

1 23 4

Simultaneous Segmentation and Superquadrics Fitting in Laser-Range Data

Ricardo Pascoal, Vitor Santos, *Member, IEEE*, Cristiano Premebida
and Urbano Nunes, *Senior Member, IEEE*

Abstract—This work presents a method for simultaneous segmentation and modeling of objects, detected in range data gathered by a laserscanner mounted on-board ground-robotic platforms. Superquadrics are used as model for both segmentation and object shape fitting. The proposed method, which we name Simultaneous Segmentation and Superquadrics Fitting (S3F), relies on a novel global objective function which accounts for the size of the object, the distance of range points, and for partial-occlusions. Results on experimental 2D range data, collected from indoor and outdoor environments, are qualitatively and quantitatively analyzed. Results are compared with those from popular and state-of-the-art segmentation methods. Moreover, we present results on 3D data obtained from an in house setup and also from a Velodyne LIDAR. This work finds applications in areas of mobile robotics and autonomous vehicles, namely object detection, segmentation and modeling.

I. INTRODUCTION

OBJECT detection and modeling is a long-standing research topic for the mobile robotics community, with the majority of the approaches relying on data from laser range sensors and cameras mounted on-board a robot. While perception systems using cameras have extensive applications, laser-based solutions constitute a field of particular importance in robotics, with many applications focussing on intelligent transportation, as evidenced in [1], [2], [3], [4], [5] and [6].

Common to most on-board perception systems using laser range data, is the use of a segmentation

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

R. Pascoal is with IEETA, University of Aveiro, Campus Universitario Santiago, 3810-193 Aveiro, Portugal, e-mail: rpascoal@ua.pt.

V. Santos is with the Dept. of Mechanical Engineering and IEETA, University of Aveiro, e-mail: vitor@ua.pt.

C. Premebida and U. Nunes are with Institute of Systems and Robotics (ISR-UC), Dept. of Electrical and Computer Engineering, University of Coimbra, Portugal, e-mail: cpremebida,urbano@isr.uc.pt.

process as the primary step towards detection and tracking of moving objects or landmarks detection. Generally, laser range data segmentation, or clustering, is the process of grouping range-points, belonging to a given scan, to a limited number of sets. Points within a set (a cluster or a *segment*) share common spatial characteristics, usually defined in terms of a distance criterion. In the robotics community, methods for laser-range data segmentation have been reported in [7], [4] and [8]. In general, segmentation is carried out independently of other processes, however, some authors have proposed methods which perform, concurrently, segmentation and geometric primitive extraction. So far, the most common geometric primitives used in the context of mobile robotics are lines [9] and circles [10].

Usually, object tracking is conducted based on the object's centroid and its relative shape, thus one of the concerns is how to model the shape (appearance) of the segment which characterizes a given object under tracking. Depending on the environment where the robot moves, the problem of assigning a proper geometric model to a segment is not trivial. This problem becomes more challenging under the presence of partial information, in non-structured scenarios, and with moving objects. In order to mitigate the problems arising from such situations, we propose to use superquadrics as a general geometric primitive upon which additional hypothesis may be set forth. Superquadrics, when compared to ellipsoids, provide an interesting increase of the degree of modeling capability at the cost of very few additional parameters and, on the other hand, it is simple to constrain the parameters in order to obtain a specific range of primitive shapes.

Superquadrics, as defined in [11], are a family of geometric shapes with many applications in computer vision, computer graphics and, recently, in robotics as well. The research works [12] and

[13] are examples of using superquadrics (or superellipses in the two-dimensional case) in computer vision. The work of Solina and Bajcsy [12] still constitutes an important reference on superquadrics fitting applied to range images. In robotics and haptics, most of the applications using superquadrics are devoted to object modeling for the purpose of manipulation and grasping, such as 3D hand tracking [14]. On the other hand, the use of superquadrics in mobile robotics and autonomous vehicle applications has been underexplored so far. Exceptions to this observation are [15], which employs superquadric surfaces to model free configuration spaces for autonomous navigation, and [16] which uses superquadrics to model an indoor environment, as perceived by a mobile robot, with the purpose of detecting changes in such environment.

In this work we expand on this topic and propose a method for concurrent segmentation and superquadric fitting in range data collected in indoor-outdoor scenarios. This method, named Simultaneous Segmentation and Superquadric Fitting (S3F), explores superquadrics formulation as an intrinsic model to be used in both segmentation and shape fitting. Additionally, we discuss and propose a solution to deal with self-occlusions and report numerical experiments on 2D and 3D field data, allowing for a better understanding of the problem of fitting superquadrics in laser-range data collected from moving ground-robotic platforms.

This paper is organized as follows. Background material, superquadric models and their properties are presented in Section II. In Section III, we address specific formulations necessary to support the new proposed objective function used in our method. Experiments, comparisons and discussions are presented in Section IV. Finally, conclusions are given in Section V.

II. BRIEF REVIEW ON SUPERQUADRICS

For the sake of completeness, the canonical and parametric formulation associated with superquadrics, and existing objective functions, are briefly presented in the sequel (please see [17] for a more complete description). Superquadrics have great potential in modeling a large set of shapes, some of them are illustrated in Fig. 1, and have been used as geometrical primitives in computer graphics, computer vision, and object modeling for robotics applications [13].

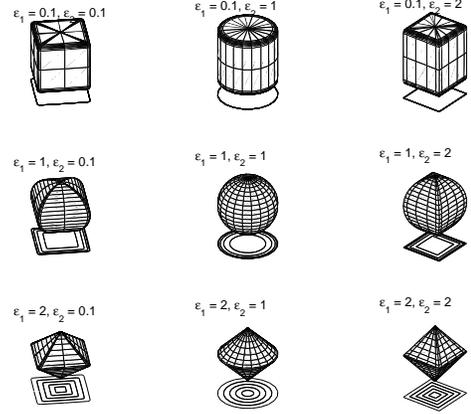


Fig. 1: Examples of superquadric shapes and their 2D projection.

Superquadrics may be represented in canonical implicit form by the contour surface of a function [12]. This function, for a fixed set of size and shape parameters, respectively a_i and ϵ_j , with $\{a_i | i=1,2,3, \epsilon_j | j=1,2 : a_i \in \mathbb{R}_{>0} \text{ and } \epsilon_j \in [0.1, 2]\}$, is given by

$$F(\mathbf{x}) = \left(\left(\frac{x_1}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{x_2}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{x_3}{a_3} \right)^{\frac{2}{\epsilon_1}}, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^3$ with $\mathbf{x} = [x_1, x_2, x_3]^T$. This function is often termed *inside-outside function* because for $F(\mathbf{x}) < 1$ a given point \mathbf{x} lies in the volume enclosed by the superquadric surface, and for $F(\mathbf{x}) > 1$ the point lies in the outer volume (visible side). On the other hand, if a point denoted by \mathbf{x}_s lies on the surface of a superquadric then $F(\mathbf{x}_s) = 1$. For $\mathbf{x} \in \mathbb{R}^2$, *i.e.* 2D shapes, equation (1) simplifies to:

$$F(\mathbf{x}) = \left(\frac{x_1}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{x_2}{a_2} \right)^{\frac{2}{\epsilon_2}}. \quad (2)$$

Equations (1) and (2) are valid for superquadrics centered at the origin of a local reference frame (x_1, x_2, x_3) . Considering the coordinate system shown in Fig. 2, where \mathbf{X}_p is a point in the sensor (laser) coordinate system (X_1, X_2, X_3) , the origin of the local reference frame is given by the coordinates of the vector μ . Therefore, in order to cope with superquadrics having arbitrary orientation and centered at a distance of μ from the sensor reference frame, a rotation (expressed by \mathbf{R}) is applied to

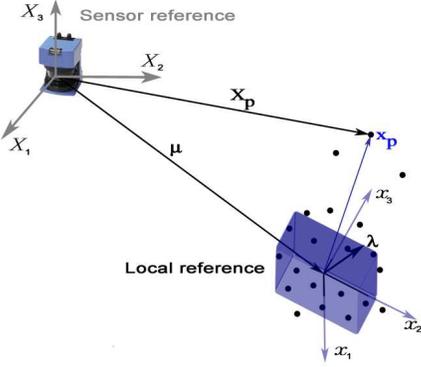


Fig. 2: Coordinate systems associated with the sensor (laser) and the local (superquadric) reference system. The centroid of the superquadric is given by μ .

the superquadric local reference frame and then followed by a translation (expressed by μ). As suggested in [12], these operations may be collected into a homogeneous transformation as follows:

$$\begin{bmatrix} \mathbf{X}_p \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} \mathbf{x}_p \\ 1 \end{bmatrix}, \text{ with } \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mu \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (3)$$

where \mathbf{x}_p is a point in the local reference system. The rotation is performed based on Euler-Rodrigues parameters [18]

$$\mathbf{R} = \begin{bmatrix} 1 - 2(e_2^2 + e_3^2) & 2(e_1e_2 + e_0e_3) & 2(e_1e_3 - e_0e_2) \\ 2(e_1e_2 - e_0e_3) & 1 - 2(e_1^2 + e_3^2) & 2(e_2e_3 - e_0e_1) \\ 2(e_1e_3 + e_0e_2) & 2(e_2e_3 - e_0e_1) & 1 - 2(e_1^2 + e_2^2) \end{bmatrix}, \quad (4)$$

with $e_0 = \cos(\theta/2)$, $e_{i=\{1,2,3\}} = \lambda_i \sin(\theta/2)$, and θ is the angle of rotation about the direction vector λ with $\|\lambda\| = 1$. Using Euler-Rodrigues parameters enables an unconstrained and singularity free transformation.

For the purpose of graphical representation, the parametric representation is necessary. Given the size and shape parameters (a_i, ε_j) , the 3D parametric representation is of the form

$$\begin{aligned} x_{s1} &= a_1 \operatorname{sgn}(\cos(\omega)) \cos^{\varepsilon_1}(\eta) |\cos(\omega)|^{\varepsilon_2} \\ x_{s2} &= a_2 \operatorname{sgn}(\sin(\omega)) \cos^{\varepsilon_1}(\eta) |\sin(\omega)|^{\varepsilon_2} \\ x_{s3} &= a_3 \operatorname{sgn}(\sin(\eta)) |\sin(\eta)|^{\varepsilon_1} \end{aligned} \quad (5)$$

where (x_{s1}, x_{s2}, x_{s3}) are the coordinates of the points on the surface of the superquadric, $\{\eta : -\frac{\pi}{2} \leq \eta \leq \frac{\pi}{2}\}$ and $\{\omega : -\pi \leq \omega < \pi\}$. The parametric form allows one to identify that there exists a bounding

box $[-a_1, a_1] \times [-a_2, a_2] \times [-a_3, a_3]$ and a one-to-one correspondence between points belonging to the surface and the parameters (angles) η and ω . Because of the one-to-one correspondence, the angular parameters corresponding to points on the surface may be retrieved as

$$\begin{aligned} \eta &= \operatorname{atan2} \left(\operatorname{sgn}(x_{s3}) \left| \frac{x_{s3}}{a_3} \right|^{1/\varepsilon_1}, \right. \\ &\quad \left. \left(\left(\frac{x_{s1}}{a_1} \right)^{2/\varepsilon_2} + \left(\frac{x_{s2}}{a_2} \right)^{2/\varepsilon_2} \right)^{\varepsilon_2/2\varepsilon_1} \right) \\ \omega &= \operatorname{atan2} \left(\operatorname{sgn}(x_{s2}) \left| \frac{x_{s2}}{a_2} \right|^{1/\varepsilon_2}, \operatorname{sgn}(x_{s1}) \left| \frac{x_{s1}}{a_1} \right|^{1/\varepsilon_2} \right). \end{aligned} \quad (6)$$

A. Common objective functions

The importance of selecting an adequate objective function for superquadric fitting has been addressed in [19], [20] and [21], while [19] and [21] have performed numerical experiments specifically to characterize the behavior of objective functions used in fitting of 3D superquadrics. Dense synthetic range data is considered in [19], while field data is used in [21]. Aspects such as concavity of the objective function towards the minimum and precision on the recovered parameters were analyzed. In particular, the conclusion of the study reported in [21] was that the ‘‘mean square error and minimal volume’’ objective function used in [12], expressed by

$$G = \prod_{i=1}^3 a_i \cdot \sum_j \left(1 - F^{\varepsilon_1}(\mathbf{x}_p^j) \right)^2, \quad (7)$$

is less adequate than the ‘‘radial Euclidean’’ objective function proposed by [19]:

$$G = \sum_j \left(\|\mathbf{x}_p^j\| \left(1 - F^{-\varepsilon_1/2}(\mathbf{x}_p^j) \right) \right)^2, \quad (8)$$

where the summation is over all points being fitted, which may be the complete laser scan or just part of the scan. The analysis presented in [19] shows that the radial objective function outperformed other measures in terms of precision, sensitivity to parametric changes and the concavity towards the minimum.

The objective function from the landmark paper [12] is expressed by the product between a volume penalty, $G_V = \prod_{i=1}^3 a_i$, and a quadratic error, $G_F = \sum_j \left(1 - F^{\epsilon_1}(\mathbf{x}_p^j)\right)^2$, resulting in (7). Both expressions, (7) and (8), do not fulfill all the requirements to solve the problem being studied. More specifically, we noted that a null error in G_F will in fact preclude a desirable further minimization of G_V . For instance, in indoor environments, a null distance error is easily induced by a straight wall segment (in the form of a line-segment), or a planar patch in the 3D case, and this will not allow a proper volumetric minimization. The objective function (8), showed to be more robust to outliers but also evidenced difficulties in properly minimizing the superquadric volume. Notice that line-segments (or planar patches) would constitute ill posed problems if the inclusion of a volume minimization term was not properly chosen.

While volume minimization and distance error are quite common subjects in the literature dedicated to superquadrics, visibility is not commonly addressed as part of a recovery process but, instead, as a condition for shading in computer graphics. Objects represented by superquadrics are frequently shaded using information on the angle between the superquadric normal and the light source. Unless information about the viewpoint is available, or estimated, the superquadric fitting procedure may produce surfaces fitting to points which would actually be self-occluded from the laserscanner (considering such surface actually exists). This is a problem, not considered in previous publications, that occurs when no penalty is given to points on an object occluded by a recovered superquadric. In fact, when using solely distance and volume objective functions, a superquadric could fit just as well to scanned points with one of its self-occluded sides along a segment. Without penalizing points which lie on the self-occluded side, the fitting procedures will often fail to correctly recover even the simplest partially represented object. This finding led us to propose a method comprising a new objective function which is presented in the sequel.

III. SIMULTANEOUS SEGMENTATION AND SUPERQUADRIC FITTING IN RANGE DATA

An appropriate segmentation stage is a key step towards detecting and modeling objects using data

from laserscanners. Usually, segmentation is performed independently of shape extraction and, as consequence, there is no guarantee of consistency between the hypothesis used to perform the segmentation and the shapes extracted from the segments (objects). This is pointed out, *e.g.*, in [22] where it is stated that, performing segmentation as an independent process, is not adequate for use with robot manipulators because segmentation itself will fulfill constraints other than those imposed by fitting superquadrics.

In this work, the characteristics and the conditions under which our dataset was collected considerably differ from the related works mentioned in previous sections. Due to sparseness of the data, type of environment, scale variations, noise, and partial information, a new optimization procedure was needed to cope with typical mobile robotic and vehicular environments. The shortcomings mentioned in Section II have motivated us to propose an objective function based on partial costs, so that the volume of the superquadric, the distance error and the visibility are all taken into account (in the 2D case, it is adapted accordingly to area, distance and visibility). The proposed objective function takes the form of a weighted sum of all the portions to be minimized:

$$G = \alpha_1 G_V + \alpha_2 G_F + \alpha_3 G_V, \quad (9)$$

where the partial costs G_V , G_F and G_V refer to volume, distance and visibility respectively, and the weight hyperparameters α_i control the behavior and influence of each cost parcel of (9). The volume penalty is important once the other partial costs have low values, more specifically in situations of self-occlusion which would result in an undeterminate volume.

We begin by considering the extracted volume, which is penalized in the form of the product

$$G_V = \prod_{i=1}^n a_i, \quad (10)$$

where $n = 3$ if (1) is considered (3D case) or $n = 2$ if (2) is used (2D case). Defined when (1) was established, a_i represent the size parameters. Taking \mathbf{x}_p^i to be the i^{th} scanned point from a set of N points represented in the local reference frame, and considering a bounding sphere with diameter $d = \max_{j,k}(\|\mathbf{x}_p^j - \mathbf{x}_p^k\|)$, where it is assumed that the

superquadric is inscribed by such sphere, $[0, d^3[$ is a conservative interval for the values of G_V .

The distance part of our objective function is based on [19] and is expressed as:

$$G_F = \frac{1}{N} \sum_{i=1}^N \left(\|\mathbf{x}_p^i\| \left(1 - F^{-\varepsilon_1/2}(\mathbf{x}_p^i, \Lambda) \right) \right)^2, \quad (11)$$

where $[0, d^2[$ is assumed as a conservative interval for the values of G_F , with zero being the case when the scanned points are exactly on the surface. The vector of unknowns is $\Lambda = \{a_i, \varepsilon_m, \mu, \lambda_1, \lambda_2, \theta\}$ *i.e.*, a total of 11 components in the 3D case.

Potential self-occlusion can be readily verified by using the dot product between the superquadric surface normal and the unit vector defined from the sensor to the range point. The dot product should be negative at the time of detection. When a parametric formulation exists, it is common to calculate the surface normal as the cross-product between the tangents using derivatives with respect to the parameters (angles). However, this procedure would require the solution of (6) to recover the parameters η and ω in (5) that correspond to a particular range point. The additional equations to recover the correct quadrant of η and ω , and the use of the cross-product, can be easily avoided. In fact, if ε_1 or ε_2 are extended to be functions of \mathbf{x} (see [23], [24]), there are additional benefits in avoiding the use of η and ω since they result in implicit equations. Instead, the gradient of the canonical representation has been used to calculate the surface normal directly at the range points, $\mathbf{n} = \frac{\nabla F}{\|\nabla F\|}$. This is possible because the gradient of the canonical representation, calculated at a range point $\mathbf{x}_p = \beta \mathbf{x}_s$, with β a scalar and \mathbf{x}_s on the surface of the superquadric, results in

$$\nabla F(\mathbf{x}_p) = \beta^{2/\varepsilon_1} \nabla F(\mathbf{x}_s), \quad (12)$$

i.e., the gradient of the canonical representation has the same direction (orientation and sense) at radially aligned points.

The gradient of an extended superquadric, $\nabla F = [\partial_{x_1} F, \partial_{x_2} F, \partial_{x_3} F]$ *i.e.*, the general case which considers that the exponents are themselves function of

the coordinates (see [23], [24]), is given by

$$\begin{aligned} \partial_{x_1} F &= (f_1 + f_2)^{\frac{\varepsilon_2}{\varepsilon_1}} \left(\frac{1}{\varepsilon_1} \frac{\partial \varepsilon_2}{\partial x_1} - \frac{\varepsilon_2}{\varepsilon_1^2} \frac{\partial \varepsilon_1}{\partial x_1} \right) \ln(f_1 + f_2) \\ &+ (f_1 + f_2)^{\frac{\varepsilon_2}{\varepsilon_1} - 1} \left(f_1 \left(\frac{2}{x_1 \varepsilon_1} - \frac{1}{\varepsilon_2 \varepsilon_1} \frac{\partial \varepsilon_2}{\partial x_1} \ln(\hat{x}_1^2) \right) \right. \\ &\left. - \frac{f_2}{\varepsilon_2 \varepsilon_1} \frac{\partial \varepsilon_2}{\partial x_1} \ln(\hat{x}_2^2) \right) - \frac{f_3}{\varepsilon_1^2} \frac{\partial \varepsilon_1}{\partial x_1} \ln(\hat{x}_3^2), \end{aligned} \quad (13)$$

$$\begin{aligned} \partial_{x_2} F &= (f_1 + f_2)^{\frac{\varepsilon_2}{\varepsilon_1}} \left(\frac{1}{\varepsilon_1} \frac{\partial \varepsilon_2}{\partial x_2} - \frac{\varepsilon_2}{\varepsilon_1^2} \frac{\partial \varepsilon_1}{\partial x_2} \right) \ln(f_1 + f_2) \\ &+ (f_1 + f_2)^{\frac{\varepsilon_2}{\varepsilon_1} - 1} \left(f_2 \left(\frac{2}{x_2 \varepsilon_1} - \frac{1}{\varepsilon_2 \varepsilon_1} \frac{\partial \varepsilon_2}{\partial x_2} \ln(\hat{x}_2^2) \right) \right. \\ &\left. - \frac{f_1}{\varepsilon_2 \varepsilon_1} \frac{\partial \varepsilon_2}{\partial x_2} \ln(\hat{x}_1^2) \right) - \frac{f_3}{\varepsilon_1^2} \frac{\partial \varepsilon_1}{\partial x_2} \ln(\hat{x}_3^2), \end{aligned} \quad (14)$$

$$\begin{aligned} \partial_{x_3} F &= -\frac{\varepsilon_2}{\varepsilon_1^2} \frac{\partial \varepsilon_1}{\partial x_3} (f_1 + f_2)^{\frac{\varepsilon_2}{\varepsilon_1}} \ln(f_1 + f_2) \\ &+ f_3 \left(\frac{2}{x_3 \varepsilon_1} - \frac{1}{\varepsilon_1^2} \frac{\partial \varepsilon_1}{\partial x_3} \ln(\hat{x}_3^2) \right), \end{aligned} \quad (15)$$

with $\hat{x}_i = x_i/a_i$ and $f_1 = \hat{x}_1^{\frac{2}{\varepsilon_1}}$, $f_2 = \hat{x}_2^{\frac{2}{\varepsilon_2}}$ and $f_3 = \hat{x}_3^{\frac{2}{\varepsilon_3}}$.

The pure visibility constraint results in a discontinuous function at the superquadric surface (visible, on the surface, and not visible). This characteristic is highly undesirable in the context of iterative optimization and thus an additional contribution of the present work is to propose a smooth objective function to improve convergence. The self-occlusion objective function has been designed using the directional derivative ∇_r , and the hyperbolic tangent as a replacement for the discontinuous Heaviside function. This replacement ensures a smooth transition zone with controllable width for the shift from occluded to visible. The hyperbolic tangent is a C^∞ function and has also been suggested for other optimization problems [25]; thus, the self-occlusion objective function is expressed by

$$G_V = \frac{1}{2} + \frac{1}{2N} \sum_{i=1}^N \tanh(\alpha_0 \nabla_r F(\mathbf{x}_p^i, \Lambda)) \quad (16)$$

where G_V takes values in the interval $]0, 1[$, with zero being the limit value when all points lie on the visible side of the surface and one takes $\lim_{\alpha_0 \rightarrow \infty} G_V$ (*i.e.* the hyperbolic tangent converges to a Heaviside). The gradient is calculated in the superquadric local reference frame and the rotations to the sensor reference are performed afterwards. Indeed, the

parameters associated with rotation and translation are constants and do not need to be applied before differentiation. Translation itself is not necessary for the analysis of gradients or directional derivatives. The parameter α_0 controls the transition width of the hyperbolic tangent and thus its similarity with the Heaviside function. The directional derivative of the *inside-outside function* used in equation (16), $\nabla_r F$, is determined by $\nabla_r F = \nabla F \cdot \mathbf{X}_p / \|\mathbf{X}_p\|$. To calculate this directional derivative it is important to have data from the laser scanner in the sensor reference frame at the time of scanning (even if the reference frame is moving - which is the case in this work). Finally, applications using 2D data require the dimension of the vector of unknowns to be adequately reduced. Because the proposed objective function is well behaved, the Levenberg-Marquardt algorithm was used to achieve the results reported in this work.

The Levenberg-Marquardt algorithm requires the objective function Jacobian. Special care in establishing the cost function has enabled stable numerical differentiation and, additionally, it allows calculation of the analytical Jacobian. Numerical conditioning of the analytical Jacobian and the use of other optimization methods, which may be considered in evaluating trade-offs between memory usage and processing time, are considered important topics that require extensive and additional numerical experiments which exceeds the scope of this work. Furthermore, our method is iterative and needs initial values for the shape, size and orientation of the superquadric. These values have been estimated based on a bounding box which encompasses the whole scan, *i.e.* a rounded corner rectangular superquadric has been used and aligned with the direction of motion of the mobile platform.

A. Implementation

Our method concurrently segments and extracts superquadrics from 2D range data and can be applied as a self contained 2D method, or expanded to the 3D case (for selected environments). Concurrent, in this context, means that segmentation is based on successfully fitting superquadrics thus, at some point in the segmentation process, an extracted superquadric is obtained automatically while a segment is identified. For the 2D case, which is detailed in this section, equation (2) is used as

part of the solution in both the segmentation and the goodness of fit, thus ensuring consistency of the model used simultaneously for segmentation and shape fitting. In 3D environments, the same procedure applies when the data contains sufficient information allowing a consistent projection to a 2D plane. Notice that, for the 3D case, the formulation given in (1) is used to recover the parameters of the superquadrics.

Considering a set S of N 2D range-points \mathbf{x}_i in the local coordinate system (where $i = 1, \dots, N$), the main steps of the method, which is summarized in **Algorithm 1**, are the following:

- 1) Calculate the midpoints $\hat{\mathbf{x}}_m$ (where $m = 1, \dots, N - 1$) between each two consecutive angularly ordered scanned points.
- 2) Fit a superquadric to the scanned points by minimizing the objective function (9).
- 3) Check the value of the objective function against the threshold (Thr_G) and the minimum number of points (equal to 2 in our case). If less than the threshold, then a valid superquadric and a segment have been found and its parameters Λ are added to a list. Remove from S the points belonging to the extracted superquadric, and jump to step 5.
- 4) Calculate the most-interior midpoint ($b1$) *w.r.t.* the superquadric, and the point with worst ‘shear’ ($b2$). Midpoint $b1$ corresponds to the smallest value of F and point $b2$ to the worst difference between gradients at successive points (interpreted as forces which would ‘cut’ the superquadric). Select $b2$ unless $b1$ has no neighbor within a distance d . Distance d is a scene preservation parameter which is used only to reject superquadrics which have good fit but are bridging gaps greater than d (fixed in our experiments as $d = 1.5m$). Return to step 2 with the subset that ends with the point closest to the segmenting point.
- 5) Group unclassified points within unprocessed subsets (if they exist), thus making a new initial set, and jump to 2 if the number is greater than a certain minimum (set to 1).
- 6) The algorithm stops when all points have been assigned to a given superquadric or labeled as outlier.

The reason for using midpoints at step 4, and

Algorithm 1 Simultaneous segmentation and superquadric fitting algorithm

Input: S : set of 2D range-points (\mathbf{x}) sorted by the angular values, where $\mathbf{x} \in S$;
Output: Ω : set of parameters (Λ) for all recovered superquadrics;
 1: Λ_L and Λ_U : element-wise lower and upper bounds for Λ ;
 2: $\Lambda^* = \{\Lambda : \Lambda_L \leq \Lambda \leq \Lambda_U\}$: element-wise operators assumed;
 3: G : measure of the goodness of fit, see (9); Thr_G : adopted threshold on the goodness of fit;
 4: i : the index of a given laser-point \mathbf{x}_i ;
 5: m : the index of a given midpoint $\hat{\mathbf{x}}_m = 0.5(\mathbf{x}_i + \mathbf{x}_{i+1})$;
 6: b : the index of a given laser-point selected as break point \mathbf{x}_b ;
 7: S^s : set of range-points which do not have an associated superquadric;
 8: S^* : set of points belonging to a superquadric that complies with Thr_G ;
 9: $S^* \leftarrow S$;
 10: **while** $|S^*| > 1$ **do**
 11: **while** $G > Thr_G$ **do**
 12: $\Lambda : \min_{\Lambda \in \Lambda^*} G : i.e. \ \Lambda$ which minimizes G ;
 13: **if** $G \leq Thr_G$ **then**
 14: $S^s \leftarrow S^s$;
 15: $\Omega \leftarrow \Omega \cup \Lambda$;
 16: **else**
 17: $b1 = \min_{\hat{\mathbf{x}}_m \in S^*} F(\hat{\mathbf{x}}_m, \Lambda)$;
 18: $b2 = \max_{\mathbf{x}_i \in S^*} |\partial_{x_1}(F(\mathbf{x}_i, \Lambda) - F(\mathbf{x}_{i+1}, \Lambda))| + |\partial_{x_2}(F(\mathbf{x}_i, \Lambda) - F(\mathbf{x}_{i+1}, \Lambda))|$;
 19: $b = select(b1, b2)$;
 20: $S^* \leftarrow S^* \setminus \{\mathbf{x}_i | b < i \leq |S^*|\}$;
 21: **end if**
 22: **end while**
 23: $S^* \leftarrow S \setminus S^s$;
 24: **end while**

not the range-points as proposed by [26]¹, is to reduce the chance of inappropriate segmentation which may happen when range-points lie on the superquadric boundary. This modification was found necessary, in particular to cope with parallel lines from detected walls (e.g. in a tunnel or corridor).

To illustrate, sequentially, the effects of each step of the algorithm, an example of recovering a superquadric from a 2D range-scan is shown in Fig. 3. The result of the step 2 is shown in the left part of Fig. 3, where the segment having the breakpoint is chosen, according to step 4, to break the scan into a new sub-scan. The procedure is repeated for the remaining sub-scans until a good fit is found. The superquadric shown in the right part of Fig. 3 is, for this example, the first to be recovered. The range-points belonging to this superquadric form the first segment and are not considered for further iterations, indicating the simultaneity behavior. Finally, the algorithm restarts at step 1 and the procedure is repeated until the fitting criteria are no longer met.

As demonstrated, our method works from global to local, has a deterministic behavior and was designed towards dealing primarily with 2D data. These aspects are in contrast with most of the

published algorithms using superquadric models, such as the *Segmentor* [17], which works from local to global, rely on a large number of initial random (stochastic) seeds and are designed exclusively for 3D data. Moreover, in the applications we are dealing with, the CPU-time of the *Segmentor* seems to be prohibitive.

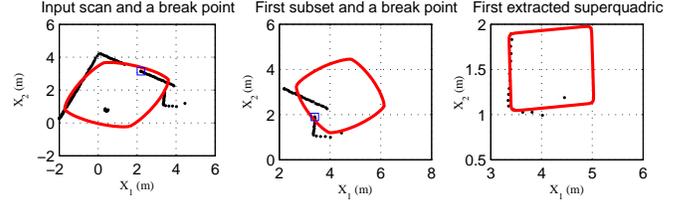


Fig. 3: Sequence illustrating the extraction of a superquadric from a scan of 2D range-points.

Due to the visibility constraint imposed by the present procedure, which tends to force the superquadric to go behind the points as seen by the sensor, our method tends to extract a superquadric in such a way that the ‘hidden’ part of a segment with a convex-like shape (for example, the L-shape of a vehicle as perceived by a laserscanner) is filled by the superquadric shape. Conversely, in segments with a concave-like appearance (e.g., interior wall corners) the method tends to extract two superquadrics instead of one, as shown in Fig. 4 by the highlighted segments.

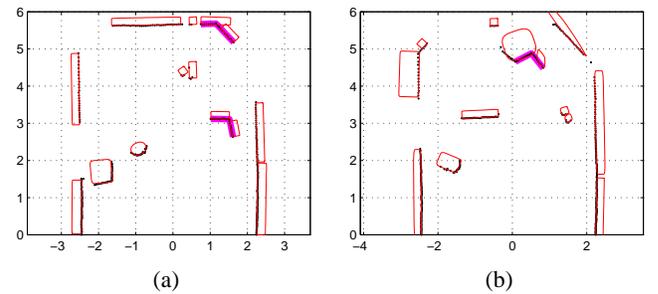


Fig. 4: Segmented indoor scenes with some concave regions.

IV. EXPERIMENTS

The numerical experiments reported in this article aim at demonstrating the performance of the proposed method in terms of segmentation and superquadric fitting to laser range data. To support analysis with real data, a set of laser scans collected

¹Nevertheless [26] studied superquadrics in a different application field.

in indoor and outdoor scenarios is used. Most of the data were obtained with our robotic vehicles, AtlasCAR [27] and the ISRobotCar [28], both shown in Fig. 5.



Fig. 5: AtlasCAR and ISRobotCar instrumented robotic vehicles.

A. Experiments on 2D laser-range data

We have evaluated our method in a set of 2D range-data collected from indoor and outdoor environments with laserscanners mounted on mobile platforms. The weighting parameters intrinsic to the S3F method, α_0 in (16) and α_i in (9), have been adjusted and fixed throughout the experiments as $\alpha_{0..3} = \{2, 1, 80, 30\}$. The inequality which is verified during optimization, comparing the cost function against a threshold, is invariant to a non-zero scale factor applied simultaneously to the explicit hyperparameters of (9) and the threshold. It is thus important to keep the ratio between those hyperparameters and the threshold Thr_G which provides the optimal results. The value of α_1 may be set to one and the other hyperparameters may be chosen as multiples of α_1 .

The ratio between α_2 and α_3 is especially important because α_2 is responsible for controlling the fit to the set of points while α_3 is responsible for recovering superquadrics which are consistent with the sensor viewpoint. It has been found through numerical experiments that $\frac{\alpha_2}{\alpha_3}$ close to 2.7 and α_2 close to 30 provide good results. The hyperparameter α_0 is important to the numerical stability of the optimization, and it should be chosen as large as possible to allow for a precise self-occlusion boundary but small enough to dampen unwanted oscillations in the search. This parameter controls the transition width of the visibility constraint and is not invariant to the above mentioned non-zero

scale factor. Before discussing the quantitative evaluation detailed in Section IV-B, it is appropriate at this point to present some qualitative analysis. This consideration comes from the fact that there are some interesting details of fitting superquadrics to real data that deserves some preliminary discussions. Macroscopic quantitative analysis tend to occlude some cases of difficult fitting which appear to be simple. For instance, in [4], peculiarities of real outdoor data are briefly discussed and, before presenting statistical results, we will further build upon those.

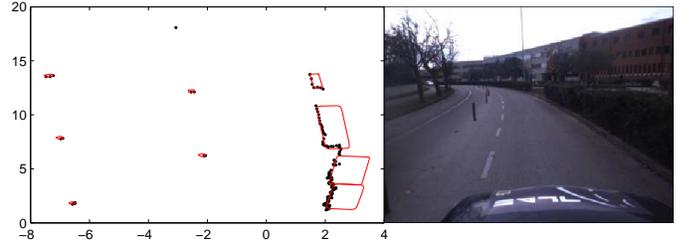


Fig. 6: Segmentation of a scene containing road, poles, trees and bushes.

Results obtained on range-data collected in urban-like scenarios are shown in Fig. 6 to Fig. 8. The scan appears to be noisy but mostly this is not due to intrinsic sensor noise, rather to characteristics of the objects being scanned and because of ego-motion, constituting one of the challenging aspects of real-world data. In Fig. 6, tree trunks and road poles have been properly segmented and the extracted superquadrics, which depend on the number of support points available for fitting, resemble rounded squares and ellipses. The road side bushes were also segmented and the openings between them have been correctly preserved. The S3F algorithm works from global to local and the final fit has only local support (*i.e.*, no verification is made with respect to occlusion between recovered superquadrics) moreover, line-segments do not constitute an ill posed problem since our volume minimization provides the necessary additional equations.

Figure 7 shows a road scene where a vehicle is approaching from the opposite direction with respect to the ego-vehicle. On the left, an overview of the scan and respective extracted superquadrics are depicted. On the bottom-right, a detailed view of the scene with the oncoming vehicle and some poles nearby are shown, giving evidence of the robustness of the proposed method under sparse data.

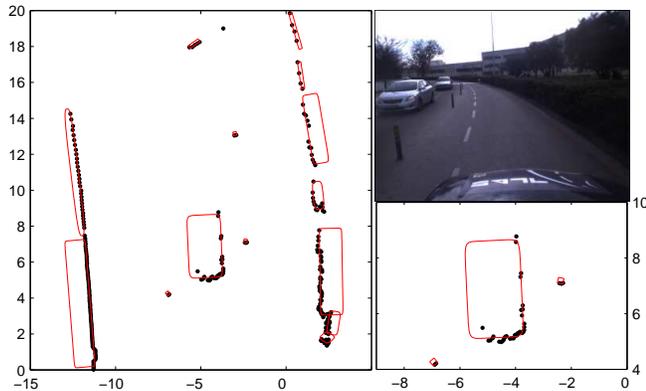


Fig. 7: An example of an approaching car in the opposite direction of the ego-vehicle.

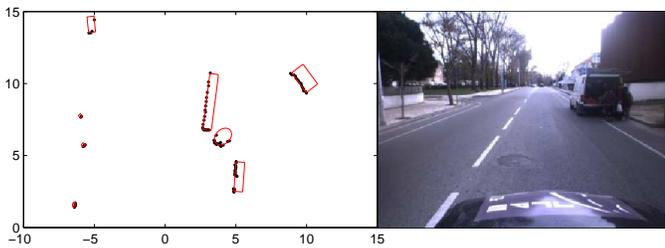


Fig. 8: Scene with pedestrians, vehicle, trees and sign poles.

The road scene of Fig. 8 is characterized by several road agents, some of them with partial data, including the presence of pedestrians in very close proximity. Typically, trees and lamp posts have been segmented and modeled by superquadrics resembling square shapes with rounded corners, while the vehicle and some nearby pedestrians have been segmented by a single superquadric.

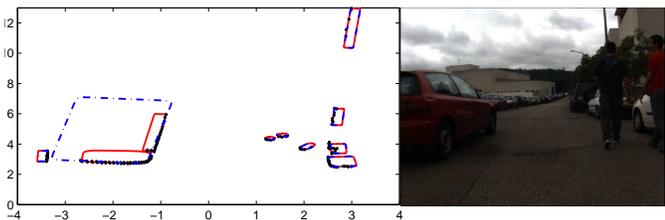


Fig. 9: On the left, superquadrics resulting from **Algorithm 1** with $Thr_G = 2$ (red solid line) and $Thr_G = 5$ (blue dashed line). On the right, an image of the scene.

Finally, a road scene with pedestrians and a moving vehicle (at the left) are shown in Fig. 9.

Using different values of Thr_G in the **Algorithm 1**, namely $Thr_G = 2$ and $Thr_G = 5$, we obtained the superquadrics in red-solid and blue-dashed lines, respectively. As Thr_G increases, the number of detected segments and respective superquadrics decreases, indicating a tendency of merging points. This behavior is intrinsic since high thresholds will tend to merge segments while lower thresholds have the opposite effect, establishing a tradeoff. It has been noticed, like in [4], that vehicles pose a particular problem to segmentation algorithms, normally leading to oversegmentation due to heavy discontinuity in the scan of a vehicle's fender areas.

B. Performance evaluation

The methodology used to quantitatively evaluate segmentation algorithms is described as follows. We have manually labeled 3098 segments in order to compose the set Γ . The labeling process was performed on the laserscanner Cartesian space with the aid of image-frames collected from the on-board cameras. For each scan s , the objects (segments) of interest were labeled according to the following criteria:

- 1) The objects of interest are: vehicles, poles, tree-trunks, sign-posts and pedestrians.
- 2) Points belonging to rigid body objects, such as vehicles, were labeled as a single object.
- 3) For any pair of clearly spaced groups of range-points that correspond to the legs of a given pedestrian, the pair is labeled as two independent segments. Otherwise, if they appear too close, a single segment is added to the ground-truth.
- 4) A group of objects appearing together in an image-frame and in the corresponding scan, especially people close to each other, was usually labeled by a number of segments less than the actual number of objects, since a more accurate evaluation could not be established.

Let Γ_s denote the set of labeled segments for a given scan s , where $n_s = |\Gamma_s|$ is the number of segments belonging to Γ_s ; wherein $|\cdot|$ denotes the cardinality of a set. Each labeled segment in Γ_s is denoted by S_i (with $i = 1, \dots, n_s$), where the number of range-points in each segment S_i is designated by T_i . Similarly, let K_s be the number of superquadrics extracted from s and M_j the number of points in a

given superquadric Q_j . For the purpose of evaluation, each superquadric is referred as a ‘segment’. For each segment we calculate $R_i = \sum_{j=1}^{K_s} |S_i \cap Q_j|$ as the total number of points of S_i which have the range-coordinates in Q_j . Thus, if there is a perfect match between S_i and Q_j , $R_i = T_i = M_j$, if S_i has no association with any Q_j , $R_i = 0$, while if there exists a partial-match $R_i < T_i$.

Let \mathcal{P}_i be the set of superquadrics which contains at least one point associated to S_i and $P_i = |\mathcal{P}_i|$ be the number of superquadrics matched to S_i . Denoting by M_i the total number of points associated with the superquadrics in \mathcal{P}_i thus, based on the above notations, we propose the following ‘performance measure’ calculated per each i^{th} -segment:

$$D_i = \frac{2R_i}{(M_i + T_i)P_i}, \quad (17)$$

where $D_i \in]0, 1]$. In (17), P_i penalizes oversegmentation since, if this happens, $\frac{1}{P_i} < 1$. On the other hand, undersegmentation and undetected fractions of ground truth affect (17) by $\frac{2R_i}{(M_i + T_i)} < 1$. The per-scan performance measure is expressed by

$$L_s = \sum_{i=1}^{n_s} \log(D_i) \quad (18)$$

and the global measure L , considering all scans of the dataset ($N=1081$), reduces to $L = \sum_{s=1}^N L_s$. The maximum value of (18) ($L_s = 0$) represents a perfect match, which means that each segment in Γ is represented unambiguously by a single superquadric. For purpose of comparison with other methods, the value of L should be normalized by the total number of segments in Γ (equivalent to using a geometric mean).

We benchmark the S3F method against two popular segmentation methods used by the robotic community and a state-of-the-art method. The first one is a simple but effective method designated Jump Distance Segmentation (JDS), which considers an approximation of the Euclidean distance $JD = |r_i - r_{i+1}|$ between consecutive range-points (r_i, r_{i+1}) and a threshold Thr_j . In the JDS method, a break-point is detected if $JD > Thr_j$. The second evaluated method, called KF-based Segmentation (KFS), uses the Kalman filter in conjunction with a statistical test, under a Chi-square validation region, to detect breakpoints. The stochastic model used to describe the spatial-dynamic evolution of the range

measurements as well the transition matrices are described in [29]. Basically, a breakpoint is detected if the *normalized innovation squared* exceeds a threshold Thr_χ according to a χ_1^2 distribution table. The model used in the KFS framework is not restrictive with respect to shape thus, it assumes a constant rate of change between the range-distance and the angle (also termed constant speed model). The third method considered here, called Segmentation using Invariant Parameters (SIP), has been recently developed by Fortin *et al.* [4] where the authors reported experimental results on vehicle detection. The SIP method was primary developed to deal with laser measurements in Polar coordinates. This approach is founded on the use of lines to model segmented objects, and leads to the definition of a criterion of line-segment detection that only depends on the sensor intrinsic parameters and range measurement noise. In [4], a confidence interval in a Mahalanobis-based merging procedure and a scene preservation distance are considered design parameters. Finally, after adjusting these design parameters by means of validation tests, we perform experiments with the SIP method considering the standard deviation (Thr_s) of range measurements as the variable parameter.

For $N=1081$ scans, Fig. 10 shows the values of L as function of normalized thresholds, being Thr_{maxL} the threshold which provides maximum value of L (the thresholds were normalized in order to align the maximum of each curve at a common abscissa). One may notice that all curves have identical trends, which is explained by the fact that distance-based criteria form the basis of the methods.

Furthermore, a variety of qualitative analysis have been performed based on performance measures summarized in Table I. The nomenclature used for indicating these evaluation measures is as follows.

- $N.Det.seg$ = number of detected (extracted) segments used to represent the total of 3098 labeled-segments of Γ .
- $N.Match$ = total number of extracted segments with unique and complete ground-truth correspondence.
- $N.Overseg$ = total number of over-segmentation instances.
- $N.Underseg$ = total number of under-segmentation instances.

The results presented in Table I and were obtained using thresholds whose values correspond to the

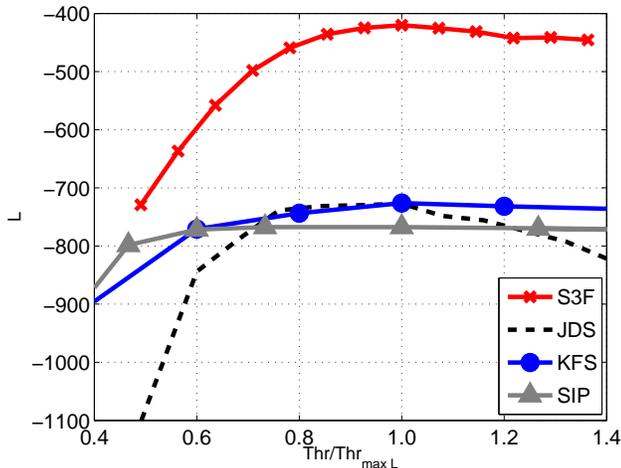


Fig. 10: Results of the global measure L as function of the normalized threshold for S3F, JDS, KFS and SIP methods.

maximum of the global measure L . The maximum values for each method correspond to the absolute thresholds $Thr_G = 5.5$, $Thr_J = 0.1$, $Thr_\chi = 2.0$ and $Thr_S = 0.3$, respectively for S3F, JDS, KFS and SIP. According to the results given in Table I, the S3F method outperforms all the aforementioned methods. The KFS slightly outperforms the JDS in terms of perfect match (given by $N.Match$), yet additionally providing less undersegmentation and better stability near the maximum L (Fig. 10), while the JDS method presents lower oversegmentation compared to the KFS method. Besides vehicles segmentation and detection, which was the primary application presented in [4], here SIP was benchmarked in a real-world dataset with a variety of objects with different sizes and shapes. Under these circumstances the performance of SIP was mainly penalized by its higher rate of oversegmentation, induced by laser-data missing returns and by intrinsic variability of objects forms and shapes.

Complementary results of the S3F method are provided in Fig. 11, with values of L shown as function of normalized hyperparameters α_2 and α_3 , chosen for their very distinctive roles in the optimization. Parameters were varied one at a time and normalized by $\alpha_{i_{max}L}$, the respective values of $\alpha_{i=2,3}$ which result in maximum L for $Thr_G = 5.5$. One can verify that α_3 has a strong influence on segmentation performance, where low values of α_3 alleviate visibility constraints at the cost of undersegmentation, while large values induce

oversegmentation in order to impose visibility.

TABLE I: Summary of the experimental results of the algorithms. The ground truth has 3098 segments.

Method	$N.Det.seg$	$N.Match$	$N.Overseg$	$N.Underseg$
S3F	3308	2270	309	142
KFS	3523	1949	555	203
JDS	3419	1853	458	222
SIP	3588	1332	615	222

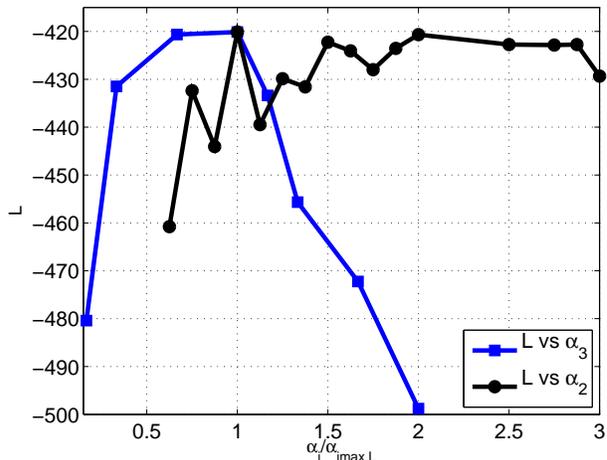


Fig. 11: Global measure L achieved by S3F as function of normalized α_2 and α_3 .

Finally, we report performance assessment based on precision-recall curves that characterize segmentation results in terms of over and undersegmentation [30]. In this work, we define precision Pr and recall Re measures by

$$Pr_{(Thr)} = \frac{N.Match}{N.Match + N.Overseg} \quad (19)$$

$$Re_{(Thr)} = \frac{N.Match}{N.Match + N.Underseg}, \quad (20)$$

where each value of (Thr) gives a point in the curve. We have varied the thresholds according to $Thr_G \in [0.1, 14]$, $Thr_J \in [0.005, 0.22]$, $Thr_\chi \in [0.005, 9.0]$ and $Thr_S \in [0.083, 3.3]$, with the resulting curves shown in Fig. 12. Among the methods, JDS was most sensitive to the increase of the threshold level (measured as a fraction of the optimal value) and thus it is more prone to low recall (high undersegmentation). On the other hand, compared to JDS, the KFS method presents better behavior. However, Thr_χ acts in a more complex manner than Thr_J . S3F exhibits robustness with respect to

precision and shows a more stable behavior around the optimal-operating point of the $Pr-Re$ curve (the upper right point), while SIP presents the highest stability near the optimal operating point.

All code has been implemented in script language and without exercising special care in respect to resource management (memory and calls to the processor). For timing purposes, the stopping criteria for the optimization in **Algorithm 1** have been set to 0.01 tolerance on estimated parameters (one centimeter in the case of size parameters) and the optimal threshold. The initial values were determined as discussed in Section III.

In these conditions, extracting up to 10 two-dimensional superquadrics using our method demands on average 1.5 s of CPU-time running in a single thread-single core (Intel T9300 2.5 GHz CPU, Ubuntu 12.04) in Matlab environment. However, most of the time of the algorithm is spent in operations whose number can be reduced if the implementation exchanges information on the breakpoints between iterations; in this case single threaded execution time is expected to fall to around 0.2 s. Further reductions require optimization of resources. Both the KFS and the JDS are several times faster. The version of SIP applied to our data runs in 0.05 s on average.

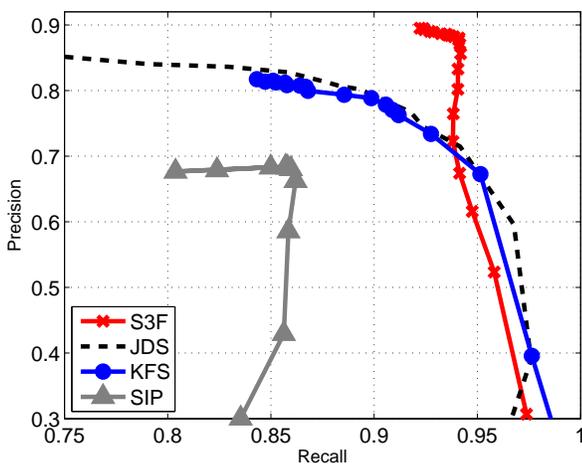


Fig. 12: Results of precision-recall for S3F, JDS, KFS and SIP methods.

C. Experiments on 3D laser-range data

Processing of 3D point clouds normally results in computer-intensive procedures. The projection of

3D point clouds onto a 2D plane can often be used as a means to reduce problem complexity and to make 2D superquadric based segmentation applicable as a pre-processing step for the 3D problem. This is the main reason why 2D representations for 3D scenes have been chosen and successfully applied in other studies, *e.g.* [31], [32].

In this section we have used 3D point clouds available from the KITTI dataset [33] and from an in-house platform that rotates a 2D range finder, as shown in Fig. 13. The 3D point clouds from the KITTI dataset were generated by a Velodyne sensor and have an intrinsic ordering while from the in-house setup they are not ordered.

Raw point clouds are sparse and noisy, the scenes themselves are unstructured and it becomes complex to fit general models for the objects. Direct superquadric fitting would be prone to errors and would be highly time consuming. To overcome these problems, we first project the 3D point-cloud onto a 2D plane and then apply the S3F algorithm. The decision on how to perform the projection, or if indeed there is a projection that greatly simplifies the segmentation, will depend on the type of scene and it is out of scope of this work to satisfy all situations.



Fig. 13: The in-house 3D laser platform.

We assume that the correct projections are vertical onto a plane that fits the ground. The plane is described by all points in space satisfying:

$$\mathbf{n} \cdot (\mathbf{x}_p - \mathbf{x}_0) = 0, \quad (21)$$

with \mathbf{n} the plane normal and \mathbf{x}_0 a point known to belong to that plane. Such point is chosen to be the intersection between the plane and the vertical axis of the reference frame moving with the vehicle (relative reference frame, *i.e.* x_3). Thus, there are three unknowns, two normal components, and one offset. The unknown unit normal components have

been chosen as n_1 and n_2 with n_3 assumed to be positive.

We start by determining the plane that best fits to the points on the ground, if they exist. Assuming that geometric moments and principal directions of sparse clouds, conditioned on partial information, should induce errors, we have decided to fit the ground-plane using nonlinear least squares and a cost G_{∇} similar to the self-occlusion proposed for the superquadrics. This cost states that the plane cannot have many scanned points under it (some assumptions have been made to account for sensor noise).

Once the plane is found, a 2D description is sought and required to be compatible with the concurrent extraction procedure. The 3D point cloud is projected vertically onto the previously determined plane and a synthetic 2D scan is obtained by using a grid procedure, similar to occupancy grids presented in [34], whose grid size depends on the smallest element we wish to detect. The grid size in our work has been chosen to be 5 cm.

Results are given here for a simple harbor scene scanned by the in-house scanner and for more complex scenes available from the KITTI dataset. 2D superquadrics are used to determine the points that will contribute to a 3D fit, by calculating the inside-outside function for all candidates and considering a specified neighborhood, which we have fixed arbitrarily at 0.1 m. In case overlapping exists, those overlapping points are attributed in a greedy manner to the largest superquadric. Since 2D superquadrics are a subset of the 3D, they are also used to estimate the initial parameters for the 3D. Fitting each 3D superquadric to centimeter tolerance takes, on average, 0.2 s on raw point clouds from the KITTI dataset.

In the top-right part of Fig. 14 (in a harbor environment), 2D extracted superquadrics are shown. Finally, the resulting 3D fit is shown on the bottom of the same figure. It can be seen that the 2D segmentation has been well performed and the 3D superquadrics are adequate to model the containers. Some results on 3D point clouds from the KITTI dataset are given in Fig. 15 and Fig. 16. An image of the scene is presented on the top, while the resulting 3D superquadrics are shown on the bottom part. In Fig. 15 not all vehicles have been correctly recovered, in particular the dark ones on the right side of the road, however, this is a problem on

the sensor side, since the light emitted by the laserscanner, and which then hits the dark vehicles, provides a very weak return signal, eventually not discernable from noise and tagged as missing.

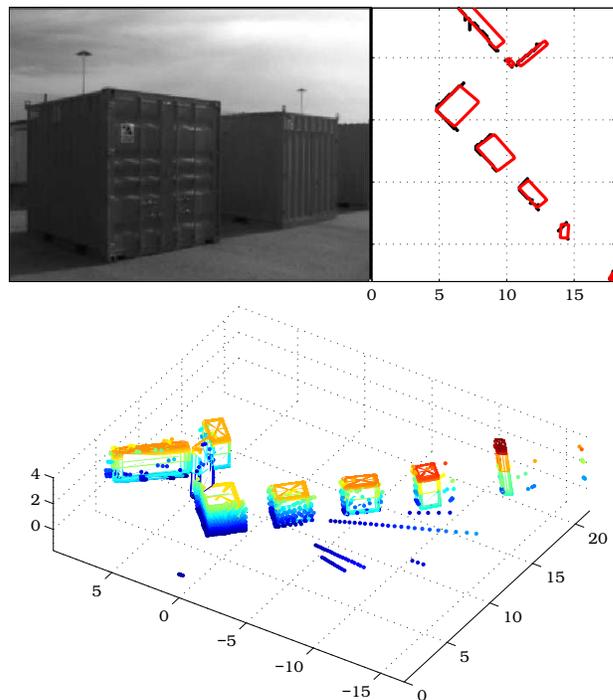


Fig. 14: Harbor scene as described by 2D and 3D superquadrics. In the left-down part, the extracted 2D superquadrics are shown, while the final 3D superquadrics are illustrated on the right.

V. CONCLUSION

In this paper we have proposed a method for multiple object segmentation and modeling, using superquadrics formulation, in range data collected from laserscanners mounted onboard ground-robotic platforms. The proposed method is directly applicable to 2D cases and is based on the minimization of an objective function and on segmentation criteria introduced in this work. The objective function accounts for the size of the recovered object, the distance between the recovered superquadric and the range points, and also for partial-occlusions. The criteria adopted for segmentation are based on the superquadric *inside-outside function* and its gradient. As consequence, the method preserves consistency between the formulation used for segmentation and fitting.

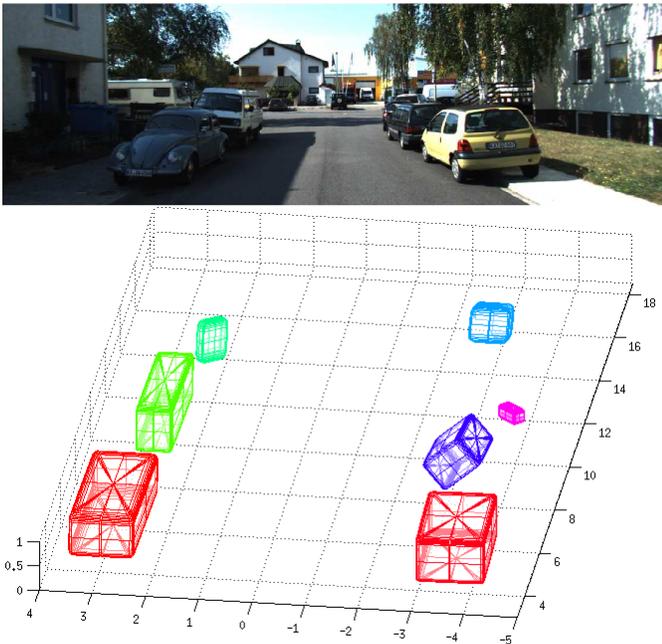


Fig. 15: Scene from the KITTI dataset with vehicles.

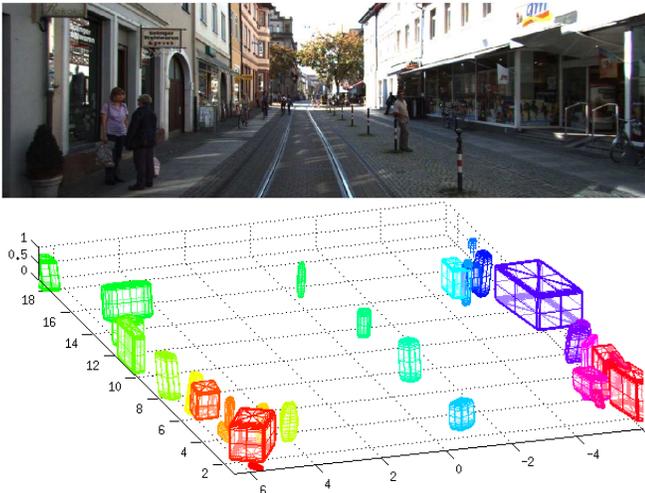


Fig. 16: City scene from the KITTI dataset with pedestrians, poles, bicycle and buildings.

Considering the results obtained on a labeled experimental dataset, comprising 2D range scans from indoor and outdoor environments, the segmentation aspect of the proposed method outperformed two popular and one state-of-the-art segmentation approach in terms of a variety of segmentation performance measures; particularly one, introduced in this work, which penalizes over and undersegmentation. Regarding object modeling, which is an integral part of the method, we can conclude that, at the cost of few additional shape parameters, superquadrics have

a strong capability of representing multiple shapes of interest within a unified functional approach.

Finally, the algorithm used in the 2D case has been applied as the primary preprocessing stage in 3D range data, preserving the consistency between 2D and 3D formulations and providing a convenient solution towards a more complete 3D object segmentation and modeling.

ACKNOWLEDGMENT

The first author would like to thank the Portuguese Foundation for Science and Technology (FCT) for funding under post-graduation grant SFRH/BPD/69555/2010 (Funds from POPH-QREN, co-financed by the European Social Fund and MEC National Funds). This work has also been supported by FCT and COMPETE program (grant PDCS10: PTDC/EEA-AUT/113818/2009). We would like to thank Dr. Benoît Fortin for sharing the original scripts of the SIP method with pre-clustering and merging. We would like to thank the anonymous reviewers, their comments have led to a significant improvement of the manuscript.

REFERENCES

- [1] S. Gidel, P. Checchin, C. Blanc, T. Chateau, and L. Trassoudaine, "Pedestrian detection and tracking in an urban environment using a multilayer laser scanner," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 579–588, 2010.
- [2] C. Prenebida and U. Nunes, "Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 371–384, 2013.
- [3] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Enhanced road boundary and obstacle detection using a downward-looking lidar sensor," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 3, pp. 971–985, 2012.
- [4] B. Fortin, R. Lherbier, and J.-C. Noyer, "Feature extraction in scanning laser range data using invariant parameters," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 9, pp. 3838–3850, August 2012.
- [5] S. Rodriguez, V. Fremont, P. Bonnifait, and V. Cherfaoui, "An embedded multi-modal system for object localization and tracking," *Intelligent Transportation Systems Magazine, IEEE*, vol. 4, no. 4, pp. 42–53, 2012.
- [6] J. Choi, J. Lee, D. Kim, G. Soprani, P. Cerri, A. Broggi, and K. Yi, "Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 974–982, 2012.
- [7] T.-D. Vu and O. Aycard, "Laser-based detection and tracking moving objects using data-driven markov chain monte carlo," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 3800–3806.

- [8] C. Premebida and U. Nunes, "Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications," in *In 5th National Festival of Robotics Scientific Meeting (Robotica 2005)*, Coimbra, Portugal, April 2005.
- [9] V. Nguyen, S. Gchter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, 2007. [Online]. Available: <http://dx.doi.org/10.1007/s10514-007-9034-y>
- [10] S. Zhang, M. Adams, F. Tang, and L. Xie, "Geometrical feature extraction using 2d range scanner," in *Control and Automation, 2003. ICCA '03. Proceedings. 4th International Conference on*, 2003, pp. 901–905.
- [11] A. H. Barr, "Superquadrics and angle preserving transformations," *IEEE Comput. Graph. Appl.*, vol. 1, no. 1, pp. 11–23, 1981.
- [12] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: the case for superquadrics with global deformations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 2, pp. 131–147, 1990.
- [13] D. Katsoulas, C. Bastidas, and D. Kosmopoulos, "Superquadric segmentation in range images via fusion of region and boundary information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 781–795, 2008.
- [14] R. Cipolla, B. Stenger, A. Thayananthan, and P. Torr, "Hand tracking using a quadric surface model," in *Mathematics of Surfaces, LNCS 2768*, 2003, pp. 129–141.
- [15] D. Conner, H. Choset, and A. Rizzi, "Towards provable navigation and control of nonholonomically constrained convex-bodied systems," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*, May 2006.
- [16] P. Drews, P. Nunez, R. Rocha, M. Campos, and J. Dias, "Novelty detection and 3d shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 3635–3640.
- [17] A. Jaklič, A. Leonardis, and F. Solina, *Segmentation and Recovery of Superquadrics*, ser. Computational imaging and vision. Dordrecht: Kluwer, 2000, vol. 20, ISBN 0-7923-6601-8.
- [18] T. I. Fossen, *Marine Control Systems - Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*. Trondheim: Marine Cybernetics, 2002.
- [19] A. H. Gross and T. E. Boulton, "Error of fit measures for recovering parametric solids," in *Second International Conference on Computer Vision*, 1988, pp. 690–694.
- [20] E. V. Dop and P. Regtien, "Fitting undeformed superquadrics to range data: Improving model recovery and classification," in *Internat. Conf. Computer Vision and Pattern Recognition*, 1998, pp. 396–401.
- [21] Y. Zhang, "Experimental comparison of superquadric fitting objective functions," *Pattern Recognition Letters*, vol. 24, no. 1, pp. 2185–2193, 2003.
- [22] G. Biegelbauer and M. Vincze, "Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 1086–1091.
- [23] L. Zhou, C. Kambhamettu, and R. Kambhamettu, "Extending superquadrics with exponent functions: Modeling and reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 73–78.
- [24] L. Zhou and R. Kambhamettu, "Representing and recognizing complete set of geons using extended superquadrics," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2002.
- [25] D. Skanda and D. Lebedz, "An optimal experimental design approach to model discrimination in dynamic biochemical systems," *Bioinformatics*, vol. 26, no. 7, pp. 939–945, 2010.
- [26] T. S. Bhabhrawala, "Shape recovery from medical image data using extended superquadrics," Master's thesis, Department of Mechanical and Aerospace Engineering, State University of New York at Buffalo, 2004.
- [27] V. Santos, J. Almeida, E. Avila, D. Gameiro, M. Oliveira, R. Pascoal, R. Sabino, and P. Stein, "Atlascar - technologies for a computer assisted driving system on board a common automobile," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, September 2010, pp. 1421 – 1427.
- [28] M. Silva, F. Moita, U. Nunes, L. Garrote, H. Faria, and J. Ruivo, "Isrobotcar: The autonomous electric vehicle project," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, October 2012, pp. 4233–4234.
- [29] G. A. Borges and M.-J. Aldon, "Line extraction in 2d range images for mobile robotics," *J. Intell. Robotics Syst.*, vol. 40, no. 3, pp. 267–297, Jul. 2004.
- [30] F. Estrada and A. Jepson, "Quantative evaluation of a novel image segmentation algorithm," in *Proceedings of IEEE Society Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, USA, 2005, pp. 1132–1139.
- [31] O. Wulf, D. Lecking, and B. Wagner, "Robust self-localization in industrial environments based on 3d ceiling structures," in *Proceedings of the 2006 IEEE International Conference on Intelligent Robots Systems*. IEEE, 2006, beijing, China.
- [32] D. Ferguson, M. Darms, C. Urmson, and S. Kolski, "Detection, prediction, and avoidance of dynamic obstacles in urban environments," in *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium*. Eindhoven: IEEE, June 2008, pp. 1149–1154.
- [33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [34] M. Himmelsbach, F. von Hundelshausen, and H.-J. Wünsche, "Lidar-based perception for offroad navigation," in *Proceedings of FAS 2008, Fahrerassistenzsysteme Workshop 2008*. Walting, Germany: C. Stiller and M. Maurer, April 2008.