ORIGINAL RESEARCH PAPER

Uncertainty analysis for optimum plane extraction from noisy 3D range-sensor point-clouds

**Kaustubh Pathak · Narunas Vaskevicius ·
Andreas Birk**

Abstract We utilize a more accurate range noise model for 3D sensors to derive from scratch the expressions for the optimum plane fitting a set of noisy points and for the combined covariance matrix of the plane's parameters, viz. its normal and its distance to the origin. The range error model used by us is a quadratic function of the true range and also the incidence angle. Closed-form expressions for the Cramér–Rao uncertainty bound are derived and utilized for analyzing four methods of covariance computation: exact maximum likelihood, renormalization, approximate least-squares, and eigenvector perturbation. The effect of the simplifying assumptions inherent in these methods are compared with respect to accuracy, speed, and ease of interpretation of terms. The approximate least-squares covariance matrix is shown to possess a number of desirable properties, e.g., the optimal solution forms its null-space and its components are functions of easily understood terms like the planar-patch's weighted centroid and scatter. It is also fast to compute and accurate enough in practice. Its experimental application to real-time range-image registration and plane fusion is shown by using a commercially available 3D range sensor.

K. Pathak (✉) · N. Vaskevicius · A. Birk
Department of Computer Science, Jacobs University Bremen,
28759 Bremen, Germany
e-mail: k.pathak@jacobs-university.de; kaust@ieee.org

N. Vaskevicius
e-mail: n.vaskevicius@jacobs-university.de

A. Birk
e-mail: a.birk@jacobs-university.de

1 Introduction

Spatial cognition is a key capability for autonomous intelligent robots to be useful in daily life scenarios. The availability of fast 3D sensors is pushing the envelope of faster algorithms for 3D perception and mapping. The algorithms developed for 2D mapping using Laser Range Finders (LRF) do not always scale well to 3D, e.g. the generalization of traditional occupancy-grid 2D maps to three dimensions is hindered by the disproportionate increase in storage and computation requirements.

The input data for a typical 3D mapping algorithm is a "point-cloud" obtained from a range-sensor like a rotating LRF [16] or a time-of-flight sensor like the Swiss-ranger [2] and the PMD [12]. Mapping methods based directly on point-clouds are usually off-line [7]. There have been recent attempts to use feature-based 3D mapping, where the features are usually planar patches [6,18,19] extracted from the point-clouds by a variation of the region-growing algorithm [13]. This results in a compact semantic representation of the environment which is faster to compute and requires much less storage than a voxel-grid. 3D Plane Simultaneous Localization and Mapping (SLAM) was recently introduced in [9], which utilizes large planar patches to generate 3D maps. Some example maps are shown in Fig. 1.

3D Plane SLAM consists of the following steps, which are repeated at each sampling step:

1. Acquisition of 3D range scans.
2. Extraction of planes including uncertainties.
3. Registration of scans based on plane sets (plane-matching).
4. Embedding of the registrations in a pose-graph.
5. Loop detection and relaxation of pose-graph.

Fig. 1 a–e show data being
collected in a robot test arena in
the form of a multi-story
high-bay-rack. **f–g** show the
resulting plane-based 3D map.
The robot's location is shown in
the map by its to-scale 3D avatar
with a red chassis. The color of
the planar surfaces was chosen
by the plane-matching algorithm
to show surface-correspondence
across scans. **c** Robot collecting
data, **f** The *top view* of the 3D
Plane Registration map. The
robot starts at top left and goes
clockwise around, **g** the tilted
top view showing the windows at
the far end which are visible in **a**
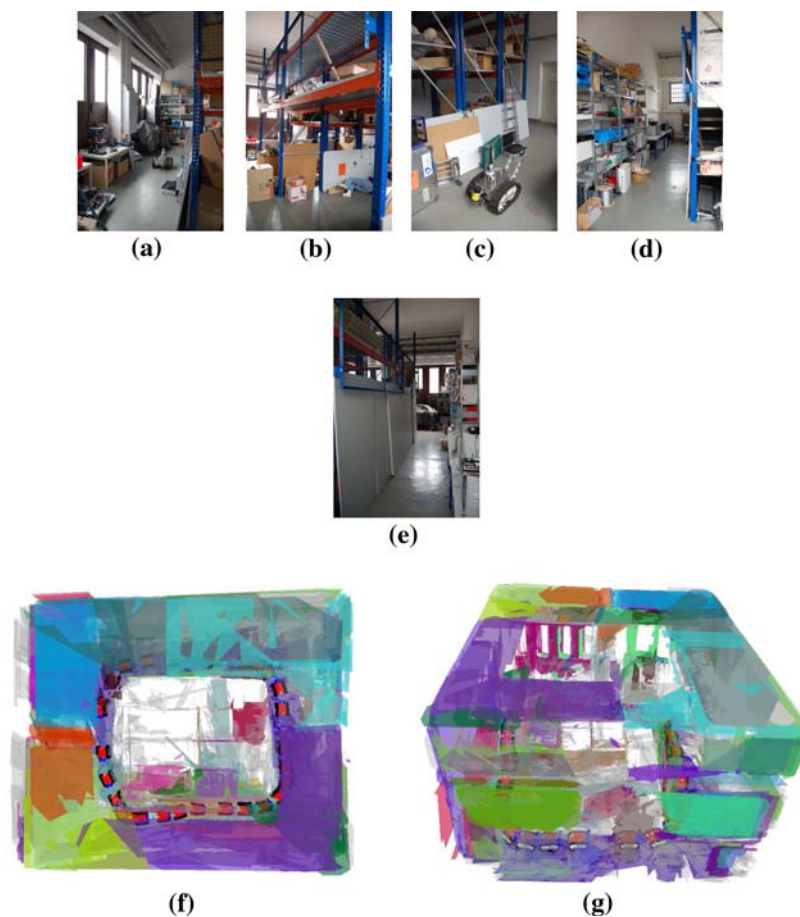
In this article, we concentrate on a discussion of the second step, i.e., an efficient extraction of the planes and the related uncertainties, which forms the very basis for 3D Plane SLAM.

Given a set of noisy points known or hypothesized to lie on a plane, the "optimum" plane can be extracted from them using methods surveyed in [17]. The main methods are that of least-squares [17,19] and renormalization [4,5]. The latter uses a more detailed error-model for the 3D points which results in better estimations with respect to the assumed model. This is achieved, however, at the cost of a computationally expensive iterative method which is not suited for a real-time application. The main drawback of this method is the use of an outdated range resolution model which does not incorporate the effect of incidence-angle, and wherein the range standard-deviation is a linear rather than a quadratic function of the true range. Another aspect which needs reexamination is the computation of covariance matrix of the estimated plane parameters from the point of view of a trade-off between accuracy and computation-time.

This article is organized as follows: in Sect. 2, we formulate the maximum likelihood and least-squares plane extraction problems and derive useful properties of the exact

solution based on an accurate range noise model. In Sect. 3, we derive an approximate solution suitable for real-time applications along with analytical expressions for the its covariance. In Sect. 4 another analytical solution for the covariance is derived using the eigenvector perturbation method. These two methods are numerically compared to the method of renormalization in Sect. 5. Finally, an application of the plane covariance matrices for plane fusion using real-world data is presented in Sect. 6 followed by conclusions in Sect. 7.

2 Problem formulation

In this section, we briefly rederive the optimum plane parameters from scratch because this serves to highlight the various assumptions made along the way and the effects of using a new range sensor noise model. The equation of a plane is $\hat{\mathbf{n}} \cdot \mathbf{r} = d$, where $\hat{\mathbf{n}}$ is the plane's unit normal and d the distance to the origin. Assume that the sensor returned a point-cloud $\mathbf{r}_j = \rho_j \hat{\mathbf{m}}_j$, $j = 1 \ldots N$, where, $\hat{\mathbf{m}}_j$ are the measurement directions for the sensor, usually accurately known, and ρ_j are the respective ranges which are noisy.

2.1 Maximum likelihood and least squares plane estimation

We make the assumption of radial Gaussian noise, i.e. $\rho_i \sim \mathcal{N}\{\bar{\rho}_i, \sigma^2\{\bar{\rho}_i, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i\}\}$, where $\{\cdot\}$ encloses function arguments, and $\bar{\rho}_i = d/(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i)$ is the *true* range of ith measurement. This implies a covariance matrix of $\mathbf{r}_j$ of the form

$$\mathbf{C}_{\mathbf{r},j} = \sigma^2\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\}\, \hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T, \tag{1}$$

where the range standard-deviation $\sigma\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\}$ is explicitly a (usually quadratic) function of $\bar{\rho}_j$ and has been found to be inversely proportional to $\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j$, i.e. the cosine of the incidence-angle [1].

The likelihood of plane-parameters $(\hat{\mathbf{n}}, d)$ given range sample ρ_i along measurement direction $\hat{\mathbf{m}}_i$ is

$$p(\rho_i \mid \hat{\mathbf{n}}, d, \hat{\mathbf{m}}_i)$$
$$= \frac{1}{\sqrt{2\pi}\,\sigma\{\bar{\rho}_i, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i\}} \exp\left\{-\frac{1}{2}\frac{(\rho_i - d/(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i))^2}{\sigma^2\{\bar{\rho}_i, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i\}}\right\} \tag{2}$$

Then, considering a sequence of samples $i = 1 \ldots N$, ignoring constant terms, and defining a *binary switch* $\beta = 0/1$, the log-likelihood function to be maximized is as follows

$$\max_{\hat{\mathbf{n}},d} \mathcal{L}_{\text{GMLP}} = -\beta \sum_{i=1}^{N} \log \sigma\{\bar{\rho}_i, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i\}$$
$$-\frac{1}{2}\sum_{j=1}^{N} \frac{(\rho_j(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j) - d)^2}{(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2 \sigma^2\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\}}, \tag{3}$$

where GMLP stands for *General Maximum Likelihood Problem* for $\beta = 1$. The *General Least Squares Problem* (GLSP) can be obtained from GMLP by simply setting $\beta = 0$. For GLSP, the above can be equivalently rewritten as

$$\min_{\hat{\mathbf{n}},d} \sum_{j=1}^{N} \frac{(\rho_j(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j) - d)^2}{(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2 \sigma^2\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\}}. \tag{4}$$

We note that the above expression is exactly the same as the weighted least-squares cost function in [5, Eq. (8)], except that it is rewritten in our notation on substitution of (1). No particular functional dependence has been enforced on $\sigma^2\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\}$ yet.

The optimization (3) cannot be handled analytically, especially as σ is a function of $\bar{\rho}_j = d/(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i)$ and $\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_i$. One approach is to handle the problem numerically using iterations—as is done in the renormalization method of [5]. It has the advantage of having higher accuracy. Additionally, it assumes a *linearly varying* standard-deviation $\sigma\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\} = \kappa_1 \bar{\rho}_j$, *independent* of the incidence-angle, and simultaneously estimates the value of κ_1 during the iterations. However, there are two main reservations regarding this approach.

Firstly, an iterative method is usually not suitable for real-time applications. Secondly, and more importantly, as shown

in [1,8,14], the standard-deviation of commonly available 3D sensors like the Swiss-ranger and Laser-range-finder (LRF) is more accurately modeled by a function of the form

$$\sigma\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\} = \frac{\hat{\sigma}\{\bar{\rho}_j\}}{|\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j|}, \text{ where,}$$

$$\hat{\sigma}\{\bar{\rho}_j\} \triangleq \kappa\bar{\rho}_j^2 \equiv \frac{\kappa d^2}{|\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j|^2} \tag{5}$$

where $\hat{\mathbf{n}}$ is the local normal to the surface the point $\mathbf{r}_j$ lies on. The coefficient $\kappa > 0$ can be estimated by doing initial calibration experiments with the sensor. In fact, on substitution of the LHS of (5) in (3), we get,

$$\max_{\hat{\mathbf{n}},d} \mathcal{L}_{\text{EMLP}} = -\beta \sum_{j=1}^{N} \log \frac{\hat{\sigma}\{\bar{\rho}_j\}}{|\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j|} - \frac{1}{2}\sum_{j=1}^{N}\frac{(\hat{\mathbf{n}} \cdot \mathbf{r}_j - d)^2}{\hat{\sigma}^2\{\bar{\rho}_j\}}, \tag{6}$$

where EMLP stands for *Exact Maximum Likelihood Problem*. If $\beta = 0$, we get the *Exact Least Squares Problem* (ELSP). The ELSP expression is both more accurate and simpler-looking than the cost function considered in [5], which is equivalent to the cost in (4) with $\sigma\{\bar{\rho}_j, \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j\} \equiv \kappa_1 \bar{\rho}_j$. On expanding (6) using RHS of (5), and ignoring constants, we get

$$\max_{\hat{\mathbf{n}},d} \mathcal{L}_{\text{EMLP}} = -2\beta N \log d + 3\beta \sum_{j=1}^{N} \log |\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j|$$
$$-\frac{1}{2}\sum_{j=1}^{N}(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4 \frac{(\hat{\mathbf{n}} \cdot \mathbf{r}_j - d)^2}{\kappa^2 d^4}. \tag{7}$$

As $\hat{\mathbf{m}}_j$ are the measuring directions of the sensor, from arguments of geometric consistency of points on a plane observed along these directions we take $|\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j| \equiv \hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j$ in what follows.

2.2 Properties of the solution

We denote the *ground-truth* values of $\hat{\mathbf{n}}, d$ by $\bar{\mathbf{n}}, \bar{d}$. To save space, we define the error in the plane-equation for some estimate $\hat{\mathbf{n}}, d$ at sample $\mathbf{r}_j$ as

$$\epsilon_j \triangleq \hat{\mathbf{n}} \cdot \mathbf{r}_j - d. \tag{8}$$

Although only a numerical solution of (7) is possible, we can show that unless $\beta = 0$, the solution is biased. To this end, we define a Lagrangian $\mathcal{L} = L_{\text{EMLP}} - \frac{1}{2}\lambda(\hat{\mathbf{n}}^T \hat{\mathbf{n}} - 1)$, where λ is the usual Lagrange multiplier. Since at the optimum estimate $\hat{\mathbf{n}}_\star, d_\star$, the gradient vanishes,

$$\frac{\partial \mathcal{L}}{\partial d} = -\frac{2\beta N}{d} + \frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4 \left(\epsilon_j + \frac{2\epsilon_j^2}{d} \right), \quad (9a)$$

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{n}}} = -\lambda \hat{\mathbf{n}} + 3 \sum_{j=1}^{N} \beta \frac{\hat{\mathbf{m}}_j}{\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j}$$
$$- \frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^3 \left(2\epsilon_j^2 \hat{\mathbf{m}}_j + \epsilon_j (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j) \mathbf{r}_j \right).$$
$$(9b)$$

At $\hat{\mathbf{n}}_\star, d_\star$, the above expressions equate to zero and

$$\lambda_\star = 3\beta N - \frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4 \left(2\epsilon_j^2 + \epsilon_j (\hat{\mathbf{n}} \cdot \mathbf{r}_j) \right). \quad (9c)$$

The Hessian of (7) at $\lambda = \lambda_\star, \hat{\mathbf{n}} = \hat{\mathbf{n}}_\star, d = d_\star$ can be computed as follows after some algebra

$$\frac{\partial^2 \mathcal{L}}{\partial d^2} = -\frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4$$
$$\times \left(1 + \frac{8\epsilon_j}{d} + \frac{10\epsilon_j^2}{d^2} \right) + \frac{2\beta N}{d^2} \quad (10a)$$

$$\frac{\partial^2 \mathcal{L}}{\partial d \, \partial \hat{\mathbf{n}}} = \frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} 4\epsilon_j (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^3 \left(1 + \frac{2\epsilon_j}{d} \right) \hat{\mathbf{m}}_j$$
$$+ \frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4 \left(1 + \frac{4\epsilon_j}{d} \right) \mathbf{r}_j, \quad (10b)$$

$$\frac{\partial^2 \mathcal{L}}{\partial \hat{\mathbf{n}}^2} = -\lambda_\star \mathbf{I} - \frac{1}{\kappa^2 d^4} \sum_{j=1}^{N} (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4 \mathbf{r}_j \mathbf{r}_j^T$$
$$- \frac{4}{\kappa^2 d^4} \sum_{j=1}^{N} \epsilon_j (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^3 \left(\hat{\mathbf{m}}_j \mathbf{r}_j^T + \mathbf{r}_j \hat{\mathbf{m}}_j^T \right)$$
$$- \frac{6}{\kappa^2 d^4} \sum_{j=1}^{N} \epsilon_j^2 (\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2 \hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T$$
$$- 3\beta \sum_{j=1}^{N} \frac{\hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T}{(\hat{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2} \quad (10c)$$

$$\mathbf{H}_{\text{EMLP}} \triangleq \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \hat{\mathbf{n}}^2} & \left(\frac{\partial^2 \mathcal{L}}{\partial d \, \partial \hat{\mathbf{n}}} \right) \\ \left(\frac{\partial^2 \mathcal{L}}{\partial d \, \partial \hat{\mathbf{n}}} \right)^T & \frac{\partial^2 \mathcal{L}}{\partial d^2} \end{bmatrix}. \quad (10d)$$

Using results from [4,15], we can find the 4×4 plane-parameter covariance matrix

$$\mathbf{C}_{\text{EMLP}} \equiv -\mathbf{H}_{\text{EMLP}}^+, \quad (11)$$

i.e., the negative of the *Moore–Penrose generalized inverse* of the Hessian. The generalized inverse is necessary because the Hessian may be singular.

2.2.1 The Cramér–Rao Lower Bound on Covariance

The Cramér–Rao (CR) lower bound, denoted $\bar{\mathbf{C}}_{\text{EMLP}}$, is applicable only to unbiased estimators [4], for which the optimum solution is the ground-truth $\bar{\mathbf{n}}, \bar{d}$. It can be computed as the negative inverse of the *expected-value* of the Hessian derived in (10). Since the ground-truth is not known in practice, this bound cannot be found. However, *in simulations* this ideal value can be used to compare the estimated covariances found using various methods. We note that its derivation in [10] was not complete: we now proceed to derive the correct expression for it.

We would first need to compute the following expected values. Using Eqs. (1), (5), and (8), we can compute the following expectations

$$E(\epsilon_j) = 0, \quad (12a)$$

$$E(\epsilon_j^2) = \bar{\mathbf{n}}^T \mathbf{C}_{\mathbf{r},j} \bar{\mathbf{n}} \equiv \hat{\sigma}_j^2 \equiv \frac{\kappa^2 \bar{d}^4}{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4}, \quad (12b)$$

$$E(\mathbf{r}_j) \equiv \bar{\mathbf{r}}_j = \frac{\bar{d} \, \hat{\mathbf{m}}_j}{\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j}, \quad (12c)$$

$$E(\mathbf{r}_j \mathbf{r}_j^T) = E(\mathbf{C}_{\mathbf{r},j}) + \bar{\mathbf{r}}_j \bar{\mathbf{r}}_j^T$$
$$\equiv \left(\frac{\kappa^2 \bar{d}^4}{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^6} + \frac{\bar{d}^2}{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2} \right) \hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T, \quad (12d)$$

$$E(\epsilon_j \mathbf{r}_j) = E((\mathbf{r}_j \mathbf{r}_j^T)\hat{\mathbf{n}} - d \, \mathbf{r}_j) \equiv \frac{\kappa^2 \bar{d}^4}{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^5} \hat{\mathbf{m}}_j. \quad (12e)$$

We now substitute the above expectations in (10), and after some tedious algebra, we obtain

$$E(\lambda_\star) = 3\beta N - 3N, \quad (13a)$$

$$E\left(\frac{\partial^2 \mathcal{L}}{\partial d^2} \right) = -\frac{1}{\kappa^2 \bar{d}^4} \sum_{j=1}^{N} (\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^4 - \frac{N(10 - 2\beta)}{\bar{d}^2}, \quad (13b)$$

$$E\left(\frac{\partial^2 \mathcal{L}}{\partial d \, \partial \hat{\mathbf{n}}} \right) = \sum_{j=1}^{N} \frac{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^3}{\kappa^2 \bar{d}^3} \hat{\mathbf{m}}_j + \sum_{j=1}^{N} \frac{12}{\bar{d}(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)} \hat{\mathbf{m}}_j, \quad (13c)$$

$$E\left(\frac{\partial^2 \mathcal{L}}{\partial \hat{\mathbf{n}}^2} \right) = -\sum_{j=1}^{N} \frac{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2}{\kappa^2 \bar{d}^2} \hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T$$
$$- \sum_{j=1}^{N} \frac{15 + 3\beta}{(\bar{\mathbf{n}} \cdot \hat{\mathbf{m}}_j)^2} \hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T, \quad (13d)$$

$$E(\mathbf{H}_{\text{EMLP}}) \triangleq \begin{bmatrix} E(\frac{\partial^2 \mathcal{L}}{\partial \hat{\mathbf{n}}^2}) & E\left(\frac{\partial^2 \mathcal{L}}{\partial d \, \partial \hat{\mathbf{n}}} \right) \\ E\left(\frac{\partial^2 \mathcal{L}}{\partial d \, \partial \hat{\mathbf{n}}} \right)^T & E(\frac{\partial^2 \mathcal{L}}{\partial d^2}) \end{bmatrix}. \quad (13e)$$

The first terms on the right-hand side of Eqs. (13b), (13c), and (13d) are the ones which were given in [10]. They need to be corrected by the corresponding second terms. We can

write these terms as

$$E(\mathbf{H}_{\text{EMLP}}) = \bar{\mathbf{H}}_1 + \bar{\mathbf{H}}_2, \tag{14}$$

where $\bar{\mathbf{H}}_1$ contains the first terms and $\bar{\mathbf{H}}_2$, the second. As noted in [10], $\bar{\mathbf{H}}_1$ has the interesting property that $[\bar{\mathbf{n}}^T, \bar{d}]^T$ lies in its null-space.

Finally, the Cramér–Rao covariance bound is

$$\bar{\mathbf{C}}_{\text{EMLP}} \equiv -E(\mathbf{H}_{\text{EMLP}})^+. \tag{15}$$

Note that $-E(\mathbf{H}_{\text{EMLP}})$ is also called the Fisher Information matrix $\mathbf{F}$.

Another way of finding this bound is using the terms in the Gaussian of Eq. (2). We define the parameter vector $\boldsymbol{\theta} \triangleq \begin{pmatrix} \hat{\mathbf{n}} \\ d \end{pmatrix}$, and compute

$$-\mathbf{F} \equiv E(\mathbf{H}_{\text{EMLP}})$$
$$= -\sum_{k=1}^{N} \left[\frac{\partial \bar{\rho}_k}{\partial \boldsymbol{\theta}} \sigma_k^{-2} \left(\frac{\partial \bar{\rho}_k}{\partial \boldsymbol{\theta}} \right)^T + \frac{1}{2} \sigma_k^{-2} \frac{\partial \sigma_k^2}{\partial \boldsymbol{\theta}} \sigma_k^{-2} \left(\frac{\partial \sigma_k^2}{\partial \boldsymbol{\theta}} \right)^T \right]$$
$$\tag{16}$$

We have verified that this leads to the same results as in Eqs. (13). Furthermore, the first summation in (16) turns out to be $\bar{\mathbf{H}}_1$ and the second $\bar{\mathbf{H}}_2$. For the noise-model considered, the contribution of $\bar{\mathbf{H}}_2$ is numerically found to be significantly less than that of $\bar{\mathbf{H}}_1$, and thus it continues to be true that eigenvector corresponding to the smallest eigenvalue of $E(\mathbf{H}_{\text{EMLP}})$ is almost parallel to $[\bar{\mathbf{n}}^T, \bar{d}]^T$.

3 Approximate solution and computation of covariance matrix

The objective of this work is to explore if, under reasonable simplifying assumptions, one can obtain a fast estimation of the plane parameters and their covariance matrix without iterations. When extracting planar patches from a range-image using region-growing, the plane-fitting algorithm is called many times per range-image, and an iterative algorithm is unsuitable for this. The solution of ELSP cannot be done analytically– hence, in the sequel we explore ways of solving it in an approximate but fast manner.

To proceed with the approximate solution, we make the assumption that $\hat{\sigma}(\bar{\rho}_j) \approx \hat{\sigma}(\rho_j)$. We define $\hat{\sigma}_i \triangleq \hat{\sigma}(\rho_j)$ and note that $\hat{\sigma}_i$ is now no longer a function of $\bar{\rho}_j$ and hence of $\hat{\mathbf{n}}$ and d. Using Eq. (6) and setting $\beta = 0$, we get

$$\max_{\hat{\mathbf{n}}, d} \mathcal{L}_{\text{ALSP}} = -\frac{1}{2} \sum_{j=1}^{N} \frac{(\hat{\mathbf{n}} \cdot \mathbf{r}_j - d)^2}{\hat{\sigma}_j^2}, \tag{17}$$

where ALSP stands for *Approximate Least Square Problem*.

3.1 Solution of ALSP and covariance estimation

ALSP is a well studied problem, see for example [18]. Although the solution is well-known, the covariance matrix computation is done in various ways in the literature. The method presented in [18] is particularly redundant as it solves the optimization problem twice: It first finds the optimum plane in the $(\hat{\mathbf{n}}, d)$ space, then rotates the plane such that the z axis is parallel to $\hat{\mathbf{n}}$, and subsequently translates the plane such that $d = 0$. Finally, another best-fit is done using the plane-equation of the form $z = \beta_0 + \beta_1 x + \beta_2 y$. The problem is essentially *re-solved* for this form by least-squares to obtain the covariance matrix for $\beta_0, \beta_1, \beta_2$. The procedure is justified by saying that it provides an analytical way of obtaining the covariance and it avoids the matrix singularity in $(\hat{\mathbf{n}}, d)$ space.

We show that the covariance matrix can be obtained analytically *also* in the $(\hat{\mathbf{n}}, d)$ space. We see the fact that the null-space of the covariance matrix consists of the optimal solution as an asset which can be used to efficiently fuse planes. In fact, we show that this covariance matrix also contains other summarized information about the point cloud from which it was extracted, like its center and scatter.

As before, we have a constrained optimization problem which can be solved using Lagrange multipliers. Defining the Lagrangian

$$\mathcal{L} = \mathcal{L}_{\text{ALSP}} + \lambda(\hat{\mathbf{n}}^T \hat{\mathbf{n}} - 1) \tag{18}$$

Setting $\partial \mathcal{L} / \partial d = 0$ gives the well-known solution

$$d_\star = \hat{\mathbf{n}}_\star \cdot \mathbf{r}_G, \quad \mathbf{r}_G \triangleq \left(\sum_{j=1}^{N} \hat{\sigma}_j^{-2} \right)^{-1} \sum_{j=1}^{N} \left(\hat{\sigma}_j^{-2} \mathbf{r}_j \right), \tag{19}$$

where $\mathbf{r}_G$ is the weighted center of the point-cloud. Substituting the above in (17) gives

$$\hat{\mathbf{n}}_\star = \arg \min_{\|\hat{\mathbf{n}}\|=1} \hat{\mathbf{n}}^T \left(\sum_{j=1}^{N} \hat{\sigma}_j^{-2} (\mathbf{r}_j - \mathbf{r}_G)(\mathbf{r}_j - \mathbf{r}_G)^T \right) \hat{\mathbf{n}}$$
$$\triangleq \arg \min_{\|\hat{\mathbf{n}}\|=1} \hat{\mathbf{n}}^T \mathbf{S} \hat{\mathbf{n}}, \tag{20}$$

where $\mathbf{S}$ is the positive semi-definite weighted scatter matrix. The solution $\hat{\mathbf{n}}_\star$ is then the eigenvector corresponding to the minimum eigenvalue of $\mathbf{S}$.

As done in Sect. 2.2, the covariance matrix of the optimal solution can simply be obtained by finding the Hessian of the Lagrangian (18) at $\hat{\mathbf{n}}_\star, d_\star, \lambda_\star$. Its expressions are

$$\lambda_\star = \frac{1}{2} \hat{\mathbf{n}}_\star^T \mathbf{S} \hat{\mathbf{n}}_\star, \tag{21a}$$

$$\frac{\partial^2 \mathcal{L}}{\partial d^2} = -\sum_{j=1}^{N} \frac{1}{\hat{\sigma}_j^2} \triangleq -\mu, \tag{21b}$$

$$\frac{\partial^2 \mathcal{L}}{\partial d\,\partial \hat{\mathbf{n}}} = \sum_{j=1}^{N} \frac{\mathbf{r}_j}{\hat{\sigma}_j^2} \equiv \mu\,\mathbf{r}_G, \tag{21c}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \hat{\mathbf{n}}^2} = -\sum_{j=1}^{N} \frac{\mathbf{r}_j \mathbf{r}_j^T}{\hat{\sigma}_j^2} + 2\lambda_\star \mathbf{I},$$

$$\equiv -\mathbf{S} - \mu\,\mathbf{r}_G \mathbf{r}_G^T + \left(\hat{\mathbf{n}}_\star^T \mathbf{S} \hat{\mathbf{n}}_\star\right) \mathbf{I}. \tag{21d}$$

Using the above, the ALSP Hessian is computed by

$$\mathbf{H}_{\text{ALSP}} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial \hat{\mathbf{n}}^2} & \frac{\partial^2 \mathcal{L}}{\partial d\,\partial \hat{\mathbf{n}}} \\ (\frac{\partial^2 \mathcal{L}}{\partial d\,\partial \hat{\mathbf{n}}})^T & \frac{\partial^2 \mathcal{L}}{\partial d^2} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{nn} & \mathbf{H}_{\hat{\mathbf{n}}d} \\ \mathbf{H}_{\hat{\mathbf{n}}d}^T & \mathbf{H}_{dd} \end{bmatrix}_{\text{ALSP}} \tag{21e}$$

Finally, the covariance matrix is

$$\mathbf{C}_{\text{ALSP}} = -\mathbf{H}_{\text{ALSP}}^+. \tag{21f}$$

Using Eqs. (21), it is easily verified that

$$\mathbf{H}_{\text{ALSP}} \begin{bmatrix} \hat{\mathbf{n}}_\star \\ d_\star \end{bmatrix} = \mathbf{0}, \quad \Rightarrow \mathbf{C}_{\text{ALSP}} \begin{bmatrix} \hat{\mathbf{n}}_\star \\ d_\star \end{bmatrix} = \mathbf{0}. \tag{22}$$

It is a remarkable property of the ALSP solution, i.e. it lies precisely in the null-space of its covariance matrix.

4 Method of eigenvector perturbation for covariance computation

Another method to compute covariances was derived in a different context in [20]. We would like to confirm the computation of Hessian based covariance with their method. For this, however, we need to reformulate the problem similar to [5]. We define a unit vector in parameter space

$$\hat{\boldsymbol{v}} = \begin{pmatrix} v_1 & v_2 & v_3 & v_4 \end{pmatrix}^T = \frac{1}{\sqrt{\|\hat{\mathbf{n}}\|^2 + d^2}} \begin{pmatrix} \hat{\mathbf{n}} \\ d \end{pmatrix}. \tag{23}$$

Then the equation of the plane is

$$\hat{\boldsymbol{v}} \cdot \begin{pmatrix} \mathbf{r} \\ -1 \end{pmatrix} \triangleq \hat{\boldsymbol{v}} \cdot \boldsymbol{\varrho} = 0, \quad \mathbf{C}_{\boldsymbol{\varrho},i} = \begin{pmatrix} \mathbf{C}_{\mathbf{r},i} & \mathbf{0} \\ \mathbf{0} & 0 \end{pmatrix}, \tag{24}$$

where (1) has been used. Then, the ALSP (17) can be reformulated to the *Eigenvector Perturbation Problem* (EVPP)

$$\min_{\|\hat{\boldsymbol{v}}\|=1} \mathcal{L}_{\text{EVPP}} = \sum_{j=1}^{N} \hat{\boldsymbol{v}}^T \frac{\boldsymbol{\varrho}_j \boldsymbol{\varrho}_j^T}{\hat{\sigma}_j^2} \hat{\boldsymbol{v}} \triangleq \hat{\boldsymbol{v}}^T \mathbf{M} \hat{\boldsymbol{v}}. \tag{25}$$

The solution $\hat{\boldsymbol{v}}_\star$ is simply the eigenvector corresponding to the minimum eigenvalue of the symmetric positive semi-definite matrix $\mathbf{M}$. The uncertainty in $\hat{\boldsymbol{v}}_\star$ can be computed using the method of [20]. The specialization of this method to the present problem is, however, derived by us, and we have not found it elsewhere.

The symmetric positive semi-definite matrix $\mathbf{M}$ has the decomposition $\mathbf{M} = \mathbf{V}\Lambda\mathbf{V}^T$, where the orthonormal matrix

$\mathbf{V}$ contains the eigenvectors corresponding to the eigenvalues λ_i in the diagonal matrix Λ. The eigenvectors are assumed to be sorted in ascending order, so that λ_1 is the eigenvalue corresponding to $\hat{\boldsymbol{v}}_\star$. If $\mathbf{M}$ is perturbed by a small disturbance Δ_M, $\hat{\boldsymbol{v}}_\star$ is perturbed by

$$\boldsymbol{\delta}_{\hat{\boldsymbol{v}}_\star} = \mathbf{V}\Delta_1 \mathbf{V}^T \Delta_M \hat{\boldsymbol{v}}_\star, \quad \text{where,} \tag{26}$$

$$\Delta_1 = \text{diag}\left\{0, (\lambda_1 - \lambda_2)^{-1}, (\lambda_1 - \lambda_3)^{-1}, (\lambda_1 - \lambda_4)^{-1}\right\}.$$

$$\therefore \mathbf{C}_{\hat{\boldsymbol{v}}_\star} = \mathcal{E}[\boldsymbol{\delta}_{\hat{\boldsymbol{v}}_\star} \boldsymbol{\delta}_{\hat{\boldsymbol{v}}_\star}^T] = \mathbf{V}\Delta_1 \mathbf{V}^T \mathcal{E}[\Delta_M \hat{\boldsymbol{v}}_\star \hat{\boldsymbol{v}}_\star^T \Delta_M^T] \mathbf{V}\Delta_1 \mathbf{V}^T$$
$$= \mathbf{V}\Delta_1 \mathbf{V}^T \mathbf{C}_{\Delta_M \hat{\boldsymbol{v}}_\star} \mathbf{V}\Delta_1 \mathbf{V}^T, \tag{27}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. Let the perturbation $\mathbf{r}_i = \bar{\mathbf{r}}_i + \delta_{\mathbf{r}_i}$, $\delta_{\mathbf{r}_i} \equiv \begin{pmatrix} \delta_{x_i} & \delta_{y_i} & \delta_{z_i} \end{pmatrix}^T$. One can easily derive that

$$\Delta_M = \sum_{i=1}^{N} \frac{1}{\hat{\sigma}_i^2}$$
$$\times \begin{bmatrix} 2\delta_{x_i} x_i & x_i \delta_{y_i} + y_i \delta_{x_i} & x_i \delta_{z_i} + z_i \delta_{x_i} & -\delta_{x_i} \\ x_i \delta_{y_i} + y_i \delta_{x_i} & 2\delta_{y_i} y_i & y_i \delta_{z_i} + z_i \delta_{y_i} & -\delta_{y_i} \\ x_i \delta_{z_i} + z_i \delta_{x_i} & y_i \delta_{z_i} + z_i \delta_{y_i} & 2\delta_{z_i} z_i & -\delta_{z_i} \\ -\delta_{x_i} & -\delta_{y_i} & -\delta_{z_i} & 0 \end{bmatrix}. \tag{28}$$

Using this one can derive a matrix $\mathbf{N}_i\{\hat{\boldsymbol{v}}_\star\}$, a function of $\hat{\boldsymbol{v}}_\star$, such that,

$$\Delta_M \hat{\boldsymbol{v}}_\star = \sum_{i=1}^{N} \frac{1}{\hat{\sigma}_i^2} \mathbf{N}_i\{\hat{\boldsymbol{v}}_\star\} \delta_{\mathbf{r}_i}, \quad \text{which gives the result}$$

$$\mathbf{C}_{\Delta_M \hat{\boldsymbol{v}}_\star} = \sum_{i=1}^{N} \frac{1}{\hat{\sigma}_i^2} \mathbf{N}_i\{\hat{\boldsymbol{v}}_\star\} \mathbf{C}_{\mathbf{r},i} \mathbf{N}_i^T\{\hat{\boldsymbol{v}}_\star\}. \tag{29}$$

On substitution in (27), we get the required covariance $\mathbf{C}_{\hat{\boldsymbol{v}}_\star}$.

We can now recover our optimum solution in the $(\hat{\mathbf{n}}, d)$ space by applying the following reverse transform to $\hat{\boldsymbol{v}}_\star$

$$\begin{pmatrix} \hat{\mathbf{n}} \\ d \end{pmatrix} = \mathbf{g}\{\hat{\boldsymbol{v}}\} \triangleq \frac{1}{\sqrt{1 - \hat{v}_4^2}} \hat{\boldsymbol{v}}. \tag{30}$$

Transforming $\mathbf{C}_{\hat{\boldsymbol{v}}_\star}$ to the $(\hat{\mathbf{n}}, d)$ space by the usual Jacobian method presents an interesting paradox, which has not been addressed by previous researchers [5], who have used a similar transform.

4.1 A paradox in covariance matrix transformation

Given a symmetric positive semi-definite matrix $\mathbf{C}$, its eigenvectors are the unit-directions $\hat{\boldsymbol{v}}$ in the space of definition, where the quadratic product $\hat{\boldsymbol{v}}^T \mathbf{C} \hat{\boldsymbol{v}}$ achieves a minimum or maximum [3]. Since a *unit-vector* is identical in the $(\hat{\mathbf{n}}, d)$ space as well as in the space of $\hat{\boldsymbol{v}}$ of (23), we expect that $\mathbf{C}_{\hat{\boldsymbol{v}}_\star}$ will have the same eigenvectors (though not eigenvalues) as

the covariance $\mathbf{C}_{\hat{\mathbf{n}},d}$ in $(\hat{\mathbf{n}}, d)$ space. We have verified numerically that this is true (details in Sect. 5, Figs. 2b, c, 3b, c). However, these two matrices are also supposed to be related by

$$\mathbf{C}_{\hat{\mathbf{n}},d} = \mathbf{J}\mathbf{C}_{\hat{\mathbf{v}}_\star}\mathbf{J}^T. \qquad (31)$$

An obvious way to compute the Jacobian $\mathbf{J}$ is

$$\mathbf{J} \triangleq \frac{\partial \mathbf{g}}{\partial \hat{\mathbf{v}}} = (1 - \hat{\mathbf{v}}_4^2)^{-1/2} \begin{pmatrix} 1 & 0 & 0 & \frac{\hat{\mathbf{v}}_1\hat{\mathbf{v}}_4}{1-\hat{\mathbf{v}}_4^2} \\ 0 & 1 & 0 & \frac{\hat{\mathbf{v}}_2\hat{\mathbf{v}}_4}{1-\hat{\mathbf{v}}_4^2} \\ 0 & 0 & 1 & \frac{\hat{\mathbf{v}}_3\hat{\mathbf{v}}_4}{1-\hat{\mathbf{v}}_4^2} \\ 0 & 0 & 0 & \frac{1}{1-\hat{\mathbf{v}}_4^2} \end{pmatrix}. \qquad (32)$$

The same Jacobian was used in [5], but with the bottom right corner element negated as they had defined $\hat{\mathbf{v}} = (\|\hat{\mathbf{n}}\|^2 + d^2)^{-1/2}\begin{pmatrix}\hat{\mathbf{n}} \\ -d\end{pmatrix}$ instead of as in (23). However, unless $\hat{\mathbf{v}}_4 = 0$, this Jacobian rotates the eigenvectors of $\mathbf{C}_{\hat{\mathbf{v}}_\star}$, which *contradicts our earlier observation that the eigenvectors should remain unrotated*. In fact, this Jacobian is incorrect because it assumes that the components of $\hat{\mathbf{v}}$ can be independently perturbed, whereas this is not true as $\|\hat{\mathbf{v}}\| = 1$. The correct Jacobian is

$$\mathbf{J} \triangleq (1 - \hat{\mathbf{v}}_4^2)^{-1/2}\mathbf{I}_4, \qquad (33)$$

which merely scales and which will be used in this paper along with (31) to transform the covariance to the $(\hat{\mathbf{n}}, d)$ space.

5 Comparison of methods

We will compare four methods: renormalization [5], ALSP (Sect. 3), EVPP (Sect. 4), and the EMLP (Sect. 2.2 with $\beta = 1$). The comparison is done in simulation because the "ground-truth" is known accurately in simulation and hence the lower-bound covariance matrix $\bar{\mathbf{C}}_{\text{EMLP}}$ of Sect. 2.2.1 can be computed.

Let the eigen-pair of $\bar{\mathbf{C}}_{\text{EMLP}}$ in which the eigenvector is almost parallel to the solution be denoted as $\bar{\lambda}_0, \bar{\tau}_0$, and the rest of the eigen-pairs are sorted according to the eigenvalues and and denoted as $\bar{\lambda}_j, \bar{\tau}_j, \ j = 1\ldots 3$.

Given the covariance matrix estimation of method i as $\mathbf{C}_i$ with sorted eigenvalues and their eigenvectors as $\lambda_{i,j}, \hat{\tau}_{i,j}$, $j = 0\ldots 3$, and i = renormalization, ALSP, EVPP, and EMLP. We define the following three error metrics which are plotted for all the methods in Figs. 2, 3, and 4:


Fig. 2 Squared Mahalanobis distance ε_1 from the ground-truth based on the Cramér–Rao bound on the covariance: **a** renormalization, **b** ALSP, **c** Eigenvector perturbation, **d** Exact Maximum Likelihood (EMLP)
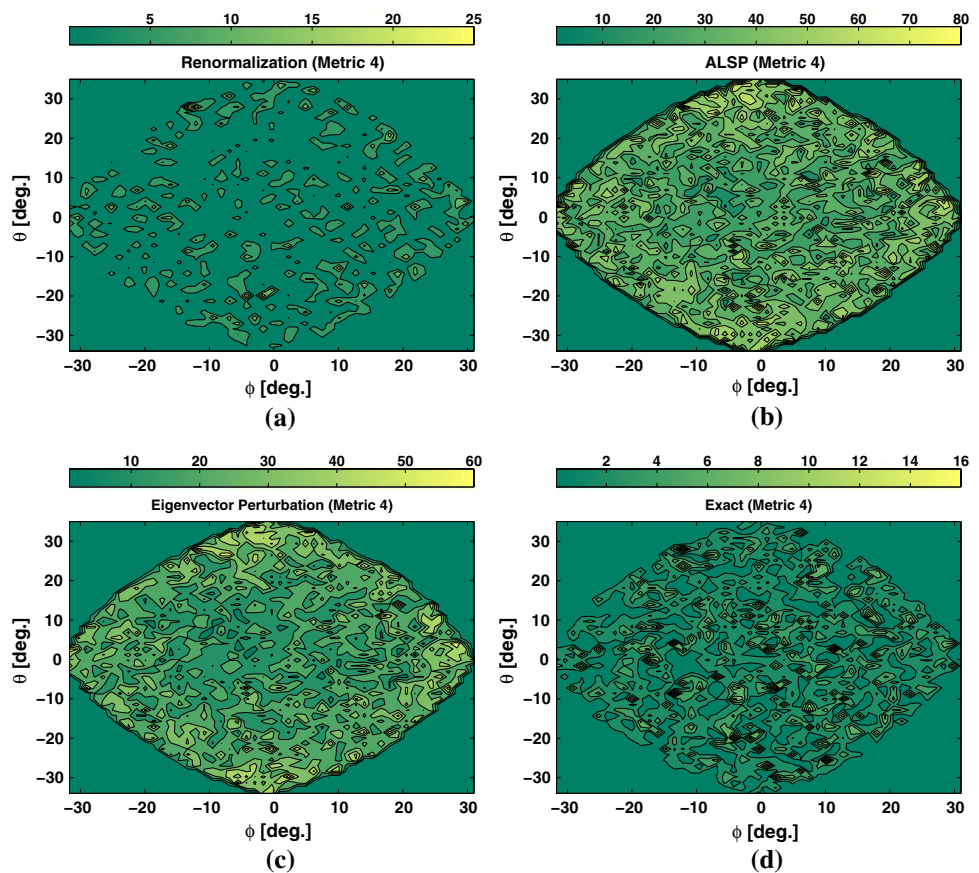
Fig. 3 Principal Uncertainty Directions Error ε_2. The obviously strong effect of the different range noise model assumed by renormalization on the principal uncertainty directions: **a** renormalization, **b** ALSP, **c** Eigenvector perturbation, **d** Exact Maximum Likelihood (EMLP)
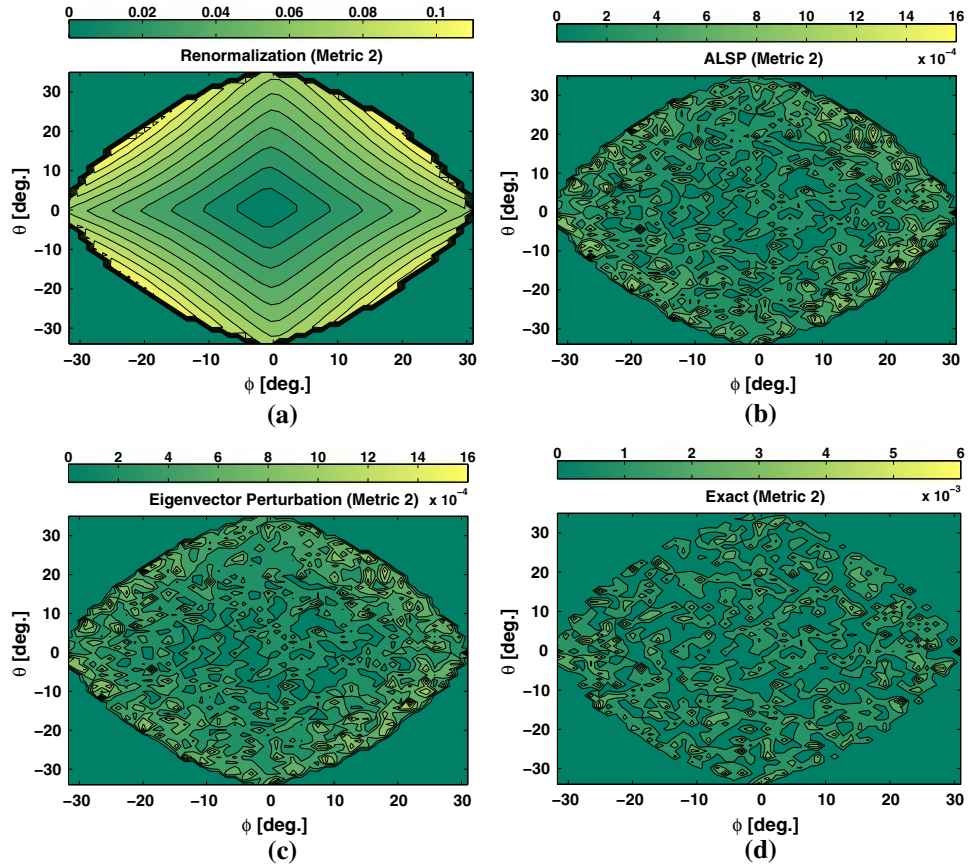
Fig. 4 Relative Total Uncertainty ε_3. In **a**, note that lowest value is below the theoretically least value of unity because the algorithm assumes a different sensor error model. Note also from the color-scale that the values are much worse than the other two methods: **a** renormalization, **b** ALSP, **c** Eigenvector perturbation, **d** Exact Maximum Likelihood (EMLP)
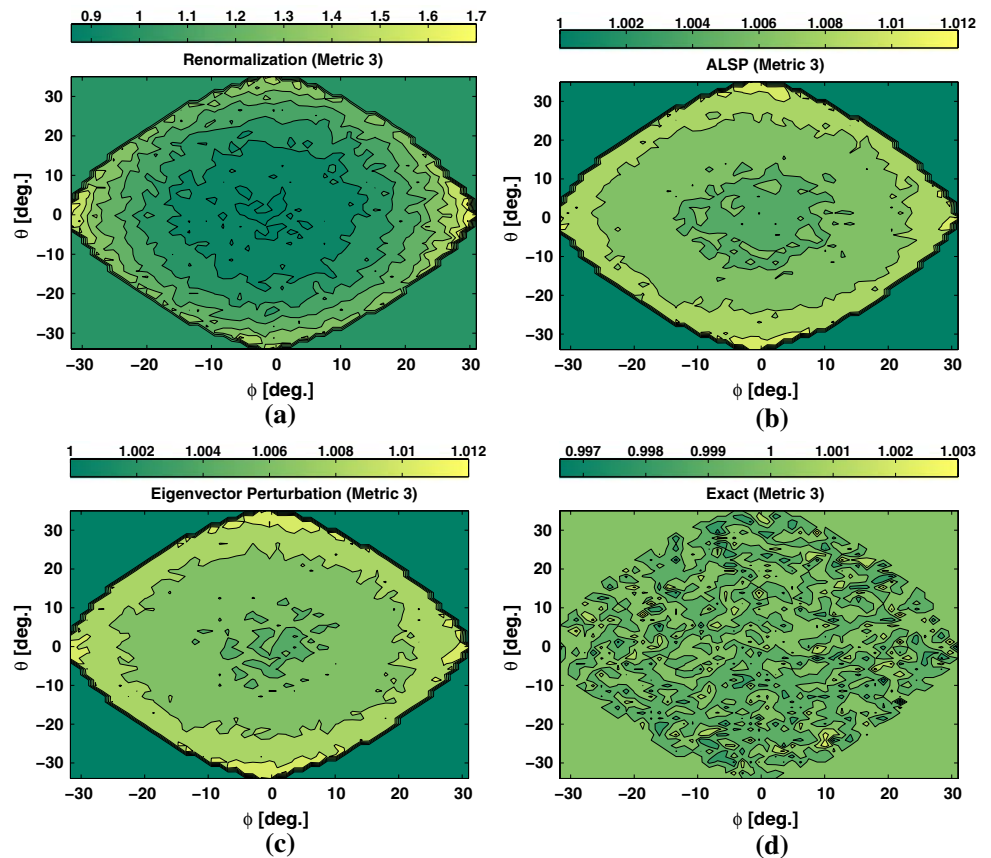
Square of Mahalanobis Distance ε_1 This metric, ideally zero, is defined as follows for the ith method

$$\varepsilon_1 \triangleq (\hat{\tau}_{i,0} - \bar{\tau}_0)^T \bar{\mathbf{C}}_{\text{EMLP}}^+ (\hat{\tau}_{i,0} - \bar{\tau}_0). \tag{34}$$

Principal Uncertainty Directions Error ε_2 Weighted sum of angular errors in the principal uncertainty directions of the ith method, which should be ideally zero.

$$\varepsilon_2 \triangleq \sum_{j=1}^{3} w_j \cos^{-1}(\bar{\tau}_j \cdot \hat{\tau}_{i,j}), \quad w_j \triangleq \frac{\bar{\lambda}_j}{\bar{\lambda}_1 + \bar{\lambda}_2 + \bar{\lambda}_3}. \tag{35}$$

Relative Total Uncertainty ε_3 Lower bounded by unity.

$$\varepsilon_3 \triangleq (\lambda_{i,1}\lambda_{i,2}\lambda_{i,3})/(\bar{\lambda}_1\bar{\lambda}_2\bar{\lambda}_3). \tag{36}$$

We simulated the commercial 3D sensor Swiss-ranger and took its experimentally determined parameter $\kappa = 0.0018$ (with all lengths in meters) from [1]. The measuring directions $\hat{\mathbf{m}}_j$, $j = 176 \times 144$ were taken to be the same as that of the real device. Samples were generated using the noise model of (5). We considered infinite planes at a constant distance from origin $d = 4$ m but with varying normals as a function of spherical coordinates

$$\hat{\mathbf{n}} = (\cos\theta\cos\phi \; \cos\theta\sin\phi \; \sin\theta)^T. \tag{37}$$

In Figs. 2, 3, and 4, the three error metrics are plotted as a function of θ and ϕ. The four corners of the plots have been truncated as they correspond to samples for which the range is more than the device's maximum range of 7.5 m. The color-scale is different for all plots and should be noted carefully for proper comparison.

Discussion

Figure 2 shows that as expected, EMLP, which is based on iterative constrained nonlinear optimization, is the most accurate method. Renormalization also is relatively accurate, which shows that the solution (though not the estimated covariance) is not strongly dependent on the noise model. The non-iterative ALSP and EVPP method have similar accuracy.

Figure 3 shows that the wrongly assumed model in renormalization leads to a systematic error pattern for ε_2. The ALSP and EVPP methods are surprisingly accurate in estimating the principal error directions.

In Fig. 4, one notices that for the Renormalization, ALSP, and EVPP methods, the ratio of the estimated uncertainty to the least possible uncertainty steadily worsens as the incidence angle to the plane increases. EMLP does not show this effect and is also the most method where the ratio stays closest to unity. Interestingly, the ratio sometimes is even slightly less than unity, which suggests that EMLP may be a biased estimator, since the Cramér-Rao bound is only applicable

to unbiased estimators. This bias, if any, is quite small however, as is clear from Fig. 4. Renormalization also sometimes reports ratios less than unity: however, this is not trustworthy because it is based on a different noise model.

We compared the computation times for the three algorithms in MATLAB, although ALSP has been implemented in C++ also. In MATLAB, the average computation time (in 840 runs) per sample of 176×144 points took 0.0052 s, renormalization 0.4393 s, EVPP took 0.3332 s, and the iterative non-linear optimization for EMLP took about 15 s. Renormalization is slower as expected because it is iterative. EVPP is non-iterative and could possibly be made faster by a more efficient implementation. For run-times of a C++ implementation of a full-fledged region-growing algorithm with embedded ALSP, see Sect. 6. In summary, ALSP seems to be a good alternative for covariance computations in real-time applications. Another advantage of the covariance of the ALSP is that its components are known in terms of easily understood quantities like point-cloud scatter and their weighted mean—this semantic information is lost in the EVPP covariance, although numerically its performance is quite close to that of ALSP. EMLP is the most accurate method, but also computationally the most expensive. It is therefore recommended only for offline optimization.

6 Application: plane fusion after scene registration

In this section, we provide an application of the covariance estimate provided by ALSP for fusing two estimates of plane-parameters. Let the point-clouds corresponding to the same physical plane be sampled from two sensor reference frames $\mathcal{F}_\ell$ and $\mathcal{F}_r$. The coordinates of the same physical point observed from these frames are denoted by $\mathbf{r}_\ell$ and $\mathbf{r}_r$ respectively, and are related by

$$\mathbf{r}_\ell = \mathbf{R}\,\mathbf{r}_r + \mathbf{t}, \tag{38}$$

where $\mathbf{R}$ is the relative rotation and $\mathbf{t}$ is the relative translation, both assumed known. The plane parameters vector $\boldsymbol{\xi} \triangleq (\hat{\mathbf{n}}^T d)^T$ found in the two frames are related by the transform

$$\boldsymbol{\xi}_r = \mathbf{U}\,\boldsymbol{\xi}_\ell, \quad \mathbf{U} \triangleq \begin{pmatrix} \mathbf{R}^T & \mathbf{0}_{3 \times 1} \\ -\mathbf{t}^T & 1 \end{pmatrix}. \tag{39}$$

There are two intermediate problems which are not covered in this article, but have been addressed by the authors in other works

1. Planes extraction from range-images by region-growing [13].
2. Finding which planes in one scene correspond to which planes in the second, and subsequently finding the least

squares scene registration. This issue has been addressed in [11].

Once the planes have been matched and registered together, we want the corresponding plane parameters to be probabilistically fused together as well. The estimated plane parameters and their ALSP covariances satisfy $\mathbf{C}_\ell \boldsymbol{\xi}_\ell = 0$ and $\mathbf{C}_r \boldsymbol{\xi}_r = 0$, as shown earlier.

Now, $\mathbf{U}\boldsymbol{\xi}_\ell$ with covariance $\mathbf{U}\mathbf{C}_\ell\mathbf{U}^T$, and $\boldsymbol{\xi}_r$ with covariance $\mathbf{C}_r$ can be considered to be two observations in frame $\mathcal{F}_r$ for the same plane. The usual equation for fusing two Gaussians estimates, which is analogous to the Kalman filter update, would give

$$\mathbf{C}_1^+ = (\mathbf{U}\mathbf{C}_\ell\mathbf{U}^T)^+ + \mathbf{C}_r^+, \tag{40}$$

$$\boldsymbol{\xi}_1 = \boldsymbol{\xi}_r + \mathbf{C}_r(\mathbf{C}_r + \mathbf{U}\mathbf{C}_\ell\mathbf{U}^T)^+(\mathbf{U}\boldsymbol{\xi}_\ell - \boldsymbol{\xi}_r). \tag{41}$$

This approach has two problems: first, there is no guarantee that the *first three components* of $\boldsymbol{\xi}_1$ form a unit vector, and secondly, the property $\mathbf{C}_1\boldsymbol{\xi}_1 = \mathbf{C}_1^+\boldsymbol{\xi}_1 = 0$ may not be satisfied. We therefore propose that the fused covariance matrix be found by modifying the matrix in (40) as

$$\mathbf{C}_f^+ = \mathbf{C}_1^+ - \lambda_{\min}\{\mathbf{C}_1^+\}\,\hat{\boldsymbol{v}}_f\hat{\boldsymbol{v}}_f^T, \tag{42}$$

where $\lambda_{\min}\{\mathbf{C}_1^+\}$ is the minimum eigenvalue of $\mathbf{C}_1^+$ and the corresponding normalized eigenvector is $\hat{\boldsymbol{v}}_f$. Then the fused plane parameters $\boldsymbol{\xi}_f$ are found by using the reverse transform (30).

In fact, $\mathbf{C}_f^+$ is the matrix closest in Frobenius norm to $\mathbf{C}_1^+$ which satisfies both of the aforementioned properties and $\mathbf{C}_f$ is positive semi-definite. All fused quantities are with respect to the right frame. Another reason for preferring (42) over (41) is that the latter does not work if $\boldsymbol{\xi}$ is negated, which represents the same plane.

The results of the application of the above fusion strategy to real-life data is shown in Fig. 7b. The plane-patches were extracted using the ALSP method coupled with a region-growing algorithm for point-clouds obtained from a Swiss-ranger mounted on a robot turning in-place. As shown in Figs. 5 and 6, six successive views were analyzed by a registration algorithm and corresponding planes identified. The details of the registration algorithm, which makes extensive use of the computed covariance matrices, can be found in [11]. Finally, the corresponding plane-patches were rotated in the global frame and fused using the method described in this section. There were on an average 10 planar patches per view. The average time per view-pair for plane-patch extraction by region-growing, registration, and fusion was about 0.25 s


Fig. 5 Intensity images for successive views captured from a Swiss-ranger
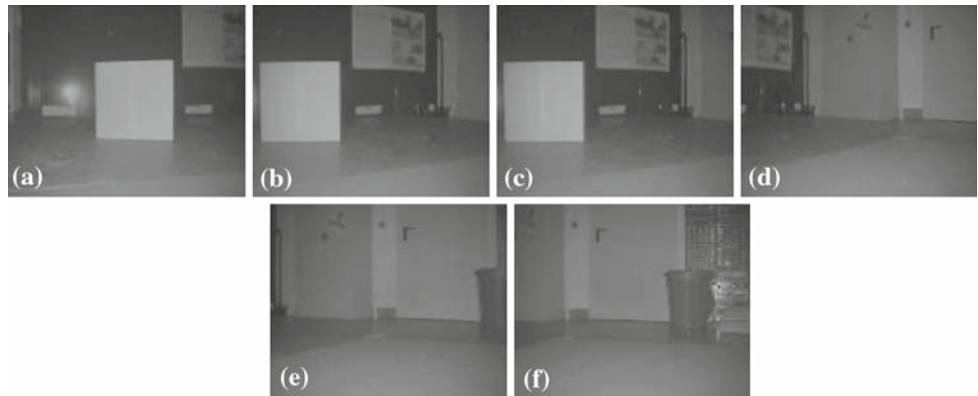
Fig. 6 Swiss-ranger point-cloud views corresponding to Fig. 5c. **a** *Front view* the color corresponds to intensity, **b** zoomed in *side view* to show noise normal to the planar surface
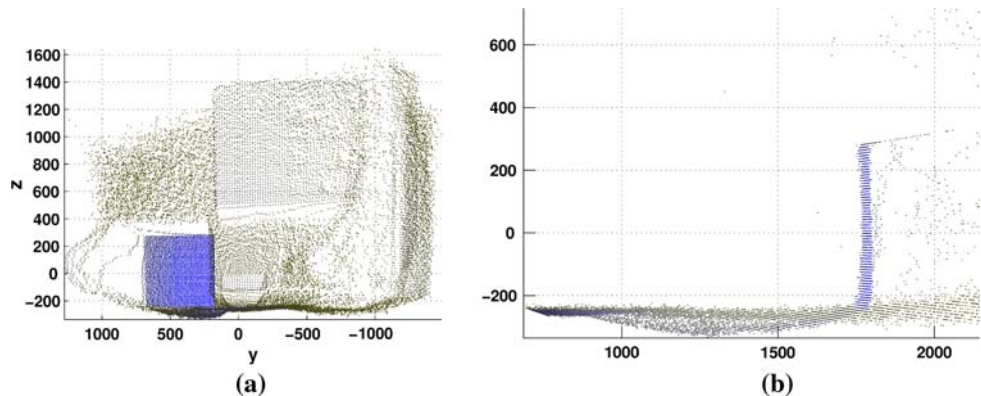
Fig. 7 Results for planes extracted from the views of Fig. 5. In **a**, planes were matched and aligned using our registration algorithm. Corresponding matched planes are drawn with the same color. Unmatched and filtered out planes are shown grayed out. **b** shows the result of applying plane fusion based on the methodology described in Sect. 6 on **a**. **a** registered planes (*front view*), **b** registered and fused planes (*front view*), **c** registered planes (*top view*), **d** registered and fused planes (*top view*)
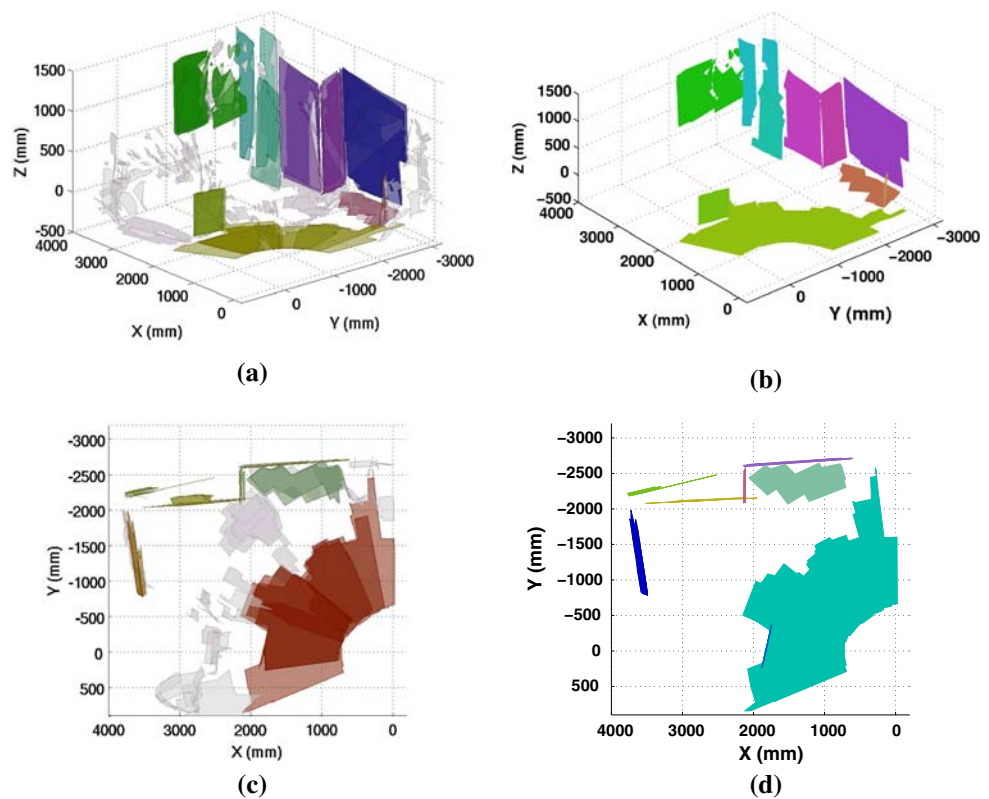
by a C++ implementation on a Pentium 4 HT, 3 GHz, with 1 GB of memory.

7 Conclusions

We presented several new results for the best-fit plane using a new range noise model. The covariance matrix for the plane parameters is derived in several different ways, all of which are entirely analytical. Some important properties of the null-space of this matrix are derived and used for plane fusion. Both simulation and real sensor data were presented to support our theoretical results. We conclude that the approximate least-squares plane-fit (ALSP) is a suitable method for real-time applications.

References

1. Anderson D, Herman H, Kelly A (2005) Experimental characterization of commercial flash ladar devices. In: International conference on sensing technologies. Palmerston North, New Zealand. http://www.frc.ri.cmu.edu/~alonzo/pubs/papers/icst05FlashLadar.pdf
2. CSEM (2006) The SwissRanger, Manual V1.02. 8048 Zurich, Switzerland. http://www.swissranger.ch
3. Horn RA, Johnson CR (1985) Matrix analysis. Cambridge University Press, Cambridge
4. Kanatani K (2005) Statistical optimization for geometric computation. Dover Publications, Inc., New York. ISBN 0486443086
5. Kanazawa Y, Kanatani K (1995) Reliability of fitting a plane to range data. IEICE Trans Inform Syst E78-D(12):1630–1635
6. Kohlhepp P, Pozzo P, Walther M, Dillmann R (2004) Sequential 3D-SLAM for mobile action planning. Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings of the 2004 IEEE/RSJ International Conference, vol 1, pp 722–729 doi:10.1109/IROS.2004.1389438
7. Nüchter A, Lingemann K, Hertzberg J (2007) 6D SLAM—3D mapping outdoor environments. J Field Robot 24(8/9):699–722
8. Pathak K, Birk A, Schwertfeger S, Poppinga J (2007) 3d forward sensor modeling and application to occupancy grid based sensor fusion. In: International conference on intelligent robots and systems (IROS). IEEE Press, San Diego, USA
9. Pathak K., Birk A., Vaskevicius N., Pfingsthorn M., Schwertfeger S., Poppinga J (2010) Online Three-Dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. J Field Robot (to appear). doi:10.1002/rob.20322
10. Pathak K, Vaskevicius N, Birk A (2009) Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds. In: International conference on robotics and automation (ICRA), pp 1631–1636. IEEE Press, Kobe, Japan. doi:10.1109/ROBOT.2009.5152502
11. Pathak K, Vaskevicius N, Poppinga J, Pfingsthorn M, Schwertfeger S, Birk A (2009) Fast 3d mapping by matching planes extracted from range sensor point-clouds. In: International conference on intelligent robots and systems (IROS). IEEE Press
12. PMD Technologies (2008) PhotonIC (R) PMD 3k-S. http://www.pmdtec.com
13. Poppinga J, Vaskevicius N, Birk A, Pathak K (2008) Fast plane detection and polygonalization in noisy 3D range images. In: IEEE

international conference on intelligent robots and systems (IROS). Nice, France

14. Prieto F, Redarce T, Boulanger P, Lepage R (1999) CAD-based range sensor placement for optimum 3D data acquisition. In: Second international conference on 3-D imaging and modeling (3DIM'99), p 0128. IEEE Computer Society, Los Alamitos, CA, USA. http://doi.ieeecomputersociety.org/10.1109/IM.1999.805343

15. Sivia DS (1996) Data analysis: a Bayesian tutorial. Oxford University Press, Oxford. ISBN 0198518897

16. Surmann H, Nuechter A, Hertzberg J (2003) An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. Robot Auton Syst 45(3–4): 181–198

17. Wang C, Tanahashi H, Hirayu H, Niwa Y, Yamamoto K (2001) Comparison of local plane fitting methods for range data, vol. 1, pp I-663–I-669. doi:10.1109/CVPR.2001.990538

18. Weingarten J (2006) Feature-based 3D SLAM. Ph.D. thesis, EPFL, Lausanne, Switzerland. http://library.epfl.ch/theses/?nr=3601

19. Weingarten J, Siegwart R (2006) 3D SLAM using planar segments. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Beijing

20. Weng J, Huang TS, Ahuja N (1989) Motion and structure from two perspective views: algorithms, error analysis, and error estimation. IEEE Trans Pattern Anal Mach Intell 11(5): 451–476. doi:10.1109/34.24779