

3D Forward Sensor Modeling and Application to Occupancy Grid based Sensor Fusion

Kaustubh Pathak, Andreas Birk, Jann Poppinga, and Sören Schwertfeger

Abstract—This paper presents a new technique for the update of a probabilistic spatial occupancy grid map using a forward sensor model. Unlike currently popular inverse sensor models, forward sensor models can be found experimentally and can represent sensor characteristics better. The formulation is applicable to both 2D and 3D range sensors and does not have some of the theoretical and practical problems associated with the current approaches which use forward models.

As an illustration of this procedure, a new prototype 3D forward sensor model is derived using a beam represented as a spherical sector. Furthermore, this model is used for fusion of point-clouds obtained from different 3D sensors, in particular, time-of-flight sensors (Swiss-ranger, laser range finders), and stereo vision cameras. Several techniques are described for an efficient data-structure representation and implementation. The range beams from different sensors are fused in a common local Cartesian occupancy map. Experimental results of this fusion are presented and evaluated using Hough-transform performed on the grid.

I. INTRODUCTION

Mobile robot based mapping using occupancy grids is a widely used method at present [1], [2]. The aim of this paper is to come up with a scheme of map update which allows for easy incorporation of a given sensor’s physical characteristics. Not only is the resulting map more accurate, but it also can be fused more systematically with another map generated using a sensor based on an entirely different operating principle. This applies in particular to the variety of 3D sensors currently in use.

Sec. I-A introduces the basic terminology. The current state of the art and its shortcomings are touched upon in Sec. I-B. The novel scheme for reformulation of forward sensor model and its corresponding update equations are presented in Sec. II. Application of this scheme to currently popular 3D sensors is done in Sec. III, which includes a general sensor model formulation and grid based sensor fusion. Efficiency issues of implementation are covered in Sec. IV. Sec. III presents some experimental results, followed by conclusion.

A. Basic Terminology and Formulation

In the occupancy grid paradigm, the state to be estimated consists of a discrete map which is a collection of cells $\mathcal{G} \triangleq [m_i], i = 1 \dots N_{\text{map}}$, each of which can be occupied $m_i = 1$, or unoccupied $m_i = 0$. This state-space is binary and also static for a given robot location and orientation.

This work was supported by the German Research Foundation (DFG).

The authors are with the Dept. of Electrical Engineering and Computer Science, Jacobs University Bremen, 28751 Bremen, Germany. (k.pathak, a.birk)@jacobs-university.de

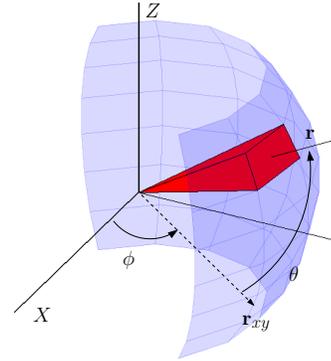


Fig. 1. Spatial beam parameterization. \mathbf{r}_{xy} refers to the beam’s projection on the XY plane.

Therefore, it is commonly estimated directly using the binary static version of the Bayes filter [3].

The probability that a particular cell is occupied is $p(m_i = 1) \equiv p(m_i)$. The probability that the cell is unoccupied is $p(m_i = 0) \equiv p(\neg m_i) \equiv 1 - p(m_i)$. We write vectors and lists in bold. Therefore, $\mathbf{p}(m) \equiv [p(m_i)]$. In this paper, we use only probability mass functions (pmf’s), also called discrete probability distribution functions (d-pdf’s).

Each sensor observation is a list $\mathcal{Z} = [\mathbf{r}_k = r_k \hat{\rho}_k \mid k = 1 \dots N_{\text{beams}}]$ of the range r_k along various lines-of-sight (LOS) directions (beams) denoted by unit vectors $\hat{\rho}_k$. A general beam, as shown in Fig. I, would be denoted without the subscript as $\mathbf{r} = r \hat{\rho}$. Each such beam carries information about only a list $\mathbf{m}(\mathbf{r}) = [m_i \mid i = 1 \dots N_{\text{cells}}]$ of cells, and $\mathbf{m}(\mathbf{r}) \subseteq \mathcal{G}$. The parameter N_{cells} would be different for different beams. The list $\mathbf{m}(\mathbf{r})$ is arranged in *ascending order* of the distance to the sensor.

We recall some standard definitions [1]. The conditional probability $p(\mathbf{r}_k \mid m_i)$ is called the *forward sensor model*. It gives the probability of observing a reading \mathbf{r}_k , when the state of the i th cell is m_i . Similarly, $p(m_i \mid \mathbf{r}_k)$ is the *inverse sensor model*, since it goes from effects (\mathbf{r}_k) to causes (m_i). Forward models have the advantage that they can be easily determined experimentally and can characterize a sensor in a more straightforward manner. A classical forward beam model applicable to laser range finders (LRFs) is given in [1] and plotted in Fig. 2(a).

B. Popular Cell-update Approaches and their Disadvantages

Let us consider a list of cells $\mathbf{m}(\mathbf{r})$ along a beam. Elfes [4] assumed that the occupancy probability of each cell in the grid is independent of the other. We make the same

assumption and proceed as in [3]. $p(m_i)$ already reflects the information in the last sample \mathbf{r}^- at time instant $t - 1$. We now wish to update $p(m_i)$ using the new sample \mathbf{r}^+ at time instant t . The sensor samples are also considered independent of each other, so $p(\mathbf{r}^+ | m_i \wedge \mathbf{r}^-) = p(\mathbf{r}^+ | m_i)$. From Bayes theorem, we get

$$p(m_i | \mathbf{r}^+ \wedge \mathbf{r}^-) = \frac{p(\mathbf{r}^+ | m_i)p(m_i | \mathbf{r}^-)}{p(\mathbf{r}^+ | m_i)p(m_i | \mathbf{r}^-) + p(\mathbf{r}^+ | \neg m_i)p(\neg m_i | \mathbf{r}^-)} \quad (1)$$

One current approach is to compute the forward model $p(\mathbf{r}^+ | m_i = 1)$ using a formalism such as that given in [1]. This model has two disadvantages: Firstly, it inherently assumes that the beam-range \mathbf{r} returned by a sensor when cell m_i is occupied is not influenced by the occupancy state of other cells. This is not true because if a cell $m_j, j < i$ is occupied, it will short circuit m_i so that $p(\mathbf{r} | m_i = 1) \rightarrow 0$. The model attempts to partially heuristically rectify this by having an exponentially decreasing term. Secondly, (1) has the term

$$p(\mathbf{r}^+ | \neg m_i) \neq 1 - p(\mathbf{r}^+ | m_i). \quad (2)$$

This term is quite difficult to interpret and compute in practise. This issue seems to be mostly unaddressed in the literature.

Another approach suggested in [5] would be to consider all possible permutations of cell occupancies $\mathcal{X}(m_i) = [m_1, \dots, m_{i-1}, m_{i+1}, \dots, m_{N_{\text{cells}}}]$, $m_j = 0/1$, as follows

$$p(\mathbf{r}^+ | m_i) = \sum_{\mathcal{X}(m_i)} p(\mathbf{r}^+ | m_i \wedge \mathcal{X}(m_i))p(\mathcal{X}(m_i) | m_i).$$

There are $2^{N_{\text{cells}}-1}$ such permutations and it is computationally expensive to use this definition. In both of the above approaches, the authors finally resort to using a reverse sensor model by reformulating (1) to get

$$\frac{p(m_i | \mathbf{r}^+ \wedge \mathbf{r}^-)}{p(\neg m_i | \mathbf{r}^+ \wedge \mathbf{r}^-)} = \frac{p(m_i | \mathbf{r}^+) p(\neg m_i)}{p(\neg m_i | \mathbf{r}^+) p(m_i)} \frac{p(m_i | \mathbf{r}^-)}{p(\neg m_i | \mathbf{r}^-)} \quad (3)$$

This has the advantage that hard to compute terms like (2) cancel out. However, it loses the ability to use a forward sensor model which can be experimentally found. Finding an inverse sensor model is difficult: refer to [1] for a neural-network based method for estimating a sensor reverse model.

A third approach used by [2, Eqs. (8,9,10)] is to use the following formula, where $\bar{p}(x) \equiv 1 - p(x)$.

$$p(m_i | \mathbf{r}^+ \wedge \mathbf{r}^-) = \frac{p(\mathbf{r}^+ | m_i) p(m_i | \mathbf{r}^-)}{p(\mathbf{r}^+ | m_i)p(m_i | \mathbf{r}^-) + \bar{p}(\mathbf{r}^+ | m_i)\bar{p}(m_i | \mathbf{r}^-)} \quad (4)$$

On comparing this with (1) and noting (2), it can be concluded that (4) is incorrect.

Finally, the approach of Thrun [6] which uses forward models employs a time-consuming iterative Expectation-Maximization (EM). Furthermore, it was designed for application to wide beam-cone sensors like the SONAR.

One of the main contributions of this paper is to come up with a modified definition of sensor forward model and a systematic scheme for beam cell-probability update which does not have the disadvantages noted in the other approaches noted in this section.

II. REDEFINING FORWARD SENSOR BEAM MODEL

A. A Cell Occupancy Probability Mapping

We present here an alternate definition of the sensor forward model, partially based on [4]. Instead of dealing directly with the cell occupancy probabilities, we introduce intermediate variables $B_i, i = 1 \dots N_{\text{cells}} + 1$ whose probabilities have a one-to-one nonlinear mapping $\mathbb{R}^{N_{\text{cells}}+1} \rightarrow \mathbb{R}^{N_{\text{cells}}}$ to those of the cells' occupancy. These variables are analogous to the correspondence variables of [6]: however, unlike that work, our approach does not require EM iterations for cell occupancy updates.

Definition 2.1 (B_i): For a list $\mathbf{m}(\mathbf{r})$ of cells, B_i is defined as the *event* that the cell m_i is occupied but all cells before it are unoccupied. The state of the cells after the i th cell is irrelevant.

$$B_i \triangleq \underbrace{000 \dots 000}_{i-1 \text{ free cells. } m_i = 1} \underbrace{1}_{m_i = 1} \underbrace{xxx \dots xxx}_{\text{Short-circuited cells.}} \quad (5)$$

In this formulation, $i = 1 \dots N_{\text{cells}} + 1$ and $B_{N_{\text{cells}}+1} \equiv 000 \dots 000$, i.e. the event that all cells are unoccupied.

We define the operator $\|B_i\|$ to return the radial distance of the center of the cell m_i from the sensor origin. This is the range an ideal sensor will return if $p(B_i) = 1$. \triangleleft

The events $B_i, i = 1 \dots N_{\text{cells}} + 1$ have the following properties.

- 1) They are *exhaustive* and *mutually exclusive*. Therefore, their probabilities $p(B_i)$ are such that

$$p\left(\bigcup_{i=1}^{N_{\text{cells}}+1} B_i\right) = \sum_{i=1}^{N_{\text{cells}}+1} p(B_i) = 1. \quad (6)$$

- 2) The occupancies of all the cells are assumed to be *independent* of each other. This means that $p_{i \neq j}(m_i \wedge m_j) = p(m_i)p(m_j)$. The probabilities $p(B_i)$ have a one-to-one relationship with the cell occupancy probabilities $p(m_i)$. This can be stated as

$$p(B_i) = p(m_i) \prod_{j=1}^{i-1} p(\neg m_j), i = 1 \dots N_{\text{cells}},$$

$$p(B_{N_{\text{cells}}+1}) = \prod_{j=1}^{N_{\text{cells}}} p(\neg m_j). \quad (7)$$

- 3) Note that the event B_i does not contain any information about the occupancy status of the cells $m_j, j > i$. This can be seen when we invert (7). After performing some algebra, one gets

$$p(m_i) = p(B_i) \left[1 - \sum_{j=1}^{i-1} p(B_j) \right]^{-1}. \quad (8)$$

If a denominator number k in Eqs. (8) vanishes, all the subsequent ones vanish also. *This means that there is no information in the list $\mathbf{p}(B)$ about the probabilities of those cell numbers $i > k$, and they are unaffected.* This is another advantage of using the intermediate variables B_i .

B. Modified Forward Sensor Beam Model

In light of the reasons outlined in Sec. I-B, our motivation for the definitions of events B in Sec. II-A becomes clear, namely, that they are exhaustive and mutually-exclusive, and that they capture the state of *all* the cells in the beam at a time. In terms of these new variables, the *forward sensor beam model* may be written as a *pmf* defined on the discrete set $\mathcal{D} \triangleq \{\|B_j\|\hat{\rho} \mid j = 1 \dots N_{\text{cells}}\}$.

$$p(\mathbf{r} \in \mathcal{D} \mid B_i), \quad i = 1 \dots N_{\text{cells}} + 1. \quad (9)$$

Conveniently enough, we can still use some of the classical beam models like the one described in [1]. We would just need to reinterpret $p(\mathbf{r} \mid m_i = 1)$ to mean $p(\mathbf{r} \mid B_i)$. What *has* changed, however, is that we cannot update the cell probabilities the old way. This is shown next.

C. Beam Occupancy Probability Update

The event space B is not binary but has $N_{\text{cells}} + 1$ members. Therefore, the cell probability update equation is now different. When we want to update the occupancy probabilities of cells $\in \mathbf{m}(\mathbf{r}^+)$ along a new sensor sample \mathbf{r}^+ , given the previously computed values $p(m_i \mid \mathbf{r}^-)$, we need the following steps.

- 1) Compute $p(B_i \mid \mathbf{r}^- \in \mathcal{D})$, $i = 1 \dots N_{\text{cells}} + 1$ by substituting $p(m_j \mid \mathbf{r}^-)$, $1 \dots N_{\text{cells}}$ in the RHS of Eq. (7). If we have a predefined number of beams, the values $\mathbf{p}(B_i)$ can be cached, making this entire step unnecessary. In this case, $p(m_j \mid \mathbf{r}^-)$ will already be consistent.
- 2) Apply Bayes theorem to compute the new values $p(B_i \mid \mathbf{r}^+ \in \mathcal{D})$ as follows $\forall i = 1 \dots N_{\text{cells}} + 1$

$$p(B_i \mid \mathbf{r}^+) = \frac{p(\mathbf{r}^+ \mid B_i) p(B_i \mid \mathbf{r}^-)}{\sum_{j=1}^{N_{\text{cells}}+1} p(\mathbf{r}^+ \mid B_j) p(B_j \mid \mathbf{r}^-)} \quad (10)$$

$$= \eta_B p(\mathbf{r}^+ \mid B_i) p(B_i \mid \mathbf{r}^-). \quad (11)$$

Here, η_B is the normalizer and $p(\mathbf{r}^+ \mid B_i)$ is the sensor forward beam model as mentioned in (9). Note that $p(B_i \mid \mathbf{r}^+)$ already satisfy (6). This update step requires $N_{\text{cells}} + 1$ forward model evaluations per cell update (followed by N_{cells} evaluations of (8)). *Eq. (11) is important because it lets us apply an occupancy update directly on $p(B_i)$ using a forward model without needing the difficult to calculate terms like in (2).*

- 3) Compute $p(m_j \mid \mathbf{r}^+)$ from $p(B_i \mid \mathbf{r}^+)$ using Eq. (8). As already mentioned, depending on the actual values of $p(B_i \mid \mathbf{r}^+)$ some cell probabilities may remain unchanged because no new information concerning them is contained in the new sample.

TABLE I
COMPARISON OF 3D SENSORS

Information	Swiss Ranger	Stereo Camera
Manufacturer	CSEM	Videre Design
Range [r_{\min}, r_{\max}]	[600, 7500] mm	[686, ∞] mm
Horiz. FOV	47°	65.5°
Vert. FOV	39°	51.5°
Resolution	176 × 144	640 × 480

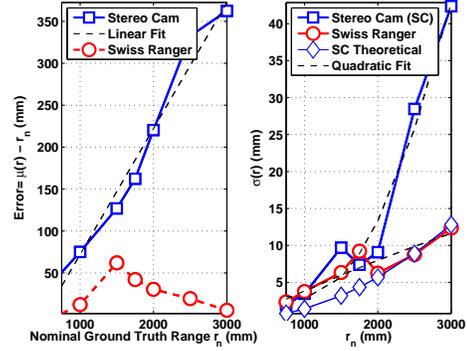


Fig. 3. Comparing the error in the means value $\mu(r)$ and the standard deviation $\sigma(r)$ of the Swiss Ranger (SR) vs. the Stereo Cam (SC). Refer to Table II for the polynomials fitted on this data.

These steps are repeated recursively at each time instant for each beam returned by the sensor. This procedure has none of the disadvantages of the approaches mentioned in Sec. I-B. On applying Eq. (11) to the classic beam model of Fig. 2(a), one obtains the graphs shown in Figs. 2.

III. APPLICATION TO 3D SENSORS

To motivate the empirical derivation of the 3D sensor models to be presented subsequently, we take a brief digression to introduce two 3D sensors being used in our robotics lab.

There are several options for 3D data acquisition like 3D laser scanners [7], [8], [9] or stereo cameras [10], [4]. In our experiments, a time-of-flight based SwissRanger SR-3000 [11] and a Stereo camera Stereo-on-Chip (STOC) [12] were used. Their locations on the robot are indicated in Fig. 5(a). The technical details of these sensors are summarized in Table I. Their relative accuracy is compared in Fig. 3. As described in [12], the range-resolution Δr of a stereo-camera is proportional to the square of the range and has a systematic error in the mean error which is linear with respect to the range. Refer to Fig. 3 for a comparison of this dependence with the experimental observations. The Swiss-ranger shows a similar decay but to a lesser degree. Good forward sensor models of stereo cams are hard to come by [10].

A. Forward Beam Modeling for 3D Sensors

In this section, we restrict ourselves to the 3D sensors already mentioned in Sec. III, viz. the Swiss ranger and the stereo camera.

Remark 3.1 (3D Sensors Characteristics): Experimental readings as shown in Fig. 3 show that the beam model as

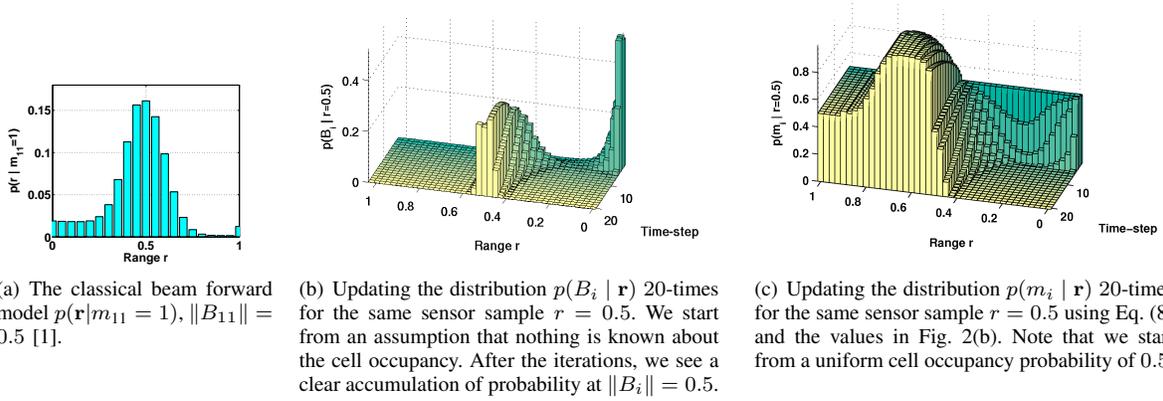


Fig. 2. Application of Eq. (11) to the classic beam model shown in Fig. 2(a). Note that the time-step axis increases towards the front.

TABLE II
CORRECTION POLYNOMIALS $q(r_n) = a_2 r_n^2 + a_1 r_n + a_0$.

	a_2 .	a_1 .	a_0 .
q_μ^{sr}	0.0	1.0	0.0
q_μ^{sc}	0.0	1.150204	-78.158607
q_σ^{sr}	-0.243897×10^{-6}	0.486625×10^{-2}	-0.7691149
q_σ^{sc}	0.947909×10^{-5}	-0.178817×10^{-1}	11.186380

in Fig. 2(a) is not entirely valid for commonly used 3D sensors. The main differences are as follows.

Both the sensors are not able to detect any obstacles with range less than r_{\min} . This means that the forward model $p(\mathbf{r} | B_i)$ is *not exhaustive*, meaning that $\sum_{\mathbf{r} \in \mathcal{D}} p(\mathbf{r} | B_i) d\mathbf{r} = 0$, if $\|B_i\| < r_{\min}$, where, $\|B_i\|$ is defined in Def. 2.1 and the set \mathcal{D} was defined before Eq. (9).

The false beams reporting maximum range as found in the classical model seem to be mostly absent from the given 3D sensors' samples. In their place, we now have to deal with *false negatives*, i.e., cases where the sensor does not report an obstacle even when one is present within the maximum range. This implies that $\sum_{\mathbf{r} \in \mathcal{D}} p(\mathbf{r} | B_i) d\mathbf{r} < 1$, if $r_{\min} \leq \|B_i\| \leq r_{\max}$. This again shows that the model is not exhaustive even when $r_{\min} \leq \|B_i\| \leq r_{\max}$.

As illustrated in Fig. 3, the standard deviation σ is a function of the ground-truth nominal range r_n . This effect is more pronounced for the stereo camera. This function can be approximated by a polynomial $q_\sigma(r_n)$. Refer to Table II.

As is clear from Fig. 3, the stereo camera has a systematic linear degradation of the mean accuracy with the range. This function too can be approximated by a polynomial $q_\mu(r_n)$. Refer to Table II.

Although the theoretical maximum range for the Swiss-ranger SR3000 is quite large and that of the stereo camera STOC is unlimited (Table I), we restrict it to a finite value r_{\max} for practical computational purposes. We note that r_{\max} can be chosen to be the maximum $\|B_{N_{\text{cells}}}\|$ among all the beams considered.

A sensor range sample of $r > r_{\max}$ carries information which should not be discarded. It shows that all cells along that beam are probably free. This case shows another advantage of our reformulation of the sensor forward model,

namely, that we have an explicit event $B_{N_{\text{cells}}+1}$ which handles it. Extending (9), we could say that for $i = 1 \dots N_{\text{cells}}$,

$$p(\mathbf{r} \wedge r > r_{\max} | B_i) \equiv 0. \quad (12)$$

$$p(\mathbf{r} \in \mathcal{D} | B_{N_{\text{cells}}+1}) \equiv \epsilon, \quad (13)$$

$$p(\mathbf{r} \wedge r > r_{\max} | B_{N_{\text{cells}}+1}) \equiv 1 - \epsilon. \quad (14)$$

Here, $1 \gg \epsilon > 0$ is a small value found experimentally. \triangleleft

We now formulate a forward 3D sensor model which is inspired by their above mentioned characteristics.

Definition 3.1 (Normal Probability Mass Distribution):

$$\mathcal{N}(\mathbf{r} \in \mathcal{D}, \|B_i\|, \sigma^2) \triangleq \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(r - \|B_i\|)^2}{2\sigma^2}\right\} \quad (15)$$

We recall that the normal distribution has most of its probability (99.73%) within $\mu \pm \Delta\sigma$, $\Delta = 3$. \triangleleft

Finally, Algorithm III.1 gives the resulting general 3D sensor forward model $p(\mathbf{r} \in \mathcal{D} | B_i)$ defined by using the polynomial definitions in Table II.

Algorithm III.1: $p(\mathbf{r} | B_i)$

```

 $\mu_i \leftarrow q_\mu(\|B_i\|), \quad \sigma_i \leftarrow q_\sigma(\|B_i\|)$ 
 $r_{\text{limit}} \leftarrow q_\mu(\|B_{N_{\text{cells}}}\|) + \Delta q_\sigma(\|B_{N_{\text{cells}}}\|) \quad (1)$ 
if  $i \neq N_{\text{cells}} + 1$ 
  then  $\left\{ \begin{array}{l} \text{if } \|B_i\| < r_{\min} \\ \text{then } p \leftarrow 0 \\ \text{else } \left\{ \begin{array}{l} \text{if } r \in [\mu_i - \Delta\sigma_i, \mu_i + \Delta\sigma_i] \\ \text{then } p \leftarrow (1 - \epsilon)\mathcal{N}(\mathbf{r}, \mu_i, \sigma_i^2) \\ \text{else } p \leftarrow \epsilon \end{array} \right. \end{array} \right.$ 
  comment:  $B_{N_{\text{cells}}+1}$ 
else  $\left\{ \begin{array}{l} \text{if } r > r_{\text{limit}} \\ \text{then } p \leftarrow (1 - \epsilon) \quad (2) \\ \text{else } p \leftarrow \epsilon \quad (3) \end{array} \right.$ 
return  $(p)$ 

```

Discussion: In Algorithm III.1, the region of influence (ROI) of B_i is set to a factor Δ of the corresponding standard deviation of its normal probability distribution. Line 1 sets the limit of the influence of the furthest B_i . If a range reading r exceeds this limit (Line 2), it means that all the cells in $\mathbf{m}(\mathbf{r})$ were free. Otherwise, Line 3 states that the probability of all cells being free is zero. This last part is a bit simplified because an occupied cell beyond our region of interest can also cause a range reading to fall within this region. This can be easily taken into account by considering certain “outside” cells whose ROI falls within $\mathbf{m}(\mathbf{r})$.

Limitations of this Model: 3D sensors like the stereo camera STOC and Swiss-ranger SR3000 are quite susceptible to environmental factors like ambient light intensity. The stereo camera STOC, in particular, cannot detect planar surfaces with very low texture. Therefore, it performs better in texture-rich outdoor environments. Since the set of all possible environments is too large, its effect is not taken into account in the model. To a limited extent, this can be accounted by including a uniform random error distribution within the model.

B. Fusing Sensor Data from Different Sensors

We now deal with the fusion of occupancy 3D data from the previously mentioned sensors which have different physical characteristics and operating principles. The aim is to merge the sensor data into a local structure like an occupancy grid centered around a given robot position and orientation, and subsequently merge this local representation into a global structure which takes into account the overall robot motion between the samples. In this paper, we consider the robot stopped at the location which needs to be mapped. The state-space to be estimated hence becomes static and the previously introduced occupancy grid paradigm can be used. The global data merging is outside the scope of this paper.

A recent tutorial in [13] covers many aspects of sensor fusion. In our implementation, the fusion is done using Eq. (16) [4], which is based on a *Superbayesian Independent Opinion Pool* formula. It is applicable for the case when separate occupancy grids are maintained for each sensor and fusion is done at a later stage by composing cell probabilities.

$$p(m_i) = \frac{p_1(m_i) p_2(m_i)}{p_1(m_i) p_2(m_i) + p_1(\neg m_i) p_2(\neg m_i)} \quad (16)$$

It has also been used recently in [2]. We use it in this paper for the results shown in Sec. V. There are also non-Bayesian fusion methods like taking the maximum or applying De Morgan’s law, etc., which have been mentioned in [1] and can also be used but have been left out due to lack of space.

IV. THE OVERALL FRAMEWORK IMPLEMENTATION

We assume that the robot stops occasionally, takes a few dozen samples of 3D data using its sensors, and fuses them into a local occupancy grid. This local grid is used for making decisions about motion and it can also be merged into a global grid while the robot is in motion. If this whole procedure can be done fast enough with respect to robot’s speed, an actual stop of the robot may not be necessary.

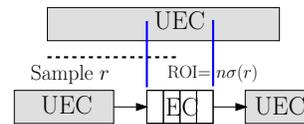


Fig. 4. The splitting of a composite beam due to a new sample.

The various range sensors are positioned on the robot at relative spatial offsets with respect to each other. This means that their *lines of view* or beams are also different. Whenever two beams from two or more different sensors intersect we have a region where the two readings can be fused.

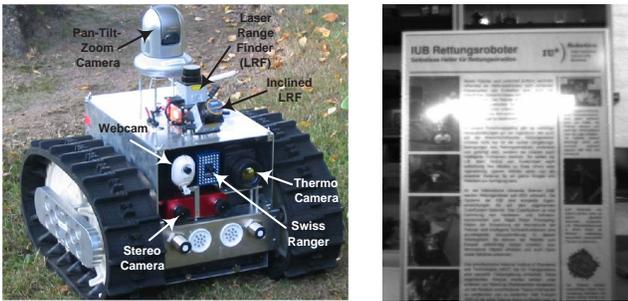
As shown in Fig. I, we assign a fixed tessellation of the beam angular space to *each* sensor. Each beam consists of cells which have occupancy probabilities. Based on the 3D point clouds returned by the sensor, each of these beams is then updated using the Bayesian update formula given in (11). After a requisite number of samples have been merged into the respective beam space tessellation, cell occupancy probabilities are computed using (8) and these values are written to a local occupancy-grid using ray-tracing, keeping in mind the relative offsets of the sensors. It is at this point that the occupancy values from different sensors are fused using (16).

A. Implementation Efficiency Issues

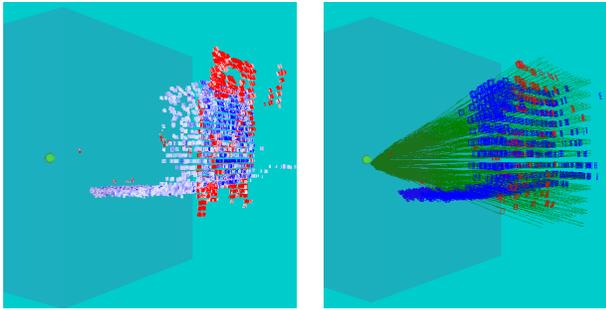
For any mobile robot implementation using 3D point clouds, efficiency is of paramount importance to keep the computational load low. To this end, we have relied on specific data-structures in software. This is discussed next.

Beam Representation: Along every beam, we would like to have a small cell size (typically 1 – 2 mm). The current range of popular 3D sensors is 8 meters or more. This would make a Bayesian update (11) quite expensive due to the large number of cells involved per beam. To reduce this number we use an *unrolled linked-list* (ULL) data-structure to represent a beam. Each element of this list is a *composite cell* made out of an integer number of individual cells. A composite cell can be in one of two states: *exploded cell* (EC) or *unexploded cell* (UEC). During Bayesian update, all constituent cells of an EC are updated. A UEC cell is updated as a whole– which saves computational effort.

As mentioned in Algo. III.1, each sensor range reading along a beam is also associated with a region-of-influence (ROI= $n\sigma$, $n = 6$). On each sample in which a UEC cell overlaps the ROI, it is split to a maximum of three sub-cells, with one of them becoming an EC. This is illustrated in Fig. 4, where arrows represent links within the ULL. To update a UEC as a whole, one can still use (11) except that one needs to use a *composite forward sensor model* for such cells. Let the UEC i be implicitly N_{UEC} individual cells long. Because $B_{ij}, j = 1 \dots N_{\text{UEC}}$ are exhaustive and mutually exclusive, $p(B_i) = p(B_{i1} \vee B_{i2} \vee \dots \vee B_{iN_{\text{UEC}}}) = \sum_{j=1}^{N_{\text{UEC}}} p(B_{ij})$, where \vee denotes the logical *OR* operation. Next, we need to find a composite forward sensor model $p(\mathbf{r} | B_i)$. This can be found as follows by applying Bayes rule. $p(\mathbf{r} | B_i) = p(\mathbf{r} | B_{i1} \vee \dots \vee B_{iN_{\text{UEC}}})$. This expression



(a) The robot with onboard sensors. (b) A flat obstacle with visual texture.



(c) Occupied points ($p > 0.1$ only) (d) The fused points shown in color coded by sensor. Gray-scaled green. No gray-scaling done to occupancy probabilities.

Fig. 5. Results of fusion for a flat obstacle with visual texture. In Fig. 5(c) the red points are from the stereo-cam, the blue from the Swiss-ranger. The color gradient from red/blue to white denotes the occupancy probability decreasing to a minimum of 0.1. The width of a point depicts the width of the beam it belongs to at that distance. In Fig. 5(d) additionally the fused points are shown with occupancy probability < 0.4 and the point width has not been changed according to the beam width.

for $p(\mathbf{r} | B_i)$ can be further expanded as:

$$= p(\mathbf{r}) \frac{\sum_{j=1}^{N_{UEC}} p(B_{ij} | \mathbf{r})}{\sum_{j=1}^{N_{UEC}} p(B_{ij})} = \frac{\sum_{j=1}^{N_{UEC}} p(\mathbf{r} | B_{ij}) p(B_{ij})}{\sum_{j=1}^{N_{UEC}} p(B_{ij})} \quad (17)$$

Eq. (17) shows that the composite forward sensor model is just the weighted average of the forward models of the implicit constituent cells. Therefore, composite forward sensor model for UECs can be calculated by finding a numerical weighted average of the first, middle, and last implicit individual cell. Simpson's three-point rule is well suited for this. This method thus constitutes a large saving of both memory and computation time.

V. EXPERIMENTAL RESULTS

In our experiments, we used the Swiss-ranger and stereo camera mounted on the robot as shown in Fig. 5(a). The origin of the spatial coordinate system is at the Swiss-ranger and the stereo camera is at an offset of about 25 mm to the left and 80 mm down, looking out. Fig. 5 shows a basic experiment. Fig. 5(d) shows that most of the fusion took place in the empty region. Furthermore, only the Swiss-ranger captured the floor.

Finding nearby planes and their inclination is important for autonomous robots capable of climbing up and down

ramps or stairs. To extract this information, a variation of the Hough Transform can be applied on the leaves of the Octree. The plane normal is parameterized by angles ϕ, θ mentioned in Fig. 1. The third parameter is the distance to the origin d . The algorithm consists of defining a scalar field γ over the $\mathcal{P}(\phi, \theta, d)$ space which is proportional to the probability of there being a plane corresponding to each point in \mathcal{P} . The algorithm proceeds by visiting each spatial cell m_{ijk} in the Octree and computing a distance to the origin d for each point in the tessellation of the \mathcal{P} -space. Subsequently, $\gamma(\phi, \theta, d)$ is incremented by $p(m_{ijk})$. At the end, points in \mathcal{P} with maximum γ are the most likely planes. For the flat board shown in Fig. 5 with ideal nominal values ($\phi = 180^\circ, \theta = 90^\circ, d = 1750$ mm), we obtained a plane ($\phi = 177^\circ, \theta = 83^\circ, d = 1762$ mm) using the above mentioned method.

VI. CONCLUSION

A method for generating local occupancy grid based maps applicable to popular 3D sensors was presented. It requires a reformulation of the sensor forward model in terms of certain intermediate variables. Special emphasis was put on an efficient implementation, which was described in some detail. Future work includes adding more sensors to the framework and extracting useful information from the fused data.

REFERENCES

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2005.
- [2] P. Stepan, M. Kulich, and L. Preucil, "Robust data fusion with occupancy grid," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, no. 1, pp. 106–115, 2005.
- [3] H. Moravec, "Sensor Fusion in Certainty Grid for Mobile Robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.
- [4] A. Elfes, "3," in *Data fusion in robotics and machine intelligence*, M. A. Abidi and R. C. Gonzalez, Eds. Academic press, 1992, ch. Multi-source spatial data fusion using Bayesian reasoning, pp. 137–160.
- [5] P. Payeur, P. Hebert, D. Laurendeau, and C. Gosselin, "Probabilistic octree modeling of a 3D dynamic environment," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 1997, pp. 1289–1296 vol.2.
- [6] S. Thrun, "Learning occupancy grids with forward sensor models," *Autonomous Robots*, vol. 15, pp. 111–127, 2003.
- [7] O. Wulf and B. Wagner, "Fast 3D-Scanning Methods for Laser Measurement Systems," in *International Conference on Control Systems and Computer Science (CSCS14)*, 2003.
- [8] H. Surmann, A. Nuechter, and J. Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, vol. 45, no. 3–4, pp. 181–198, 2003.
- [9] O. Wulf, C. Brenneke, and B. Wagner, "Colored 2D maps for robot navigation with 3D sensor data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3. IEEE Press, 2004, pp. 2991–2996 vol.3.
- [10] D. Murray and J. J. Little, "Using Real-Time Stereo Vision for Mobile Robot Navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 161–171, 2000.
- [11] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *Quantum Electronics, IEEE Journal of*, vol. 37, no. 3, pp. 390–397, 2001.
- [12] K. Konolige and D. Beymer, *SRI Small vision system, user's manual, software version 4.2*, February 2006. [Online]. Available: <http://www.ai.sri.com/~konolige>
- [13] S. Challa and D. Koks, "Bayesian and Demster-Shafer fusion," *Sadhana*, vol. 29, no. 2, p. 145174, 2004.