimation, we can guarantee that the system has a solution if there are at least $3(N-1)$ correspondences which fulfill the rotation and translation conditions between pairs of cameras. Depending on the type of sensors in the system, a solution may exist even with less line correspondences, but such analysis is out of the scope of this paper. Note also that we will be interested to obtain considerably more line correspondences than the minimal set in order to improve the accuracy. Nevertheless, the minimal solution is of interest to remove outlier line correspondences using RANSAC (Section II-C).

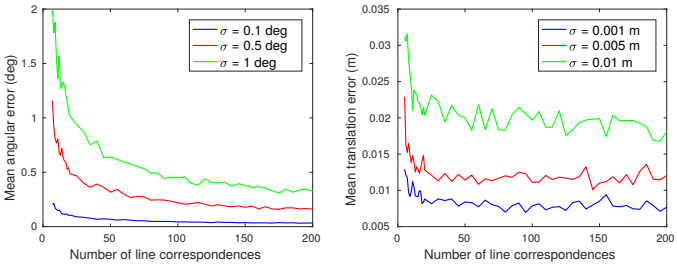

Fig. 4: Simulation results of the calibration accuracy for the rotation (*left*) and translation (*right*), given by the mean error with respect to the ground truth, over the number of line correspondences at three different noise levels.

V. EXPERIMENTAL VALIDATION

We have performed experiments in simulation and with real multi-camera systems. With the former we provide an analysis of performance and robustness to noise with regard to the number of line correspondences. Next, the real case scenarios show the validity and applicability of our method to the real world.

A. Simulation

We present the results of calibration of a pair of depth and RGB cameras with a relative pose T from D to C given by a rotation of $(0.2, -\pi/4 - 0.2, 0.1)$ in Euler angles and a translation of $(-0.06, 0.03, 0.1)$ in meters, for different numbers of line correspondences and different noise levels in their observations. We have also tested different ground truth poses with similar results. For each experiment, we generate randomly $N_L = 5..200$ lines in 3D space and obtain their observation parameters in both cameras. For the analysis of the rotation we add unbiased Gaussian noise to the vectors $\mathbf{v}$ and $\mathbf{n}$ to rotate them slightly. We analyse the calibration accuracy for three noise levels with standard deviation $\sigma = \{0.1, 0.5, 1\}$ degrees. For the analysis of the translation, the point $\mathbf{p}$ is also translated with Gaussian noise $\sigma = \{0.001, 0.005, 0.01\}$ meters. We find these values to be realistic for the case of RGB-D sensors like Asus Xtion Pro Live (XPL).

We show the accuracy of the rotation and the translation separately in Fig. 4, with the accuracy of the calibration

measured by the angular error of our estimated rotation (left), and the translation error measured in meters as the norm of the difference with respect to the ground truth (right). The mean errors decrease asymptotically with the number of line correspondences. This behavior was expected, since having more data should improve the performance when the noise in the measurements is Gaussian. We see how the translation error is more sensitive to noise in comparison with the rotation error. This experiment shows promising results, since the mean error values remain small with a reasonably low number of correspondences.

B. Real case scenarios

Four sensor combinations are tested to show the successful performance of our method under different challenging situations:

1) Fisheye to Depth:
This experiment describes the extrinsic calibration of a regular fisheye camera with FOV over 180° and the depth sensor of an Asus XPL, shown in Fig. 5 (a). This system is useful to combine the large FOV of the fisheye camera with the real scale provided by the depth [3]. We compare our calibration results to the ones obtained using the method from [23], based on a traditional planar checkerboard calibration which we use as ground truth. Note that we do not make use of the color information from the Asus XPL in this experiment in order to be consistent with our comparison with [23].

We have recorded a set of 65 image pairs from our office desktop which contain a good number of lines, thus constituting a good source of information for our method, (see Fig. 3). We selected manually 77 line matches to perform calibration and test its accuracy. From the set of line correspondences, we extract random sets of $N_L = \{10, 20, 30, 40, 50\}$ and measure the average angular and translational errors with respect to the ground truth from 100 calibration runs for each set, using the same metrics of the simulation results. The error values in Table I seem to corroborate the results from simulation, since the error decreases as number of line correspondences rises. The residuals from the optimization are also quite low, as we could expect since no outliers are introduced with the manual selection of correspondences. In Fig. 3 (d) there is the reprojection of the 3D planes and lines on the fisheye image after the calibration.

We also test the performance of the automatic line-matching via RANSAC (Section II-C) compared to manual matchings. For that, we only consider as correspondence candidates those pairs of lines with a relative rotation below 5° (i.e.: $|\mathbf{n} \cdot \mathbf{v}| < cos((90-5) * \pi/180))$ and a relative translation of 10 cm (i.e.: $|\mathbf{n} \cdot \mathbf{p}| < 0.1$), where we have used the identity as the initial estimation of the relative pose, obtaining a total of $N_L = 152$ candidate line matches. Notice that this type of heuristic filtering of candidate correspondences may be applied to any system where we have some rough information about the sensor set-up. We apply RANSAC to remove outliers and perform the calibration, whose results are shown in the lower part of the Table I. We can see that the automatic approach achieves better accuracy in comparison to the case with manually selected correspondences.

TABLE I: Mean rotation and translation errors with respect to the ground truth and residuals in the calibration case of fisheye and depth cameras. Results for both manual and automatic (via RANSAC) matching of lines. The acccuracy is analyzed respectively with the number of lines (N_L) or iterations (N_{iter}).

		Mean error with GT		Mean residual error	
	N_L	Rotation (degrees)	Translation (meters)	Rotation (degrees)	Translation (meters)
MANUAL	**10**	1.9796	0.0483	0.0132	0.0053
	20	1.2006	0.0288	0.0158	0.0050
	30	0.8901	0.0216	0.0169	0.0051
	40	0.7103	0.0179	0.0172	0.0050
	50	0.5741	0.0150	0.0176	0.0051
	N_{iter}	Rotation (degrees)	Translation (meters)	Rotation (degrees)	Translation (meters)
RANSAC	**100**	1.1922	0.0376	0.0009	0.0007
	1000	0.7270	0.0202	0.0009	0.0008
	10000	0.5545	0.0127	0.0010	0.0008

(a) (b)

Fig. 5: *(a)*: Fisheye with RGB-D camera system. *(b)*: Omnidirectional camera rig.

2) Kinect calibration: RGB to Depth:
In this case we show the performance of our calibration system using images from the NYU2 RGB-D public dataset [21]. This dataset is thought to be used in segmentation tasks instead of calibration, so all images are from common indoor scenes (e.g. living rooms, kitchens, offices). A few examples of the line and plane extractions are shown in Fig. 2. We use the provided parameters of extrinsic calibration of the camera Kinect as ground truth to compare our results. The Kinect has a relative pose from the depth camera to the RGB camera close to the identity, with a translation in the X axis of around 2.5 cm. We use the identity as initial rotation matrix, with initial translation equal to zero.

For the experiment we use a recorded sequence in a study room, which was one of the less cluttered (third image in Fig. 2). We use the automatic line-matching with prefiltering of 0.5° for the rotation and 5 cm for the translation, obtaining an initial set of 2229 line-matchings. The RANSAC returns 328 inliers, for which in the optimization we got a rotation error of 0.7578° and a translation error of 1.27 cm (the translation result in X is 3.16 cm). The residuals of the optimization are very low (under 10^{-5} for the rotation and the translation).

We can consider this results satisfactory considering the difficulty of using images from an external dataset which

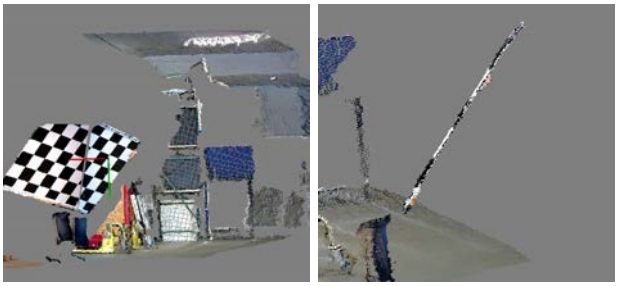

Fig. 6: Visual evaluation of the RGB to depth camera calibration with a checkerboard with the point clouds reprojected to a common reference frame (general and lateral views).

TABLE II: Residual errors in the extrinsic calibration of an RGB camera to a depth camera experiment and the omnidirectional RGB-D camera rig for different number of line correspondences.

	RGB to Depth no overlap		Omnidirectional RGB-D rig	
N_L	Rotation residual (degrees)	Translation residual (meters)	Rotation residual (degrees)	Translation residual (degrees)
10	0.0882	0.0319	0.0378	0.0557
20	0.0849	0.0336	0.0354	0.0405
30	0.0632	0.0271	0.0267	0.0160
40	0.0553	0.0229	0.0205	0.0110
50	0.0489	0.0193	0.0211	0.0193

was thought for another task. In particular, this dataset has very high levels of clutter which introduces many outliers in the set of line correspondences. Besides, many of the useful line correspondences are from far distances, where the plane extraction is less accurate due to the higher levels of noise from the depth camera. Other methods achieve better accuracy (such as [24]), but they need to build a calibration pattern and capture images for this specific purpose.

3) RGB to depth with non-overlapping FOVs:
In this experiment we have used an omnidirectional camera rig formed by 8 Asus Xtion Pro Live cameras arranged in a radial configuration, see Fig. 5 (b). For this particular experiment we only use two adjacent cameras from the rig to calibrate the RGB of one of the cameras to the Depth of the other. Adjacent cameras have a relative rotation of 45°, which we use as initial estimate of relative pose, and a relative translation of less than 10 cm.

The average value of the residuals after the optimization for different numbers of line correspondences (N_L) are shown in Table II (columns 2-3). As in simulation, we observe that a higher number of line correspondences generally improves the results in both rotation and translation. Such improvement stabilizes after a few tens of lines, with a similar trend as the simulation above.

In order to evaluate the accuracy of the system it is desirable to have the ground truth of the calibration of our camera rig. Since this is not available, we employ a big planar checkerboard in a way that each camera observes the portion of the checkerboard not visible by the other camera to evaluate the accuracy of our calibration. First we perform a qualitative evaluation by visualizing the image stitching together with the point cloud reconstructed after calibration from different perspectives (Fig. 6), showing the consistency of the different views. For a quantitative evaluation, we extract the 3D points of the square corners from the checkerboard and place them into the same reference frame given by the calibration (as it is commonly done for intrinsic calibration [24], [23]). Then, we measure the distance of the corners between both cameras to compare them to the real measurements. We compute the average distance between the most distant corners for each row. The average size of the checkerboard squares is of 118.3 mm in the calibrated images, which is similar to the real dimension of 120 mm. We can also estimate the plane equations from each side and compare angles of their normals and the differences in distances to the origin. We obtained an angular difference of 1.7° and a distance difference of 2.4 mm.

4) Omnidirectional rig of 8 RGB-D cameras:
In this experiment we calibrate the relative positions among all cameras from the camera rig shown in Fig. 5 (b). Since the cameras have an approximate vertical FOV of 45°, the eight camera rig achieves an horizontal FOV of 360°. We calibrate this rig following III-C. Table II (columns 4-5) shows the residuals of the optimization according to the number of correspondences extracted between pairs of adjacent cameras (e.g. $N_L = 10$ corresponds to 10 correspondences per pair and 80 for the full rig). A comparison with the two-camera case reveals that the residuals are smaller because of the global optimization.

In this case we cannot obtain any ground truth, so we evaluate the accuracy qualitatively by visual verification. The different RGB images are stitched into a panorama by projecting the individual 3D point clouds transformed to a common reference frame. Ideally, the images should merge seamlessly for a good calibration. In Fig. 7 there are two examples of our image stitching, where it can be observed that the relative positions are well recovered. Compared to [15], which uses the same camera rig, we got better results in the image stitching. The main reason for that is that we use information coming from the color camera and not only depth.

VI. CONCLUSIONS

We propose a novel solution to calibrate different combinations of range and conventional cameras. In contrast to previous alternatives to solve the problem for sensors systems without overlapping FOVs, our solution is considerably easier to apply, it does not have unobservable parameters and it allows to calibrate different sensor combinations with reasonable accuracy. We also present an observability analysis of the problem, providing relevant information regarding the number of observations necessary for our method to perform properly. Our experiments in simulation and real multi-camera systems prove the validity of the method and its applicability to real cases.

REFERENCES

[1] B. Soheilian, O. Tournaire, N. Paparoditis, B. Vallet, and J.-P. Papelard, "Generation of an integrated 3D city model with visual landmarks for autonomous navigation in dense urban areas," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 304–309.

Fig. 7: Panoramic views from two different scenes obtained with the 8 RGB-D camera rig. The views have been obtained by reprojecting the 3D points transformed to a common reference frame with the relative poses obtained following our multi-camera calibration method.

[2] R. Martins, E. Fernandez-Moral, and P. Rives, "Dense accurate urban mapping from spherical RGB-D images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6259–6264.

[3] A. Pérez-Yus, G. López-Nicolás, and J. J. Guerrero, "Peripheral expansion of depth information via layout estimation with fisheye camera," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 396–412.

[4] R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *International Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 251–258.

[5] R. I. Hartley, "Camera calibration using line correspondences," in *Proc. DARPA Image Understanding Workshop*, 1993, pp. 361–366.

[6] G. H. Lee, "A minimal solution for non-perspective pose estimation from line correspondences," in *European Conference on Computer Vision*. Springer, 2016, pp. 170–185.

[7] B. Přibyl, P. Zemčík, and M. Čadík, "Absolute pose estimation from line correspondences using direct linear transformation," *Computer Vision and Image Understanding*, 2017.

[8] A. F. Habib, M. Morgan, and Y.-R. Lee, "Bundle adjustment with self–calibration using straight lines," *The Photogrammetric Record*, vol. 17, no. 100, pp. 635–650, 2002.

[9] Y. Zhang, L. Zhou, H. Liu, and Y. Shang, "A flexible online camera calibration using line segments," *Journal of Sensors*, 2016.

[10] J.-E. Ha, "Extrinsic calibration of a camera and laser range finder using a new calibration structure of a plane with a triangular hole," *International Journal of Control, Automation and Systems*, vol. 10, no. 6, pp. 1240–1244, 2012.

[11] F.-A. Moreno, J. Gonzalez-Jimenez, J.-L. Blanco, and A. Esteban, "An instrumented vehicle for efficient and accurate 3D mapping of roads," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 6, pp. 403–419, 7 2013.

[12] J. Brookshire and S. J. Teller, "Extrinsic calibration from per-sensor egomotion," in *Robotics: Science and Systems*, 2012.

[13] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1793–1800.

[14] S. Schneider, T. Luettel, and H.-J. Wuensche, "Odometry-based online extrinsic sensor calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 1287–1292.

[15] E. Fernandez-Moral, J. González-Jiménez, P. Rives, and V. Arévalo, "Extrinsic calibration of a set of range cameras in 5 seconds without pattern," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 429–435.

[16] E. Fernández-Moral, J. González-Jiménez, and V. Arévalo, "Extrinsic calibration of 2D laser rangefinders from perpendicular plane observations," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1401–1417, 2015.

[17] R. Gomez-Ojeda, J. Briales, E. Fernandez-Moral, and J. Gonzalez-Jimenez, "Extrinsic calibration of a 2D laser-rangefinder and a camera based on scene corners," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3611–3616.

[18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[19] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[20] J. Bermudez-Cameo, G. Lopez-Nicolas, and J. J. Guerrero, "Automatic line extraction in uncalibrated omnidirectional cameras with revolution symmetry," *International Journal of Computer Vision*, vol. 114, no. 1, pp. 16–37, 2015.

[21] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012, pp. 746–760.

[22] J.-A. Fernández-Madrigal and J. L. B. Claraco, *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods*. Information Science Reference, 2013.

[23] A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero, "A novel hybrid camera system with depth and fisheye cameras," in *IAPR International Conference on Pattern Recognition (ICPR)*, 2016, pp. 2789–2794.

[24] D. Herrera C, J. Kannala, and J. Heikkilä, "Joint depth and color camera calibration with distortion correction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2058–2064, 2012.