Wavelet Occupancy Grids: a Method for Compact Map Building ⋆

Manuel Yguel, Olivier Aycard, and Christian Laugier

1   ε-motion, GRAVIR-UJF-INRIA-INP Grenoble, France
    `firstname.lastname@inrialpes.fr`
2   Inria Rhône-Alpes, 655 avenue de l'Europe - Montbonnot 38334 Saint Ismier
    Cedex, France

1 Introduction

The capacity to know about the environment is a major requirement for robots and automated systems. Real scenes are complex to perceive, dynamic and large. Therefore to perceive those scenes, robots have access to complementary sensors such as sonar/laser range-finders, cameras or bumpers, but all these are always noisy. In automated navigation, the internal representation of the environment is used for all fundamental tasks: localization[14], path-planning[6], obstacle avoidance[1] or target tracking[2]. But this modelling can require huge memory space, especially when the level of abstraction is fairly low and the representation is close to the sensor data. In these cases the time of data analysis is also prohibitive, making it necessary to change sophisticated algorithms for naive ones. This paper addresses the problem of data representation and data storage for large maps, under the constraints of multi-sensor real-time updates and hierarchical representation. Thus multi-scale algorithms could be applied to very large maps with efficient calculating time.

The first step when mapping the environment is obviously to define how to map. Geometrical properties are powerful guidelines to do it. A focus on the ability to reach a point from an other leads to the class of topological maps and the associated representation is the graph of connected places. Topological maps are semantically rich, can be drawn even without an accurate localisation of the robot and their size is very small. However it is difficult to build and update them automatically and they do not allow direct complex data manipulations. The possibility to index all the spatial information in a common reference frame, via a global coordinate system, leads to the class of metric maps. In an inversely symmetrical manner, metric maps are easy

---

to build and update and they allow direct complex interactions with the environment such as obstacle avoidance, but they are semantically poor, they require an accurate localisation of the robot and the size of data is critically huge. Among metrical maps, occupancy grids (OGs) are a classical approach [9] with the essential property, that they make it straightforward to integrate noisy measurements of different sensors over time. To deal with the sensor uncertainty is a main requirement for a mapping framework, because chip sensors are very noisy. And an unified theory for sensor integration allowed sensor networks, that achieve robustness for a robotic system. Convincing new world representations have recently used topological and metric maps in intricate manners to combine their advantages [13], [5]. Now, both of these new world representations resort to OGs for precise mapping. Therefore focusing on the main drawback of OGs which is the data size is a major challenge for robotic.

The wavelet framework [3], [8] is very interesting for non linear signal approximation as images. Image and occupancy grid processing are two related subjects as pointed out by Elfes in [4]. Moreover wavelets offer a rigorous mathematical framework for multi-resolution representations and signal compression. Thus unlike mixed representations as metric-topologic ones, we propose a new unified framework for multi-resolution map building based on wavelets, which we call the wavelet occupancy grid (WavOG). Pai and Reissell [10] have shown that wavelets could be used to represent 3D static maps, and also how to devise efficient path planning through rough terrain with this kind of representation. Drawing upon this, Sinopoli *et al.* [?] have used this static wavelets representation for global path-planning while using traditional 3D updatable OG for local navigation. What we propose is a synthesis of wavelet representation and OG representation in order to do OG building in the wavelet space. And also, we describe a probabilistic formulation for every scaled information provided by the wavelet transform.

In the rest of this paper we first present the wavelet mathematical framework in simple terms. Second, we explain how to link the theoretical building of OGs with wavelets in order to on-line construct occupancy grids in wavelet space. We will then pay attention to the multi-resolution occupancy meaning of WavOGs. Fourth, we describe how WavOGs are implemented and a compact representation is obtained. This new building was validated through a mapping tour with the cycab autonomous robot. Finally we present these results in a comparison with a standard occupancy grid, which proves the compactness of the new representation.

2 The wavelet mathematical framework

The function which is discretized by a 2D-occupancy grid is the continuous occupancy function:

$$\mathbb{R}^2 \longrightarrow [0;1]$$
$$(x,y) \longmapsto p(x,y, occupied)$$

which is the probability that an obstacle lies in the point (x,y) (Fig. 1).
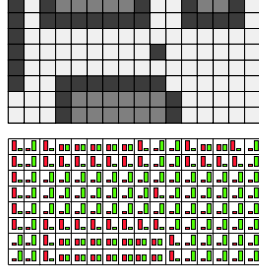

Fig. 1. Occupancy grid (in grey level) and its corresponding probability distribution. The green boxes represents the probability that the cell is empty while the red ones the probability that the cell is occupied.

The aim of this work is to compress the occupancy function. The main idea at work in several compression schemes is to project the function onto a set of elementary functions which is a basis for the *vector space* of approximation functions. For example the Fourier transformation projects the functions onto the infinite set of sines and cosines. Then, in this case, the approximation process consists in selecting a finite set of the lowest frequencies and rejects high ones, which are almost considered as noise. But this leads to poor compression results, especially for non linear functions as OGs certainly are. Indeed, There exists a similarity between OGs and images Fig. 1 and there exist some approximation spaces that are useful for this kind of signals called wavelet spaces [3].

We will now present a 1-dimensional wavelet decomposition using Mallat's algorithm [7]. Then we will present the wavelet notation and the Haar basis we use.

2.1 Example of the Haar wavelet transform in 1D

We will now focus on 1-dimensional wavelet decomposition using Haar basis.

A 1D function p, which is regularly discretized over n points $\{x_0, \ldots, x_n\}$, is seen as a vector $[p(x_0), \ldots, p(x_n)]$ (Fig. 2).

One elementary step of the Haar wavelet transform (HWT) uses two neighboring samples $p(x_i)$ and $p(x_{i+1})$ to generate a scale s_i coefficient and a detail coefficient d_i:

Table 1. elementary step of direct and inverse Haar transform

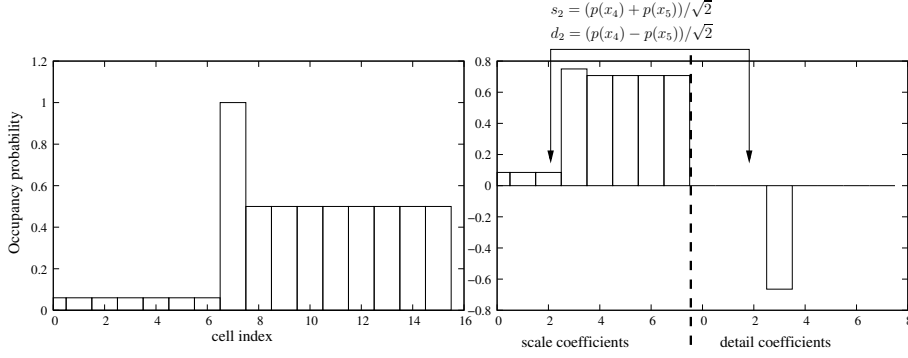| Haar wavelet transform:	Haar inverse wavelet transform :
$s_i = (p(x_{2i}) + p(x_{2i+1}))/\sqrt{2}$	$p(x_{2i}) = (s_i + d_i)/\sqrt{2}$
$d_i = (p(x_{2i}) - p(x_{2i+1}))/\sqrt{2}$	$p(x_{2i+1}) = (s_i - d_i)/\sqrt{2}$

Fig. 2. left : mono-dimensional OG: an observer in the first cell views an obstacle in the 7th cell. Rigth : first iteration of a 1D Haar transform; scaled and detail coefficients are grouped together. There is only one nonzero detail coefficient corresponding to the impacted cell.

Except for a constant, the scale coefficient is the average of the neighboring samples and the detail coefficient is the difference between them. It is clear that it exploits correlation for better encoding, because the more similar $p(x_{2i})$ and $p(x_{2i+1})$ are, the closer to zero d_i is.

The principle of the HWT is to apply the HWT elementary step recursively on the scale coefficients (Tab. 1, Fig. 2). The process can be stopped at every iteration but the best encoding is obtained when there is only one scale coefficient.

2.2 The Haar wavelet transform in 2D

It is straightforward to deduce HWT in 2 dimensions by alternating one step of the 1d wavelet transform on raws (Fig. 3(b)) and then on columns (Fig. 3(c)).

One step of the 2D HWT divides the space in 4 parts: the scale, raw detail, column detail and diagonal detail spaces (Fig. 3(c)). Then one square of 2×2 pixels in the original image produces 4 coefficients by one step of the 2D HWT (Fig. 4). These coefficients are the weights of special functions, called the Haar basis functions, in the Haar wavelet representation (Fig. 5). Performing HWT is just iterating this step on the previous scalle space until the size of the resulting scale space is one.

Wavelet notation

Here we consider the 2D Haar basis. It is built upon 2 types of basis functions: the scale and detail functions.

- The scaling mother function is of unit value over the unit square:

$$\Phi(x, y) = 1 \text{ for } (x, y) \in [0, 1]^2, \text{ zero elsewhere}$$
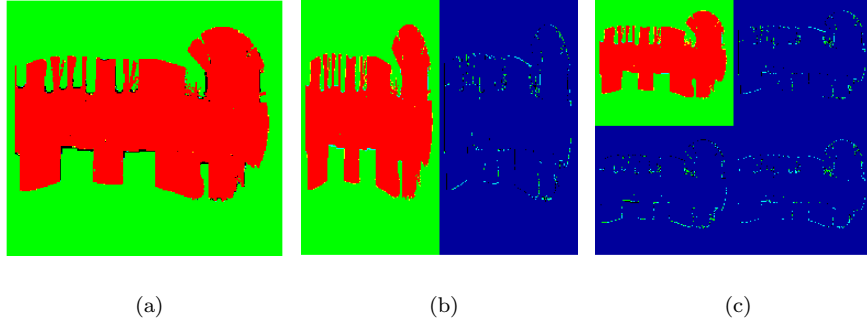
(a)   (b)   (c)

Fig. 3. One step of the Haar wavelet transform in 2d. From the original image (a): first (a): one 1d wavelet transform on each raw (b), then one 1d wavelet tranform on each column (c). In the result: from up to bottom, left to right: there are the scale, raw detail, column detail and diagonal detail spaces.

Fig. 4. 2D Haar wavelet transform and inverse transform: elementary step algorithm. The direct and inverse transform are the same algorithm.

- The three wavelet mother functions over the unit square[3]:

$$\Psi_{01}(x,y) = \quad \Psi_{10}(x,y) = \quad \Psi_{11}(x,y) =$$

Fig. 5. These graphically-defined functions are +1 where white and -1 where black in the unit square shown and implicitly zero outside that domain.

We define the Haar basis at scale s as the union of the set of scaled functions: $\{\Phi^{sij}|(i,j) \in \mathbb{Z}^2\}$ and the set of details functions: $\{\Psi_M^{lij}|l = 0,\dots,s;(i,j) \in \mathbb{Z}^2\}$. Where :

$$\Phi^{lij} = 2^{-l}\Phi(2^{-l}x - i, 2^{-l}y - j) \tag{1}$$

$$\Psi_M^{lij} = 2^{-l}\Phi(2^{-l}x - i, 2^{-l}y - j) \tag{2}$$

---

[3] the type M take three values -01, 10 or 11- corresponding to one of the three mother wavelets for horizontal, vertical and diagonal differencing.

Each triplet (s, i, j) defines a *wavelet square* at scale s and offset (i, j). The scale s is just the number of steps in the HWT. Squares at scale s are disjoint and have an area of $4^l u$ where u is the area of a square at scale 0. We use the term *coarser* to describe basis functions in square at higher scale and *finer* for those at lower scale.

Thanks to the orthogonality of the Haar basis, the weight of a basis vector function, *e.g.* Φ^{sij}, is formally given by the scalar product (noted $< . \mid . >$) with the occupancy function:

$$< p(x, y) \mid \Phi^{sij} > = \int_{x, y \in \mathbb{R}^2} p(x, y) \Phi^{sij}(x, y) dx dy \tag{3}$$

3 Occupancy Grids and Wavelets

We first present the bayesian definition of occupancy grids. As the purpose is to build dynamic maps, we show how occupancy grids are updated. Then as the wavelet framework is a vector space framework wich special constraints, we show how to combine OGs and wavelets to perform updates in wavelet space.

We will now introduce OGs as a statistical representation of the distribution of obstacles over space (Fig.1). Each cell (x) corresponds to a binary random variable (E_x) whose value is either *occupied* (occ) or *empty* (emp)[4]. Each cell is considered independent from all others, which makes it possible to define the consequences of an observation for each specific cell. This leads to efficient update computation, linear in the number of cells in the sensor range.

3.1 Relation between sensor measurement and cell occupancy

To build a robust modelisation, we use a statistical framework to capture measurement uncertainty. Let Z be a random variable which takes values among all possible sensor values. Let $P(Z|E_x)$ be the probability distribution over Z knowing the occupancy state of cell x; $P(Z|E_x)$ is called the sensor model (Fig. 6(a) for E_x "occupied" and Fig. 6(b) for E_x "empty").

We define the joint distribution over Z and E_x as $P(Z, E_x) = P(E_x)P(Z|E_x)$. For a new sensor measure z_t, the Bayes rule gives:

$$p(e_x|z_t) = \frac{p(e_x)p(z_t|e_x)}{p(z_t)} \tag{4}$$

---

[4] We use capital letters for random variables and normal case letters for their realisation. We use the P capital letter for probability distributions and the p normal case letter for probability values.

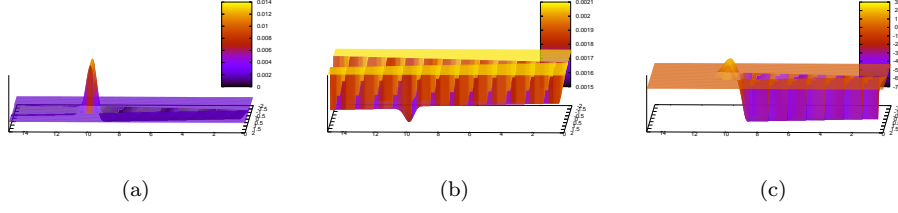(a)   (b)   (c)

Fig. 6. Sensor models with a laser-range finder located in $(0, 0)$ the sensor measures an impact in $(0, 10)$. (a) Sensor model for occupied cells, (b) Sensor model for empty cells, (c) Log ratio of previous sensor models.

where the measurement probability is equal to a marginalisation term:

$$p(z_t) = p(occ)p(z_t|occ) + p(emp)p(z_t|emp) \qquad (5)$$

So from an occupancy state $P(E_x)$ at time $t : p_t(e_x)$, a new observation gives a new occupancy state $P(E_x|[Z = z_t])$. This process defines a bayesian filter which can be indexed by time:

$$p_{t+1}(e_x) = p(e_x|z_t) = \frac{p_t(e_x)p(z_t|e_x)}{\sum_{e_x} p_t(e_x)p(z_t|e_x)} \qquad (6)$$

In a mapping process the robot gets a series of observations $\mathcal{Z} = \{z_0, z_1, \ldots, z_t\}$. Therefore $p_t(e_x)$ represents $p(e_x|z_0, \ldots, z_{t-1})$ in the above equation. The process is initialized with a prior for $P_{t=0}(E_x)$, it could be the map obtained by a previous mapping process or uniform distribution if the aera is unknown. Thus Eq. 6 provides a general framework for updating a map each cell apart.

3.2 Logarithmic-form for occupancy grid updates

We will now link OGs to wavelets. We have seen that we can project a huge representation of a function in a wavelet *vector* space in order to compress it. However an occupancy update (Eq. 6) is not a linear operation. Thus, we present now a well known logarithmic transformation which makes it possible to perform the entire update operation with only sums. Since $p_t(occ) = 1 - p_t(emp)$, we can summarize $p_t(occ)$ and $p_t(emp)$ with only one number q_t:

$$q_t = p_t(occ)/p_t(emp) \qquad (7)$$

As can be seen[5], $p_t(occ)$ and $p_t(emp)$ are easy to compute from q_t.

---

[5] $p_t(occ) = \frac{q_t}{1+q_t}$ and $p_t(emp) = \frac{1}{1+q_t}$

This leads to the elimination of the marginalisation term in (6), so that the time process appears to be:

$$q_t = \frac{p(z_{t-1}|occ)}{p(z_{t-1}|emp)}q_{t-1} = q_0 \prod_{i=0}^{t-1} \frac{p(z_i|occ)}{p(z_i|emp)} \qquad (8)$$

where q_{t-1} was recursively evaluated.

Now, the products can be changed into sums by using a logarithm. Let $odds_t = log(q_t)$ then it leads to *log-form* expressions of the occupancy grid updates:

$$odds_t = \log(\frac{p(z_{t-1}|occ)}{p(z_{t-1}|emp)}) + odds_{t-1} = odds_0 + \sum_{i=0}^{t-1} \log(\frac{p(z_i|occ)}{p(z_i|emp)}) \qquad (9)$$

In (9) $\log(\frac{p(z_{t-1}|occ)}{p(z_{t-1}|emp)})$ is the observation term of the update; we note it $Obs(z_{t-1})$. $odds_{t-1}$ is the a priori term so that the observation term corrects it.

Thus, updating a WavOG comes down to adding the wavelet transform of observation terms to the wavelet representation of the map.
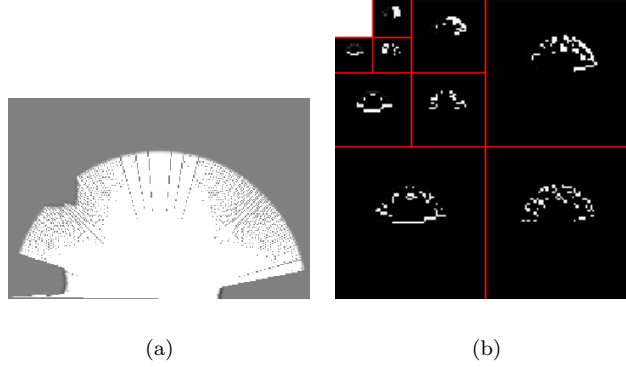

(a)   (b)

Fig. 7. mapping obtained by a single laser range-finder in OG and WavOG: (a) classical OG, (b) the corresponding 3 first detail spaces of the WavOG.

4 Multi-resolution

The wavelet representation is a hierarchical one, and WavOGs are represented from the coarser scales to the finer. So the coarse resolution information must be interpreted in order to use multi-scale algorithms.

Let us analyse the scaled coefficients of a WavOG. As we have seen in sec. 2, it is given by a scalar product eq. (3). It is the integral of a product between the log-form of the occupancy grid function and the scale basis function $\Phi_s^{(i,j)}$, where s indexes the scale.

$$< \sum_x \log(\frac{p(occ)}{p(emp)}(x))\delta_x | \Phi_s^{(i,j)}(x) > = \sum_x log(\frac{p(occ)}{p(emp)}(x)).\Phi_s^{(i,j)}(x)$$

$$= \log(\frac{\prod_x p([E_x = occ])^{\Phi_s^{(i,j)}(x)}}{\prod_x p([E_x = emp])^{\Phi_s^{(i,j)}(x)}}) = k\log(q^s) \qquad (10)$$

In the case of the Haar wavelet basis (Eq. 2) $\Phi_s^{(i,j)}$ is constant over a square S and zero elsewhere. Let k this constant[6]:

$$k\log(q^s) = k\log(\frac{\prod_{x \in S} p([E_x = occ])}{\prod_{x \in S} p([E_x = emp])}) \qquad (11)$$

$$q^s = \frac{\prod_{x \in S} p([E_x = occ])}{\prod_{x \in S} p([E_x = emp])} \qquad (12)$$

let us define an aera of n cells: $\mathcal{A} = \{c_0, \ldots, c_n\}$, then let us define 2 events: e^{full} when all the cells in $\mathcal{A}$ are occupied and e^{open} when all the cells in $\mathcal{A}$ are empty. So the weight for the scaled function leads us immediately to the probability of e^{full} or e^{open} when the system must make a choice between those two possibilities:

$$P(e^{open}|e^{open} \vee e^{full}) = \frac{\prod_{x \in \mathcal{A}} p([E_x = emp])}{\prod_{x \in \mathcal{A}} p([E_x = occ]) + \prod_{x \in \mathcal{A}} p([E_x = emp])} = \frac{1}{1 + q^s}$$

and in a symetrical way, $P(e^{full}|e^{open} \vee e^{full})$ is obtained. So each resolution of the map provides a continuous information about the occupancy of the map.

The pixel (i, j, s) in the scale space of a WavOG is proportional to the mean over the square (i, j, s)[7] of the log-ratio occupancy values (Fig. 8).

5 Implementation

We have implemented a classical mapping process using a WavOG with the Cycab. The Cycab is an autonomous robot devoted to urban transport equiped with a laser range-finder : SICK LMS-291. In the mapping loop, there is first a localisation step using a SLAM algorithm and a second step where the map is updated from the current position using sensor values. We use the classical SLAM algorithm [11] to get the absolute position.

---

[6] with value: 2^{-s}.

[7] The square has is up left corner at coordinate: $(2^s i, 2^s j)$ and has a side of 2^s pixels.

(a)   (b)   (c)   (d)

Fig. 8. Scaled view of OG provided in WavOG representation (a) scale $\frac{1}{16}$: 192×128 pixels, $s = 2$. (b) scale $\frac{1}{64}$: 96×64 pixels, $s = 3$. (c) scale $\frac{1}{256}$: 48×32 pixels, $s = 4$. (d) scale $\frac{1}{1024}$: 24×16 pixels, $s = 5$.

The laser range-finder has an 8-meter range. The side of the finer scale is $6.25cm$ then sizes double over 5 scales. Thus the side of the coarsest cells is 1 meter and so there still exist many coarser cells which appear totally open at this scale.

We consider a square window which encloses the sensor view. We maintain an area of this size as a buffer. As long as the laser impacts belong to this window, the updates occur in this buffer. Then we apply a 2D Haar wavelet transform to the buffer and add it to the WavOG. The buffer is flushed and the process is repeated. Waiting for an entire sick scan at least before doing the wavelet transform gives a large enough area to see regular fields appear. So that the result of the wavelet transform is sparse and it is only necessary to perform wavelet transforms at distant time intervals.

After experimental studies we set a compression threshold which is a compromise between data fitting and sparsity. *i.e.* after updating a wavelet detail coefficient, if it is lower than this threshold, it is removed from the WavOG.

6 Results

As a validation of our method we compare the number of cells in a classical occupancy grid with a WavOG (Fig. 9(a), Fig. 9(b)).

Fig. 9 shows the results obtained on the car park of INRIA. These experimental results clearly shows that we have obtained a significant reduction of the size of the model (about 80% relatively to the *OG* model), and that the interesting details are still represented (such as the beacons represented by dark dots in Fig. 9(c)). It should be noticed that the coarser model give a quite good representation of the empty space (see Fig. 9(d)); this model could be used for path planning, and refined when necessary using the wavelet model. In the previous experiments, the map building has been done in real-time. This kind of compression is however weaker than we expected, the reason seems to be du to the logarithm use. Indeed the very weak probabilities which must appear to be the same in human eye, are very different in logarithm space *e.g.* consider
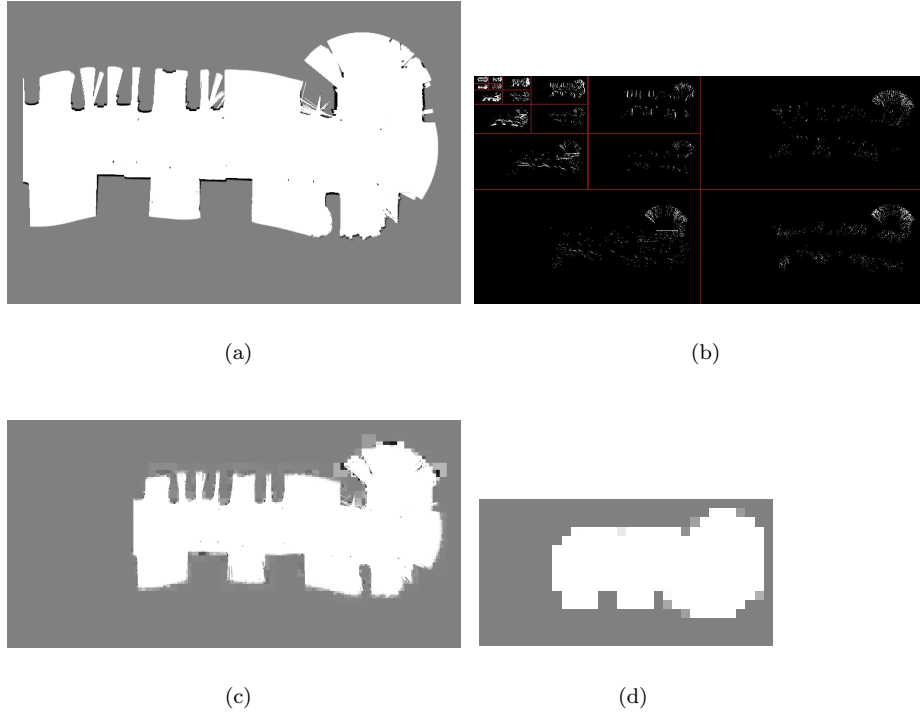
(a)

(b)

(c)

(d)

Fig. 9. (a) classical occupancy grid : $393,126$ cells. (b) wavelet occupancy grid : $78,742$ cells. (c) reconstructed occupancy grid from the WavOG. (d) Semantics for the coarser scaled of the WavOG.

the difference between 10^{-100} and 10^{-50} is huge in log space whereas it is negligible in a standard representation of probabilities. Then two conclusions arise: one it is necessary to work with bounded probabilities, which make the map reactive and easy to compress in wavelet space, two we must observe homogeneously near area such as the duration of observation of a cell don't make a bias in the map representation and then in the map compression. So in the development of this framework, we plan to study systematic methods to achieve both of these tasks.

7 Conclusion and future works

This paper introduces the structure of wavelet occupancy grids (WavOGs) as a tool for storing occupancy grids in a compact way. We have shown that WavOGs provide a continuous semantics of occupancy through scaled spaces. In accordance with the theoretical properties of wavelets, our experiments have validated that WavOGs allow major memory gains. WavOG as a com-

pact multi-scaled tool provides an efficient framework for the various algorithms that use OGs such as robot navigation, spatio-temporal classification or multiple target-tracking. In future works we plan to apply WavOGs to the monitoring of urban traffic over large areas.

References

1. J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, - 1989.
2. C. Coué, T. Fraichard, P. Bessière, and E. Mazer. Using bayesian programming for multi-sensor multi-target tracking in automotive applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2003.
3. I. Daubechies. *Ten Lectures on Wavelets*. Number 61 in CBMS-NSF Series in Applied Mathematics. SIAM Publications, Philadelphia, 1992.
4. Alberto Elfes. Multi-source spatial data fusion using bayesian reasoning. In M. A. Abidi and R. C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, chapter 3, pages 137–163. Academic Press, 1992. ISBN 0-12-042120-8.
5. B. J. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, (119):191–233, 2000.
6. Maxim Likhachev, Geoffrey J. Gordon, and Sebastian Thrun. Ara*: Anytime a* with provable bounds on sub-optimality. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
7. Stéphane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.
8. Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
9. Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, July/August 1988. ISSN:0738-4602.
10. D. K. Pai and L.-M. Reissell. Multiresolution rough terrain motion planning. In *IEEE Transactions on Robotics and Automation*, volume 1, pages 19–33, February 1998.
11. C. Pradalier and S. Sekhavat. Simultaneous localization and mapping using the geometric projection filter and correspondence graph matching. *Advanced Robotics*, 2004.
12. Bruno Sinopoli, Mario Micheli, Gianluca Donato, and T. John Koo. Vision based navigation for an unmanned aerial vehicle. In *Proceedings of the International Conference on Robotics Automation*, May 2001.
13. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 1(99):21–71, 1999.
14. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. joint issue with Autonomous Robots 5.