Semantical Occupancy Grid Mapping Framework

Timo Korthals, Julian Exner, Thomas Schöpping, and Marc Hesse

Abstract— **In recent decades, mapping has been increasingly investigated and applied in unmanned terrain, aerial, sea, and underwater vehicles. While exploiting various mapping techniques to build an inner representation of the environment, one of the most famous remaining is occupancy grid mapping. It has been applied to all domains in a 2D/3D fashion for localization, mapping, navigation, and safe path traversal. Until now generally active range measuring sensors like LiDAR or SONAR are exploited to build those maps. With this work the authors want to overcome these barriers by presenting an occupancy mapping framework offering a generic sensor interface. The interface handles occupancy grids as inverse sensor models, which may represent knowledge on different semantical decision levels and therefore build up a semantic grid map stack. The framework offers buffered memory management for efficient storing and shifting and further services for accessing the 2D map stack in different cell-wise pre-fused and topometric ways. Within the framework, two novel techniques operating especially with occupancy grids are presented: First, a novel odds based interpolation filter is introduced, which scales grid maps in a Bayesian way. Second, a Supercell Extracted via Variance-Driven Sampling (SEVDS) algorithm is presented which, abstracts the semantical occupancy grid stack to a topometric map. While this work focuses on the framework's introduction, it is extended by the evaluation of SEVDS against state-of-the-art superpixel approaches to prove its applicability.**

I. Introduction

Mapping frameworks (MFs) are widely used within commercial products and build up the foundation in many services like navigation with Google Maps or process planning in farm management systems. Fig. 1 shows the principal concept of a MF, which consists mainly of the map storage. A map is a symbolic depiction which emphasizes relationships between elements of some space. These maps may have different representations depending on the application like topographic, metric, orthographic, etc. Data acquisition builds the map storage, whilst the preprocessing handles the transformation and embedding of incoming data. Services build up the link to other applications or clients, which invoke a processing on the maps due to their request. For the sake clarity, data acquisition and processing are explicitly depicted, but can be alternatively represented as a service as well. Within the authors' nomenclature of MFs, services are restricted to the derivation of alternative representations of stored maps which excludes navigation, task calculation, and others. Due to that definition, different mapping frameworks exist in robotics that model an inner representation of the world. Apparently, every known mapping algorithm is realized as a MF, which stores e.g. range readings into a data structure while the constructed map can be requested during navigation. The famous Simultaneous Localization and Mapping (SLAM) concept, for instance, can be interpreted as MF with feedback loop which updates a map hypothesis during data acquisition and loop closure.

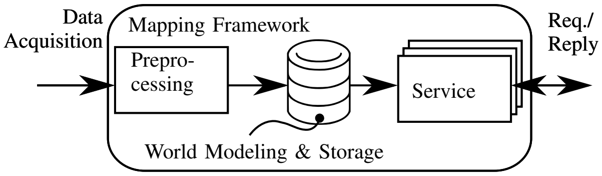

Fig. 1: Mapping framework concept

2D occupancy grids as statistical sufficient representation for an inner world model were introduced by Moravec and Elfes [1]. They have been widely applied and are still up to date for robotic mapping and SLAM applications due to their lean and efficient implementations. Semantical occupancy grid maps (SOGMs), also known as inference grids [2], [3], on the other hand got left behind since their first promotion. This could be of lacking applications or implementation complexity, because for each sensor detection algorithms, a mapping from raw sensor data to a 2D grid-based semantical interpretation of the environment has to be realized. On the other hand, nowadays enormous robotic sensor setups demand new approaches where SOGMs could be the next breakthrough.

With this definition of SOGMs and MFs in mind, the publication presents an Robot Operating System (ROS) implementation of a map centric, semantical occupancy grid mapping framework with generic interfaces to fuse inverse sensor models. First, related work is presented in Section II. In Section III the authors' framework is introduced where the occupancy mapping is depicted and generous sensor interfaces, interpolation, scan matching, services, and the implementation are defined. Section IV gives an evaluation for a supercell clustering service in a static table top scenario. Finally, a view on future applications in Section V is given where the proposed framework will be exploited.

II. Related Work

In the ongoing chapters the authors focus on robotic mapping applications for environment traversal, recapitulate common approaches and bring them into the given nomenclature.

Bielefeld University, Cluster of Excellence Cognitive Interaction Technologies, Cognitronics & Sensor Systems, Inspiration 1, 33619 Bielefeld, Germany, `http://www.ks.cit-ec.uni-bielefeld.de/`, {`tkorthals, jexner, tschoepp, mhesse`} `@techfak.uni-bielefeld.de`

A. Data Acquisition and Preprocessing

In the context of robotics achieving safe environment traversal, areas which are not occupied by an object are of interest. Active sensors concepts like LiDAR or SONAR, which are inherently able to detect the range to objects, are commonly used for this particular task [4]. Furthermore, passive and indirect measurements by cameras, in a stereo or inverse perspective mapping setup, became famous in automotive applications due to their inexpensiveness [5]. Range measurements are preprocessed, such that they are first transformed into the proper coordinate system of the map. After this, an inverse sensor model is applied, which is the deduction from measurements to causes [6].

B. World Modeling

The world modeling storage focuses the format in which the information is obtained. Nüchter et al. sorts maps used in robotics for representing environments in the following descending order from level of detail to compactness [7]: *Grid maps* spatialize the environment into an equidistant grid which can be stored in arrays, or more efficiently in sparse environments in quad- or octal-trees. Grid maps on one hand are limited in environmental representation by their resolution which influences memory consumption drastically, but their non-parametric property makes them ideal for mapping unstructured objects. *Feature maps* represent obstacles via geometric primitive descriptions, e.g. lines, planes, or cubes. Topometric maps are *hybrid maps* consisting of a topological structure with direct spatial relationship. The highest compactness and therefore greatest abstraction is achieved by *topological maps* where no more spatial but logical relations exist.

C. Services & Existing Mapping Frameworks

Services are fundamental extensions to the mapping approach. They offer any subsequent application the interface to access the mapped data in specific ways. For a navigation approach on unstructured environments commonly a metric grid map is necessary and therefore, a service which formats the map data in such way is recommended.

Applicable mapping frameworks (MF) exist and are generally implemented within the Robotic Operating System (ROS) middleware [8]. The most common approach for the map storage and for incorporating new scans is the occupancy grid map (OGM) algorithm based on grid maps in a 2D or 3D fashion [1]. Due to that, the authors focus on application which implemented that approach.

A popular ROS package that works with a grid-based occupancy map representation is the *costmap_2d* for incorporating range measurements and offering the result to the ROS 2D navigation stack [9]. *Karto SDK* is an commercial but educationally freely available MF which incorporates mapping and additionally localization, path planning, and exploration while interfaces for other robotic middlewares (MS Robotics Studio and Robot OS) exist [10]. Kohlbrecher et al. provide a comprehensive ROS package called *hector_slam* which is maintained by Team Hector [11]. It comprises the main application *hector_mapping* which realizes a SLAM algorithm. Various subpackages extend the mapping as services, e.g. map retrieval (*hector_map_server*) or trajectory creation (*hector_trajectory_server*), which are build upon that. *Gmapping* realizes a Rao-Blackwellized particle filter for SLAM and allows access of the map via a service [12]. All former approaches exploit 2D grid maps while they are mainly designed for autonomous ground (AGV) or surface (ASV) vehicles. 3D OGM is commonly applied in autonomous aerial vehicles (AAV). A famous 3D occupancy mapping framework is *OctoMap* by Hornung [13]. More recently Droeschel et al. [14] used occupancy grid mapping to fuse multiple sensor modalities into one representation to achieve robust mapping in AAVs. Multimapping approaches have been done by Morris et al. [15] handling one static and multiple temporal maps, and Frankhauser and Hutter handling multiple elevation maps [16].

Fundamentally, all named application share the same inner representation structure which is a two dimensional OGM approach for multiple reasons: First, grid maps can be stored, searched, and accessed very efficiently by computer systems. Second, the OGM algorithm introduced in III-A allows a robust and efficient online sensor fusion and map updating.

III. Semantical Occupancy Grid Mapping Framework

Within this publication, three challenges are faced: transfering the sensor detections into a map representation of the vehicle's environment, storing, and service advertisement. First, the semantical occupancy grid mapping approach as a map server is introduced in Section III-A. Second, the generic interface definition by inverse sensor models (ISM) is further explained in Section III-B. More deeply, Section III-C proposes a novel interpolation technique for OGMs/ISMs in Section III-C.1 and scan matching in Section III-C.2. Services are introduced for accessing the current map data in a raw, fused, or topometrical way in Section III-D.1 and III-D.2. Finally, Section III-E gives the functional description of the map server application in ROS.

A. Semantical Occupancy Grid Mapping

Two-dimensional occupancy grids were originally introduced by Elfes [17]. In this representation, the environment is subdivided into a regular array or a grid of quadratic cells. The resolution of the environment representation directly depends on the size of the cells. In addition to this discretization of space, a probabilistic measure of occupancy is associated with each cell. This measure takes on any real number in the interval $[0, 1]$ and describes one of the two possible cell states: unoccupied or occupied. An occupancy probability of 0 means definitely unoccupied space, and a probability of 1 means definitely occupied space. A value of 0.5 refers to an unknown state of occupancy.

An occupancy grid is an efficient approach for representing uncertainty, fusing multiple sensor measurements on the decision level, and to incorporate different sensor models [5]. To learn an occupancy grid M given sensor information

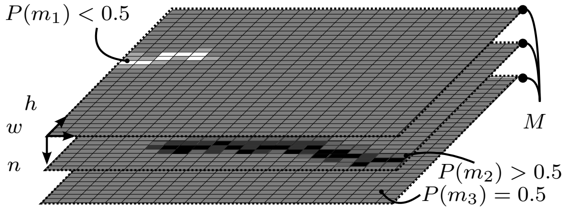

Fig. 2: SOGM with $N = 3$ layers

z, different update rules exist [18]. For the author's approach, Bayesian update rule is applied to every cell $m \in M$ at position (w, h) as follows: Given the positions x_t of a vehicle at time t, let $x_{1:t} = x_1, \ldots, x_t$ be the positions of the vehicle's individual steps until t, and $z_{1:t} = z_1, \ldots, z_t$ the environmental perceptions. Occupancy probability grids determine for each cell m of the grid the probability that this cell is occupied by an obstacle. Thus, occupancy probability grids seek to estimate

$$P\left(m|z_{1:t}, x_{1:t}\right) = \mathrm{Odd}^{-1}\left(\prod_{t=1}^{T} \underbrace{\frac{P\left(m|z_t, x_t\right)}{1 - P\left(m|z_t, x_t\right)}}_{\mathrm{Odd}(P(m|z_t,x_t))}\right) \quad (1)$$

This equation already describes the online capable, recursive update rule that populates the current measurement z_t to the grid, where $P\left(m|z_{1:t}, x_{1:t}\right)$ is the so called inverse sensor model (ISM). The ISM is used to update the OGM in a Bayesian framework, which deduces the occupancy probability of a cell, given the sensor information.

The extension to semantical occupancy grid maps or inference grids is straightforward and defined by an OGM M with W cells in width, H cells in height, and N semantical layers:

$$M : \{1, \ldots, W\} \times \{1, \ldots, H\} \rightarrow m = \{0, \ldots, 1\}^N \quad (2)$$

Compared a single layer OGM which allows the classification into three classes $\{\text{occupied}, \overline{\text{occupied}}, \text{unknown}\}$, the SOGM supports a maximum of $\left|\{\text{occupied}, \overline{\text{occupied}}, \text{unknown}\}\right|^N = 3^N$ different classes allowing much higher differentiability in environment and object recognition. The corresponding ISMs are fused via the occupancy grid algorithm in their nth associated semantical occupancy grid. The implementation of a semantical map stack as shown in Fig. 2 is further called *map server* (MS).

B. Inverse Sensor Model Handling

The ISM is the preprocessed sensor data originating from one or more sensors. It maps from causes to reasons so that information resides on a decision level (cf. [19]) as occupancy probability at the particular cell. It is commonly used for sensors with a planar sensor lobe, oriented parallel to the ground. In that case, a quite simplistic model can be applied, e.g. for a laser range finder. Each cell m that is covered by the beam of the observation z and whose distance to the sensor is shorter than the measured one, is supposed to be unoccupied. The cell in which the beam ends (the measurement point) is supposed to be occupied, and everything behind is unknown [20]. For generic implementations, however, sensors like cameras LiDAR and RADAR may be non-planary installed and thus their sensor lobes are tilted with respect to the map. Each sensor-algorithm combination requires its own ISM, converting from the algorithm's output to a 2D measurement grid representation. An ISM approach for monocular/stereo cameras, depth cameras, proximity sensors, RADAR, LiDAR and their corresponding algorithms has been published by Kragh et al. [21].

C. Preprocessing

Incorporating new sensor measurements on ISM level needs to be handled in two dependent ways: First, the resolution of the ISM is interpolated to meet the MS. Second, the ISM is matched and therefore transformed by a homography into the MS frame.

1) Interpolation: Kohlbrecher et al. states, that the discrete nature of occupancy grid maps limits the precision that can be achieved and also does not allow the direct computation of interpolated values or derivatives [11]. For this reason an interpolation scheme, allowing sub-grid cell accuracy, is necessary for estimating occupancy probabilities. Intuitively, the grid map cell values can be viewed as samples of an underlying continuous probability distribution which can be defined in various ways. A naive approach might be the nearest-neighbor interpolation. In [11], a bilinear filter is introduced for subsampling OGMs. Other approaches using higher polynomials should not be applied to OGM interpolation, due to their value range definition outside of the sampling points. This can lead to values outside of $(0, 1)$ which needs to be truncated or handled otherwise.

In this paper, a new interpolation method called *biodds interpolation* is proposed where it is assumed that not the occupancy values are linearized between the sampling points, but the amount of sensor readings. This way, the nature of the OGM algorithm in conjunction with the probability values, which are clearly non-linear as stated by Hähnel [18], is respected. To achieve this, an interpolation function is derived from a two-parametric odds model:

$$\widehat{\mathrm{Odd}}\left(R; \epsilon_0, \epsilon_1\right) = \mathrm{Odd}\left(\epsilon_0\right) \mathrm{Odd}\left(0.5 + \epsilon_1\right)^R \quad (3)$$

Defining $\epsilon_0 > 0$ and $1 - \epsilon_0$ as the lower and upper bounds and $\epsilon_1 > 0$ as the information increment, Equation 3 can be evaluated over $R \in [0, \log_{\mathrm{Odd}(0.5+\epsilon_1)}(\mathrm{Odd}(1 - \epsilon_0)/\, \mathrm{Odd}(\epsilon_0))]$ measurements. For the sake of clarity the value range R is mapped to $\hat{x} \in [0, 1]$. Fortunately, the given odds model can be approximated by function f with a vanishing error by a one-parametric hyperbolic function:

$$f(x; \alpha) = 0.5 \tanh((x - 0.5)\alpha) + 0.5 \quad (4)$$

Wrapping everything up, the interpolation m of point $\mathbf{P}$ can be calculated by applying the approximator f as depicted in

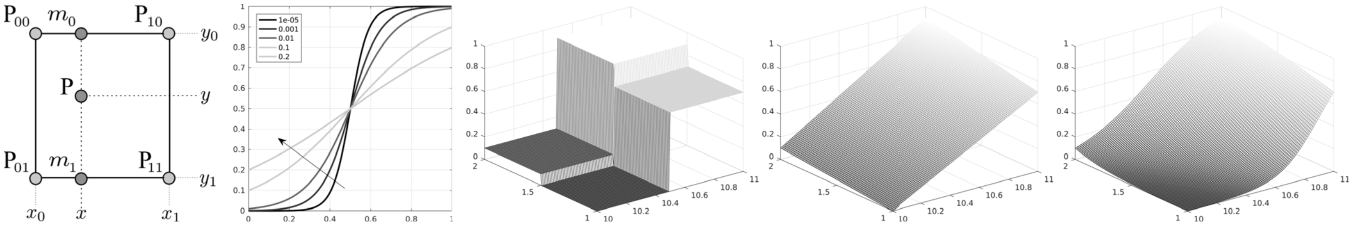

Fig. 3: From left to right: Interpolation point on a grid, odds interpolation for different ϵ_0, nearest neighbour, bilinear, and biodds interpolation

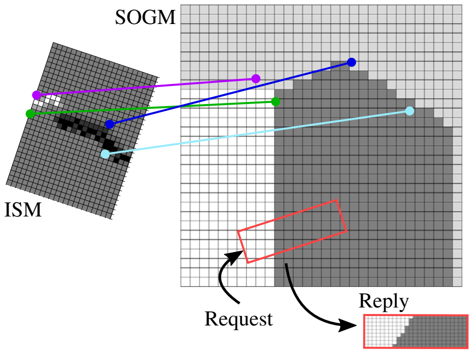

Fig. 4: Qualitative homography for a missaligned ISM into its corresponding SOGM and request of a sub map with different resolution.

Fig. 3 (left):

$$m_1 = f\left(\frac{x-x_0}{x_1-x_0} f^{-1}(m(\mathrm{P}_{11})) + \frac{x_1-x}{x_1-x_0} f^{-1}(m(\mathrm{P}_{01}))\right)$$
$$m_0 = f\left(\frac{x-x_0}{x_1-x_0} f^{-1}(m(\mathrm{P}_{10})) + \frac{x_1-x}{x_1-x_0} f^{-1}(m(\mathrm{P}_{00}))\right)$$
$$m(\mathrm{P}) \approx f\left(\frac{y-y_0}{y_1-y_0} f^{-1}(m_1) + \frac{y_1-y}{y_1-y_0} f^{-1}(m_0)\right) \tag{5}$$

Fig. 3 depicts the different interpolation models for a grid consisting of four cells. It is worth noticing that the two extrema $\epsilon_0 \to 0$ and $\epsilon_0 \to 0.5$ of the odds interpolation leads to either the nearest neighbor or the bilinear interpolation with respect to the boundaries (cf. Fig. 3).

2) Scan Matching: Scan matching is the process of aligning ISMs with each other or with an existing SOGM. This is necessary while ISMs can be produced in another coordinate frame than the one of the map, or the ISMs are misaligned due to poor odometry, registration, or ISM algorithm design. The first issue can be solved by knowing the registration between the map and the sensor frame and then applying an homography to the ISM. The latter one can be solved similar, but the missing transformation needs to be recovered. This is done by utilizing ORB feature extraction of the ISM and the corresponding SOGMs surrounding [22]. The features are matched via the distance of pairwise feature candidates which again leads to a homography that is depicted in Fig. 4.

D. Services

Services are defined, such that they process the requested type, pose, and resolution as shown in Fig. 4. Pose and resolution are features to facilitate requesting applications to outsource such post processing to the MS. Therefore, it is uniquely handled and benefits from the proposed interpolation technique from Section III-C.1.

1) Cell Wise Retrieval: Requesting preprocessed SOGMs is beneficial for applications only requiring one kind of information which can be derived from a set of SOGMs. Therefore, the authors implemented, besides the raw access of SOGMs, also the two following fusion techniques among layers: The first approach is based on a super Bayesian independent opinion pooling P_{B} [23]. It is applicable for the case when separate SOGMs with identical feature representations (e.g. set of maps for class "obstacle") are maintained. Second, a non-Bayesian fusion maximum pooling method P_{M} is applied to heterogeneous feature representations s(e.g. set of maps with varying classes). The fusion techniques are cell-wise and therefore do not introduce any clustering.

$$P_{\mathrm{B}}(m) = \frac{1}{1+\prod_N \frac{1-P(m_n)}{P(m_n)}}, \quad \tilde{P}_{\mathrm{M}}(m) = \max_n P(m_n) \tag{6}$$

2) Superpixel Clustering: Unlike single layer OGM approaches, an SOGM incorporates multiple OGMs with varying classes residing in the map storage. For further applications, respecting every grid cell is not a feasible approach due to noise, sparse data, or offset between the layers. Even worse, high bandwidth would be necessary for requesting a raw SOGM. To transform the SOGM into a parametrized form by clustering e.g. via Gaussians is unfeasible as well, due to the risk of lacking objects. The authors' approach is therefore a superpixel-like clustering inspired by computer vision to find homogeneous regions and assigning a feature vector for these. This leads to a topometric map, which is derived from the centroids of the superpixels as shown in Fig. 6. Utilizing the Superpixels Extracted via Energy-Driven Sampling (SEEDS) algorithm from Van den Bergh [24] the authors revise the formulation

$$H(s) = D(s) + \gamma G(s), \tag{7}$$

to respect the nature, s.t. probability and locality of information of SOGMs more precisely. In Equation 7 s is the superpixel of interest, D is the color distribution term and G is the contour function which can be smoothed via the scalar factor γ. As stated in the original paper a drawback of the color distribution term is the lack of respecting the variance of color values inside a superpixel. But especially for SOGM clustering applications, where data represents probabilities, this must be considered. For instance, superpixels consisting of contradicting values greater and less than 0.5 inside one layer should be avoided.

The authors introduce therefore the *Supercell Extracted via Variance-Driven Sampling* (SEVDS) algorithm, which substitutes the color distribution term of SEEDS by a variance driven formulation D'. The distribution term D' of a supercell c is defined as the sum of Eigenvalues e of the covariance matrix C of the probability histogram $h(c)$:

$$D'(c) = \sum_{n=1}^{N} e_n(\operatorname{var}(h(c))) \tag{8}$$

Each supercell c groups the cells $m \in c \subset M$ into a arbitrary but connected shape as depicted in Fig. 5. The probability range $[0, 1]$ is divided into K^N bins of size $1/K$ along every principal dimension. Thus, a bin $b_{k_1,k_2,\dots,k_N} \in \mathbb{N}$ (with $k_n = 1, \dots, K$) of a N dimensional histogram $h \in \mathbb{R}^{k\times k \dots \times k}$ for one particular supercell c is calculated by

$$b_{k_1,\dots,k_N} = \sum_{m\in c}\prod_{n=1}^{N}\Pi\left(\frac{m_n - \frac{1+2(k_n-1)}{2K}}{K}\right) \tag{9}$$

with m_n addressing a cell of the nth semantical layer and Π being the boxcar function. From that histogram, the covariance matrix $\operatorname{var}(h(c)) \in \mathbb{R}^{N\times N}$ is calculated. Further, from that covariance matrix, the N Eigenvalues e_n are calculated. Their sum is the quantifier of the distribution of occupancy probabilities in the corresponding supercell. This procedure becomes intractable with $\mathcal{O} = (K^2)^N$ for even a few semantical layers, but fortunately this derivation can be simplified twice so that the calculation becomes linear with $\mathcal{O} = KN$. First, the sum of Eigenvalues is equal to the trace and therefore no Eigenvalue decomposition needs to be performed. Second, while the covariances are no longer of interest due to the trace, the calculation of variances $\operatorname{var}(h(c_n)) \in \mathbb{R}$ of the marginal histogram is sufficient. The marginal histogram is nothing but the histogram of the a single semantic layer n in the supercell c_n (cf. Fig. 5). Finally, the distribution term can be simplified to

$$D'(c) = \sum_{n=1}^{N} e_n(\operatorname{var}(h(c))) = \sum_{n=1}^{N} \operatorname{var}(h(c_n)) \tag{10}$$

Using the variance of histograms as a distribution measurement becomes intuitive by taking the differences of probabilities in one supercell into account. Thus, the distribution of probabilities along the semantical layers is marginally respected. But with increasing contradictions along the spatial resolution the distribution term maximizes. Therefore, D' needs to be minimized to find the best supercell. This is implemented just like done by the SEEDS in a greedy fashion: For every supercell c an adjoining block c' with $\hat{c} = \{c_l, c'\}$ is taken into account for the distribution calculation.

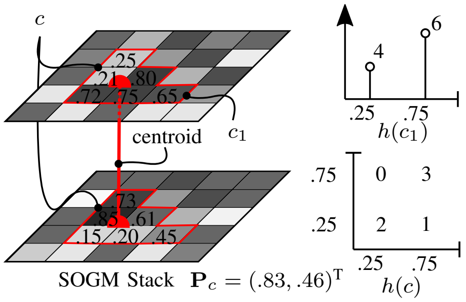

Fig. 5: Supercell with $N = 2$ layers and corresponding histrograms with $K = 2$ bins.

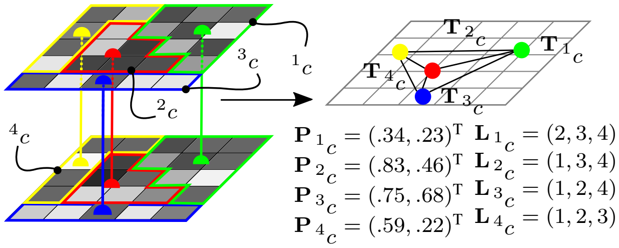
${}^{3}c$
${}^{1}c$
${}^{2}c$
${}^{4}c$
$\mathbf{T}_{{}^{2}c}$
$\mathbf{T}_{{}^{1}c}$
$\mathbf{T}_{{}^{4}c}$
$\mathbf{T}_{{}^{3}c}$
$\mathbf{P}_{{}^{1}c} = (.34, .23)^{\mathrm{T}}$ $\mathbf{L}_{{}^{1}c} = (2, 3, 4)$
$\mathbf{P}_{{}^{2}c} = (.83, .46)^{\mathrm{T}}$ $\mathbf{L}_{{}^{2}c} = (1, 3, 4)$
$\mathbf{P}_{{}^{3}c} = (.75, .68)^{\mathrm{T}}$ $\mathbf{L}_{{}^{3}c} = (1, 2, 4)$
$\mathbf{P}_{{}^{4}c} = (.59, .22)^{\mathrm{T}}$ $\mathbf{L}_{{}^{4}c} = (1, 2, 3)$

Fig. 6: Conversion of supercells to a graph of centroids labeled with feature vectors.

If $D(\hat{c}) < D(c)$ holds, the newly $\hat{c}$ will be used further. As depicted in Fig. 6, for every found supercell a tripel $\mathcal{C} = (\mathbf{T}_c, \mathbf{L}_c, \mathbf{P}_c)$ consisting of its centroid location $\mathbf{T}_c$, a list of adjunct supercell $\mathbf{L}_c$, and a feature vector $\mathbf{P}_c$ is calculated

$$\operatorname{Odd}(\mathbf{P}_c) = \left(\prod_{m\in c_1} \operatorname{Odd}(P(m)), \dots, \prod_{m\in c_N} \operatorname{Odd}(P(m))\right)^{\mathrm{T}} \tag{11}$$

Experiments revealed that supercells tend to grow uncoordinatedly. This was caused by neighbouring supercells comprising homogenous regions which leads to a total variance of zero. This can be explained by taking blocks of cells away from those zero-variance supercells their variance remains zero, while the other supercells' variance may decrease. This corner case can be circumvented by not allowing superpixels with a total variance of zero to shrink.

E. Mapping Framework Implementation in ROS

Fig. 7 depicts the composition of the mapping framework (MF) habitate which consists mainly of two components: The tile publisher keeps track of all necessary transforms of region of interest (ROI). The map server fuses incoming ISMs, offers services and debug messages.

An instance of the map server subscribes to all topics which needs to be mapped. For every topic a double buffer holding the current and last SOGM stack is allocated. To build a generic ISM interface, all messages have to have the ROS type `nav_msgs/OccupancyGrid`. It is worth mentioning that sensor fusion on the OGM level can directly be applied by letting different nodes publish their ISMs to the same topic. But further, one has to respect differing publishing frequencies and weightings which can distort a

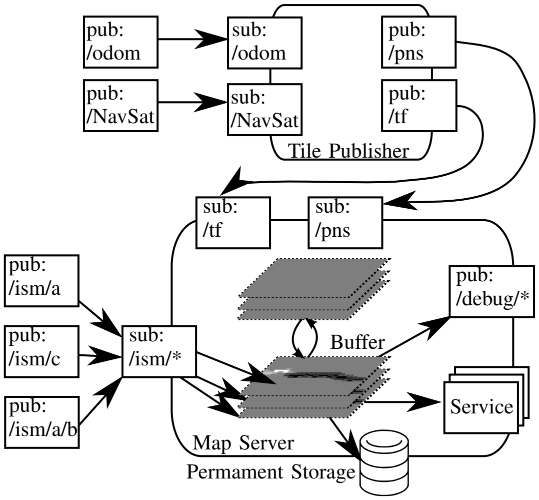

Fig. 7: Mapping framwork implementation

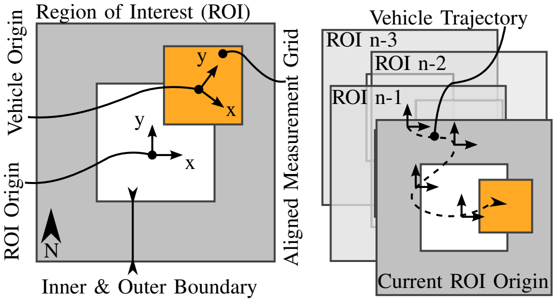

Fig. 8: 2D map centric mapping approach. The ROI comprises the horizontal and lateral ($H \times W$) dimesion of SOGM.

balanced fusion. While the amount of sensors measuring exteroceptive features is increasing and with them the derived ISMs, keeping track of all topics can be unfeasible. To overcome the unmanageable number of topics to which the map server has to subscribe to, a new scoping technique inspired by Wienke et al. [25] is introduced. It is build upon the common ROS pub./sub. model. Rather than subscribing to every single topic explicitly, the map server can be configured by a parent topic (e.g. /ism). All child topics (e.g. /ism/a, /ism/a/b, /ism/c) will be subscribed on demand after they have registered at the ROSMASTER.

In conjunction with the map server node, a tile publisher node keeps track of all map transformations depicted in Fig. 8. The tile publisher publishes transforms from the parent coordinate system to the origin of the current map tile. As the vehicle moves through the world, it will at some point exceed the current map dimension. Therefore, if an inner boundary dimension is passed, a new transform for a new map tile is published to ROS's tf tree. As proposed by *REP-105: Coordinate Frames for Mobile Platforms*, all sensor fusion has to be done in the local frame of a vehicle, because it is not affected by any discrete jumps caused by e.g. GPS. Therefore, all map tiles reside in the local frame, but are referenced in any global frame for later georeferencing or postprocessing. Just shifting the current map coordinate frame is not feasible, because discontinuities and jumps of frames are not well handled by ROS's tf API and therefore, new transforms with a new frame names are necessary. To inform the map server, first the transforms to the new map are published on the /tf scope. After a configurable delay a Pose/NavSat/String tuple (/pns) is published to inform about the current tile pose in the odometry frame, its NavSat datum for global localization, and a string containing the name of the current tile transform in ROS's tf tree. Finally, as the map server node receives a pns-tuple the double buffer of SOGMs is swapped so that the content can be independently stored in binary form, tagged with kind of content, resolution, and location to a permanent storage.

IV. Testcase and Evaluation

The proposed SEVDS is evaluated and compared against other superpixel algorithm implementations included in OpenCV 3.2.0. As metrics Intra-Cluster Variation, Explained Variation, Undersegmentation Error (UE) and Boundary Recall (BR) [24] are applied. Additionally, a pureness metric is defined as

$$g(s) = \frac{1}{|s|} \sum_{s_j} \frac{|\{m \in s_j | I(s) = \text{Label}(s_j)\}|}{|s_j|} \tag{12}$$

with $\text{Label}(s_j) = \text{argmax}_i |\{s \in s_j | I(s) = i\}|$. The Pureness of a segmentation is defined by the average pureness of its superpixels. As proposed by Stutz et al., parameters were optimized such that $(1 - \text{BR}) + \text{UE}$ is minimized. Input data originates from simulating the Autonomous Mini Robot (AMiRo) via ROS and Gazebo in a table-top scenario inspired by RoboCup@Home competitions [26], [27] (cf. Fig. 9 top). The AMiRo maps its environment with known poses, a tilted laser rangefinder and a RGB camera using ISMs from [28]. The corresponding mapping for the enlarged test case scenario in Fig. 9 is depicted in Fig. 10. The composed SOGM is interpreted as multi-channel image and provides the input for all evaluated clustering algorithms presented in this work. A ground truth segmentation was provided by a human labeled segmentation (cf. Fig. 9 bottom). Fig. 12 to Fig. 16 compare the results of the algorithms using above-mentioned metrics for all four different scenarios (scenarios are depicted in different colors). In Fig. 11 the clustering is qualitatively depicted. Compared to SEEDS, the proposed SEVDS let the supercells not degenerate that much in object border regions and therefore provides more distinctive border segmentations. Other differences can barely be noticed. All algorithms achieve very similar Pureness. It should be noted that Pureness and Explained Variation are displayed with an offset to illustrate small differences in these metrics. SEVDS produces segmentations with noticeable, significant lower Intra-Cluster Variation (ICV), due to the fact that ICV's definition is very similar to the variance distribution term SEVDS aims to minimize. Explained Variation intents to measure boundary adherence independent of human annotated ground truth (higher is better). SEVDS's segmentation yields slightly lower Explained Variation. On average the Undersegmentation Error of SEVDS is 3.41

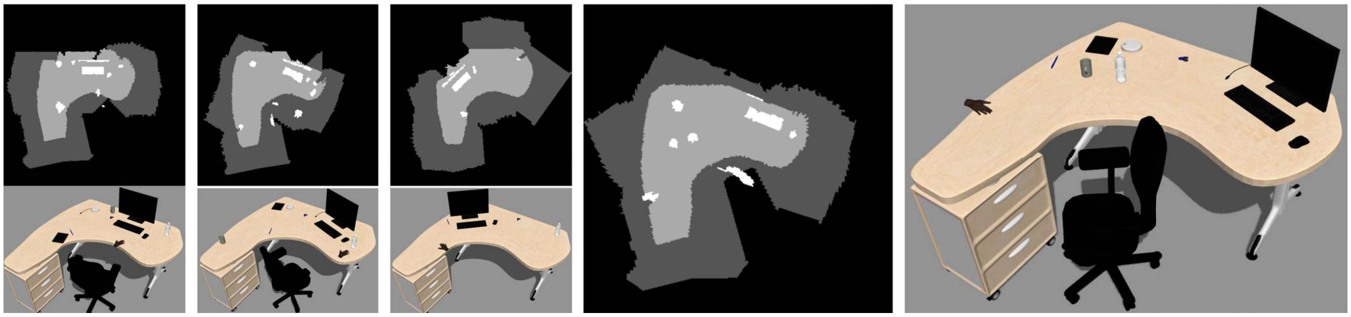

Fig. 9: Four randomized table-top scenarios used for recording maps and their corresponding ground truth map segmentation with four greyish shaded classes: unknown, floor, table, obstacle. Scenarios from left to right: first, second, third, forth.

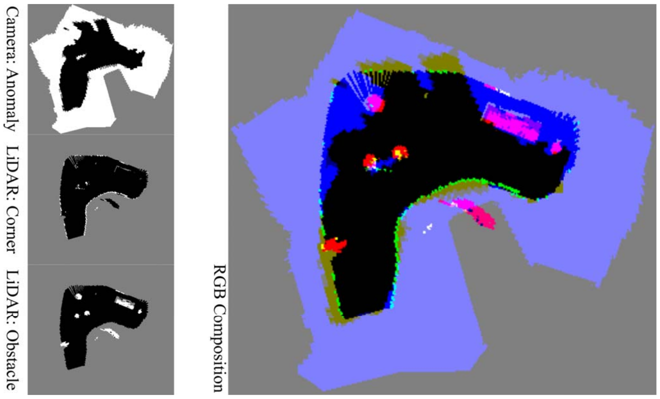

Fig. 10: Single SOGM layers produced by the "sensor: classification" set (left) for the fourth example in Fig. 9 and the composition in RGB color space that depicts the $|\{\text{occ.}, \overline{\text{occ.}}, \text{unknown}\}|^{\#\text{layer}} = 27$ possible classes.

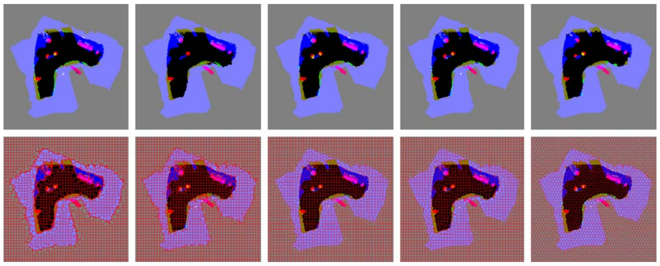

Fig. 11: Clustering of SOGM resulting from the fourth scenario (c.f. Fig. 10). From left to right: SEEDS, SEVDS, LSC, SLIC, SLICO. Top: Coloring of clusters with class majority. Bottom: Corresponding cluster segments in red.

times higher than SEEDS's. But it should be noted that the general power of all Undersegmentation Errors is pretty low (10^{-2}) and that this metric substantially depends on the provided ground truth. SEVDS achieves slightly lower Boundary Recall compared to SEEDS, but better results than the other algorithms. To conclude, SEVDS generates low Intra-Cluster Variation superpixels, with state-of-the-art Pureness and Explained Variation, but improvable boundary adherence. Anyway, this is an outstanding result which is beneficial for the author's definition on topometric features from Equation 11. Having low variation means that the derived feature vectors are highly separable. On the other hand, having low variation would result in resembling features among classes which would be hardly discriminable. A noteworthy implementation detail is, that these clustering algorithms now work on grids which have a spatial relation. Therefore, as a rule of thumb the minimal initial superpixel edge length should be of the size of smallest object to be found (s.t. 8 cm $\widehat{=}$ 8 pixel for the current evaluation).

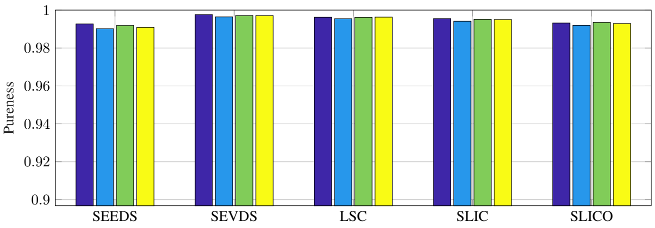

Fig. 12: Pureness

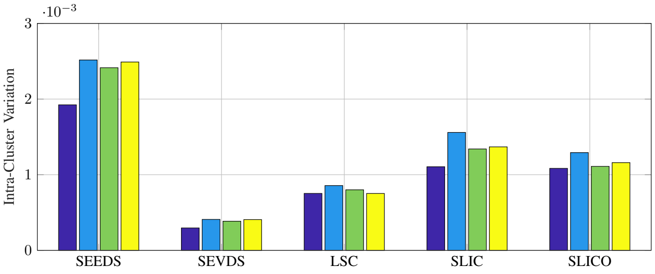

Fig. 13: Intra Cluster Variation

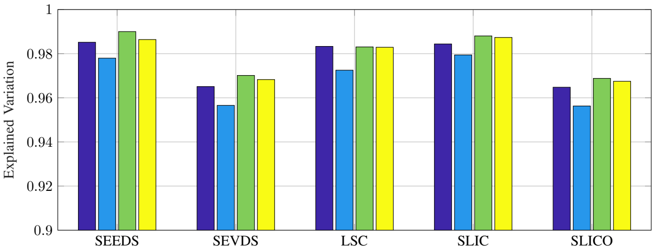

Fig. 14: Explained Variation

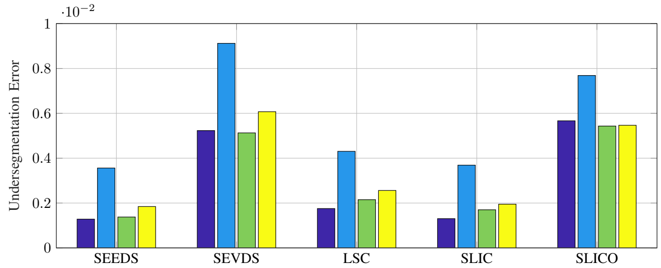

Fig. 15: Undersegmentation Error

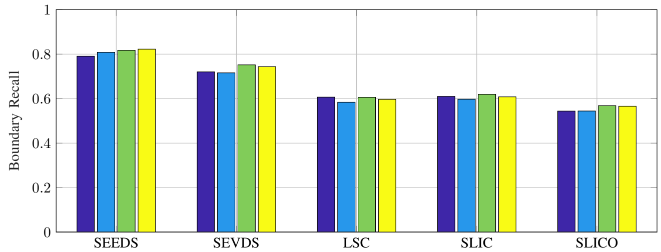

Fig. 16: Boundary Recall

V. Conclusion and Outlook

The authors present a complete semantical occupancy grid mapping framework (MF) which offers new possibilities of mapping sensor readings on a decision level in a generic way. Further, a new interpolation method has been proposed to incorporate sensor readings in a Bayesian way. Finally, services are introduced which refine multi layered, semantic maps to a single layer, or via the newly proposed SEVDS algorithm to a topometric map. Thus, this MF facilitates the possibility to handle SOGMs by standard navigation techniques. Ongoing investigations will exploit the proposed MF in dynamic agricultural and multi robotic applications to evaluate its mapping capability and quality. Further research will concentrate on applying optimized navigation techniques on SOGMs. Finally, it is intended to publish the MF as a ROS package on `https://opensource.cit-ec.de/`.

Acknowledgment

This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG) and by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalenLippe" (it's OWL) and managed by the Project Management Agency Karlsruhe (PTKA).

References

[1] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985.

[2] A. Elfes, "Dynamic control of robot perception using multi-property inference grids," 1992.

[3] ——, "Robot navigation: Integrating perception, environmental constraints and task execution within a probabilistic framework," *Lecture Notes in Computer Science*, vol. 1093, pp. 93–130, 1996.

[4] S. Thrun, "Robotic Mapping: A Survey," *Science*, vol. 298, no. February, pp. 1–35, 2002.

[5] H. Winner, *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Wiesbaden: Vieweg+Teubner Verlag, 2015.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Mass.: MIT Press, 2005.

[7] A. Nüchter, "Semantische dreidimensionale Karten für autonome mobile Roboter," Ph.D. dissertation, University of Bonn, 2006.

[8] A. Koubaa, *Robot Operating System (ROS): The Complete Reference*. Springer International Publishing, 2016, vol. 1, no. 1.

[9] E. Marder-Eppstein, D. V. Lu, and D. Hershberg, "costmap_2d." [Online]. Available: http://wiki.ros.org/costmap{_}2d

[10] KARTO, "KARTO," 2017. [Online]. Available: https://www.kartorobotics.com/

[11] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," *9th IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2011*, pp. 155–160, 2011.

[12] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, 2013.

[14] D. Droeschel, M. Nieuwenhuisen, M. Beul, D. Holz, J. Stöckler, and S. Behnke, "Multilayered Mapping and Navigation for Autonomous Micro Aerial Vehicles," *Journal of Field Robotics*, vol. 33, no. 4, 2016.

[15] T. Morris, F. Dayoub, P. Corke, G. Wyeth, and B. Upcroft, "Multiple map hypotheses for planning and navigating in non-stationary environments," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2765–2770, 2014.

[16] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," *Studies in Computational Intelligence*, vol. 625, pp. 99–120, 2016.

[17] A. Elfes, "Occupancy Grids: A Stochastical Spatial Representation for Active Robot Perception," in *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, 1990.

[18] D. Hähnel, "Mapping with Mobile Robots," Ph.D. dissertation, University of Freiburg, 2004.

[19] M. E. Liggins, D. L. Hall, and D. Llinas, *Handbook of multisensor data fusion*, 2001.

[20] C. Stachniss, "Robot Mapping Features vs . Volumetric Maps Grid Maps Example."

[21] M. Kragh, P. Christiansen, T. Korthals, T. Jungeblut, H. Karstoft, and R. N. Jørgensen, "Multi-Modal Obstacle Detection and Evaluation of Occupancy Grid Mapping in Agriculture," in *International Conference on Agricultural Engineering*, Aarhus, 2016.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.

[23] K. Pathak, A. Birk, J. Poppinga, and S. Schwertfeger, "3D Forward sensor modeling and application to occupancy grid based sensor fusion," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp. 2059–2064, 2007.

[24] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An Evaluation of the State-of-the-Art," *CoRR*, vol. abs/1612.0, 2016.

[25] J. Wienke and S. Wrede, "A middleware for collaborative research in experimental robotics," *2011 IEEE/SICE International Symposium on System Integration (SII)*, pp. 1183–1190, dec 2011.

[26] S. Herbrechtsmeier, T. Korthals, T. Schöpping, and U. Rückert, "AMiRo: A Modular & Customizable Open-Source Mini Robot Platform," in *ICSTCC*, 2016.

[27] S. Meyer zu Borgsen, T. Korthals, F. Lier, and S. Wachsmuth, "ToBI Team of Bielefeld: Enhancing Robot Behaviors and the Role of Multi-Robotics in RoboCup@Home," 2016.

[28] T. Korthals, T. Krause, and U. Rückert, "Evidence Grid Based Information Fusion for Semantic Classifiers in Dynamic Sensor Networks," *Machine Learning for Cyber Physical Systems*, vol. 1, no. 1, p. 6, 2015.