

RMAP: a rectangular cuboid approximation framework for 3D environment mapping

Sheraz Khan · Athanasios Dometios · Chris Verginis · Costas Tzafestas · Dirk Wollherr · Martin Buss

Received: 6 June 2013 / Accepted: 5 February 2014 / Published online: 28 February 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract This paper presents a rectangular cuboid approximation framework (RMAP) for 3D mapping. The goal of RMAP is to provide computational and memory efficient environment representations for 3D robotic mapping using axis aligned rectangular cuboids (RC). This paper focuses on two aspects of the RMAP framework: (i) An occupancy grid approach and (ii) A RC approximation of 3D environments based on point cloud density. The RMAP occupancy grid is based on the Rtree data structure which is composed of a hierarchy of RC. The proposed approach is capable of generating probabilistic 3D representations with multiresolution capabilities. It reduces the memory complexity in *large scale* 3D occupancy grids by avoiding explicit modelling of free space. In contrast to point cloud and fixed resolution cell representations based on beam end point observations, an approximation approach using point cloud density is presented. The

proposed approach generates variable sized RC approximations that are memory efficient for *axis aligned surfaces*. Evaluation of the RMAP occupancy grid and approximation approach based on computational and memory complexity on different datasets shows the effectiveness of this framework for 3D mapping.

Keywords 3D mapping · Point cloud approximation

1 Introduction

An accurate 3D map of the environment is an essential requirement for all autonomous robots to perform navigation and collision avoidance. Many recent research works focus on mapping and localization. The environment representation can be topological (Thrun 1998) or metric (Elfes 1989; Thrun 2003; Hornung et al. 2013). Topological maps utilize graph structures to represent the environment whereas metric maps capture its area or volume. This paper presents the RMAP framework which generates metric maps using axis aligned rectangular cuboids (RC). The goal of this framework is to provide computational and memory efficient environment representations for 3D robotic mapping.

A commonly used environment representation for metric mapping is an occupancy grid (Moravec and Elfes 1985; Elfes 1989; Thrun 2003), which has been used extensively for navigation, localization and exploration in the field of robotics. 2D NDT (Biber and Straßer 2003) (Normal Distribution Transform) additionally models the grid cell centers and their uncertainty using Gaussian distributions. Some approaches model the 2D environment using geometric primitives such as lines (Nguyen et al. 2007) which are computationally less expensive compared to grid representations. The main drawback of 2D environment representations is that

S. Khan (✉) · D. Wollherr · M. Buss
Institute of Automatic Control Engineering,
Technische Universität München, Munich, Germany
e-mail: sheraz.khan@tum.de

D. Wollherr
e-mail: dw@tum.de

M. Buss
e-mail: mb@tum.de

A. Dometios · C. Verginis · C. Tzafestas
Division of Signals, Control and Robotics,
School of Electrical and Computer Engineering,
National Technical University of Athens (NTUA), Athens, Greece
e-mail: ath.dometios@gmail.com

C. Verginis
e-mail: chrisverginis@gmail.com

C. Tzafestas
e-mail: ktzaf@softlab.ntua.gr

they are applicable in case of planar environments. Besides 2D environment representations, some grid structures also store the height corresponding to each cell leading to 2.5D height maps (Herbert et al 1989) however they are unable to model the explicit shape of the environment.

The recent surge in 3D sensing technology with the influx of Kinect and Velodyne has shifted the focus of the robotics society from 2D to 3D environment representations. A common approach for 3D environment representation is utilizing point clouds or 3D occupancy grids. An extension of 2D occupancy grid concepts directly to 3D leads to a large overhead in terms of memory and computational cost due to the explicit modelling of free space. In contrast to standard occupancy grids some techniques keep a list of occupied cells (Ryde et al. 2012, 2010). MVOG (Multi volume occupancy grids) have been presented in (Dryanovski et al. 2010) which groups observations in vertical volumes over a 2D occupancy grid. The vertical volumes represent positive (obstacle) and negative (free) readings. A 3D probabilistic occupancy grid is formed by merging these volumes. Multi level surface maps (Triebel et al 2006) in contrast to elevation maps use intervals for each 2D grid cell to represent vertical surfaces and hence are more suitable for navigation. The work presented in (Einhorn et al. 2011) proposes a formulation which adapts the resolution of the grid in an online manner based on measurements. A fully probabilistic grid structure based on octrees titled Octomap (Wurm et al. 2010; Hornung et al. 2013) has been presented which generates accurate 3D environment representations. Recently an extension of 3D NDT concepts titled NDT-OM (Occupancy mapping) has been presented (Saarinen et al. 2013) which has similar computational complexity to Octomap (Wurm et al. 2010; Hornung et al. 2013), whereas the memory complexity issue is not discussed.

Since different approaches are available for 3D environment representation, an explicit comparison of each of them to the RMAP framework is essential. This paper focuses on two aspects of the RMAP framework: (i) An occupancy grid approach and (ii) A RC approximation approach based on point cloud density. To the author's best knowledge no prior work in 3D robotic mapping focuses on extracting axis aligned RC approximation based on point cloud density. The most similar work uses bounding volume hierarchies for collision detection in computer graphics (Figueiredo et al. 2010). Hence a comparison of the RMAP occupancy grid with other approaches is discussed. Point cloud representation is common in robotics, however it requires a large amount of memory since each point is stored and does not allow fusion of data in a probabilistic manner. Height maps (Herbert et al 1989) and multi level surface maps (Triebel et al 2006) do not model the explicit shape and hence cannot represent any arbitrary 3D shaped environment. In comparison to Octomap (Wurm et al. 2010; Hornung et al. 2013), the RMAP occupancy grid approach avoids explicit free space modelling hence it does

not differentiate between free and unknown space in the grid. This approach leads to advantages in terms of memory and computational cost and does not limit the applicability of this framework for most robotic applications such as localization, registration and even navigation, exploration as discussed in Sect. 4.

Motivation and contribution of RMAP

The goal of the RMAP framework is to provide computational and memory efficient representations for 3D mapping using axis aligned RC. 3D representations commonly used in robotics can be classified into two categories:

- (1) Representations with free and unmapped space modelling
- (2) Representations without free and unmapped space modelling

Mapping techniques which fall in these categories and their characteristics are discussed below.

Standard occupancy grid

The standard occupancy grid falls within the first category. Figure 1a shows an exemplary occupancy grid representation composed of occupied, free and unmapped cells. In this paper the term cell based on context is used abstractly for squares, rectangles, cubes (3D) and RC (3D). Given the robot position (shown in green) all cells inside the field of view (FOV) of the sensor (shown in red) are marked as free or occupied whereas all cells outside this FOV are unmapped areas. Occupancy grid formulations are very common in robotics because they generate probabilistic representations that allow to cope with sensor noise and provide a principled formulation for sensor fusion. Differentiating free and unmapped areas is important for exploration and navigation. In context of safe navigation, the robot path should only be planned through regions covered by sensor measurements.

A major obstacle in application of occupancy grids for large scale 3D mapping is efficiency, specifically computational and memory efficiency. As mentioned in (Hornung et al. 2013) in the context of occupancy grids “the memory consumption is often the bottle neck for 3D mapping”. The majority of cells in large scale 3D occupancy grids are composed of free space. This can be visualized by considering the usage of a standard 3D occupancy grid of 10–20 cm resolution for the 1st laser scan of Freiburg campus¹ which contains beam lengths up to 50 m range shown in Fig. 2a. In case of large scale 3D mapping the robot accumulates multiple scans, hence the efficiency issue becomes a dominating factor.

¹ Courtesy of Steder and Kümmerle, available at <http://ais.informatik.uni-freiburg.de/projects/datasets/octomap/>.

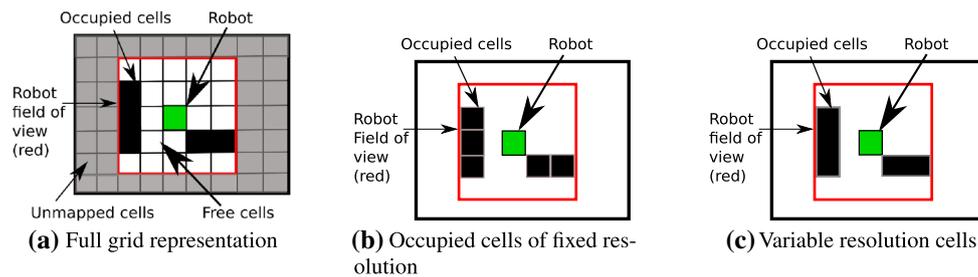


Fig. 1 Different 2D environment representations. **a** A standard occupancy grid with explicit free space modelling. The regions inside the field of view (shown in red) based on the robot position (green) are marked as occupied or free. In *large scale 3D mapping* explicit free space modelling is not a feasible option in context of efficiency as majority of cells in large scale 3D occupancy grids are composed of free space. **b** The RMAP occupancy grid focuses on the transition from **a** to **b** with

ray tracing and implicit free space modelling. The RMAP occupancy grid can be considered as an extension of **b**. **c** A variable resolution environment representation which is memory efficient in comparison to **b**. The RMAP RC approximation based on point cloud density can be considered as the transition from a fixed resolution representation based on beam end points shown in **b** to a variable resolution representation shown in **c** (both approaches based on beam end points)

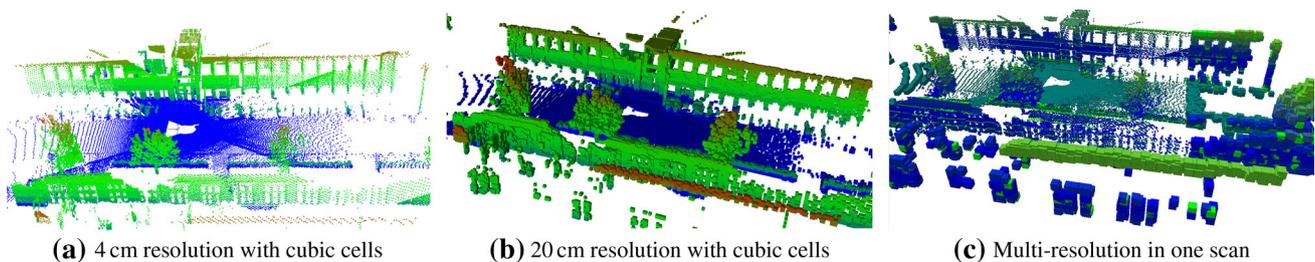


Fig. 2 **a–b** Different resolution views of the 1st scan from the Freiburg campus dataset. Using a standard 3D occupancy grid of 10–20 cm resolution for a scan consisting of beam lengths upto 50 m range would mostly constitute of free space for the scenario shown above. Free space modelling becomes an important aspect in context of efficiency for *large scale 3D occupancy mapping*. The question to be raised here is

if explicit modelling of free space is essential. **c** The RMAP occupancy grid is capable of generating multiresolution representations within a scan. In the figure shown, all regions close to the robot are shown in high resolution whereas regions far away are shown with lower resolution. All images shown above are height colored

Since the majority of the cells in large scale occupancy grids are composed of free space the questions arises *if explicit modelling of free space in occupancy grids is essential*. The goal is the transition from Fig. 1a to 1b which involves *ray tracing*, however avoiding explicit free space modelling. Differentiation of free and unmapped areas can be based on the robot path (during 3D mapping), sensor characteristics (maximum range and FOV) and the map generated by the robot as discussed in Sect. 4.1. The RMAP occupancy grid can be considered as an extension of Fig. 1b which implicitly models free space and allows multiresolution capabilities. *The transition from Fig. 1a to 1b based on the RMAP occupancy grid and the variation in computational and memory complexity due to this transition is the focus of Sect. 2.2.*

Point cloud and occupied cell representations

3D representations without free and unmapped space modelling can be useful in different robotic applications such as registration and localization. Such representations are mostly used when the sensor is quite accurate and the environment is static (dynamic environments require additional processing).

Point cloud representations store beam end point observations and fall within the second category. Point cloud representation is useful but requires a large amount of memory as all points are stored. In context of this subsection a fixed resolution cell representation of Fig. 1b using *beam end points* is possible based on the number of sensor observations for a specific cell. A cell is marked as occupied if the occupancy probability is beyond a certain threshold. An important point to specify here is that once a cell has been marked as occupied it remains occupied as ray tracing is not performed. The question to be raised here is *if such representations can be further improved*.

The focus is the transition from Fig. 1b to 1c (both approaches without raytracing) which uses variable resolution cells. In the example shown in Fig. 1, the fixed resolution representation requires five grid cells whereas the variable resolution approach utilizes two grid cells. It is obvious that variable resolution representations require less memory as fewer cells need to be saved. Once a variable resolution representation has been developed the computational cost of 3D reconstruction is reduced because fewer cells need to be accessed. Finding the best resolution of grid cells for a spe-

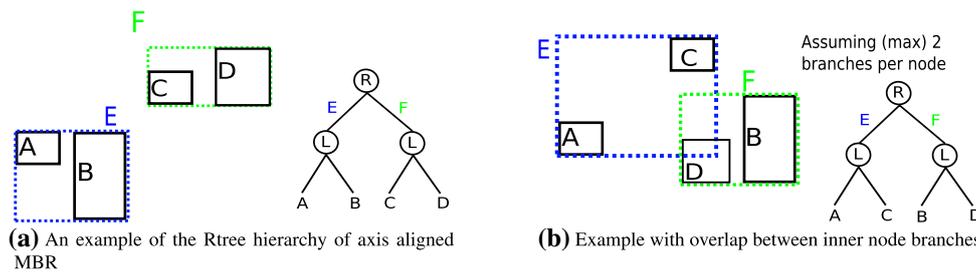


Fig. 3 Examples illustrating the Rtree hierarchy. **a** The Rtree data structure is a hierarchy of minimum bounding axis aligned rectangles (2D). Search for a specific rectangle is carried out by performing containment/overlap tests throughout the hierarchy of the tree. **b** An exem-

plary Rtree hierarchy in which the inner node branches overlap. This overlap plays an important role in the performance of the Rtree data structure in context of 3D mapping

plary Rtree hierarchy in which the inner node branches overlap. This overlap plays an important role in the performance of the Rtree data structure in context of 3D mapping

- sific environment is a challenging problem because surface shapes can be complicated. In this paper an approach based on point cloud density is presented (Sect. 2.3) which can be used to extract axis aligned RC approximations of the 3D environment. The proposed approximation focuses on axis aligned planar surfaces as they can be approximated accurately by RC.
- Based on the discussion above, the main contributions of this paper are highlighted below:
- An evaluation of the Rtree data structure in context of 3D robotic mapping on a publically available dataset.
 - The RMAP occupancy grid approach with implicit free space modelling for large scale 3D mapping. The proposed approach generates probabilistic 3D representation with multiresolution capabilities.
 - An axis aligned RC approximation of 3D environments based on point cloud density. The proposed approach allows memory efficient representations in contrast to fixed resolution voxels and point clouds representations for axis aligned surfaces.

The remainder of this paper is organized as follows: The RMAP occupancy grid and variable size RC approximation approach is presented in Sect. 2. In Sect. 3, results are presented by conducting different simulation and experimental evaluations. Discussion and comments related to each section are presented in Sect. 4. Conclusions and future work is presented in Sect. 5.

2 Formulation of the RMAP framework

In this paper two aspects of the RMAP framework are discussed. Firstly an occupancy grid formulation and secondly an axis aligned RC approximation approach of 3D environments based on point cloud density. The RMAP occupancy grid is an application of the Rtree data structure (Guttman 1984) for 3D robotic mapping. The Rtree data structure is

play Rtree hierarchy in which the inner node branches overlap. This overlap plays an important role in the performance of the Rtree data structure in context of 3D mapping

a hierarchy of minimum bounding axis aligned rectangles (MBR) as depicted in a simplified example in Fig. 3a. The term MBR and minimum bounding axis aligned rectangular cuboids (MBRC) will be used for 2D and 3D environments respectively. The tree structure depiction in Fig. 3a, b is different than the normal convention in order to facilitate the discussion about RMAP in the experimental section. In order to define the Rtree node structure three different terms will be used such as leaf, root and inner nodes. As shown in Fig. 3a, some nodes in the tree are labelled L, R to denote leaf and root nodes respectively. Figure 3a does not show inner nodes however, as the tree structure expands inner nodes are added as well. All branches connected to the leaf, inner and root node are termed leaf, inner and root branches respectively. The root, inner branches contain information regarding the MBR or rectangle in case of leaf branches. An Rtree of order (n, M) , has the following characteristics (Guttman 1984; Nanopoulos et al. (2006)):

- Each leaf node can have a maximum of M branches and a minimum of n where $n \leq \frac{M}{2}$. The leaf node branches contain the elements (Rectangle, Object). The object in context of the RMAP occupancy grid (Sect. 2.2) represents the occupancy probability of the rectangle. In case of RC approximations (Sect. 2.3) it contains the density of points in the rectangle. As the Rtree is height balanced all leaf nodes are at the same height.
- Each inner node can contain a maximum of M branches and a minimum of n entries. Each inner branch consists of a MBR which contains all the MBR/rectangles of its child node branches.
- The root node can have a minimum of two branches unless it is a leaf node.

The insertion of a rectangle into the Rtree structure involves searching for an inner branch that leads to the *least expansion* of the MBR. In case the number of branches in a node is greater than M the node splits. Figure 3b shows an exemplary Rtree hierarchy assuming that each node can

have a maximum of 2 branches. Initially rectangles A and B are inserted, such that the Rtree structure consists of a single node. If further rectangles C and D are added the node splits increasing the height of the structure and forms overlapping MBR (E and F) as shown in Fig. 3b. The splitting strategy shown in Fig. 3b is just an illustration. The strategy used in this paper is termed as quadratic splitting (Guttman 1984) and was chosen due to its better quality of split in comparison to linear splitting. The important aspect is that the MBR of the branches in the Rtree structure can *overlap*. This overlap between inner branches plays a important role in the performance of the Rtree in context of 3D mapping. The effect of overlaps is that multiple nodes might need to be searched during a spatial query or least expansion during insertion. Given any arbitrary query rectangle, the number of rectangles contained within it can be found by performing overlap/containment tests throughout the hierarchy of the tree structure. The focus of this paper is on 3D mapping, hence the term RC will be used for leaf branches and MBRC for inner and root branches.

2.1 Properties of the Rtree data structure

In this section different properties of the Rtree data structure are mentioned. The following aspects are important:

- **Number of branches per node (M):** The Rtree inner or leaf nodes can have an integer number of branches M . For a fixed number of leaf branches increasing M generate's tree structures containing fewer inner nodes and less height reducing the memory required for representation but creates more overlaps. Consider the scenario shown in Fig. 3b in which the assumption of 2 branches per node is considered. If the maximum number of branches per node is increased from 2 to 4, one node is required in the hierarchy to represent all leaf branches reducing the number of nodes and the memory respectively.
- **Assumptions of the data structure:** The tree hierarchy in the Rtree data structure is created and updated incrementally as RC are inserted which are constrained to be axis aligned. The tree hierarchy is not pre-defined hence it requires additional maintenance during insertion such as *node splitting* and checking for *least expansion*.
- **Overlap:** The most important aspect for the Rtree data structure is the overlap factor which arises indirectly due to the assumptions of the data structures. The Rtree inner branches MBRC can overlap. A disadvantage of this overlap is that search for a RC in the Rtree can become computationally costly as multiple inner nodes might need to be accessed during a query.

All the aspects listed above affect the performance of the Rtree in the context of 3D mapping. Important perfor-

mance parameters from 3D mapping perspective are insertion, extraction times and memory consumption. The extraction time corresponds to the time required to access all occupied cells once all laser scans have been inserted. The insertion time corresponds to the time required to insert beam end points as occupied and beam path as free.

The following section explains the basic aspects of the RMAP occupancy grid such as the initialization and calculation of occupancy probabilities for the RC.

2.2 RMAP occupancy grid framework

The RMAP occupancy grid tackles the computational and memory complexity issue in standard grids by making an important assumption that all space a priori is considered as free. This assumption is termed as the *free space assumption*. Unlike standard grids in which the structure is pre-allocated, the grid in the RMAP occupancy approach is not defined a priori rather created as *sensor observations are obtained*. The RMAP occupancy grid is probabilistic in nature and models the occupancy of a RC. A grid cell is initialized by inserting a RC at the beam end point. Ray tracing is done along the beam path to update the occupancy values of all RC which were initialized once. All uninitialized cells are considered as free. All RC in the leaf branches of the RMAP occupancy grid are of fixed volume (possibly cubic) based on the chosen resolution of the grid, axis aligned and do not overlap. However the inner branches MBRC can overlap. Let z represent the observation and subscript t the time instance at which the observation was recorded. The occupancy probability of any leaf branch r_i which represents the i th RC is estimated by (Moravec and Elfes 1985)

$$P(r_i | z_{1:t}) = \left[1 + \frac{1 - P(r_i | z_t)}{P(r_i | z_t)} \cdot \frac{1 - P(r_i | z_{1:t-1})}{P(r_i | z_{1:t-1})} \cdot \frac{P(r_i)}{1 - P(r_i)} \right]^{-1},$$

which is a commonly used sensor model in robotic mapping. $P(r_i | z_{1:t})$ represents the occupancy probability of the i th RC given all observations. $P(r_i)$ represents the occupancy probability of a RC prior to any observations. $P(r_i | z_t)$ and $P(r_i | z_{1:t-1})$ represent the probability given the most current observation z_t and observations since the beginning of time until time $t - 1$ respectively. To clarify, the RC initialization and update based on the beam end point observation is explained. Given the RC to which the beam end point belongs (mapping a point to the grid), overlap and containment tests are performed throughout the Rtree hierarchy to determine if the leaf branch corresponding to the RC exists. If it exists its prior probability is updated otherwise a leaf branch is added to the hierarchy (RC is initialized) and the Rtree hierarchy is updated. To prevent grid cells from being over confident about its state, a clamping/saturation probability threshold α

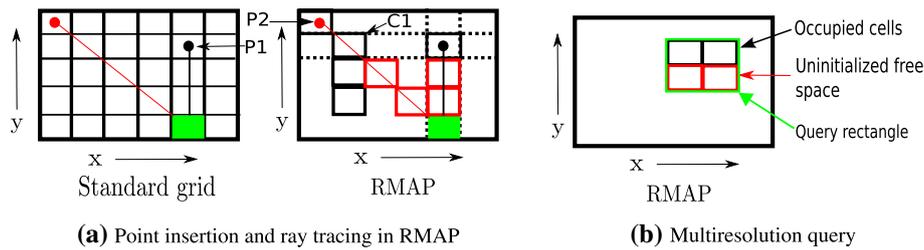


Fig. 4 Point insertion and multiresolution query mechanism. **a** The figure shows implicit free space modelling of the RMAP occupancy grid using the beam end point observations $P1$ and $P2$. The approach updates initialized cells (shown in *black*) based on beam end point observations whereas uninitialized cells (shown in *red*) do not exist in the grid and are assumed to be free based on the free space assumption. It is possible that cells like $C1$ get initialized due to noise or dynamics in the

environment and are always updated assuming that they are in the field of view of the sensor. In the figure the robot position is marked in green. **b** The occupancy probability of any query rectangular cuboid is based on all initialized and uninitialized cells that are contained in it. Given the resolution of the grid, size of the query cuboid and the number of occupied cells returned by the RMAP occupancy grid it is possible to determine the number of uninitialized cells

and $1 - \alpha$ is utilized, after which the cell is no longer updated. Given the probabilities of the leaf branches, the occupancy probability of any query rectangular cuboid q_i is calculated by

$$P(q_i) = \frac{1}{N} \sum_{i=1}^N P(\bar{r}_i), \quad (1)$$

where \bar{r}_i represents all RC contained in q_i which have been initialized until time t as well as all uninitialized free space regions based on the chosen resolution of the grid. In principle any criterion can be chosen for defining the occupancy probability of the query RC, such as the $\max_i P(\bar{r}_i)$. The main motivation for choosing an average probability of all RC is because of the smooth variation in the probability as the size of the query cuboid is increased. The query cuboid can be used in different regions of interest in the map with respect to the global frame of reference and allows arbitrary adaptation of grid resolution as shown in Fig. 2c.

The illustration of point insertion and multi-resolution query process in the RMAP occupancy grid is shown in a simplified 2D scenario in Fig. 4. Initially the entire region to be mapped by the proposed approach is considered to be free space. Now consider the insertion of a point $P1$ into a standard grid and the RMAP occupancy grid as shown in Fig. 4a. The standard occupancy grid directly updates the probability of the cell to which the point belongs and updates all cells that lie in the beam path. In contrast, the RMAP occupancy grid initializes a grid cell with 0.5 occupancy probability based on the beam end point observation. Ray tracing is performed along the beam path to update only those cells which have been initialized. Once initialized a cell is always updated. Uninitialized cells (shown in red) do not exist in the RMAP occupancy grid and based on the free space assumption are assumed to be free, hence they are implicitly modelled in the RMAP occupancy grid. *In this way the proposed approach avoids explicit modelling of free space.* It is possible that cells

such as $C1$ shown in Fig. 4a get initialized due to noise or dynamics in the environment. Such cells are always updated during ray tracing as shown in the case of sensor observation $P2$. Hence the RMAP occupancy grid is composed of two kind of cells, firstly initialized cells which are always updated if they are in the FOV of the sensor. Secondly uninitialized cells which are implicitly modelled hence they do not exist in the grid and are assumed to be free. The occupancy grid can also contain cells which are initialized due to noise or dynamics in the environment and are updated if they are in the FOV of the sensor. Fig. 4b shows the multiresolution query process in the RMAP occupancy grid. Overlap/containment tests are performed throughout the hierarchy of the Rtree data structure to find cells contained in the query cell shown in green. The query cell is constrained to be axis aligned and an integer multiple of the chosen resolution of the grid. In the scenario of Fig. 4b the RMAP occupancy grid will return the occupancy probabilities of the two initialized cells marked in black. However based on the fixed resolution of the grid, the RMAP occupancy approach can determine the two uninitialized cells (assumed to be free) that do not exist in the grid shown in red in Fig. 4b. Based on the initialized and uninitialized cells the RMAP approach can calculate the occupancy probability of the query cell using (1).

In this subsection the RMAP occupancy grid has been presented which allows probabilistic representations with multiresolution capabilities. The proposed approach avoids explicit modelling of free space in order to reduce the memory complexity of standard occupancy grids. The structure of the occupancy grid is created incrementally based on observations. The proposed occupancy grid can be used for localization, registration and even navigation, exploration. In context of exploration and navigation differentiating free and unknown space is important. The proposed approach does not explicitly differentiate between free and unknown space in the grid. The advantages and disadvantages of the free space assumption in context of navigation and exploration

are discussed in Sect. 4. Given the robot trajectory (during mapping), sensor characteristics (maximum range and FOV) and the map generated by the robot it is possible to differentiate between free and unmapped areas in context of exploration and navigation as discussed in Sect. 4.

2.3 Rectangular cuboid approximation of point clouds based on density

In this section an approach is presented that generates variable sized RC approximations based on point cloud density. The proposed approach has applications in robotics such as localization in static 3D environments or generating memory efficient 3D approximations using RC. The approach makes two important assumptions: (i) The point density is fairly uniform (not heavily skewed) for all scanned surfaces and (ii) Majority of the structure in the point cloud is axis aligned. The second assumption is present because the RMAP frame work is based on RC that are constrained to be axis aligned.

The proposed approach initializes by generating a minimal bounding RC of the given 3D point cloud. The basic premise is that the initial RC is not a good approximation of the point cloud and that it contains smaller regions of high point density. The approach splits the RC until it finds high point density regions and the volume satisfies certain constraints. Although the splitting strategy is naive as it always splits the RC into eight equal parts based on its center and recomputes the RC based on the point cloud, nevertheless it yields good results if the axis aligned assumption is satisfied as shown in Sect. 3.

The pseudocode of the approximation algorithm is shown in Fig. 5. The input to the algorithm is the point cloud \mathbf{p} to be approximated, the approximation percentage σ and the minimum volume ω threshold. The output is a set of RC R which approximate the point cloud. The algorithm starts by checking if the number of points are more than 3 (line 1) because it is the minimal number of points required to describe a volume. The algorithm then calculates the minimal bounding RC of the point cloud based on the minimal and maximal points (line 2). In addition the algorithm calculates the volume V of the minimal bounding RC and the point density (line 3 and 4). The density and volume are compared to the density and minimum volume threshold β , ω respectively (line 5). If the density and the volume are greater than β and ω respectively, the RC is stored in the set R , otherwise the algorithm splits the RC into 8 equal parts based on the center (line 7). After splitting, the points in the point cloud are reassigned to each RC. Each split point cloud is passed recursively to the **3D_approximation** (line 8) algorithm until the threshold is satisfied.

Figure 6 shows the splitting strategy defined in the pseudocode shown in Fig. 5 for a simplified 2D case. The top row shows the splitting strategy in case the entire point cloud

```

3D_approximation( $\mathbf{p}$ ,  $\sigma$ ,  $\omega$ )
Input:
  Point cloud  $\mathbf{p}$ ,
  Approximation percentage  $\sigma$ ,
  Density threshold  $\beta = \frac{\text{initial density of } \mathbf{p}}{(1 - \sigma)}$ ,
  Minimum volume threshold  $\omega$ 
Output:
  Set of rectangular cuboids  $R = \{r_1, r_2, \dots, r_n\}$ 
Procedure:
1  if (points in  $\mathbf{p} \leq 3$ )
   1.1 store as undefined points;
2  Compute minimal bounding rectangular cuboid  $r_i$ 
   of point cloud  $\mathbf{p}$ ;
3  Compute volume  $V$  of rectangular cuboid  $r_i$ ;
4  Compute point density  $d$ , where  $d = \frac{\text{points in } \mathbf{p}}{V}$ ;
5  if ( $d \geq \beta$ ) and ( $V \geq \omega$ ) then
6     $R = R + r_i$ 
   else
7    split  $r_i$  into 8 equal parts with respect to
   the center and reassign points of the
   point cloud  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_8\}$ ;
   for all  $\mathbf{p}_j$  do
8     3D_approximation( $\mathbf{p}_j$ ,  $\sigma$ ,  $\omega$ );
   end
   end

```

Fig. 5 Rectangular cuboid approximation pseudocode

is axis aligned. The bottom row shows the rough approximation in case the structure is non axis aligned. The initial minimal bounding RC is shown in red for both cases. The initial RC is not a good approximation as the density of points (given the volume) is low. Hence the RC splits and recomputes the minimal bounding RC shown in green for both cases. In the bottom row the difference is the non axis aligned structure which causes the RC to split until a high density region is found and it satisfies the volumetric constraint. The splitting strategy used in Fig. 5 is similar to an Octree hierarchy in which the cube is split into 8 equal parts. The differentiating factor is the recomputation of minimal bounding RC (line 2) which generates variable resolution RC. Additionally the initial minimal bounding RC is not constrained to be a cube.

This section presented a formulation that generates variable sized RC approximations based on point cloud density. The approach works well if the point density for the scanned surfaces is uniform (without noisy measurements) and the majority of the point cloud structure is axis aligned. The approximation over estimates the volume in case a few noisy measurements are close to a high density region. The main objective of the approach is to search for high point density regions which satisfy certain volumetric criterion. The proposed approach models the surface based on beam end point observations without *ray tracing*. It can be useful for robotic application such as localization in static environments or generating memory efficient 3D approximations.

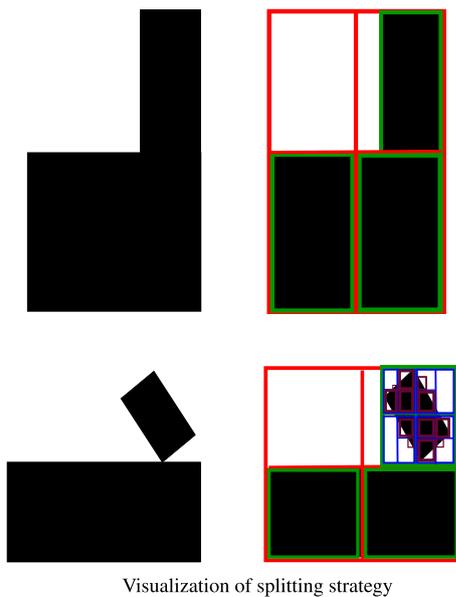


Fig. 6 Visualization of the splitting strategy for axis aligned point cloud (*top row*) and the rough approximation (*bottom row*) if the point cloud contains non axis aligned regions. The initial minimal bounding RC of the point cloud is shown in *red* for both rows. The initial RC is not a good approximation, hence based on the density threshold a split will take place. The recomputed minimal bounding RC after splitting is shown in *green*. In case of non axis aligned structures (*bottom row*) the split takes place until the density and volumetric constraint is satisfied

3 Experimental evaluation

In this section the RMAP framework is evaluated in terms of computational cost and memory efficiency. This section is further divided into three main subsections, the first dealing with an evaluation of the Rtree data structure in context of 3D mapping. The second subsection deals with the RMAP occupancy grid and its evaluation in comparison to the latest version (1.6.1) of Octomap (Wurm et al. 2010; Hornung et al. 2013). The final subsection deals with the evaluation of the RMAP RC approximation approach and memory complexity analysis to other approaches on real world data sets (Freiburg and Bremen city center² dataset). The experiments mentioned in this section were carried out on an Intel(R) Core i5-2500K, 3.30 GHz processor with 16 GB memory.

3.1 Evaluation of the Rtree data structure

An evaluation of the Rtree data structure is performed in context of 3D mapping with respect to insertion, extraction times and varying M (number of branches per node) on 70 scans of the Freiburg campus using the RMAP occupancy

grid as shown in Fig. 7. The insertion time corresponds to the time taken to update the beam end point as occupied and beam path as free for 100,000 points. The access time corresponds to the time taken to access all occupied cells once all laser scans have been inserted. As discussed in Sect. 2.1, for a fixed number of leaf branches increasing M generates compact tree representation (less height and contains less number of nodes) and correspondingly less memory as can be seen in Fig. 7c. The memory for $M = 8$, $M = 16$ and $M = 32$ at 10 cm resolution for 70 scans of the Freiburg campus is 200 MB, 182 MB and 172 MB respectively. It can be seen from the figure that the variation in memory (due to M) decreases with increasing grid cell size. Compact tree representations (with increasing M) allow faster access times for occupied cells as can be seen in Fig. 7a however due to increased overlaps the insertion time increases. The memory of RMAP for 64 bit architectures is calculated based on

$$memory_{RMAP} = N_{inner} \times f(M) + N_{leaf} \times f(M),$$

which gives the memory in bytes. N_{inner} , N_{leaf} denotes the number of inner and leaf nodes whereas $f(M)$ is a value dependent on the number of branches allowed per node. For the RMAP occupancy grid each inner or leaf node, contains 2 integers (one for the number of its branches and one for the current height of the Rtree) utilizing 8 bytes (4 bytes + 4 bytes = 8 bytes) and an array of branches. A branch contains 6 integers defining the MBRC or RC and a union which consists of a pointer to the next node or the object in case of inner or leaf branch respectively consuming 32 bytes in total. Given $M = 8$, $M = 16$ or $M = 32$, the value of $f(M)$ is 264 (8 bytes + 8×32 bytes = 264 bytes), 520 or 1032 bytes respectively.

3.2 Comparison of the RMAP occupancy grid with octomap

In this section the Octomap approach is compared with the RMAP occupancy grid. The objective of this section is to validate the free space assumption by comparing it with Octomap to give a scale of memory and computational savings for *large scale* 3D mapping. The comparison of the RMAP occupancy grid with Octomap (Wurm et al. 2010; Hornung et al. 2013) version 1.6.1 was done for different resolutions in terms of insertion, extraction time and memory costs on 70 scans of the Freiburg campus. Figure 8 shows the insertion, access times and memory consumption of both approaches. An important point here is that the access time and memory consumption are shown in a semi-log plot in the Figures. The RMAP occupancy grid has better access and memory consumption as can be seen in Fig. 8a, c. The large difference in memory is due to the *free space assumption and implicit free space modelling*. The RMAP occu-

² Courtesy of Dorit Borrmann and Jan Elseberg available at the Osnabrück robotic 3D scan repository, <http://kos.informatik.uni-osnabrueck.de/3Dscans/>.

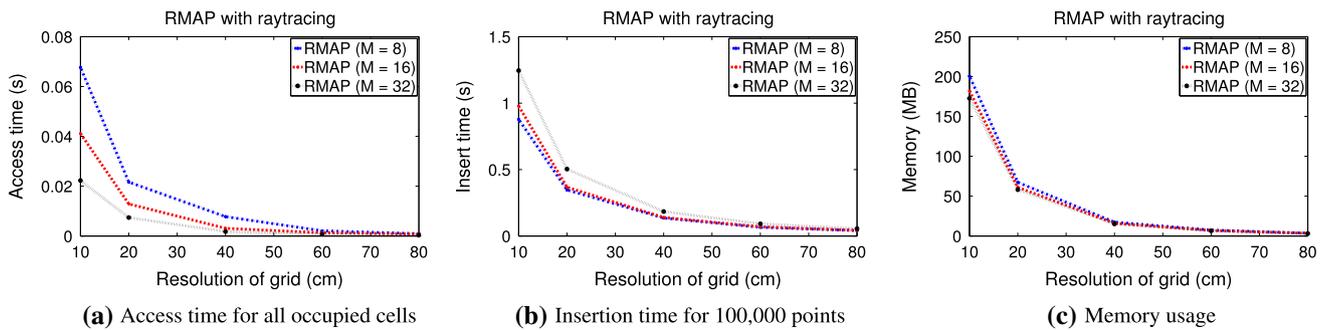


Fig. 7 An evaluation of the Rtree data structure (the RMAP occupancy grid with raytracing) in context of 3D mapping on 70 scans of Freiburg campus. **a** The access time of the Rtree data structure reduces with increasing M and increasing grid cell size because the Rtree generates tree representations of less height and contains less number of nodes

(Sect. 2.1). **b** The insertion time of the Rtree data structure increases due to additional overlaps with increasing M . **c** The memory consumption decreases with increasing M and increasing grid cell size as less number of nodes are required for representation

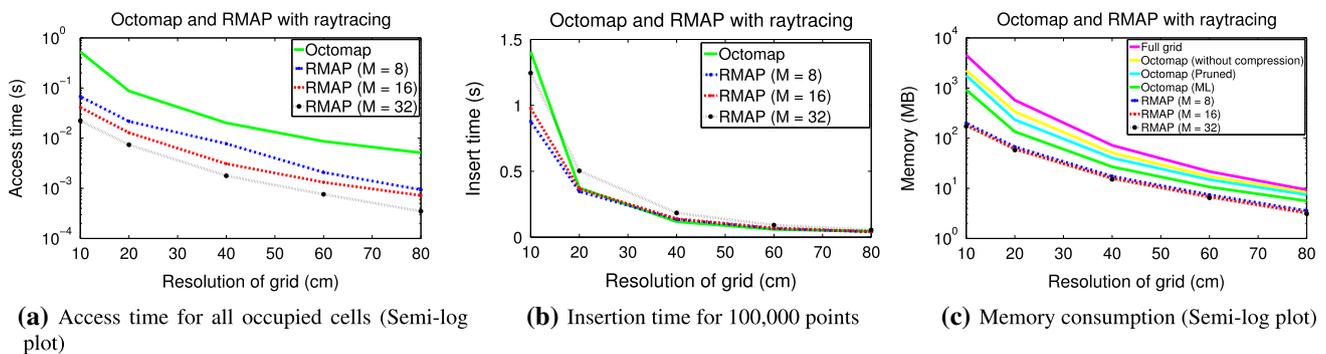


Fig. 8 An evaluation of the RMAP occupancy grid and Octomap on 70 scans of the Freiburg campus. **a** The extraction times of the RMAP occupancy grid are better than Octomap because the RMAP occupancy grid implicitly models free space and generates compact tree representations. Due to implicit free space modelling the tree contains fewer nodes which can be traversed quickly. **b** The insertion times of the RMAP occupancy grid are comparable to Octomap. The RMAP occupancy grid updates initialized cells and assumes uninitialized cells to

be free whereas Octomap generates and updates all cells (below the clamping threshold) at each insertion. **c** The large difference in memory is due to *implicit free space* modelling of the RMAP occupancy grid. The memory of Octomap is mentioned for three different cases: Without compression, Pruned and Maximum likelihood (lossy compression). The full grid corresponds to the minimal grid required to represent the same information (Wurm et al. 2010; Hornung et al. 2013)

pancy grid updates initialized regions and assumes uninitialized region to be free. In contrast Octomap generates and updates large volumes at each insertion. The insertion time of the RMAP occupancy grid is better or comparable to Octomap for $M = 8$ and $M = 16$ whereas worse for $M = 32$. Based on the insertion time mentioned all 70 scans of the Freiburg campus can be processed in *less than 2 min* by the RMAP occupancy grid. The access, insertion time of the RMAP occupancy grid is less than 7 ms, 1.25 s (for all M) respectively and the memory consumption is about 200 MB for a 10 cm resolution grid. In comparison the access, insertion time for Octomap is about 0.52 and 1.40 s. The memory consumption is 2249 MB without compression (64 bit architecture), 1778.08 MB based on pruning and 923.78 MB based on maximum likelihood (lossy compression). The memory of Octomap was calculated based on the nodes of the structure and confirmed by the *graph2tree* tool provided

in the implementation. The memory formula in (Hornung et al. 2013) adapted to 64 bit architectures is

$$memory_{octomap} = N_{inner} \times 80 \text{ bytes} + N_{leaf} \times 16 \text{ bytes},$$

where N_{leaf} corresponds to the number of leaf nodes and N_{inner} corresponds to the inner nodes in the Octomap structure. The memory of the RMAP occupancy grid is calculated based on the formula discussed in the previous section. Figures 9 and 10 shows the 3D map of the Freiburg and Bremen city centre dataset generated by the RMAP occupancy grid.

In addition to the comparison above the accuracy of the generated model by the RMAP approach is evaluated on the Freiburg campus dataset in a similar manner as in Wurm et al. (2010), Hornung et al. (2013). The accuracy is measured as the percentage of correctly mapped cells in the occupancy grid. A cell is correctly mapped if its state in the generated map is the same as in the evaluated scan. Therefore this

process involves inserting the scan being evaluated into an already built map requiring the beam end point to be occupied and beam path to be free space. A cell in the map is considered as occupied if its occupancy probability is greater than 0.9 whereas all uninitialized regions are considered as free space. Every 5th scan of the Freiburg campus is used as an evaluation scan to determine the number of correctly mapped cells. The evaluation is carried out for 20 cm resolution grid and the accuracy of the model is 99 %. The remaining 1 % error might be due to sensor noise in the scan or discretization effects during ray tracing.

In Sect. 3.1, an evaluation of the Rtree structure in context of 3D mapping showed that increasing M (number of branches per node) reduces memory and access times due to compact tree representation. In contrast it increases overlaps and causes the insertion time to increase. An evaluation of the RMAP occupancy grid and Octomap showed that the free space assumption in the RMAP occupancy grid leads to large

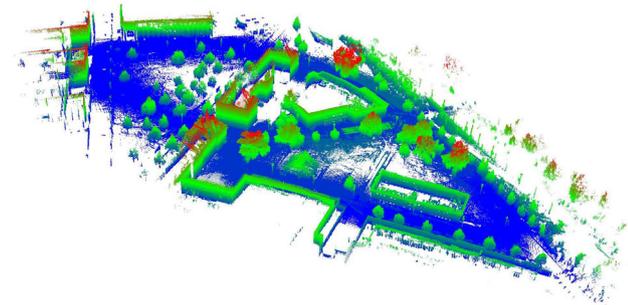


Fig. 9 Freiburg campus using the RMAP occupancy grid ($319 \times 202 \times 29$)

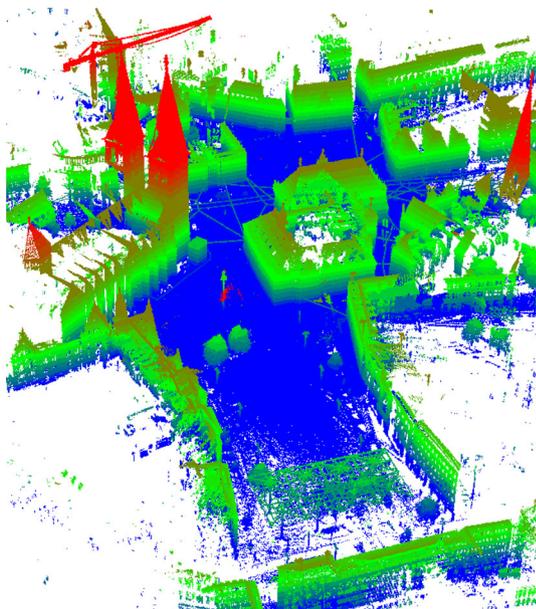


Fig. 10 Bremen city center using the RMAP occupancy grid

memory savings. The RMAP occupancy grid in comparison to Octomap is shown to have better access times and comparable insertion times. This is experimentally validated by comparing both approaches on the Freiburg campus dataset. Additionally an evaluation to determine the accuracy of the generated model is also performed.

3.3 Rectangular cuboid approximation of point clouds based on density

In this section the pseudo-code in Fig. 5 is evaluated on simulation and real world datasets. The aim of this approach is to develop memory efficient variable sized RC approximations based on point cloud density. This section is divided into two subsections. The first subsection deals with a simulation dataset which shows how the algorithm works by varying the approximation threshold. The second subsection evaluates the approximation approach on real world datasets based on memory complexity in comparison to point cloud and fixed resolution cell representations.

3.3.1 Simulation dataset

In this section an evaluation of the approximation algorithm on a simulation dataset is presented. Figure 11 shows the approximation of a sphere using RC in which σ defines the detail of the approximation. As can be seen in the figure that as σ is increased, the RC split further generating a more accurate representation. Figure 12a, b show the approximation time taken by the pseudo-code of Fig. 5 and the increasing number of RC generated as a function of the approximation threshold. In case of $\sigma = 98\%$ around 1500 RC are required to approximate the sphere.

3.3.2 Real world datasets

In order to evaluate the approximation approach, a memory consumption analysis is performed on point cloud datasets in which the majority of the structure is axis aligned. Table 1 shows a comparison of different approaches on the Freiburg campus (10 scans) and Bremen city center dataset (4 scans).

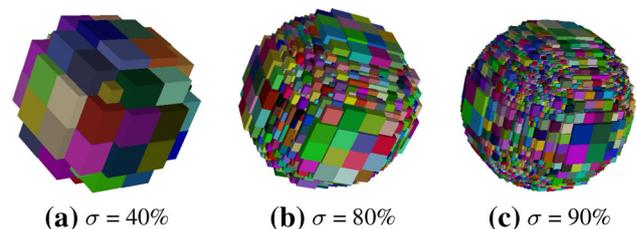


Fig. 11 A simulation dataset showing the RC approximation with different approximation thresholds σ . Increase in σ causes the RC to split in order to generate a finer approximation

Fig. 12 Computation time and number of RC as a function of σ for the sphere. An increase in σ causes an increase in computation time and number of RC required for representation

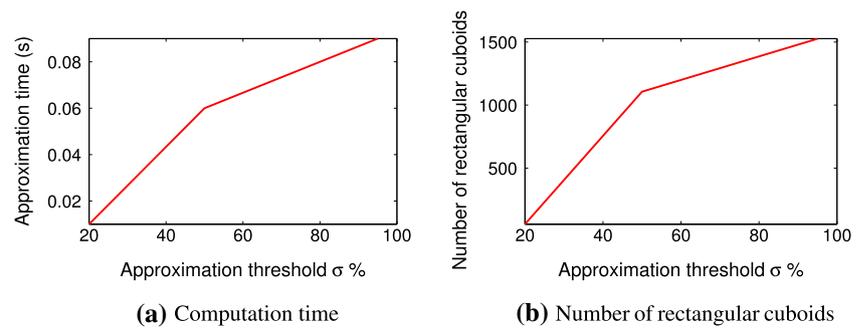


Table 1 Experimental results of the RC approximation for 10 scans of Freiburg campus and 4 scans of Bremen city center dataset

Approach	Resolution (cm)/ σ	Dataset memory (MB)	
		Freiburg campus	Bremen city center
Point cloud	–	18.73	119.98
Modified octomap	20	10.53	83.57
	40	3.06	25.59
	60	1.41	12.79
	80	0.85	7.82
Modified octomap-pruned	20	9.89	81.02
	40	2.86	24.75
	60	1.27	12.35
	80	0.77	7.54
Variable resolution RC	$\sigma = 95\%$	0.09	0.18
	$\sigma = 98\%$	0.18	0.45
	$\sigma = 99\%$	0.25	0.59
Storing variable resolution RC in Rtree (M = 8)	$\sigma = 95\%$	0.36	0.69
	$\sigma = 98\%$	0.72	1.76
	$\sigma = 99\%$	0.98	2.29
Storing variable resolution RC in Rtree (M = 16)	$\sigma = 95\%$	0.32	0.63
	$\sigma = 98\%$	0.66	1.61
	$\sigma = 99\%$	0.89	2.11
Storing variable resolution RC in Rtree (M = 32)	$\sigma = 95\%$	0.31	0.60
	$\sigma = 98\%$	0.64	1.55
	$\sigma = 99\%$	0.86	2.00

The comparison shows that even for high approximation threshold σ the RC approximation requires a small amount of memory. A visual comparison shows that the proposed approach captures similar details and generates memory efficient 3D approximations for point cloud with axis aligned structures. The Octomap implementation was modified to update based on beam end points only, hence ignoring the beam path. The modified Octomap implementation can be considered as the storage of fixed resolution cells in a hierarchy

The Bremen city center dataset is downsampled using a grid of 5 cm resolution leading to 10484513 points in the first 4 scans of the dataset. The point cloud memory is calculated by storing 3 floats for each point leading to 12 bytes. To perform a comparison the Octomap implementation was modified to update based on beam end points only, hence ignoring the beam path. The modified Octomap implementation can be considered as the storage of fixed resolution cells in a hierarchy. It can be seen from Table 1 that even for high approximation thresholds (σ), the variable resolution RC approximation can generate memory efficient representations in comparison to other approaches. The memory of the variable resolution representation was calculated by stor-

ing 6 integers for each RC, leading to 24 bytes. The RC generated by the approximation approach can be stored in the Rtree structure or without a hierarchy in a list or vector. If the objective is 3D reconstruction or visualization, the RC generated by the density based approximation approach can be stored in vector or a list. The visualization process would involve accessing all elements of the vector/list and displaying them. In case spatial queries have to be carried out for (small) specific regions it is better to store the RC in the Rtree structure. The last 3 rows of Table 1 show the memory consumption of storing the RC generated by the approximation approach in a Rtree structure for different values of M (number of branches per node).

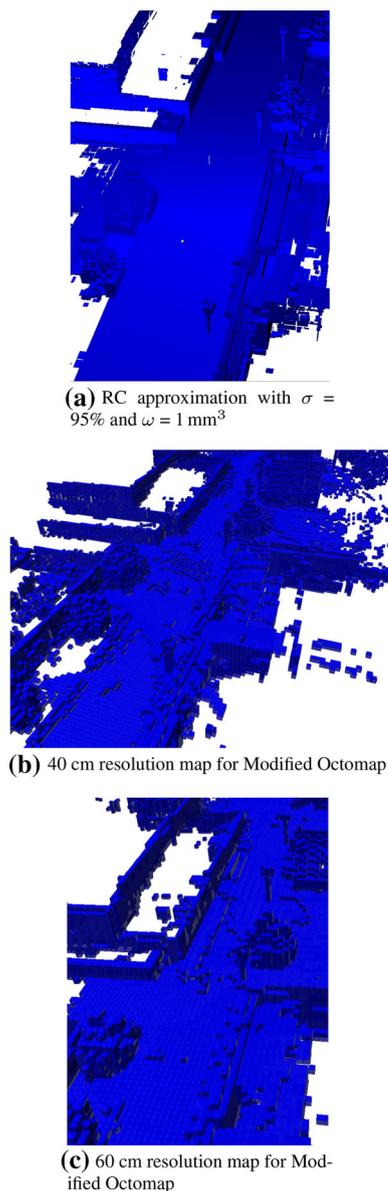


Fig. 13 a–b A visual comparison of the approximation generated by the RC approximation approach and Modified Octomap approach at 40 and 60 cm resolution. Focusing on high density axis aligned regions it can be seen that the proposed approximation captures all important details and requires less memory than its counterpart. Best viewed in color

It is difficult to define a metric that compares 3D representations which differ drastically (without ground truth). Hence a visual comparison of the RC approximation and fixed resolution cells was carried out as shown in Fig. 13. The figure shows a comparison of $\sigma = 95\%$ with the 40 and 60 cm resolution map generated by the modified Octomap implementation. It can be seen that the approximation captures similar details with 0.09 MB (memory of RC only) or 0.36 MB, 0.32 MB and 0.31 MB for $M = 8$, $M = 16$ and $M = 32$ (storing RC in Rtree) in comparison to 3.06 MB and

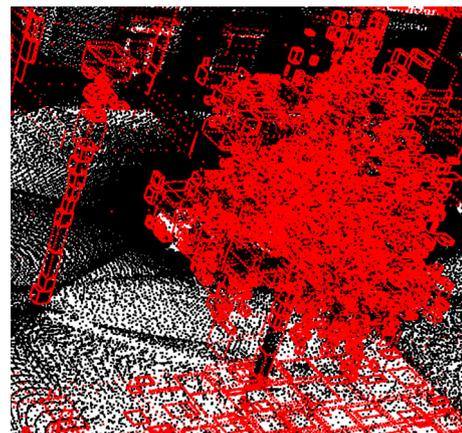


Fig. 14 RC approximation of a tree and pole. The approach is able to capture essential details and approximate the pole and the tree based on point cloud density

1.41 MB taken by the modified Octomap approach. Although it is difficult to compare the accuracy of both approaches, by focusing on high density and axis aligned surfaces it can be stated that the RC approximation generates memory efficient representations.

Figure 14 shows the RC approximation of a specific section (a tree and pole) of the Freiburg campus using a different visualization. It shows the RC and the point cloud in red and black respectively. The figure shows that the algorithm adapts the size of the RC based on the density of the point cloud to capture essential details. Figure 15a shows the first 10 scans of the Freiburg campus for $\sigma = 98\%$. The coarse approximation generated by non axis aligned regions can be visualized in Fig. 15b, specifically the approximation of the buildings. In parts of the environment that are non axis aligned, the RC split until the density threshold and volumetric constraint are satisfied. To evaluate the effect of non-axis aligned structures on memory, the point cloud from 10 files of the Freiburg campus was rotated by an increment of 15 degrees and the memory was recorded for the same approximation threshold as shown in Fig. 16. It can be seen that a larger number of RC are required to represent non axis aligned environments.

The approach is also tested for approximation of dense point clouds with complicated shapes such as the stanford repository Dragon data set³. Figure 17 shows the approximation of the dragon for different approximation thresholds σ , leading from a coarse to a finer approximation. Figure 18a–c shows the normalized (per point) approximation time, memory consumption and number of RC required for approximation. Figure 18d shows the number of unapproximated points for the dragon dataset. The computation time of

³ Courtesy of Stanford University Computer Graphics Laboratory, available at The Stanford 3D Scanning repository, <http://graphics.stanford.edu/data/3Dscanrep/>.

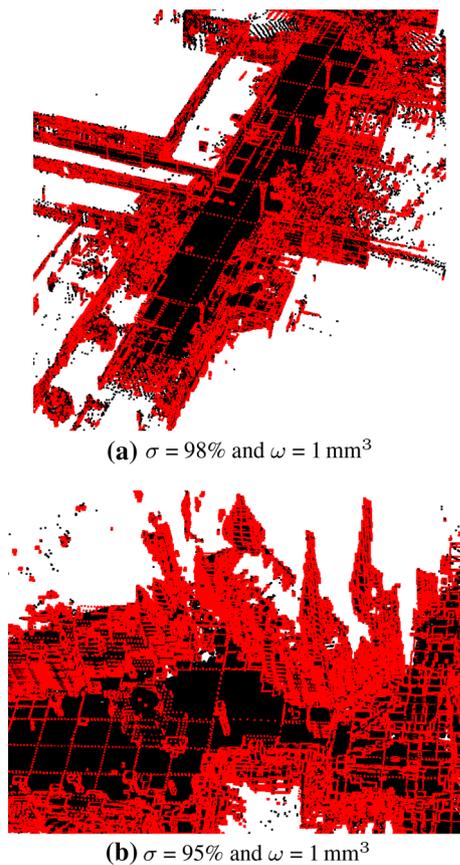


Fig. 15 RC approximation of 10 scans of Freiburg campus and 4 scans of bremen city center. **a** Visualization of the RC with the actual point cloud. It can be seen that the approach is capable of extracting variable resolution RC approximations based on density. **b** A visualization to show how the non-axis aligned structure affects the approximation. It can be seen that the non-axis aligned buildings have a rougher approximation

the RC approximation algorithm is highly dependent on the number of points and volume occupied by the point cloud.

4 Discussion and comments

This section highlights different aspects of the topics discussed in this paper.

4.1 RMAP occupancy grid

In this subsection we discuss the merits/demerits of the free space assumption in the RMAP occupancy grid for robotics applications. The important point is that as free space is assumed a priori, differentiating free and unknown space becomes tedious. This is not a major issue for most robotic applications such as localization and registration which donot require differentiation of free and unmapped space. However

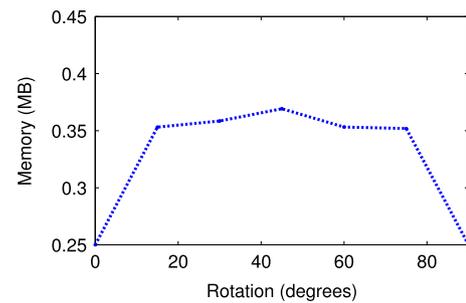


Fig. 16 Rotation effect on 10 files of Freiburg campus dataset for $\sigma = 99\%$. In case the point cloud structures are not axis aligned the RC approximation generates a rough approximation which requires more memory as can be seen in this plot

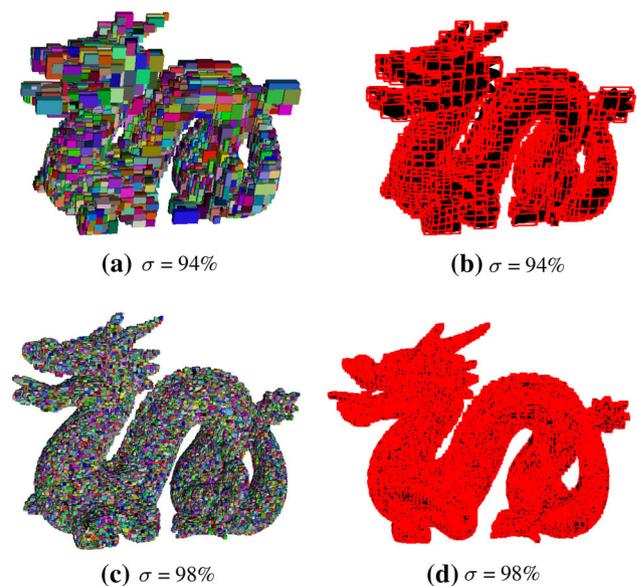


Fig. 17 a–d An evaluation of the RC approximation on a complex shape. It can be seen that as σ is increased the approximation gets finer

it can be problematic for navigation and exploration. It is briefly discussed below how these issues can be addressed.

The issue of exploration can be addressed from two perspectives. Firstly, most robotic architectures utilize two layers for navigation and exploration (global and local representations of the environment). RMAP can be used to make a global map, whereas the exploration strategy can be based on generating frontiers on the local map which can explicitly model free space and unmapped space. As the local map is a sliding window grid it is computationally and memory efficient to model free and unmapped space here. The frontiers found in the local map can then be marked on the global map to stop the algorithm from exploring already explored areas. Secondly there exist exploration algorithms (Amigoni et al. 2004) which place frontiers on the global map corresponding to a certain radius based on the robot position and the

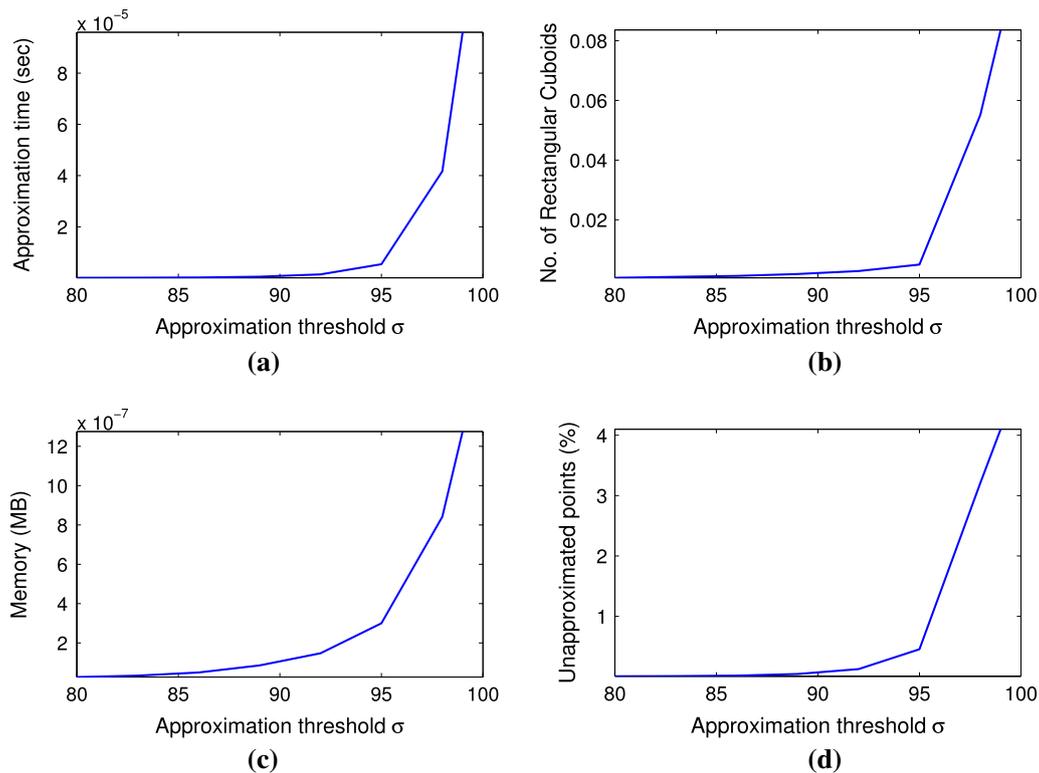


Fig. 18 a–c Normalized (per point) computation time, number of RC and memory consumption (without storing in Rtree) of the Dragon dataset as a function of approximation threshold σ . d The percentage of unapproximated points showing the information loss

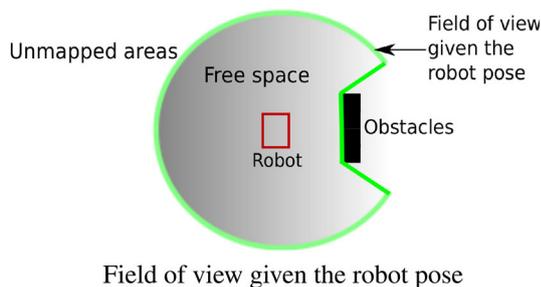


Fig. 19 Determining the map boundary. It is possible to differentiate between free and unmapped space given the robot trajectory (during 3D mapping), sensor characteristics (maximum range and FOV) and the map generated by the robot. The above information can be used to mark cells as the boundary of the map in an offline process. Once the boundary of the map has been marked, all regions beyond the boundary can be considered unmapped and the robot can be stopped from navigating in unmapped regions.

maximal FOV of the sensor and thus do not require explicit free and unknown space modelling.

In context of safe navigation the robot trajectories should be planned through regions covered by sensor measurements and should be stopped from generating trajectories in unmapped regions. In case of navigation with a known map (all explored areas) the RMAP occupancy grid is equivalent to any other grid structure. However maps containing unmapped regions can be problematic for navigation

as the proposed approach will generate optimistic trajectories in regions which are unknown assuming they are free. *In principle it is possible to differentiate between free and unmapped space given the robot trajectory (during 3D mapping), sensor characteristics (maximum range and FOV) and the map generated by robot. The above mentioned information can be used to mark cells as the boundary of the map in an offline process. Once this boundary has been marked, all regions beyond this boundary can be considered as unmapped and the robot can be stopped from navigating in these regions.* Figure 19 shows an illustration of the above mentioned details. The boundary of the map for maximum range readings can be marked by initializing cells in those regions. Another aspect specific to navigation is the occupancy probability of cells corresponding to regions containing dynamics. Consider the case in which the robot returns to a region previously mapped (implicitly) as free space. However during the current time index it contains dynamics, hence the RMAP occupancy grid will initialize grid cells in those regions whereas the standard occupancy grid will update the occupancy probability as more observations are obtained. In this case the occupancy probability will be over estimated by the RMAP approach however this is not a major issue for navigation as it will make the robot avoid these regions more than a standard occupancy grid.

4.2 Rectangular cuboid approximation based on point cloud density

The RC approximation approach utilizes density as a parameter to determine an approximation of the environment. Different sized RC are used for representation of the environment based on density. This approach is able to find RC approximations of axis aligned structures, which by intuition would consume less memory in comparison to a fixed resolution representations. Factors which affect the performance of the RC approximation of 3D environments are highlighted below:

1. Axis aligned constraint

The axis aligned constraint plays an important role in the approximation based on density. If the point cloud structure is not axis aligned in the global frame of reference the approximation will be very coarse in comparison to axis aligned regions. A natural extension of this work is an approximation based on OBB (oriented bounding box). A transition to OBB based approximation will cause an increase in computational and memory complexity. An increase in memory complexity occurs because the RMAP framework stores the minimum and maximum point of the axis aligned RC whereas in case of OBB either all corner points or transformation angles need to be stored. Additional computational cost occurs because the orientation of the point cloud structure needs to be extracted at each step. The choice between an axis aligned and OBB can be considered as a trade off between computational, memory complexity as well as accuracy.

2. Approximation percentage σ and Minimum volume threshold ω

The choice of the approximation percentage σ and the minimum volume threshold ω is highly dependent on the volume and number of points in the point cloud. As a general rule of thumb an approximation percentage σ with values within the range of 90–98 % will always yield good approximations. However forcing the threshold closer to 100 % can cause the algorithm to degenerate.

3. Density of point cloud

An important factor in the performance of the algorithm is the density of the point cloud. This factor is highly dependent on the specific sensor being used and the layout of the environment. The algorithm performs best if the point density is uniform for all scanned surface (without noisy measurements). Noisy measurements in the vicinity of high density region can cause the algorithm to over estimate the RC volume.

5 Conclusion and futurework

In this paper the RMAP framework is presented which uses axis aligned RC for 3D environment representation. This

paper presented two aspects of the RMAP framework, the RMAP occupancy grid approach and a RC approximation of point clouds based on point density. The RMAP occupancy grid with *implicit free space modelling* generates 3D probabilistic representations with multiresolution capabilities. An evaluation of the RMAP occupancy grid in comparison to Octomap is presented on the Freiburg campus dataset in context of *large scale* 3D mapping. The evaluation showed that RMAP occupancy grid generates memory efficient representations due to implicit free space modelling and allows faster access times with comparable insertion times to Octomap. In addition an evaluation of the Rtree data structure is performed in context of 3D mapping. A visual evaluation of the axis aligned RC approximation approach showed that it is capable of generating memory efficient representations for point clouds containing axis aligned surfaces. It can be stated in general that the RMAP framework presents specific advantages in terms of computational/memory complexity for 3D mapping.

Future work includes an extensive evaluation of the proposed RMAP occupancy grid in the context of exploration and navigation. Additionally an OBB based approximation algorithm will be investigated in order to determine the advantage/disadvantage in comparison to the axis aligned RC approximation.

Acknowledgments This work is part of the EU-STREP project IURO (Interactive Urban Robot), supported by the 7th Framework Programme of the European Union, ICT Challenge 2 Cognitive Systems and Robotics, contract number 248317, see also www.iuro-project.eu. Support of the TUM Institute for Advanced Study (IAS), Technische Universität München (TUM), is hereby gratefully acknowledged, see also www.tum-ias.de.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Amigoni, F., Caglioti, V., & Galtarossa, U. (2004). A mobile robot mapping system with an information-based exploration strategy. *Proceedings of ICINCO* (vol. 2, pp. 71–78).
- Biber, P., & Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. *Proceedings of IEEE International Conference on Robotics and Automation*, 3, 2743–2748.
- Dryanovski, I., Morris, W., & Xiao, J. (2010). Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1553–1559).
- Einhorn, E., Schroter, C., & Gross, H-M. (2011). Finding the adequate resolution for grid mapping-cell sizes locally adapting on-the-fly. *IEEE International Conference on Robotics and Automation (ICRA)*, (pp. 1843–1848).
- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22, 46–57.

- Figueiredo, M., Oliveira, J., Araújo, B., & Pereira, J. (2010). An efficient collision detection algorithm for point cloud models, *20th International conference on Computer Graphics and Vision* (vol. 43, p. 44).
- Guttman, A. (1984). *R-trees: A dynamic index structure for spatial searching* (Vol. 14). New York: ACM.
- Herbert, M., Caillas, C., Krotkov, E., Kweon, I. S., & Kanade, T. (1989). *Terrain mapping for a roving planetary explorer, Proceedings of International Conference on Robotics and Automation* (pp. 997–1002).
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). Octomap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*, 189–206.
- Moravec, H., & Elfes, A. (1985). High resolution maps from wide angle sonar. *Proceedings of IEEE International Conference on Robotics and Automation*, *2*, 116–121.
- Nanopoulos, A., Papadopoulos, A. N., & Theodoridis, Y. (2006). *R-trees: Theory and applications*. New York: Springer.
- Nguyen, Viet, Gächter, Stefan, Martinelli, Agostino, Tomatis, Nicola, & Siegwart, Roland. (2007). A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. *Autonomous Robots*, *23*, 97–111.
- Ryde, J., & Hu, H. (2010). 3D mapping with multi-resolution occupied voxel lists. *Autonomous Robots*, *28*(2), 169–185.
- Ryde, J., & Corso, J. J. (2012). Fast Voxel Maps with Counting Bloom Filters. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- Saarinen, J., Andreasson, H., Stoyanov, T., Ala-Luhtala, J., & Lilienthal, A. J. (2013). Normal distributions transform occupancy maps: Application to large-scale online 3D mapping. *IEEE International Conference on Robotics and Automation*.
- Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, *99*(1), 21–71.
- Thrun, Sebastian. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, *15*, 111–127.
- Triebel, R., Pfaff, P., & Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. *Proceedings of International Conference on Intelligent Robots and Systems* (pp. 2276–2282).
- Wurm, K.M., Hornung, A., Bennewitz, M., Stachniss, C., & Burgard, W. (2010). OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems, *Proceedings of the ICRA workshop on best practice in 3D perception and modeling for mobile manipulation*.



Sheraz Khan is currently pursuing his PhD (since June 2011) at Institute of Automatic Control Engineering, Technische Universität München. He received his Bachelors from National University of Sciences and Technology, Pakistan in 2007 and Masters in Advanced Robotics from Ecole Centrale de Nantes, France and Warsaw University of Technology, Poland under the European Commission scholarship program titled 'Erasmus Mundus' in 2010. His research

interests include 3D mapping, SLAM and Loop closure.



Athanasios Dometios is currently pursuing his diploma engineer degree from School of Electrical and Computer Engineering, Division of Signals, Control and Robotics, National Technical University of Athens (NTUA), Greece.



Chris Verginis is currently pursuing his diploma engineer degree from School of Electrical and Computer Engineering, Division of Signals, Control and Robotics, National Technical University of Athens (NTUA), Greece.



Costas Tzafestas is currently Assistant professor at the National Technical University of Athens, School of Electrical and Computer Engineering, Division of Signals, Control and Robotics. He received his PhD Degree from University of Pierre and Marie Curie, Paris, France in 1998. He is also an Associate editor of the *Journal of Intelligent and Robotics systems*. His research interests are primarily focused in the domain of human-robot interaction and cooperation, and more

particularly: telerobotics, virtual / augmented reality and haptics.



Dirk Wollherr is currently senior researcher and lecturer (since July 2005) and Privatdozent (since 2013) at the institute of Automatic Control Engineering, Faculty of Electrical Engineering and Information Technology, Technische Universität München, Munich, Germany. He received the diploma engineer degree in Electrical Engineering in 2000 and the Doctor of Engineering degree in Electrical Engineering in 2005 and the Habilitation degree in

2013 from the Technische Universität München, Munich, Germany. From 2001–2004 he has been research assistant at the Control Systems Group, Technische Universität Berlin, Germany. In 2004 he has been granted a research fellowship from the Japanese Society for the Promotion of Science (JSPS) at the Yoshihiko-Nakamura-Lab, The University of Tokyo, Japan. From 2006–2008 Dr. Wollherr has been General Manager of the Cluster of Excellence “Cognition for Technical Systems (CoTeSys)” and since 2005 he is principal investigator, independent junior research group leader, and research area leader within CoTeSys. He has been TUM Carl-von-Linde Junior Fellow, Institute of Advanced Studies from 2010 until 2013. Furthermore, he has been active in dissemination, e.g. General Chair of the German “Robotik 2008” conference and Finance Chair of RO-MAN 2008, as well as in several EU projects, e.g. Robot@CWE, CyberWalk, and Movement. His research interests include automatic control, robotics, autonomous mobile robots, human-robot-interaction, and humanoid walking.



Martin Buss received the Diploma Engineering degree in electrical engineering from the Technische Universität Darmstadt, Darmstadt, Germany, in 1990, and the Doctor of Engineering degree in electrical engineering from The University of Tokyo, Tokyo, Japan, in 1994. In 1988, he was a research student for one year with the Science University of Tokyo. From 1994–1995, he was a postdoctoral researcher with the Department of Systems Engineering, Australian National University,

Canberra, Australia. From 1995–2000, he was a senior research assistant and lecturer with the Institute of Automatic Control Engineering, Department of Electrical Engineering and Information Technology, Technische Universität München, Munich, Germany. From 2000–2003, he was full professor, the head of the control systems group, and the deputy director of the Institute of Energy and Automation Technology, Faculty IV, Electrical Engineering and Computer Science, Technical University Berlin, Berlin, Germany. Since 2003, he has been full professor (chair) with the Institute of Automatic Control Engineering, Faculty of Electrical Engineering and Information Technology, Technische Universität München, where he has been with the Medical Faculty since 2008. Since 2006, he has also been the coordinator of the Deutsche Forschungsgemeinschaft cluster of excellence “Cognition for Technical Systems (CoTeSys)”. His research interests include automatic control, mechatronics, multimodal human system interfaces, optimization, nonlinear, and hybrid discrete-continuous systems.