

# Estimation of Camera Calibration Uncertainty using LIDAR Data

Agustin Ortega, Ricardo Galego, Ricardo Ferreira, Alexandre Bernardino, Jose Gaspar and Juan Andrade-Cetto

**Abstract**— This paper presents an analysis of uncertainty in the calibration of a network of cameras. A global 3D scene model, acquired with a LIDAR scanner, allows calibrating cameras with non overlapping fields of view by means of the *DLT-Lines* algorithm. Once the projection matrix is computed for each camera, error sources are propagated to compute estimates of each camera position uncertainty. We validate the consistency of the uncertainty analysis with Monte-Carlo simulations, and apply the technique in a real camera network. This allows to evaluate the accuracy of *DLT-Lines* in real settings.

## I. INTRODUCTION

Networks of cameras have become ubiquitous for covering large areas. While traditional applications consist mostly of surveillance tasks [1], recently robotic guidance and human-robot collaboration has enlarged the applications portfolio of camera networks. Intrinsic and extrinsic camera calibration is essential to locate detected events into world models [2], so reliable and accurate calibration is usually required in order to successfully accomplish the desired tasks.

Different calibration methodologies behave differently in the presence of noisy data. For instance, [3] shows that the accuracy of calibration depends on the number of parameters, the type of input data and the calibration optimization criteria. In a wide variety of cases, uncertainty propagation can be analyzed using first order techniques [4].

In this work we consider the calibration methodology proposed in [5], that exploits the use of a rich LIDAR range data and images to calibrate a non-overlapped camera network [6], where a coarse calibration is performed manually, followed by fine calibration matching 3D lines to 2D image lines using the *DLT-Lines* (Direct Linear Transformation) algorithm [7]. The method provides estimates of the camera's projection matrix from the knowledge of 3D points and 2D lines. Once this projection matrix is computed, the camera pose is readily obtained by factorization [8]. In this paper

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

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

A. Ortega acknowledges support from a PhD scholarship from the Mexican Council of Science and Technology (CONACYT) and a Research Stay Grant from the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) of The Generalitat of Catalonia (2012 CTP00013). The work is also funded by the Portuguese Science Foundation (FCT) project PEst-OE / EEI / LA0009 / 2011, by the FCT project PTDC / EEACRO / 105413 / 2008 DCCAL, and by the project High Definition Analytics (HDA), QREN - I&D em Co-Promoção 13750; as well as the the Spanish Ministry of Economy and Competitiveness Project PAU+ (DPI-2011-27510) and the EU Project ARCAS (FP7-287617).

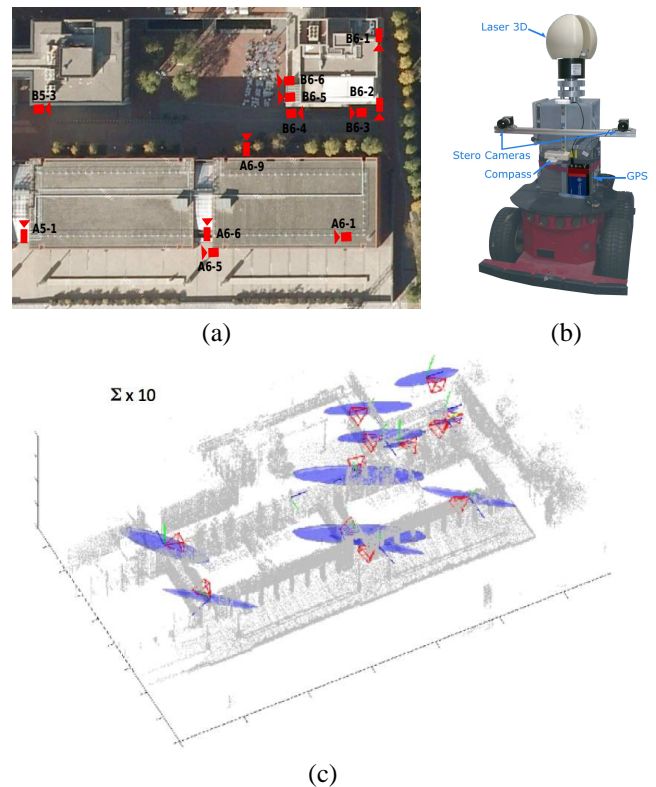


Fig. 1. (a) Barcelona RobotLab camera network used for our calibration experiments. (b) Platform used to acquire LIDAR data. (c) Result of propagating the error from the data to the projection camera matrix and then to the camera pose. The result have been augmented by a factor 10 for better visualization.

we derive expressions for the uncertainty covariance of such camera pose estimates. To do so, we compute a first order linearization of the method around the provided solution and use it to perform first order error propagation [9]. We validate the obtained uncertainty with synthetic data based on Monte-Carlo simulations, and with real data from a non-overlapped camera network. Fig. 1 illustrates real world estimates of the pose covariance for each camera in the network.

The paper is organized as follows, first we present the related work on camera calibration, uncertainty analysis, and the *DLT-Lines* algorithm. Then, we deduce the first order error propagation analysis for the camera's projection matrix and the camera pose. Next we validate these results using synthetic data and compare with Monte-Carlo simulations. We then use the main result to analyze the error in a real world camera network. Finally we argue the conclusions and discuss possible future work.

## II. RELATED WORK

Nowadays, camera networks are ubiquitous. Cameras can be installed outdoors [10], or indoors [11], and can build overlapped [12] or non-overlapped [13] shared views. Overlapping fields of view usually allow matching scene features among cameras and therefore provide extrinsic information for calibration. Conversely, non-overlapped fields of view require finding alternative sources of extrinsic, camera positioning, information. For example, some outdoor/indoors scenes have cars/people traversing the various fields of view. In [10] statistical tracking of traffic, based in image features, allows finding the network topology. In [14], a method that infers the camera network topology is proposed, defining arrival and departure regions. The system requires a priori knowledge of each camera position that is computed using GPS devices. Statistics about camera dependence are reached using vehicle recognition.

Auto-calibration methods can also be used to estimate the camera parameters using image features. In [15], 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. We propose a method that leverages a 3D map and a non-overlapped camera network where the planar assumption is eliminated due to the use of 3D information. another advantage of our method is that it does not require special auto-localization devices to obtain initial position priors for the cameras.

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. This analysis in camera calibration has been discussed in [16], concluding that the error inflicted by the bad estimation of intrinsic parameters affects the extrinsic parameter estimation. Some precedents related to uncertainty analysis are presented in [17], where the calibration is performed using essential matrices. First order error propagation is proposed between the calibration and the motion parameters. In [3], the analysis is made measuring parameter correlation. We propose instead a first order estimation analysis of the full extrinsic calibration.

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

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

where  $\doteq$  denotes equal up to a scale factor,  $P$  is a  $3 \times 4$  projection matrix,  $K$  is a  $3 \times 3$  upper triangular matrix containing the intrinsic parameters of the camera,  $R$  is a  $3 \times 3$  rotation matrix representing the orientation of the camera and  $t$  is a  $3 \times 1$  vector representing the position of the camera [18]. 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  $P$  [8].

### A. Line based Calibration, DLT-Lines

As proposed in [7], we use image lines instead of isolated points. Image processing can be used for fine tuning the location of the lines in the image and therefore automatically improve the calibration data input.

The projection of a 3D line  $L_i$  to the camera image plane can be represented by the cross product of two image points in projective coordinates

$$l_i = m_{1i} \times m_{2i}. \quad (2)$$

Any point  $m_{ki}$  lying in the image line  $l_i$  implies that  $l_i^T m_{ki} = 0$ . Hence, applying the multiplication of  $l_i^T$  to both sides of the perspective camera model, i.e.,  $l_i^T m_{ki} = l_i^T P M_{ki}$ , leads to

$$l_i^T P M_{ki} = 0, \quad (3)$$

where  $M_{ki}$  is a 3D point in projective coordinates lying in  $L_i$ . The properties of Kronecker product [19] allow to obtain a form factorizing the vectorized projection matrix:

$$(M_{ki}^T \otimes l_i^T) p = 0, \quad (4)$$

where  $p$  is the vectorization of the matrix  $P$ . Considering  $N \geq 12$  pairs  $(M_{ki}, l_i)$ , one forms a matrix  $B$ ,  $N \times 12$ , by stacking the  $N$  matrices  $M_{ki}^T \otimes l_i^T$ . Alternatively, given a 3D line  $L_i$  and its projection represented by the image line  $l_i$ , any 3D point lying on  $L_i$  can be paired with 2D line  $l_i$ . On the other hand, any image line  $l_i$  can be paired with any 3D point lying on  $L_i$ , i.e more than one image line can be paired with a 3D point.

A least squares cost function can be expressed as:

$$F(m, M, p) = p^T B^T B p, \quad (5)$$

where  $B$  is  $M_{ki}^T \otimes l_i^T$ . Which allows the formulation of the least squares problem

$$\begin{aligned} \hat{p} &= \arg \min_p F(m, M, p) \\ &\text{s.t. } p^T p = 1 \end{aligned} \quad (6)$$

The solution of the vectorized projection matrix is the right singular vector corresponding to the least singular value of  $B$ .

### B. Factorization of the Projection Matrix

Having estimated the projection matrix,  $P$ , the camera intrinsic and extrinsic parameters can be obtained using QR-decomposition [20]. More precisely, given the sub-matrix  $P_{3 \times 3}$  containing the first three columns of  $P$ , and  $S$  an anti-diagonal matrix

$$S = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad (7)$$

the QR-decomposition allows factorizing  $P_{3 \times 3}^T S = QU$ , where  $Q$  is an orthogonal matrix and  $U$  is an upper triangular matrix. Then, the intrinsic parameters and the rotation

matrices are computed as  $K = -SU^T S$  and  $R = Q^T S$ . Finally, the camera position is obtained with  $t = K^{-1}p_4$ , where  $p_4$  is a  $3 \times 1$  vector containing the fourth column of  $P$ <sup>1</sup>.

#### IV. ERROR PROPAGATION

In this section we derive first order error propagation formulas for the *DLT-Lines* calibration process. In a first step we derive the expression propagating error variance in the calibration data to error variance in the projection matrix entries. In a second step we derive error propagation from the matrix entries variance to the camera projection center.

##### A. DLT-Lines error propagation

As stated, we are interested in the least squares solution to equation (4). This section explores the sensitivity of this equation, and how measurement error propagates to the final estimate.

The optimization problem in Eq. 5 can be seen as a black box which accepts lines ( $l_i$ ) and points ( $M_i$ ) and outputs the least squares estimate of the projection matrix  $P$ :

$$P = f(l_1, \dots, l_N, M_1, \dots, M_N) \quad (8)$$

Assuming  $f$  is differentiable, a first order Covariance propagation is obtained from

$$\Sigma_P = J_f \Sigma_{l_i, M_i} J_f^T. \quad (9)$$

What remains is to compute 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 Karush-Kuhn-Tucker (KKT) conditions:

$$G(l, M, p, \lambda) = \begin{cases} 2B_{(l, M)}^T B_{(l, M)} p + 2\lambda p & = 0, \\ p^T p - 1 & = 0. \end{cases} \quad (10)$$

The implicit function theorem [9] can now be used to provide  $J_f$ :

$$J_f = - \left( \frac{\partial G}{\partial \{p, \lambda\}} \right)^{-1} \left( \frac{\partial G}{\partial \{l_i, M_i\}} \right)^T_{13 \times 6N}.$$

Note that  $J_f$  is a matrix of size  $13 \times 6N$ , while  $\Sigma_P$  is of size  $12 \times 12$ . In the calculation of  $\Sigma_P$  we just use the first twelve rows of  $J_f$  since the last row is the row equivalent to the Lagrange multiplier.

It is worth noting that the covariance of the line measurements is actually obtained from 2 image points. As such, a first order estimate for these covariances is obtained from the linearization of Eq. 2:

$$\Sigma_{l_i} = J_{l_i} \Sigma_{m_1, m_2} J_{l_i}^T \quad (11)$$

where  $J_{l_i}$  is the Jacobian of equation 2.

<sup>1</sup>If the diagonal of  $K$  contains negative values then it is corrected by post multiplying by a diagonal matrix. In Matlab/Octave  $D = \text{diag}(\text{sign}(\text{diag}(K)))$ ;  $K = K * D$ ;  $R = D * R$ ;  $t = D * t$ . In addition, since  $\pm P$  are both solutions of Eq.4, the factorization of  $P$  may imply  $\det(R) = -1$ . If  $\det(R) = -1$  then the factorization of  $P$  is repeated using  $-P$ .

##### B. Error propagation to the camera center

Propagating uncertainty from the estimated  $P$  into its decomposition  $K[R \ t]$ , or the separate intrinsic parameters (scaling, shear or principal point), involves computing the Jacobian of the transformation starting from the QR-decomposition and ending with the calibration parameter extracted from  $P$  (see Sec.III-B). Despite straightforward, this process involves lengthy expressions, where in many cases one wants just an indicator of the precision of the calibration. The uncertainty of the camera projection center is one indicator of the precision of the calibration that can be computed using concise expressions.

Denoting the projection center referred to the world coordinate system as  $M_c = [X_c \ Y_c \ Z_c]^T$ , one has that  $M_c$  projects to a point at infinity,  $m_c = [0 \ 0 \ 0]^T = \vec{0}$ , i.e.

$$\vec{0} \doteq P [M_c^T \ 1]^T. \quad (12)$$

Representing the projection matrix as a collection of columns,  $P = [p_1 \ p_2 \ p_3 \ p_4]$ , the projection center can be computed as the solution of the linear system of three equation in three unknowns,  $[p_1 \ p_2 \ p_3]M_c = -p_4$ . Considering that one wants to apply operations (derivatives) on the transformation from  $P$  to  $M_c$ , it is convenient to derive a closed form for  $M_c$ . Using the Cramer's rule to solve the system, one has

$$\begin{cases} X_c = \det([-p_4, \ p_2, \ p_3])/W \\ Y_c = \det([p_1, \ -p_4, \ p_3])/W \\ Z_c = \det([p_1, \ p_2, \ -p_4])/W \end{cases} \quad (13)$$

where  $W = \det([p_1 \ p_2 \ p_3])$ . It is interesting to note that the choice of the world coordinate system is key to obtain concise expressions. Instead of  $t = K^{-1}p_4$ , one has  $M_c = P_{3 \times 3}^{-1}p_4$  which differs from  $t$  just by a rotation<sup>2</sup> and avoids decomposing  $P = K[R \ t]$ .

Finally, one propagates the error variance of the projection matrix,  $\Sigma_P$ , to the error variance of the projection center,  $\Sigma_c$ , as an explicit formula

$$\Sigma_c = J_c \Sigma_P J_c^T, \quad (14)$$

where  $J_c$  denotes the Jacobian of the vector  $M_c$  with respect to the  $P$  matrix elements.

#### V. EXPERIMENTS

In order to validate the proposed uncertainty analysis we conduct some experiments in a synthetic environment for which one has available precise and accurate ground truth. In addition, we apply the proposed uncertainty analysis to a real setup based on an outdoor scene encompassing a number of buildings, the Barcelona RobotLab, which has been reconstructed in 3D using LIDAR data, thus providing directly the required 3D information for the *DLT-Lines* calibration methodology.

##### A. Synthetic experiments

In this section the variance of the entries of the projection matrix,  $\Sigma_P$ , predicted using the proposed uncertainty analysis

<sup>2</sup>Noting that  $P_{3 \times 3} = KR$  and  $p_4 = Kt$  one has  $M_c = (KR)^{-1}Kt = R^{-1}t$ .

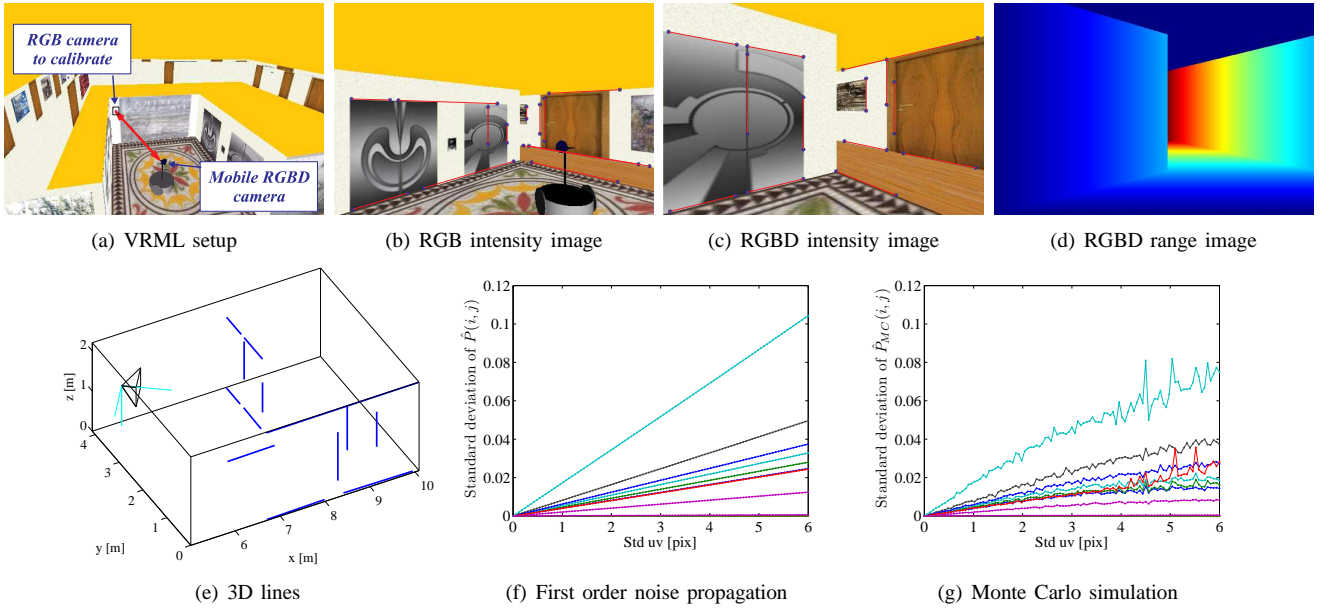


Fig. 2. Analysis of camera calibration uncertainty. (a) VRML setup. (b) RGB image. (c) RGBD intensity image. (d) RGBD range image. Each line defined in the RGBD image corresponds to a line in the RGB image, and leads to a 3D line in the world/RGBD coordinate system. (e) 3D lines form the required input 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.

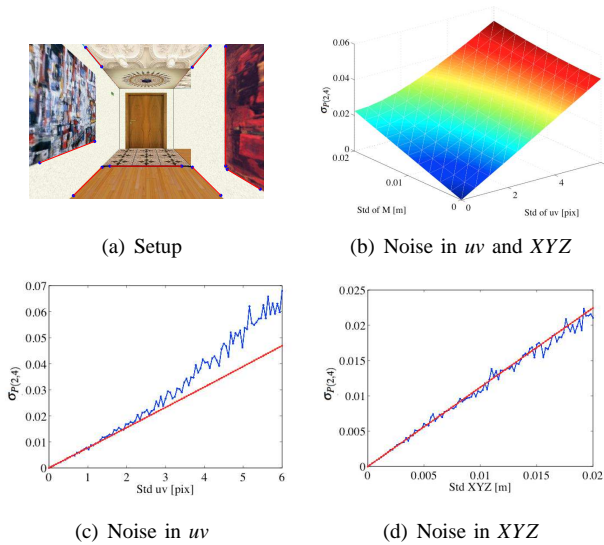


Fig. 3. Single camera setup. (a) 3D information is known for the image lines shown. (b) Standard deviation of calibration noise of  $P_{11}$ , entry (2,4) of  $P$ , given combined noise in  $uv$  and  $XYZ$ . (c) Standard deviation of calibration noise propagated from noise in  $uv$  ranging from 0 till 6 pixel, red line; compared against Monte Carlo uncertainty analysis, blue line. (d) Standard deviation of calibration noise propagated from noise in  $XYZ$  ranging 0 to 0.02 meters, red line; compared against Monte Carlo uncertainty, blue line.

(Eq.9) is compared with a Monte Carlo based uncertainty analysis. The experiments are done using synthetic data 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 two different scenarios in order to show that the proposed uncertainty propagation methodology effectively

takes into account the structure of the scenario.

The first synthetic setup is formed by two cameras, namely a mobile color-depth (RGBD) camera which collects 3D data and a fixed RGB camera. See Fig 2(a). Figure 2(b) shows a synthetic image simulated for the RGB camera, while Figs. 2(d) and (e) show synthetic intensity and range images simulated for 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 ( $\sigma_M = 0$ )<sup>3</sup>. 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.  $\Sigma_{P(i,j)}$  for  $i = 1..4$  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. 2(f). As expected some entries of  $P$  are more robust to noise than others. Figure 2(g) 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 when for values of  $\sigma_m$  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 second setup is based on a single RGBD camera.

<sup>3</sup>To improve readability, variance is written using upper case,  $\Sigma$ , and standard deviation is written using lower case,  $\sigma$ .

The setup can be seen in Fig. 3(a), which corresponds to a typical 'L' shaped corridor. Camera calibration ground truth is known and is used to assess the validity of the noise propagation estimation method. Figure 3(b) 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.

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

Plot (c) 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  $(l_i, M_i)$ . Nonlinear effects have less influence for variations of range as shown in plot (d).

### B. Experiments in real scenarios

For our experiments we use a mobile Pioneer 3AT robot equipped a 3D range sensing device consisting of a Hokuyo UTM-30LX laser mounted on a slip-ring. The laser resolution is set to 0.5 degrees in azimuth with 360 degree omnidirectional field of view, and 0.5 degrees resolution in elevation for a range of 270 degrees. Each point cloud contains 194,580 range measurements of up to 30 meters with noise varying from 30mm for distances closer to 10m, and up to 50mm for objects as far as 30m. Our robot includes also two Flea2 cameras [21]. The dataset used for the experiments is the Barcelona RobotLab dataset [6].

The computation of straight lines from the point cloud relies on identifying and intersecting planes. The method to segment planar regions is motivated by Felzenszwalb's algorithm to 2D image segmentation [22], and extended to deal with non-uniformly sampled 3D range data [23]. They image and 3d lines are associated manually.

We analyze pose error in different indoor and outdoor scenarios. The first experiment involves an indoor scenario. In this scenario, a camera calibration pattern is used to compare classical image based calibration with our 3D-2D line-based calibration scheme. We show the result in Fig. 4.

The next experiment consists in the estimation of pose uncertainty for the calibration of the camera network explained in [5]. Calibration results for a subset of 3 are shown in Fig. 1. The top frames contain the camera images, the reprojected 3D point cloud, and the 3D lines used for calibration. The bottom frame shows again the 3D lines used for calibration and the estimated 3D poses and their associated position covariances. These covariances are magnified 10 times to ease visualization.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented a methodology for estimating the calibration uncertainty of a networked camera. Calibration data has been assumed to combine 2D (image) and

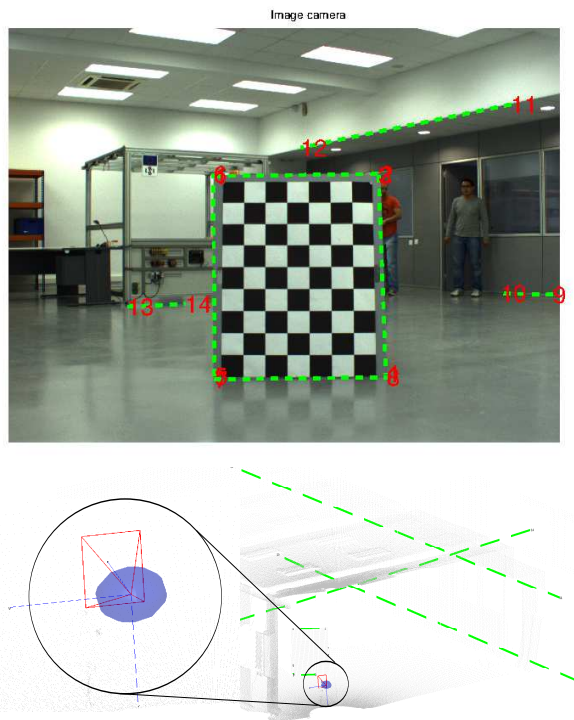


Fig. 4. Indoor experiment. We apply the calibration method using a point cloud and one of our robot cameras. The top frame shows the lines used, and the inset in the bottom frame shows the computed covariance poses.

3D data. In this work 3D data is acquired using a TOF sensor, such as LIDAR, or simply using color-depth (RGBD) cameras. More conventional scenarios can also use floor plans, where ground lines (intersections of walls and floor) combined with vertical lines (intersections of nonparallel walls) form the necessary 3D data.

Our methodology starts by estimating the camera projection matrix using 3D and image lines using the *DLT-Lines* algorithm. In other words, the projection matrix is estimated by minimizing a quadratic cost-function (MLS). The fact that calibration corresponds to the minimization of a cost function allows propagating the covariance of the calibration data to estimate the covariance of the projection matrix. Given the estimate of the covariance of projection matrix we can finally propagate the uncertainty to the camera location.

We have demonstrated that our uncertainty analysis is consistent by testing it with Monte-Carlo simulations. Besides we studied our analysis approach in a real non-overlapped camera network from the Barcelona Lab dataset.

As future work we propose exploring novel data acquisition and filtering modalities which allow characterizing lesser conservatively the error caused by extracting line features from the LIDAR range and camera sensor.

## REFERENCES

- [1] J.-H. Kim and B.-K. Koo, "Convenient calibration method for unsynchronized multi-camera networks using a small reference object," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Oct. 2012, pp. 438–444.

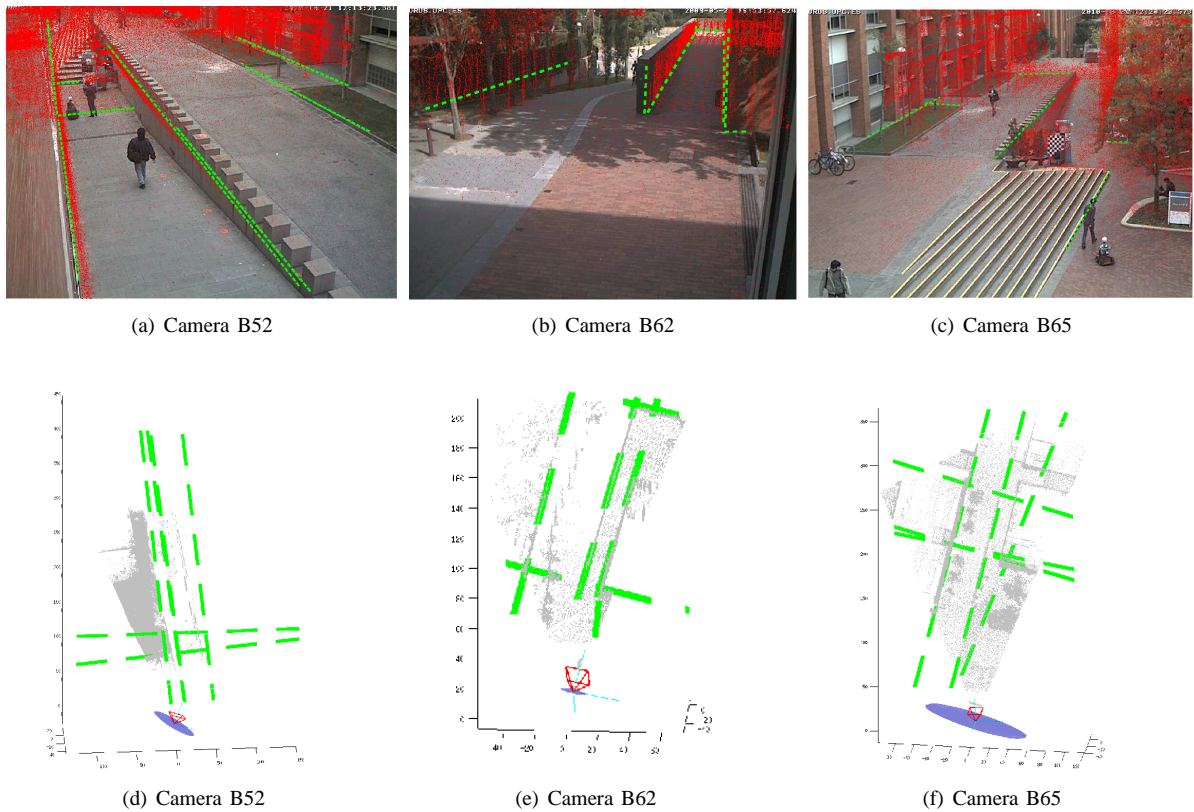


Fig. 5. Barcelona RobotLab. Top frames: Reprojected 3D point clouds and 2d image lines used for calibration for the cameras with labels B52, B62 and B65. Bottom frames: 3D lines and estimated robot locations and robot location covariances. Covariance hyper-ellipsoids have been magnified 10 times to ease visualization.

- [2] A. Sanfeliu and J. Andrade-Cetto, "Ubiquitous networking robotics in urban settings," in *Proc. IEEE/RSJ IROS Workshop Network Robot Syst.*, Beijing, Oct. 2006, pp. 14–18.
- [3] S.-W. Shih, Y.-P. Hung, and W.-S. Lin, "Accuracy analysis on the estimation of camera parameters for active vision systems," in *Proc. 13th IAPR Int. Conf. Pattern Recog.*, vol. 1, Vienna, Aug. 1996, pp. 930–935.
- [4] R. Haralick, "Propagating covariance in computer vision," in *Proc. 13th IAPR Int. Conf. Pattern Recog.*, vol. 1, Vienna, Aug. 1996, pp. 493–498.
- [5] 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. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Saint Louis, Oct. 2009, pp. 303–308.
- [6] E. Teniente, M. Morta, A. Ortega, E. Trulls, and J. Andrade-Cetto, "Barcelona Robot Lab data set," [online] <http://www.iri.upc.edu/research/webprojects/pau/datasets/BRL/php/dataset.php>, 2011.
- [7] M. Silva, R. Ferreira, and J. Gaspar, "Camera calibration using a color-depth camera: Points and lines based dlt including radial distortion," in *IROS Workshop Color-Depth Camera Fusion in Robotics*, 2012.
- [8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press, 2000.
- [9] J. Clarke, "Modelling uncertainty: A primer," University of Oxford. Dept. Engineering science, Tech. Rep. 2161/98, 1998.
- [10] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proc. 18th IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Washington, Jul. 2004, pp. 205–210.
- [11] R. Kumar, A. Ilie, J. Frahm, and M. Pollefeys, "Simple calibration of non-overlapping cameras with a mirror," in *Proc. 22nd IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, Jun. 2008.
- [12] F. Bajramovic and J. Denzler, "Global uncertainty-based selection of relative poses for multi camera calibration," in *Proc. British Mach. Vis. Conf.*, 2008.
- [13] P. Lébraly, E. Royer, O. Ait-Aider, and M. Dhome, "Calibration of non-overlapping cameras application to vision-based robotics," in *Proc. British Mach. Vis. Conf.*, 2010.
- [14] K. Tieu, G. Dalley, W. Eric, and L. Grimson, "Inference of nonoverlapping camera network topology by measuring statistical dependence," in *Proc. IEEE Int. Conf. Comput. Vis.*, Beijing, Oct. 2005.
- [15] I. Junejo, C. Xiaochun, and H. Foroosh, "Autoconfiguration of a dynamic nonoverlapping camera network," *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 4, pp. 803–816, Aug. 2007.
- [16] R. Kumar and A. R. Hanson, "Sensitivity of the pose refinement problem to accurate estimation of camera parameters," in *Proc. IEEE Int. Conf. Comput. Vis.*, Osaka, Japan, Dec. 2000.
- [17] T. Svoboda and P. Sturm, "A badly calibrated camera in ego-motion estimation, propagation of uncertainty," in *Proc. Int. Conf. Comput. Analysis Images and Patterns*, ser. Lect. Notes Comput. Sci., vol. 1296, 1997, pp. 183–190.
- [18] O. Faugeras, *Three-Dimensional Computer Vision - A Geometric Viewpoint. Artificial intelligence*. M.I.T. Press, 1993.
- [19] H. Lütkepohl, *Handbook of Matrices*. Wiley and Sons, 1996.
- [20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge: Cambridge University Press, 2004.
- [21] A. Ortega and J. Andrade-Cetto, "Segmentation of dynamic objects from laser data," in *Proc. Eur. Conf. Mobile Robot.*, Orebro, Sep. 2011, pp. 115–121.
- [22] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [23] A. Ortega, I. Haddad, and J. Andrade-Cetto, "Graph-based segmentation of range data with applications to 3D urban mapping," in *Proc. Eur. Conf. Mobile Robot.*, Dubrovnik, Sep. 2009, pp. 193–198.
- [24] *Proc. 13th IAPR Int. Conf. Pattern Recog.*, Vienna, Aug. 1996.