

Extraction of Quadrics From Noisy Point-Clouds Using a Sensor Noise Model

Narunas Vaskevicius, Kaustubh Pathak, Razvan Pascanu, and Andreas Birk
 {n.vaskevicius, k.pathak, r.pascanu, a.birk}@jacobs-university.de

Abstract—Fitting optimum quadrics on segmented noisy range-image is a challenging task. The algebraic least-squares fit usually considered in the literature is biased, although it is fast to compute. We extend our previous work in planar patch extraction using a detailed sensor noise model to the extraction of quadrics. First, the fast least-squares method is modified to detect degeneracy automatically and it is used to aid segmentation. The final segmented quadric patch is then refined using a numerical maximum likelihood approach which employs the sensor range error model. Experimental results for artificial data and two different 3D sensors are provided to show the feasibility of our approach.

I. INTRODUCTION

Extracting surfaces from “point clouds,” typically of size 10^4 – 10^6 , sampled from currently available 3D sensors leads to a dramatic saving of storage. If these surfaces are utilized for scene registration instead of using traditional points-based methods, it leads to run-time savings as well— as shown for planar surfaces in a recent work of the authors [1], [2]. To compute the uncertainty of the registration, i.e. the covariance matrix of the relative pose computed using surface matching, the uncertainties associated with the extracted surfaces needs to be known. In [3], a detailed analysis of uncertainty associated with extraction of planar surface patches was presented assuming an error model depending on the range and the incidence angle. It is natural to go one step further and try to analyze the uncertainty associated with the extraction of the simplest non-planar surfaces, namely quadrics.

Quadric-fitting has been studied by many researchers in the last two decades. Biquadratic Monge patches of form $z = f(x, y)$, where $f(x, y)$ is a quartic polynomial, were fitted to range-data in [4]. Note that Monge patches, even when $f(x, y)$ is quadratic, form a different family of surfaces from quadrics. We choose to investigate quadrics because, firstly, cylinders, cones, ellipsoids, etc. are more familiar in human created environments. Secondly, certain coordinate-frame invariant properties can be readily computed for quadrics for further use in subsequent matching of surfaces for scene registration. The basics of least-squares quadric-fitting based on algebraic error were discussed in [5], whose results were also used in [6]. Several least-squares methodologies were compared in [7]. An Akaike Information Criterion (AIC) like metric was used in [8] to distinguish three classes: plane, cylinder, and a general quadric. A nearest point geometric least-squares approach was taken by [9], based on the results of [10]. They used certain shape parameters to distinguish cylinders, cones, and ellipsoids. A more recent work [11] presents a comparative study on model selection criteria and discusses their application to the surface selection problem in range modeling.

In all of the above works, the uncertainty of the extracted quadric in terms of the uncertainty present in the range-image was neither modeled nor estimated. An exception is [12], wherein a Kalman filter based incremental estimation followed by the method of instrumental variables were used. No sensor noise model was explicitly employed.

The authors are with the Dept. of EECS, Jacobs University Bremen, 28751 Bremen, Germany.

Another important aspect of the extraction of surfaces from the range image is its segmentation, whereby possible patch boundaries are identified based on proximity and curvature discontinuities. At the same time, good seed points for region-growing may also be found. Segmentation for planar patches was discussed in [13]. A good discussion on segmentation specific to quadrics can be found in recent literature [14], [15].

There are two main contributions of this paper:

- 1) Design of a three step least-squares algorithm to aid region-growing. The advantage of this method is that it can distinguish planes, which are degenerate quadrics, from real quadrics in the second step. This allows it to not fit a full quadric to a planar surface: doing so would have led to a non-unique quadric. Thus, our approach refines the method used in [5], [6], where planes and quadrics had to be handled separately. Note that distinguishing that the underlying surface was a plane after fitting a quadric to noisy planar data using the rank condition of some coefficient matrix is difficult because the optimum fit uses the extra freedom of quadratic terms to fit to the noise.
- 2) After extraction of the surface with region-growing aided by least-squares, we present a method to refine the surface model by maximum likelihood estimation, employing a sensor error-model which we have previously employed for planes [3]. This leads to a considerable improvement in the quality of the solution while additionally providing an estimate of the uncertainty associated with the quadric parameters. To the best of our knowledge, such an uncertainty analysis for quadrics based on a sensor error model has not been addressed in the literature till now.

The rest of the paper is organized as follows: Sec. II introduces the basic quadric model; Sec. III gives the fast three-step least-squares (LS) method for extracting quadrics to aid region-growing segmentation; Sec. IV gives the details of the segmentation itself; Sec. V derives the maximum-likelihood (ML) estimation for model refinement. Experimental results are presented in Sec. VI for two different 3D sensors, and the paper is concluded in Sec. VII.

II. QUADRIC MODEL

A general quadric can be denoted by a triplet $\langle \mathbf{A} \in \mathbb{R}^{3 \times 3}$ and symmetric, $\mathbf{b} \in \mathbb{R}^3$, $c \in \mathbb{R} \rangle$, which defines the surface by the implicit equation

$$\mathbf{r}^T \mathbf{A} \mathbf{r} + \mathbf{b}^T \mathbf{r} = c, \quad (1)$$

where, $\mathbf{r} = [x \ y \ z]^T$.

Due to the Spectral Theorem, \mathbf{A} has real eigenvalues and orthonormal eigenspace. Therefore, $\mathbf{A} = \mathbf{R}^T \mathbf{D} \mathbf{R}$, where \mathbf{D} is diagonal and \mathbf{R} is a rotation matrix. We define the transform

$$\mathbf{p} \triangleq \mathbf{R} \mathbf{r} + \mathbf{t}, \quad (2)$$

which, on substitution in (1) gives

$$\mathbf{p}^T \mathbf{D} \mathbf{p} + (\mathbf{R} \mathbf{b} - 2 \mathbf{D} \mathbf{t})^T \mathbf{p} - (c + \mathbf{b}^T \mathbf{R}^T \mathbf{t} - \mathbf{t}^T \mathbf{D} \mathbf{t}) = 0 \quad (3)$$

The simplest form of the quadric, termed its canonical (principal) form, is achieved if we select

$$\mathbf{t} = \frac{1}{2} \mathbf{D}^+(\mathbf{Rb}), \quad (4)$$

\mathbf{D}^+ being the Moore-Penrose generalized inverse. For a diagonal matrix, this inverse consists of all the *non-zero* diagonal elements being reciprocated, and the zero elements being left untouched.

III. LEAST SQUARES (LS) QUADRIC FITTING

The quadric equation (1) is only unique upto a scaling factor. Therefore the ten coefficients of the quadric need to be supplemented by a constraint. In [5], this constraint was taken as $\text{trace}(\mathbf{A}\mathbf{A}^T) = 1$, based on the argument that it is invariant with respect to rigid transformations. However, it should be noted that this constraint effectively loses the ability of the quadric to represent planes. As shown in Sec. II, it is easy to compute the canonical form of a quadric which is invariant to rigid transformations as well. Therefore, we do not commit ourselves to this constraint but proceed in a general manner.

We first define the following

$$\boldsymbol{\eta} \triangleq [x^2 \quad y^2 \quad z^2 \quad 2xy \quad 2xz \quad 2yz]^T, \quad (5a)$$

$$\mathbf{a} \triangleq [A_{1,1} \quad A_{2,2} \quad A_{3,3} \quad A_{1,2} \quad A_{1,3} \quad A_{2,3}]^T. \quad (5b)$$

Eq. (1) can be reformulated as follows

$$\mathbf{a} \cdot \boldsymbol{\eta} + \mathbf{b} \cdot \mathbf{r} = c \quad (5c)$$

The cost function for the algebraic least-squares fit is

$$\zeta \triangleq \sum_{i=1}^N (\mathbf{a} \cdot \boldsymbol{\eta}_i + \mathbf{b} \cdot \mathbf{r}_i - c)^2. \quad (6)$$

We now proceed in three steps to fit the least-squares quadric, which includes the degenerate case of the plane.

1) Setting $\partial\zeta/\partial c = 0$, one gets the optimum c as

$$c_* = \frac{1}{N} \sum_{i=1}^N (\mathbf{a} \cdot \boldsymbol{\eta}_i + \mathbf{b} \cdot \mathbf{r}_i) \triangleq \mathbf{a} \cdot \bar{\boldsymbol{\eta}} + \mathbf{b} \cdot \bar{\mathbf{r}} \quad (7)$$

2) Substituting (7) in (6) gives

$$\zeta_1 = \sum_{i=1}^N (\mathbf{a} \cdot (\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}) + \mathbf{b} \cdot (\mathbf{r}_i - \bar{\mathbf{r}}))^2. \quad (8)$$

Next, we define the following matrices for later reference, the first two of which are positive semi-definite.

$$\mathbf{S} \triangleq \sum_{i=1}^N (\mathbf{r}_i - \bar{\mathbf{r}})(\mathbf{r}_i - \bar{\mathbf{r}})^T, \quad (9a)$$

$$\mathbf{Q} \triangleq \sum_{i=1}^N (\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}})(\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}})^T, \quad (9b)$$

$$\mathbf{P} \triangleq - \sum_{i=1}^N (\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}})(\mathbf{r}_i - \bar{\mathbf{r}})^T. \quad (9c)$$

Setting $\partial\zeta_1/\partial\mathbf{a} = 0$ gives

$$\mathbf{Q}\mathbf{a}_* = \mathbf{P}\mathbf{b} \quad (9d)$$

Similarly, setting $\partial\zeta_1/\partial\mathbf{b} = 0$ gives

$$\mathbf{S}\mathbf{b}_* = \mathbf{P}^T\mathbf{a} \quad (9e)$$

In theory, we can use either of (9d) or (9e) to proceed. However, we note that if the scatter matrix \mathbf{S} has a high

condition number, then 9e cannot be used to solve for \mathbf{b}_* . This case is exactly the one when the surface is being fit to an underlying plane [3]. In this case, the eigenvector of \mathbf{S} corresponding to its minimum eigenvalue is the plane's unit normal \mathbf{b}_* , we set $\mathbf{a} = 0$, compute the distance of the plane from the origin using (7), and stop. Thus, in the second step, we are able to detect if the quadric has degenerated to a plane.

3) If the condition number of \mathbf{S} is above a minimum threshold, we fit a non-degenerate quadric. We use (9e) to solve for \mathbf{b}_* , and substitute it in (8) to get

$$\zeta_2 \triangleq \sum_{i=1}^N \left\| \left((\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}})^T + (\mathbf{r}_i - \bar{\mathbf{r}})^T \mathbf{S}^{-1} \mathbf{P}^T \right) \mathbf{a} \right\|^2 \quad (10)$$

$$\equiv \mathbf{a}^T \boldsymbol{\Omega} \mathbf{a}, \text{ where,} \quad (11)$$

$$\boldsymbol{\Omega} \triangleq \mathbf{Q} - \mathbf{P}\mathbf{S}^{-1}\mathbf{P}^T \quad (12)$$

To get a unique solution, we can now enforce the constraint that $\|\mathbf{a}_*\| = 1$. In this case, the \mathbf{a}_* which minimizes ζ_2 is simply the eigenvector of $\boldsymbol{\Omega}$ corresponding to its minimum eigenvalue. The optimal vector of linear coefficients \mathbf{b}_* can now be obtained from (9e) and c_* from (7).

We have thus obtained a systematic procedure to fit a quadric which automatically detects degeneracy and fits a plane instead of a full quadric, if required. The method is quick and incremental and thus can be used to aid region-growing as described in Sec. IV. It is however a *biased* estimate of the quadric due to the quadratic term [16], and therefore needs to be refined using a maximum likelihood estimation, as discussed later in Sec. V.

IV. PREPROCESSING AND REGION-GROWING

The underlying idea of the segmentation algorithm based on region-growing is similar to the previous own work [13] presented for plane detection. In the case of quadrics, a preprocessing step is extremely important.

A. Preprocessing

The preprocessing step ensures finding of good starting points, i.e. seeds, for the newly grown segments. As a rule of thumb, we look for a fast way of finding at least several seeds per region that are as far away as possible from the boundary of the surface patch. This is done in three steps. The first two steps discard noisy measurements and divide the set of points in possible regions by detecting edge points, while the third step sorts all points according to the distance from the boundaries of the regions. Note that the heuristics used to divide the set of points in regions do not find the final segments — they only detect clear boundaries between regions and in general tend to under-segment the range image. In order to have a dimensionless measure of the thresholds used, the data are scaled by the maximum range of the sensor such that all ranges would fall between 0 and 1.

1) *Range discontinuity*: The first step in detecting noisy measurements and possible boundaries between segments is to look for sudden changes in ranges of measurements. This is an intuitive step, and is controlled by a threshold. If the ranges of neighboring measurements have a sudden change by more than the given threshold, then either we are dealing with far away objects, or a noisy measurement.

2) *Surface orientation discontinuity*: The second heuristic used to find edges is to look for sudden changes in surface orientation. For this, we consider a sliding window of a relatively small size (in this case 5×5) sliding through the range image. Locally, a surface can be approximated by a plane, thus, fitting a plane to

the measurements of a small window gives an approximate surface orientation at the center point of the window. We find all points, where the orientation of the surface normal changes more than a predefined threshold. The change is measured by computing the dot-product between consecutive normals.

3) *Selecting seed points*: Once the noisy measurements and possible edges are detected, a wave propagation algorithm is run from these points giving distances to the potential seeds. The seeds are sorted according to this distance from the largest to the smallest and are passed to the region-growing algorithm as a preference list of starting points for grown segments.

B. Region-Growing

We use a region-growing algorithm for range image segmentation. Many of its details are quite similar to the segmentation for extracting planar patches as described in [13]. The main features are summarized below.

- 1) Only one region/patch is grown at a time. The region-grower starts growing regions from certain seed points which have been found by the preprocessing step to be suitable starting points. Seeds are taken from a preference list created by the preprocessor. The growing is done at any point in the range-image by testing its neighbors using an 8-pixel neighborhood.
- 2) Equations (9) are implemented in an efficient incremental manner.
- 3) When a potential point is tested for its suitability for addition to the current region, two criteria are used: 1) The least-squares (LS) mean square error (MSE) of the region should not exceed a certain threshold, and 2) the distance of the sample point's projection on the fitted surface from itself is not beyond another threshold.
- 4) The region is allowed to grow till at least 10 points (number of degrees of freedom of quadric +1) without any tests. If the region is classified as a plane and the number of points belonging to it has exceeded a threshold, its type is fixed and cannot be changed to a real quadric. Otherwise, the region is allowed to change its type from plane to a full-quadric and continue growing.

V. MODEL REFINEMENT USING MAXIMUM LIKELIHOOD

After region-growing is finished, we have a set of regions with the corresponding surface fitted by least-squares. Since least-squares is biased and since no sensor error model was considered during its computation, these surfaces need to be refined. We formulate a maximum-likelihood refinement algorithm in this section.

Assume that the sensor returned a point-cloud $\mathbf{r}_j = \rho_j \hat{\mathbf{m}}_j$, $j = 1 \dots N$, 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. Note that in a range image, $\hat{\mathbf{m}}_j$ would form a 2D grid on the image plane. As in [3], we assume this noise to be of the form $\rho_i \sim \mathcal{N}\{\bar{\rho}_i, \sigma^2\{\bar{\rho}_i, \hat{\mathbf{n}}_i \cdot \hat{\mathbf{m}}_i\}\}$, where $\{\cdot\}$ encloses function arguments, $\bar{\rho}_i$ is the *true* range of i -th measurement, and $\hat{\mathbf{n}}_i$ is the *local* normal of the surface at $\bar{\mathbf{r}}_i \triangleq \bar{\rho}_i \hat{\mathbf{m}}_i$. This implies a covariance matrix of \mathbf{r}_j of the form

$$\mathbf{C}_{\mathbf{r},j} = \sigma_j^2 \{\bar{\rho}_j, \hat{\mathbf{n}}_j \cdot \hat{\mathbf{m}}_j\} \hat{\mathbf{m}}_j \hat{\mathbf{m}}_j^T, \quad (13)$$

where the range standard-deviation $\sigma\{\bar{\rho}_j, \hat{\mathbf{n}}_j \cdot \hat{\mathbf{m}}_j\}$ is explicitly a (usually quadratic) function of $\bar{\rho}_j$ and has been found to be inversely proportional to $\hat{\mathbf{n}}_j \cdot \hat{\mathbf{m}}_j$, i.e. the cosine of the incidence-angle [17]. Two differences to the planar case are immediately obvious:

- 1) The normal $\hat{\mathbf{n}}_i$ is now a function of the sampling direction $\hat{\mathbf{m}}_i$ as well as the quadric $\langle \mathbf{A}, \mathbf{b}, c \rangle$. Using the gradient of the implicit equation, it can be found as

$$\hat{\mathbf{n}}_i = \frac{2\mathbf{A}\bar{\mathbf{r}}_i + \mathbf{b}}{\|2\mathbf{A}\bar{\mathbf{r}}_i + \mathbf{b}\|} \quad (14a)$$

- 2) The true-range $\bar{\rho}_i$ along a given $\hat{\mathbf{m}}_i$ is also a function of the quadric. It can be found as one of the roots of the quadratic equation

$$\alpha_i \bar{\rho}_i^2 + \beta_i \bar{\rho}_i - c = 0, \quad \alpha_i \triangleq \hat{\mathbf{m}}_i^T \mathbf{A} \hat{\mathbf{m}}_i, \quad (14b)$$

$$\beta_i \triangleq \mathbf{b}^T \hat{\mathbf{m}}_i, \quad \Delta_i \triangleq \beta_i^2 + 4\alpha_i c. \quad (14c)$$

Clearly, if $\Delta_i < 0$, the ray along $\hat{\mathbf{m}}_i$ does not intersect the surface. If $\Delta_i > 0$, then there are two intersections. If one of them is negative, then the positive one is selected; if both are positive, then the one closer to the measured sample point is selected.

We now define the quadric as a unit-vector in \mathbb{R}^{10} .

$$\hat{\nu} \triangleq [\mathbf{a}^T \quad \mathbf{b}^T \quad c]^T, \quad \|\hat{\nu}\| = 1. \quad (15a)$$

Given a set of N points $\mathbf{r}_i = \rho_i \hat{\mathbf{m}}_i$ hypothesized to lie on a quadric, The maximum likelihood (ML) problem can then be formulated as a nonlinear minimization of the negative log-likelihood.

$$\min_{\|\hat{\nu}\|=1} \mathcal{L} \triangleq \sum_{i=1}^N \log(\sigma_i) + \frac{1}{2} \sum_{i=1}^N \frac{(\rho_i - \bar{\rho}_i)^2}{\sigma_i^2}, \quad (15b)$$

$$\text{subject to } \Delta_i \geq 0, \quad i = 1 \dots N. \quad (15c)$$

It should be appreciated that this is a highly nonlinear cost because σ_i is a function of $\bar{\rho}_i$ and $\hat{\mathbf{n}}_i$, which in turn are functions of $\hat{\nu}$ and $\hat{\mathbf{m}}_i$ as shown in Eqs. (14). In [3], the range standard deviation was taken as

$$\sigma_i = \frac{\kappa \bar{\rho}_i^2}{|\hat{\mathbf{n}}_i \cdot \hat{\mathbf{m}}_i|} \quad (16)$$

based on experimental data from the references cited therein. We have added the inequality constraints (15c) because quadrics which do not lead to intersections along measurement directions should not be considered by the optimizer. We solve the problem (15) using a numerical method from the MATLAB optimization toolbox, using the solution of the least-squares (Sec. III) as the initial guess. Due to the high nonlinearity of the cost function and due to the presence of inequality constraints, the maximum-likelihood fitting is computationally quite expensive. After convergence, the optimizer also provides a numerically computed constrained Hessian, which can be used to estimate the covariance of the resulting solution. Unlike the planar case in [3], analytical expressions for the Hessian cannot be obtained for the quadric case.

VI. RESULTS

A. Experiments with artificial data

To evaluate the improvement of the model refinement using maximum-likelihood, we have simulated the Swiss-Ranger (SR) [18] sensor. The sensor noise was modeled as described in the previous section based on Eq. (16). Different noise levels and quadric types were investigated.

A typical case showing the advantages of the maximum-likelihood fitting is presented in Fig. 1. The surfaces were plotted using a MATLAB plotter for implicit 3D functions — ezimplot3 [19]. Here, a hyperbolic-paraboloid was used as the ground-truth surface Fig. 1(a). The least-squares solution at the noise level of the Swiss-Ranger sensor $\kappa = 0.0018$ [17] is plotted in Fig. 1(b) and

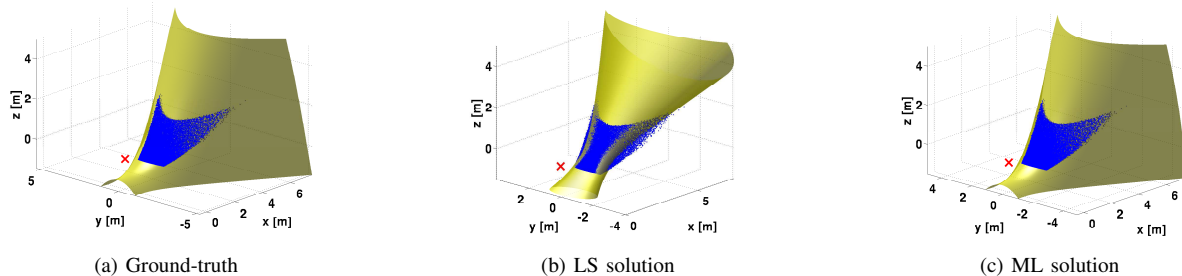


Fig. 1. Ground-truth surface and sampled points in blue with least-squares (LS) and maximum-likelihood (ML) solutions. The red cross denotes the sensor origin.

the refined solution in Fig. 1(c). This demonstrates a commonly encountered scenario in which the optimal surface fitted by the least-squares is of the incorrect type (in this case hyperboloid of one sheet) and in which the least-squares surface is subsequently refined by the maximum-likelihood fitting such that it corresponds to the ground-truth surface type. Moreover, around 10% of the points could not be back-projected on the least-squares solution since corresponding measurement directions did not intersect the optimal surface.

The dependency of the fitting error per point on the noise level is shown in Fig. 2. The noise level is controlled by changing the parameter κ in Eq. 16. The blue curve represents the average deviation of the noisy measurements from the ground-truth ranges— i.e. the fitting error explained by the noise. The red and green curves represent the fitting error for the maximum-likelihood and least-squares solutions. As can be seen, the least-squares fitting starts to introduce a structural error when the noise level increases. At the noise level of the Swiss-Ranger sensor ($\kappa = 0.0018$, the vertical dashed line in the figure), the deviation of the points from the least-squares solution was 3 times larger than the deviation from the maximum-likelihood optimal surface.

In the experiment, we made many trials at each noise level and considered only those cases where the non-linear optimization was successful, i.e. the optimization method satisfied one of the convergence criteria before the maximum number of iterations was reached. The convergence rate dependency on the noise level is given in Fig. 2. It can be immediately observed that the rate drops down in the presence of the higher noise. This is usually caused by a bad initial guess created by the least-squares fitting. The performance can probably be increased by considering alternative optimization methods.

B. Experiments with real-world data

Color-coding: In all the images that follow, the patches are color-coded based on the determined surface patch type: yellowish khaki for plane, blue for ellipsoid, green for cone, dark magenta for hyperboloid of two sheets, and red for points which could not be projected on the surface, as $\Delta_i < 0$ for them.

1) *Least Squares Fit:* We provide examples from two sensors. The first one is the Swiss-Ranger which delivers a quite noisy range image of size $176 \times 144 = 25,344$ from a field of view (FOV) of $47^\circ \times 39^\circ$. A sample range image and point-cloud for an umbrella on the floor is shown in Fig. 3 and its least-squares based segmentation is shown in Fig. 4.

The second sample shown in Fig. 5 is from an actuated laser range finder (ALRF) with much less noise. The ALRF has a horizontal field of view of 270° of 541 beams. The sensor is rotated (pitched) by a servo from -90° to $+90^\circ$ at a spacing of 0.5° . This

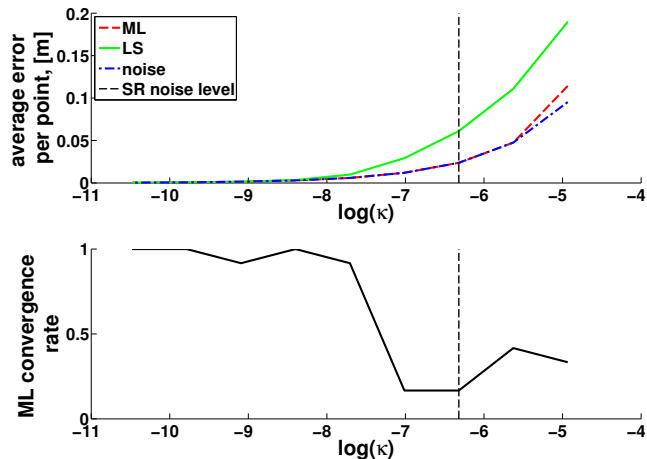


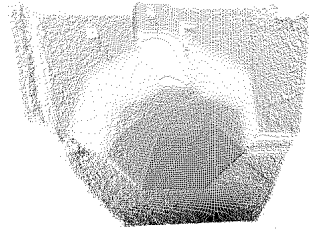
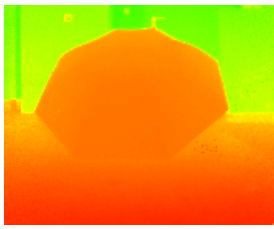
Fig. 2. The average error per point (top) and the convergence rate of the maximum-likelihood optimization (bottom) dependency on the noise level.

gives a 3D point-cloud of a total size of $541 \times 361 = 195301$ per sample. The maximum range of the sensor is about 20 meters.

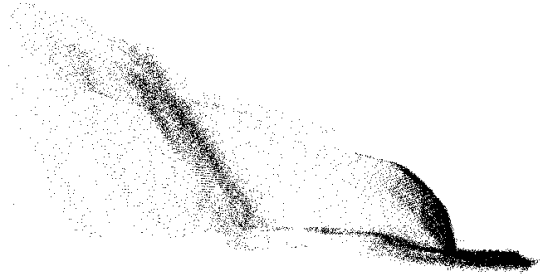
2) *Refinement using Maximum-Likelihood:* After the maximum-likelihood was computed to refine the patches found by least-squares for the Swiss-Ranger example of Fig. 3, the results improved significantly. We used the same sensor noise error model which was used in [3]. Refer to Figs. 6, 7, 8 to see the improvement. As seen, the maximum-likelihood algorithm changed the planar patch on top of the umbrella to a thin ellipsoid, and the main umbrella ellipsoid to a part of a sheet of hyperboloid of two sheets. Especially from Fig. 7, which shows the original points in black overlaid on the colored points projected on the found surfaces, it is clear that the agreement has improved significantly.

VII. CONCLUSIONS

We analyzed quadric extraction from noisy range images. The region-growing algorithm was aided by a biased but closed-form least-squares solution which could detect the case when the quadric degenerates to a plane. This was followed by a nonlinear optimization to solve the maximum-likelihood problem using the sensor range error model. Experimental results on artificial and real-world data from two different sensors were provided. We show that the maximum-likelihood fitting improved the solution considerably. However, this improvement comes at the cost of an increase in computational complexity.

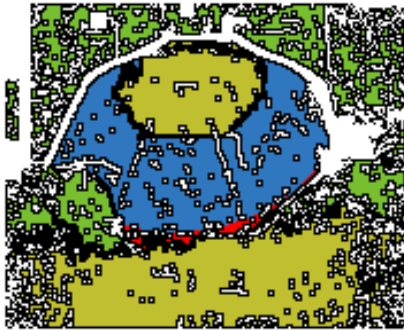


(a) The color-coded range image (b) Slanted front view of the point cloud.

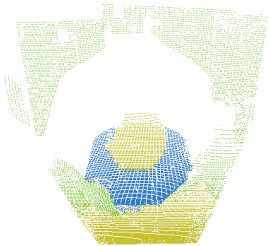


(c) Side view of the point cloud showing the noise. The top of the umbrella is seen to be quite flat.

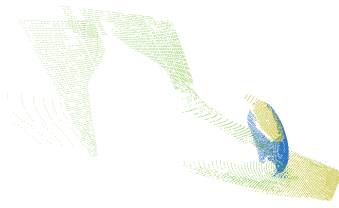
Fig. 3. The SR range image of an umbrella.



(a) The result of segmentation. The black points represent the inter-patch boundary and are not used in LS fitting.



(b) Front view.

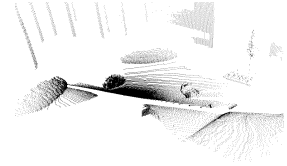


(c) Side view.

Fig. 4. Segmentation results for the sample shown in Fig. 3. Figures 4(b) and 4(c) show the point-cloud after the points were projected to the respective LS patches.



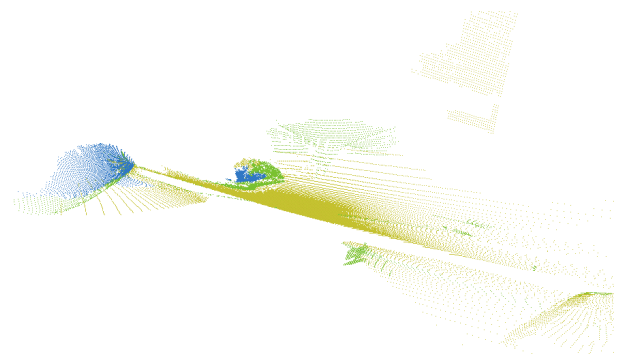
(a) A garden with undulating grassy terrain and rocks.



(b) A point-cloud collected from an actuated LRF. Perspective view.



(c) The segmented range image with the patch types color coded. Note that the due to the large field of view the range image has spherical distortion when viewed as a rectangle. The black points represent the inter-patch boundary and are not used in LS fitting.

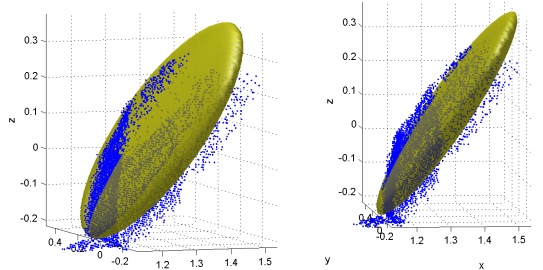


(d) The point-cloud after the points were projected to the respective patches.

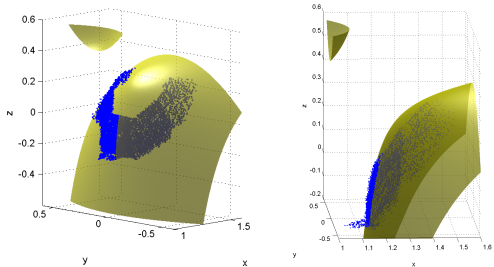
Fig. 5. A bigger sample of a landscaped Zen garden. The main surface-types found were planes, (large) cones, and ellipsoids.



Fig. 6. The segmentation image after the ML refinement step. Note that the plane on top of the umbrella in LS has been replaced by a thin ellipsoid in ML. Compare this to the LS result in Fig. 4.

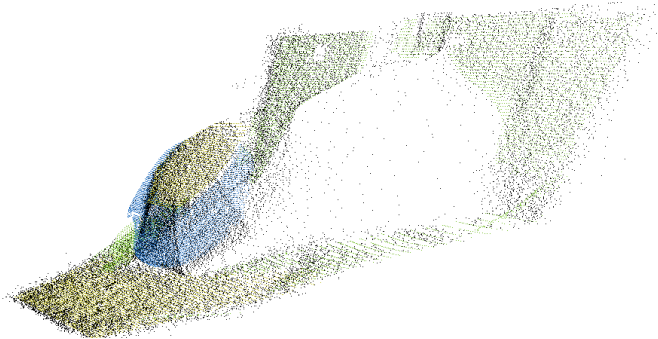


(a) LS fitted an ellipsoid to the umbrella. (b) LS ellipsoid side view.

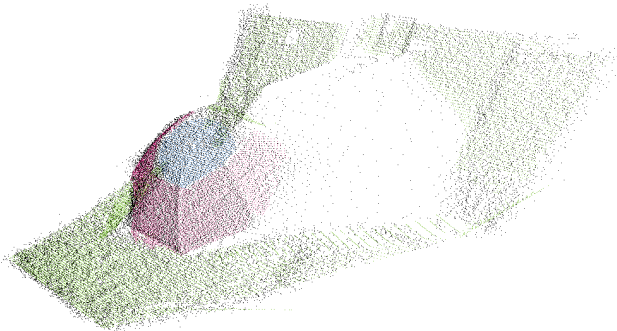


(c) ML fitted a better one sheet of the Hyperboloid of two sheets side view to the umbrella. (d) ML Hyperboloid of two sheets side view.

Fig. 7. Comparing the ML solution to that given by LS. The extracted patch is overlaid with an extension of the actual quadric found. ML fits the data much better. See also Fig. 8.



(a) The points projected on the LS surface-patches, side view. The overlap with the original points is bad in places as seen on the upper left edge of the umbrella.



(b) The points projected on the ML surface-patches, side view. There is a considerable improvement in the overlap with the original points compared to the LS.

Fig. 8. Comparing the ML solution to that given by LS. The original points in black are superimposed on the points projected on the surface. ML fits the data much better.

REFERENCES

- [1] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast Registration Based on Noisy Planes with Unknown Correspondences for 3D Mapping," *IEEE Transactions on Robotics (accepted)*, 2010. DOI: 10.1109/TRO.2010.2042989
- [2] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga, "Online Three-Dimensional SLAM by Registration of Large Planar Surface Segments and Closed-Form Pose-Graph Relaxation," *Journal of Field Robotics*, vol. 27, no. 1, pp. 52–84, 2010. DOI: 10.1002/rob.20322
- [3] K. Pathak, N. Vaskevicius, and A. Birk, "Uncertainty Analysis for Optimum Plane Extraction from Noisy 3D Range-Sensor Point-Clouds," *Intelligent Service Robotics*, vol. 3, pp. 37–48, 2010. DOI: 10.1007/s11370-009-0057-4
- [4] P. J. Besl, "Surfaces in early range image understanding," Ph.D. dissertation, University of Michigan, 1986.
- [5] O. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 27–52, 1986.
- [6] I. Bricault and O. Monga, "From volume medical images to quadratic surface patches," *Computer Vision and Image Understanding*, vol. 67, no. 1, pp. 24–38, July 1997.
- [7] X. Cao and N. Shrikhande, "Quadric surface fitting for sparse range data," in *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, Oct 1991, pp. 123–128. DOI: 10.1109/ICSMC.1991.169672
- [8] R. B. Fisher, A. W. Fitzgibbon, and D. Eggert, "Extracting surface patches from complete range descriptions," in *Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, 1997, pp. 148–155.
- [9] A. W. Fitzgibbon, D. W. Eggert, and R. B. Fisher, "High-level CAD model acquisition from range images," *Computer-Aided Design*, vol. 29, pp. 321–330, 1997.
- [10] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, November 1991.
- [11] N. Gheissari and A. Bab-Hadiashar, "A comparative study of model selection criteria for computer vision applications," *Image and Vision Computing*, vol. 26, no. 12, pp. 1636–1649, 2008, search Keywords: quadrics, model selection criteria, AIC, MDL, comparative study, surface extraction. DOI: 10.1016/j.imavis.2008.04.001
- [12] M. Baeg, H. Hashimoto, and J. B. Moore, "Pose estimation of quadratic surface using surface fitting technique," in *IEEE/R SJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1995.
- [13] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, "Fast Plane Detection and Polygonalization in noisy 3D Range Images," in *International Conference on Intelligent Robots and Systems (IROS)*. Nice, France: IEEE Press, 2008.
- [14] S. Petitjean, "A survey of methods for recovering quadrics in triangle meshes," *ACM Computing Surveys*, vol. 34, no. 2, pp. 211–262, June 2002.
- [15] A. Bab-Hadiashar and N. Gheissari, "Range image segmentation using surface selection criterion," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 2006–2017, July 2006.
- [16] J. Kuha, "Measurement error in quadratic regression models," in *Bulletin of the International Statistical Institute*, 1999. [Online]. Available: <http://www.stat.fi/isi99/proceedings/>
- [17] D. Anderson, H. Herman, and A. Kelly, "Experimental characterization of commercial flash lidar devices," in *International Conference on Sensing Technologies*, Palmerston North, New Zealand, November 2005. [Online]. Available: <http://www.frc.ri.cmu.edu/~alonzo/pubs/papers/icst05FlashLidar.pdf>
- [18] CSEM, *The SwissRanger, Manual V1.02*, 8048 Zurich, Switzerland, 2006. [Online]. Available: <http://www.swissranger.ch>
- [19] G. Morales, "Ezimplot3: a plotter for implicit 3d functions." [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/23623-ezimplot3-implicit-3d-functions-plotter>