2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Using Recursive Spectral Registrations to Determine Brokenness as Measure of Structural Map Errors

Andreas Birk
Dept. of Computer Science, Jacobs University Bremen, 28759 Bremen, Germany
a.birk@jacobs-university.de
http://robotics.jacobs-university.de

Abstract—**There are many common error sources that influence mapping, e.g., salt and pepper noise as well as other effects occurring quite uniformly distributed over the map. On the other hand, there are also errors, which occur very rarely but with severe effects. These errors influence not only the local accuracy but the overall spatial layout of the map. Concrete examples include bump noise in the robot's pose or residual errors in Simultaneous Localization and Mapping (SLAM). Brokenness is presented here to capture one form of structural errors in grid maps. Concretely, brokenness measures the degree with which a map can be partitioned into regions that are locally consistent with ground truth but "off" relative to each other. The concept of brokenness is presented in a formal way and it is shown how it can be computed in an efficient way using recursive spectral registrations. Experimental results show that the metric can indeed be used to automatically determine one structural quality aspect of a map in a quantitative way.**

I. Introduction

Given the important of mapping for mobile robotics (1), it should be of strong interest to be able to assess the quality of maps or mapping approaches in profound ways. According assessments can also be used for execution monitoring, i.e., fault detection and isolation of core functions among which mapping clearly belongs for mobile robots (2).

The benefits of a mapping algorithm or system are often presented through differences in the underlying theory, run times, and at most a qualitative visual assessment of map quality, e.g., by displaying the generated maps and a comparison basis like the results of other approaches or a ground truth plan. Surprisingly often, even the necessary references for this qualitative comparison, i.e., results from other approaches or ground truth, are omitted.

The problem can of course be simply seen from a task oriented viewpoint as formulated in (3): "We feel the ultimate test of a map is not 'does it look good?' but 'how accurately, adequately or quickly can X be performed with it?'". There is no general quality of a map or of a mapping algorithm from this viewpoint. Each user should simply run individual benchmarks based on whatever task X the user is employing the map for. There is a grain of truth in this view, but it is not a sufficient motivation to completely abandon to assess the quality of maps, respectively mapping algorithms in general ways.

There are several simple, general criteria for the quality of a map M. One is for example its *coverage*, i.e., the amount of the ground truth environment that is represented in M (4). Further general criteria for map quality (4) are its *level of detail* and its *accuracy*. The level of detail is in the predominant form of map representation, namely 2D occupancy grids (5; 6), simply determined by the resolution of M. Its coverage is determined by the number of cells that contain information. There hence remains the problem of measuring accuracy.

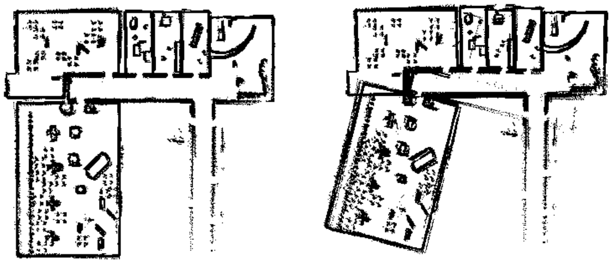
One option is to use a comparison between the ground truth trajectory of the robot and its estimate by a Simultaneous Localization and Mapping (SLAM) algorithm, respectively multiple SLAM algorithms to measure the capabilities of the algorithms (7). But ground truth measurements of the robot's trajectory are not trivial to do; they require a perfect localization system. Ground truth grid maps can be generated from several sources - including ground truth trajectories and range sensor reading - and can hence be considered as a more general case.

Given a ground truth or other reference representation R, one option is to estimate the accuracy of a map M by relating the content of each cell in M to the content of the corresponding cell in R, e.g., by using cross entropy (8) as a correlation estimate. But cross entropy only takes information from co-located cells in M and R into account. The map therefore has to be very similar to the reference to give a meaningful estimate. Cross entropy fails to provide information about the map quality in case of any larger disturbances.

A way to overcome this disadvantage is to use a measure of distance between similar points in M and R like Least Mean Squares of Euclidean Distances (LMS-ED). A disadvantage of LMS-ED is that it is expensive to compute. It is therefore not convenient to use it for whole grid maps and it is hence often applied to smaller sets of landmarks (9; 10). An alternative to LMS-ED is to use Manhattan Distances. As shown in (11), this leads to a fast computation of a map quality, which provides quantitative assessments of the level of noise in the M. Concretely, this map quality metric shows decreasing values for increasing amounts of common error sources like salt and pepper noise and the effects of translation, rotation, and scaling.

The above methods only address errors that mainly occur uniformly distributed over the map. A form of structural error is addressed here, namely the "brokenness" of a map or the degree with which a map can be partitioned into submaps that are "off" with ground truth relative to each other. Figure 1 shows a typical example. Suppose map (a) represents ground truth. Map (b) is broken in the sense that a complete partition, namely the room shown in the bottom left in this map, is rotated with respect to ground truth. When considering only this particular partition, i.e., the submap that covers this room, it is consistent within itself, i.e., it makes so to say perfect sense with respect to the according part in ground truth. But this submap is assessed to be completely false when applying for example a cell based correlation metric to measure the quality of the overall map.

The term *structural* error is used here for faults in the map that affect its global spatial layout. The alternative term "topological" is omitted for two main reasons. First of all, "topological" may be misleading with respect to the representation that is used. Topological maps in form of graphs explicitly represent a spatial structure. In grid maps, the spatial structure is only in the eye of the human observer, the underlying data structure is nothing but a uniform grid. Second, topological maps are usually based on higher level spatial information, i.e., the building blocks of the map have a semantic quality like "room", "corridor", etc. This is not the case for grid maps. It is mere coincidence that a whole room is "broken" in figure 1. Structural errors in grid maps usually do not correspond with any semantically meaningful part of the map.



(a) undisturbed map (b) "broken" map

Fig. 1. A typical example of a "broken" map: a complete partition of map (b) - the room on the bottom left - is "off" by an angular error compared to (a) as a representation of ground truth.

We consider brokenness to be an important, if not even the most important form of structural error in grid maps. The source of brokenness is a form of error, which occurs very regularly in robot mapping but which is usually neglected, namely bump noise. In contrast to Gaussian noise on sensor readings, bump noise occurs only very rarely, i.e., there is a usually very small probability p_B that at time t the value $v_B(t)$ of this noise is non-Zero. Gaussian noise is in contrast almost always non-Zero, i.e., at every moment t it is superimposed to the sensor readings. Despite its rare occurrence, bump noise has severe effects as it can lead to large "jumps" in sensor readings. In the context of mapping, bump noise can be modeled as

$$v_B(t) = (\theta, d) = \begin{cases} \theta \in \mathcal{U}[0, 2\pi], d \in \mathcal{U}[0, r] & \text{with prob.} p_B \\ (0, 0) & \text{with prob.} 1 - p_B \end{cases}$$

where θ is an angle and d is a translation by a certain distance that are both superimposed on the estimated location of a robot at time t, and $\mathcal{U}[c_1, c_2]$ is a Uniform distribution of samples over the interval $[c_1, c_2]$. Roughly speaking, $v_B(t)$ changes the estimated orientation of the robot by an arbitrary angle and shifts its estimated position within a certain radius r on rare occasion that have probability p_B to occur; but most of the time there is no effect of $v_B(t)$.

Typical sources of bump noise in mapping are slipping wheels or tracks when using odometry or the false result of a scan matcher. One may argue that state of the art Simultaneous Localization and Mapping (SLAM) algorithms are well suited to compensate this kind of error (see (12; 1) for a general overview of SLAM). But SLAM requires loop closing for this purpose and one can not assume that all parts of all maps are always a part of a loop. And even if this would be the case, it is not unlikely that a residual error remains after relaxation or filtering in the loop closing process. Please note that a single - potentially residual - bump error in just one localization estimate among the thousands that form the basis of a map can lead to the severe effects of brokenness illustrated in figure 1.

The rest of this paper is structured as follows. In section 2, the formal definition of brokenness as introduced in (13) is presented, which is supplemented by a general algorithmic way to compute the degree of brokenness in section 3. This paper then extends the own work in (13) by introducing in section 3 a new, more efficient way to compute brokenness, namely by using a spectral registration method. Experiments and results are presented in section 4. It is shown there with several maps from simulation as well as real world data that the correct degree of brokenness can indeed be computed in an efficient way with recursive spectral registration to assess the structural quality of a map. Section 5 concludes the article.

II. A formal definition of Map Brokenness

We deal here with the predominant form of maps as 2D arrays. It is assumed that each cell of a map contains binary information, which can be seen as a predicate p that is fulfilled or not for this cell. A typical example for this predicate is *occupancy*, i.e., the fact whether a location represented by a cell corresponds to free space or not. Please note that all concepts presented here can be easily extended to more general forms of maps where the cells contain more detailed information like probabilities or pixel values. But as a matter of convenience, we consider a map $\mathbf{M}$ of size $k_x \times k_y$ to consists of cells $\mathbf{m}(x, y)$ with

$$\mathbf{M} = \{\mathbf{m}(x, y) \mid \forall_{1 \le x \le k_x, 1 \le y \le k_y}\ \mathbf{m}(x, y) = p\ \vee\ \mathbf{m}(x, y) = \neg p\}$$

Based on this definition of a map $\mathbf{M}$, it is possible to consider the set $\mathbf{L^M}$ of locations (x, y) where the predicate p, for example occupancy, is fulfilled:

$$\mathbf{L^M} = \{(x, y) \mid \mathbf{m}(x, y) = p\}$$

It is obvious that there is a canonical way to get $\mathbf{M}$ from a given $\mathbf{L^M}$. In the following, we will only deal with sets of locations $\mathbf{L^M}$ with property p and use this as a synonym for map. A spatial transform $\mathcal{T}_{\theta,d}$ with a rotation θ followed by a translation d on a map $\mathbf{L^M}$ is defined as:

$$\mathcal{T}_{\theta,d}(\mathbf{L^M}) = \{(x', y') \mid \forall (x, y) \in \mathbf{L^M} : (x', y') = \mathbf{t}_{\theta,d}(x, y)\ \}$$

where $\mathbf{t}_{\theta,d}(x, y)$ is the spatial transform of point $(x, y) \in \mathbb{R}^2$ with a rotation by angle $\theta \in [0, 2\pi]$ followed by a translation by distance $d \in \mathbb{R}$.

The operator $\stackrel{\mathcal{T}_{\theta,d}}{\to}$ applied to two maps L^1 and L^2 denotes the set union of the points in L^1 with the points in L^2 after a non-trivial spatial transformation of the later ones, i.e.,

$$\mathbf{L^1} \stackrel{\mathcal{T}_{\theta,d}}{\to} \mathbf{L^2} \mathrel{\hat{=}} \mathbf{L^1} \cup \mathcal{T}_{\theta,d}(\mathbf{L^2})$$

with $\theta \neq 0 \vee d \neq 0$.

Based on the above notations, the brokenness of a map $\mathbf{M}$ can be defined as follows (13):

<u>Definition</u>: A map $\mathbf{M}$ is broken with degree $n_{BN} \geq 1$ with respect to a reference $\mathbf{R}$ if

$$\exists_{0 \le i \le n_{BN}}\ \mathbf{L^i},\ \exists_{1 \le j \le n_{BN}}\ \mathcal{T}^j\ \text{ with}$$

$$\mathbf{L^M} = \mathbf{L^0} \cup \mathbf{L^1} \cup \ldots \cup \mathbf{L^n_{BN}}$$

and

$$\mathbf{L^R} = \mathbf{L^0} \stackrel{\mathcal{T}^1}{\to} \mathbf{L^1} \ldots \stackrel{\mathcal{T}^{n_{BN}-1}}{\to} \mathbf{L^n_{BN}}$$

The degree n_{BN} of brokenness can intuitively be motivated as follows: it counts the numbers of portions into which a map $\mathbf{M}$ is partitioned due to structural errors. Each $\mathbf{L^i}$ is a consistent submap that is locally correct in the sense that it can be perfectly aligned with the corresponding part in the reference map $\mathbf{R}$. But to get a globally correct map, the portions $\mathbf{L^i}$ have to be spatially transformed to compensate the structural errors. The reference map $\mathbf{R}$ is usually a ground truth representation. This case is also referred to as the general brokenness of a map. As a matter of convenience, ground truth is assumed to be the reference if nothing else is explicitly mentioned.

Without loss of generality, it can be assumed that the "sub-maps" are proper partitions that can be ordered by their cardinality #, i.e., it can be assumed that

$$\forall i \neq j : \mathbf{L^i} \cap \mathbf{L^i} = \emptyset$$

and

$$\#\mathbf{L^0} \geq \#\mathbf{L^1} \geq ... \geq \#\mathbf{L^n_{BN}}$$

This property can be used to define more fine grain metrics for brokenness that take the sizes of the different partitions into account. Consider for example the case when two maps $\mathbf{M_1}$ and $\mathbf{M_2}$ are both broken with the same degree k. It can then be of interest to compare the size of the largest consistent partition $\mathbf{L^0_1}$ of $\mathbf{M_1}$ with the size of the largest consistent partition $\mathbf{L^0_2}$ of $\mathbf{M_2}$. Furthermore, it can be of interest to apply a norm to the parameters of the spatial transforms $\mathcal{T}$ needed to get the partitions into a globally consistent map. This norm can then be used to assess how much the different partitions are "wrong" with respect to each other.

A general algorithmic approach to compute the degree of brokenness n is presented in the following section. This generic algorithm also leads to a determination of the proper partitions - and hence their sizes - as well as the underlying spatial transforms between them.

Algorithm 1 The general algorithmic approach to determine the degree of brokenness n of a map $\mathbf{M}$ with respect to a reference $\mathbf{R}$.

1: $T = L^{\mathbf{M}}$
2: $n = 0$
3: **while** $T \neq \emptyset$ **do**
4:
5:    *< register T with reference $\mathbf{R}$ >*
6:    find ${}^{n}\mathcal{T}_{\theta,d}()$: max $\Psi_1({}^{n}\mathcal{T}_{\theta,d}(T), R')$ for $R' \subseteq L^{\mathbf{R}}$
7:    $T = {}^{n}\mathcal{T}_{\theta,d}(T)$
8:
9:    *< mask out the part of T that is well aligned with $\mathbf{R}$ >*
10:    find max $\mathbf{L^n} \subseteq T$: $\Psi_2(\mathbf{L^n}, R'') \leq c$ for $R'' \subseteq L^{\mathbf{R}}$
11:    $T = T \setminus \mathbf{L^n}$
12:
13:    *< increment the degree of brokenness >*
14:    $n = n + 1$
15:
16: **end while**

III. Computing Brokenness with Spectral Registration

A general algorithmic approach to compute the brokenness of a map can be based on the idea to employ map merging to determine the different partitions in a map with respect to a reference map. Map merging deals with the fusion of two maps based on the detection of "identical" regions in the maps where they can be "glued" together. Map merging is for example of interest in the context of multi robot mapping (14; 15; 16; 17; 18; 19; 20; 21; 22).

Map merging is also related to image registration, i.e., the search process to find a template in an image (23; 24; 25; 26). Though the task of map merging is harder than just registration. Note that instead of locating a known template in an image, an unknown region of overlap has to be identified in two maps for map merging. This is more comparable to image stitching (27), which is for example used to generate panoramic views from several overlapping photographs. Solutions to solve this problem in image processing usually need common reference points that are identified using local image descriptors (27; 28; 29).

But occupancy grids usually lack rich textures like photographs. Image stitching techniques can hence not be applied in a straightforward manner. But there are several recent advances in map merging (30; 31; 32; 33) that exactly provide the means for registering the partitions as needed here. Please note that the general formulation of the computation of brokenness is not dependent on a concrete map merging algorithm and that there are several possible choices for implementing it.

Given a map $\mathbf{M}$ and a reference map $\mathbf{R}$, e.g., ground truth, we will now present a way to compute the degree of brokenness n_{BN} as formally defined in section II.

In the most general way as formulated in algorithm 1, a metric $\Psi_1(X_1, X_2)$ is needed to determine the "similarity" between two raster data sets X_1 and X_2. The registration step in algorithm 1 is used to determine a spatial transform $\mathcal{T}_{\theta,d}()$, which finds the best, i.e., most similar, match of a partition in the reference $\mathbf{R}$ with a partition of the transformed map $\mathcal{T}_{\theta,d}(T)$ where T in the beginning holds a copy of $\mathbf{M}$. Roughly speaking, one can imagine that T is "moved around" such that an as large as possible region in it matches with a corresponding region in the reference $\mathbf{R}$.

In the following step, the matching regions are so to say masked out from the maps T and $\mathbf{R}$. Again, a similarity metric is used, here denoted with $\Psi_2()$ as it may be identical with $\Psi_1()$ but it does not have to be. In the previous registration step, the largest partition that is consistent with the reference is determined in the sense that it gets registered by a transform $\mathcal{T}()$ with the corresponding region in the reference $\mathbf{R}$. Cells $m(x, y)$ in this matching partition of $\mathcal{T}(T)$ have hence a high similarity to co-located cells $r(x, y)$ in $\mathbf{R}$. This property is now used to so to say remove this partition $\mathbf{L^n}$ from the current T and to then apply the registration and masking on the remainder $T \setminus \mathbf{L^1}$.

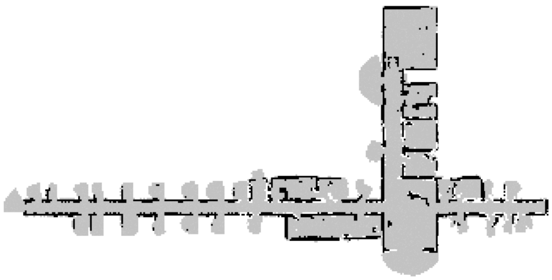
The general principle of computing brokenness can be achieved with fundamentally different implementations of the registration and masking steps. In (13), where we introduced the general idea of brokenness, a map merging algorithm introduced in (32) was used for the steps in lines 5 to 7 of algorithm 1. This map merging algorithm builds upon a (dis)similarity measure of 2D raster data sets using Manhattan distances between nearest neighbors with the same cell property (34) combined with an Adaptive Random Walk (35; 36; 37) to stochastically search the space of possible transformations. In addition, a correlation metric is used to avoid convergence to local optima.

Though this approach works reasonably well and is fully sufficient to demonstrate the general idea of computing brokenness, it has two significant disadvantage from a practical viewpoint. First of all, it is based on a stochastic search method that may not always lead to the same results for the same input data. Second, it is relatively slow, i.e., it requires a few dozen seconds up to several minutes to compute brokenness.

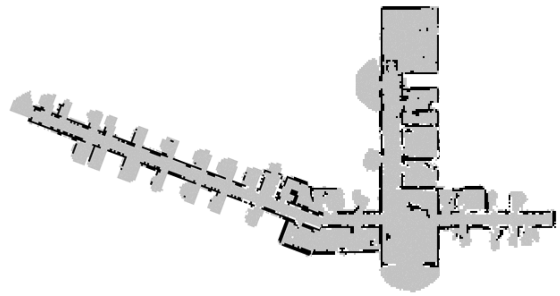
It is shown in this paper how both the long computation time and the stochasticity can be remedied by using a different registration algorithm, namely the improved Fourier Mellin (iFMI) registration (38). The iFMI is a variant of the Fourier Mellin transform for image representation and processing (39)(40), which is supplemented by the following two modifications. First, a logarithmic representation of the spectral magnitude of the FMI descriptor is used. Second, a filter on the frequency where the shift is supposed to appear is applied. A detailed description of iFMI can be found in (38).

IV. Experiments and Results

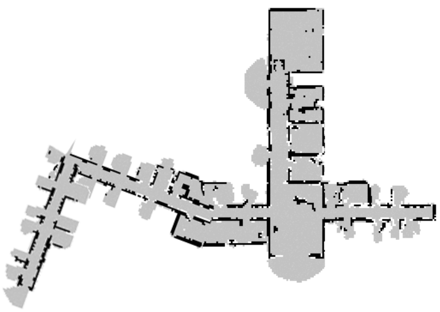
Two sets of maps are used for the following experiments. Both consist of a reference map and several versions of it with varying amounts of brokenness generated by different levels of bump noise.



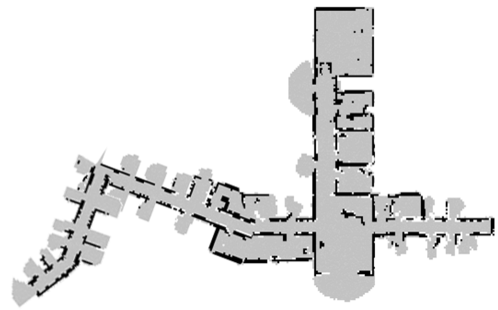
(a) undisturbed map $\mathbf{R^1}$



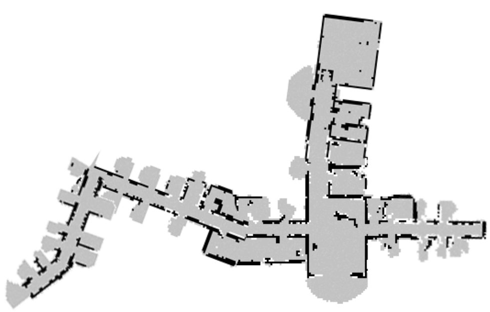
(b) $\mathbf{M_1^1}$, broken, degree $n = 1$



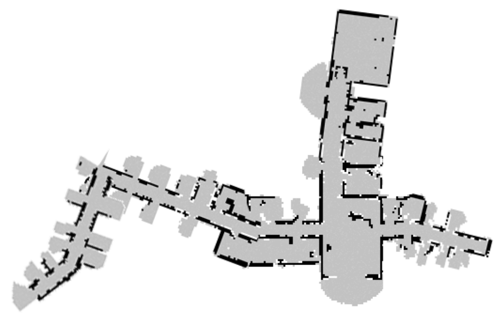
(c) $\mathbf{M_2^1}$, broken, degree $n = 2$



(d) $\mathbf{M_3^1}$, broken, degree $n = 3$

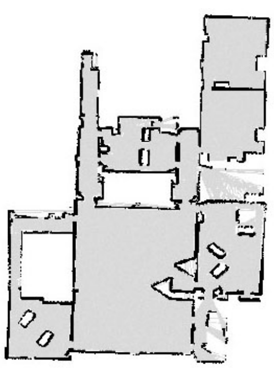


(e) $\mathbf{M_4^1}$, broken, degree $n = 4$

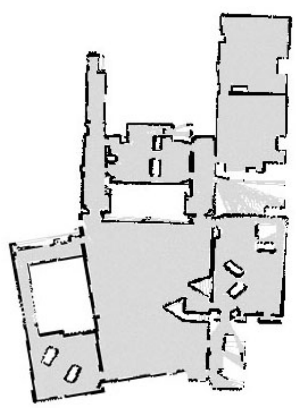


(f) $\mathbf{M_5^1}$, broken, degree $n = 5$

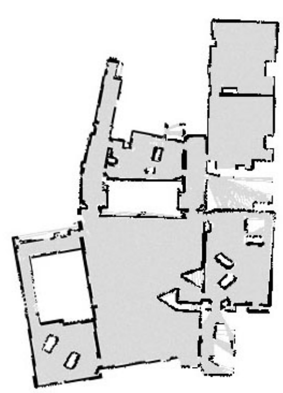
Fig. 2. An undisturbed map $\mathbf{R^1}$ as reference plus several versions of with varying degrees of brokenness due to bump noise.



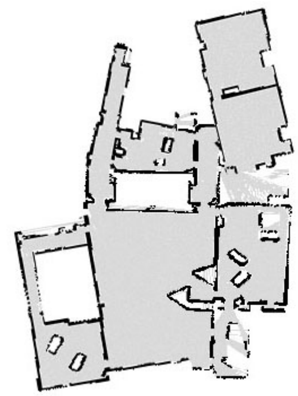
(a) undisturbed map $\mathbf{R^2}$



(b) $\mathbf{M_1^2}$, broken, degree $n = 1$



(c) $\mathbf{M_1^2}$, broken, degree $n = 2$



(d) $\mathbf{M_1^2}$, broken, degree $n = 3$

Fig. 3. A second undisturbed map $\mathbf{R^2}$ as reference plus several versions of it with varying degrees of brokenness due to bump noise.

TABLE I
RESULTS BASED ON REFERENCE $\mathbf{R^1}$

map	n_{comp}	t_{BCMM} (min:sec)	t_{iFMI} (sec)
$\mathbf{R^1}$	0	0:15	0.264
$\mathbf{M^1_1}$	1	1:16	0.350
$\mathbf{M^1_2}$	2	2:02	0.436
$\mathbf{M^1_3}$	3	4:30	0.522
$\mathbf{M^1_4}$	4	5:18	0.608
$\mathbf{M^1_5}$	5	6:14	0.694

TABLE II
RESULTS BASED ON REFERENCE $\mathbf{R^2}$

map	n_{comp}	t_{BCMM} (min:sec)	t_{iFMI} (sec)
$\mathbf{R^2}$	0	0:27	0.264
$\mathbf{M^2_1}$	1	1:07	0.350
$\mathbf{M^2_2}$	2	2:23	0.436
$\mathbf{M^2_3}$	3	3:50	0.522

The first set of maps (figure 2) is generated with USARsim (41), a high fidelity robot simulator (42). USARsim is based on the Unreal Game engine including an according physics and 3D visualization engine. It also includes realistic robot components models, e.g., with realistic noise on the sensor data. The robot-model used to generate the maps is based on the Jacobs rescue robots (43). The environment is a detailed model of the R1 research building at Jacobs University.

The second set of maps (figure 2) is based on real world data from the Robotics Data Set Repository (Radish) (44). Concretely, the maps are based on the dataset "ap_hill_07b", which contains the raw sensor data from four robots. For the experiments, the maps are generated from the raw data of the 3rd robot in the team - it simply explored most of the environment - with a state-of-the-art SLAM algorithm (45) and varying degrees of bump noise.

An undisturbed map **R** is used for both sets as reference. The levels of bump noise vary within the sets, i.e., different degrees of brokenness ranging from $n = 1$ to 5 for set 1 and from $n = 1$ to 3 for set 2 are investigated. Each map in a set is compared to the reference using the algorithm from section **??**. For completeness, the reference **R** is also compared to itself; obviously the result should be a brokenness of degree 0. The computation of the registration part of algorithm 1 (lines 5 to 7) is once based on the map merging method used in (13) and once based on iFMI. The masking operation in algorithm 1 (lines 9 to 11) is done both times with the masking step described in (13), namely by employing a local similarity measure using Manhattan distances between nearest neighbors with the same cell property (34) for Ψ_2. The experiments are carried out on a Intel Core-2 Duo 1.8 GHz processor under Linux.

The results of the experiments are shown in tables I and II. The most important fact is that for all 10 maps their correct degree of brokenness is properly determined independent of the concrete method used for the registration. But the computation times differ significantly. The map merging based method, for which the run-times are denoted with t_{BCMM}, takes in the order of dozens of seconds up to several minutes. The iFMI based method, for which the run-times are denoted with t_{iFMI}, takes in contrast a few hundred milliseconds. An other important advantage of the iFMI based registration is that it is deterministic.

V. CONCLUSIONS

The paper dealt with a contribution to assessing structural errors in grid maps, i.e., errors that disturb the large scale spatial layout of a map with respect to a reference, typically ground truth. Concretely, the concept of brokenness is presented. It is an important - if not even the most important - form of structural error. Typical sources of brokenness in maps are bump noise in robot localization, respectively residual errors in SLAM. A formal way is used to derive the degree of brokenness of a map. The introduced concepts can in general be useful for discussions of map quality; despite its formal basis, the degree of brokenness is based on intuitive notions that are also helpful to describe common properties of robot maps in informal ways.

Second, an efficient way to compute the degree of brokenness with a spectral registration method is introduced. It is one possible implementation of the general approach to recursively use registration and masking operations to determine the partitions of a map that are locally consistent but spatially transformed with respect to each other. The efficiency of the spectral based registration is shown through experiments, namely in comparison with a map merging method previously used to to compute brokenness.

REFERENCES

[1] S. Thrun, "Robotic Mapping: A Survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[2] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, pp. 73–88, 2005.

[3] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Springer, 1992.

[4] D. C. Lee, *The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Mobile Robot: An Experimental, Quantitative Evaluation*, ser. Distinguished Dissertations in Computer Science. Cambridge University Press, 2003.

[5] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[6] H. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1985, pp. 116–121.

[7] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. Tardos, "A comparison of SLAM algorithms based on a graph of relations," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 2089–2095.

[8] M. B. H. Moravec, "Learning sensor models for evidence grids," CMU Robotics Institute 1991 Annual Research Review, pp. 8–15, 1993.

[9] K. Wnuk, "Dense 3D Mapping with Monocular Vision," *SIGCSE Student Research Competition*, 2005.

[10] J.-L. Blanco, J. Gonzalez, Juan-Antonio, and Fernandez-Madrigal, Eds., *A New Method for Robust and Efficient Occupancy Grid-Map Matching*, ser. Pattern Recognition and Image Analysis. Springer, Lecture Notes in Computer Science (LNCS), 2007, vol. 4478.

[11] I. Varsadan, A. Birk, and M. Pfingsthorn, "Determining Map Quality through an Image Similarity Metric," in *RoboCup 2008: Robot WorldCup XII, Lecture Notes in Artificial Intelligence (LNAI)*, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Springer, 2009.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
[13] A. Birk, "A Quantitative Assessment of Structural Errors in Grid Maps," *Autonomous Robots*, vol. 28, pp. 187–196, 2010.
[14] S. Thrun, W. Burgard, and D. Fox, "A Real-Time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, pp. 321–328.
[15] S. Thrun, "A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.
[16] R. Madhavan, K. Fregene, and L. Parker, "Terrain aided distributed heterogenous multirobot localization and mapping," *Autonomous Robots*, vol. 17, pp. 23–39, 2004.
[17] G. Dedeoglu and G. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots," in *Distributed Autonomous Robotic Systems 4*. Springer, 2000, pp. 251–260.
[18] A. Howard, "Multi-robot mapping using manifold representations," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 4198–4203.
[19] S. Williams, G. Dissanayake, and H. Durrant-Whyte, "Towards multi-vehicle simultaneous localisation and mapping," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA*. IEEE Computer Society Press, 2002.
[20] J. Fenwick, P. Newman, and J. Leonard, "Cooperative concurrent mapping and localization," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA*. IEEE Computer Society Press, 2002.
[21] N. Roy and G. Dudek, "Collaborative exploration and rendezvous: Algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, 2001.
[22] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A Practical, Decision-theoretic Approach to Multi-robot Mapping and Exploration," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
[23] D. Stricker, "Tracking with reference images: a real-time and markerless tracking solution for out-door augmented reality applications," in *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*. ACM Press, 2001, pp. 77–82.
[24] C. Dorai, G. Wang, A. K. Jain, and C. Mercer, "Registration and integration of multiple object views for 3D model construction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 83–89, Jan., 1998.
[25] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, 1992.
[26] S. Alliney and C. Morandi, "Digital image registration using projections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 2, pp. 222–233, 1986.
[27] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[28] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," in *Proceedings of Computer Vision and Pattern Recognition*, June, 2003.
[29] L. V. Gool, T. Moons, and D. Ungureanu, "Affine/Photometric Invariants for Planar Intensity Patterns," in *Proceedings of European Conference on Computer Vision*, 1996.
[30] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, vol. 25, no. 3, pp. 305–316, 2008.
[31] ——, "Merging maps via Hough transform," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 1878–1883.
[32] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *IEEE Proceedings, special issue on Multi-Robot Systems*, vol. 94, no. 7, pp. 1384–1397, 2006.
[33] S. Carpin, A. Birk, and V. Jucikas, "On Map Merging," *International Journal of Robotics and Autonomous Systems*, vol. 53, pp. 1–14, 2005.
[34] A. Birk, "Learning Geometric Concepts with an Evolutionary Algorithm," in *Proc. of The Fifth Annual Conference on Evolutionary Programming*. The MIT Press, Cambridge, 1996.
[35] S. Carpin and G. Pillonetto, "Robot Motion Planning Using Adaptive Random Walks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 3809–3814.
[36] ——, "Learning sample distribution for randomized robot motion planning: role of history size," in *Proceedings of the 3rd International Conference on Artificial Intelligence and Applications*. ACTA press, 2003, pp. 58–63.
[37] ——, "Motion Planning Using Adaptive Random Walks," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 129–136, 2005.
[38] H. Buelow and A. Birk, "Fast and Robust Photomapping with an Unmanned Aerial Vehicle (UAV)," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 2009.
[39] Q.-S. Chen, M. Defrise, and F. Deconinck, "Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 12, pp. 1156–1168, 1994.
[40] B. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *Image Processing, IEEE Transactions on*, vol. 5, no. 8, pp. 1266–1271, 1996.
[41] USARsim, "Unified System for Automation and Robotics Simulator (USARsim)," http://usarsim.sourceforge.net/, 2006.
[42] S. Carpin, A. Birk, M. Lewis, and A. Jacoff, "High fidelity tools for rescue robotics: results and perspectives," in *RoboCup 2005: Robot Soccer World Cup IX*, ser. Lecture Notes in Artificial Intelligence (LNAI), I. Noda, A. Jacoff, A. Bredenfeld, and Y. Takahashi, Eds. Springer, 2006.
[43] A. Birk, K. Pathak, S. Schwertfeger, and W. Chonnaparamutt, "The IUB Rugbot: an intelligent, rugged mobile robot for search and rescue operations," in *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*. IEEE Press, 2006.
[44] A. Howard and N. Roy, "The Robotics Data Set Repository (Radish)," http://radish.sourceforge.net/, 2003.
[45] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2005.