Real Time Probabilistic Mapping for Sonar Sensor by Optimization

Yuxuan Liu[1], Lujia Wang[2], Ming Liu[3], *Senior Member, IEEE*

***Abstract*— Mapping with sonar sensor is highly interesting yet to be solved, due to the low-cost and high uncertainty of the sensor. A novel method for iterative probabilistic occupancy grid mapping using gradient ascent is introduced. The algorithm is based on the forward sensor model for sonar sensors. The experiments demonstrate that such method is capable of creating satisfying maps for a mobile robot in real time. It is rather robust to sensor noise and model errors and it can overcome the problem of sensor conflicts produced by some existing methods. We propose an efficient way of iterative mapping for mobile robots.**

I. INTRODUCTION

Sonar-based mapping is one of the most well-documented and inexpensive methods for mapping for mobile robots. It is usually considered to be cheaper than laser scanners and computationally more efficient than cameras [1]. Sonar sensors are also deployed on autonomous vehicles to perform obstacle detection or close range mapping.

There are several ways to represent data obtained from sonar sensors. Many researchers draw an analogy between sonar sensors and laser scanners and propose some feature-based algorithms [2], [3]. This paradigm has the potential of producing maps with high accuracy especially in corner points, given the Field of View (FOV) of a sensor is relatively narrow. However, many engineers and researchers would deploy sonar sensors with large FOVs to fully cover areas around an autonomous vehicle with less sensors [4]. Handling this type of sonar sensors is not a trivial task because they are much less informative than laser scanners. While each measurement of a laser scanner, with negligibly narrow FOV, can show the position of an obstacle with little uncertainty, a measurement from sonar sensor just indicates that there are some obstacles in FOV at some distances from the sensor. As a result, it becomes sometimes necessary to directly handle uncertainty in sonar sensor measurements.

Within the framework of probabilistic methods, one of the most commonly adopted representations of occupancy maps is occupancy grid, where each grid cell represents a continuous value between 0 and 1 indicating the probability of certain cells contain obstacles. One of the most widely appreciated algorithms under such representation is mapping via the inverse sensor model [5].

[1]Yuxuan Liu is with School of Mechanical Engineering, Zhejiang University, Hangzhou, China 3150101228@zju.edu.cn

[2]Lujia Wang is with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China lj.wang1@siat.ac.cn

[3]Ming Liu is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China. eelium@ust.hk

The log-odd update algorithm does not consider the scene where more than one obstacle is presented in the FOV of a sensor. [6] proposes a recursive, fast algorithm to obtain a more comprehensive the inverse sensor model, but the author has pointed out that information loss when using the inverse sensor model would make the output inaccurate.

Besides, some researchers have pointed out that mapping with backward model suffers from the problem of sensor conflicts [7]. This flaw stems from the fact that the original algorithm does not consider the dependencies between different grid cells and much more information is lost in the inverse sensor model. Thrun proposed a probabilistic mapping algorithm based on a much more elaborate forward sensing model [8]. The algorithm adopts an expectation-maximization (EM) scheme. It overcomes the problem of sensor conflicts in original mapping methods but the optimization of the forward sensor model is conducted by flipping individual cells and making expectation inferences to see if the expectation increases, like an inefficient greedy search. As a result, this algorithm cannot perform mapping in real time.

Another method proposed in [6] abandons the probabilistic representation and apply gradient descent to iteratively optimize the customized cost functions. It manages to construct maps with satisfying accuracy and runs in real time. However, it does not consider the case of multiple obstacles comprehensively and many manually tuned regularizing tricks are required to produce a satisfying performance.

Therefore, this paper presents a computationally efficient solution to close range mapping assuming robot poses are known. The adopted sensor model has the potential of dealing with multiple obstacles at various distances. The update algorithm is based on gradient ascent and can be implemented recursively. This method could also produce more accurate maps than traditional methods.

The paper will first introduce some relevant contributions from other researchers. Then the sensor model and the update algorithms will be described analytically and comprehensively. Finally, we will present some simulation experiments and discussions on the algorithm.

II. PRELIMINARIES

A. Problems with Mapping via Log-odd Ratio

As pointed out by [7], [9], sensor conflicts are not only raised from errors in sensor readings but they sometimes also occurs in occasions where FOV of two sensors overlap. The phenomenon can be well illustrated in Figure 1 with a doorway example.

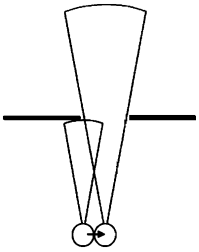


Fig. 1: Sensor conflicts when a robot moves pass a doorway.

When conducting mapping via the inverse sensor model, the update equation of each cell is derived as

$$\ell_{x,y}^{t} = log\frac{p(\mathbf{m}_{x,y})|z_t}{1-p(\mathbf{m}_{x,y}|z_t)} + log\frac{1-p(\mathbf{m}_{x,y})}{p(\mathbf{m}_{x,y})} + \ell_{x,y}^{t-1} \quad (1)$$

where $p(\mathbf{m}_{x,y})$ is the prior estimation of the occupancy value of a cell in (x, y).

In the scene shown in fig. 1, since the sonar pulse from the mobile robot could be blocked by the wall more often than passing through the narrow doorway. The output map of such method tends to lose the information of the doorway by overlooking the unblocked measurements.

This problem has also been discussed in [10]. The author suggests reducing the weights of similar readings from similar locations, which is indeed a significant improvement for the inverse sensor model. However, the output occupancy map at such regions is still largely determined by the parameters hand-crafted before the experiments thus these methods do not robustly solve the key problems of mapping with a backward model.

B. Forward Sensor Model with Deterministic Map

Intuitively, a forward probabilistic snesor model depicts the likelihood of a sensor measurement given occupancy maps and the pose of the robot.

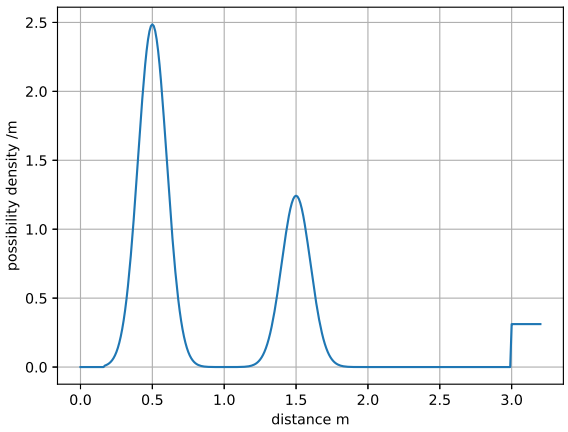


Fig. 2: The probabilistic distribution of $P(z_t^k|\mathbf{m}, x)$. Picture shows cases where there are two obstacles. The horizontal axis represents the reading of a sensor in meter while the vertical axis the probabilistic density.

A forward model can be written generally in the form

$$p(z_t^k|\mathbf{m}, x) \quad (2)$$

where z_t^k is the reading of the k th sensor at time t, $\mathbf{m}$ is the occupancy grid map and x the pose of the robot. Our sensor model consists of two basic components [1]:

1) *The hit case*. Each cell in the cone of a sensor has a possibility $p_{hit,(x,y)}^k$ of reflecting a sonar beam if the cell is occupied. $p_{hit,(x,y)}^k$ varies in each cell because of the relative distances and angles from corresponding sensors to the point (x, y) and can be predetermined by experiments. If an occupied cell reflects a sonar beam, the reading of this sensors will further be distorted by Gaussian noise. Thus, the probability density of a measurement z_t^k given the fact that the measurement is induced by an object with a distance d away is modeled to be proportional to $e^{-\frac{(z_t^k-d)^2}{2\sigma^2}}$.
2) *The un-hit case*. There will be cases when there are no occupied cells in the area covered by a sonar sensor or none of the cells reflects a sonar beam well received by the sensor. The reading of this sensor would simply be z_{max}.

A typical probabilistic density distribution with two occupied cells in a cone is shown in Figure 2.

III. Method

The problem of mapping with forward sensor model can be presented as a maximum likelihood estimation (MLE) problem, where we try to find the corresponding map that could maximize the posterior probability of the measurement.

In general, at each time steps, the target is to find a map $\mathbf{m}$ that satisfies

$$\mathbf{m} = \underset{\hat{\mathbf{m}}_t}{\operatorname{argmax}} P(\hat{\mathbf{m}}_t|Z_{1:t}, X_{1:t}) \quad (3)$$

where $\hat{\mathbf{m}}_t$ is the map we estimated at time t with continuous value in each grid cells, $Z_{1:t}$ and $X_{1:t}$ are the measurements and robotic poses at each time steps.

The optimization process adopts the expectation-maximization (EM) scheme. $P(\hat{\mathbf{m}}_t|Z_{1:t}, X_{1:t})$ will be estimated first, and gradient ascent is performed afterwards. The update part will be implemented using gradient ascent. This section would introduce the assumptions and mathematical details to implement the algorithm.

A. Assumptions

There are three major and non-trivial assumptions made to simplify this optimization-based mapping problem.

1) All information from past measurements $Z_{1:(t-1)}$ and past robot poses $X_{1:(t-1)}$ are integrated into a estimated grid map with continuous value $\hat{\mathbf{m}_{t-1}}$. This first-order markov assumption is a shortcut to an iterative representation. This assumption can be summarized as

$$P(\hat{\mathbf{m}}_t|Z_{1:t}, X_{1:t}) = P(\hat{\mathbf{m}}_t|Z_t, X_t, \hat{\mathbf{m}_{t:1}}) \quad (4)$$

2) The measurements taken from time t (Z_t) is independent of $\mathbf{m}_{\hat{t-1}}$ given $\hat{\mathbf{m}}_t$. The map constructed at time t alone should be enough to account for the measurements taken then. Such assumption is expressed as

$$P(Z_t|\hat{\mathbf{m}}_t, \mathbf{m}_{\hat{t-1}}) = P(Z_t|\hat{\mathbf{m}}_t) \tag{5}$$

where the pose of the robot is neglected for simplicity.

3) The output of the k th sensors at time t, denoted as z_t^k, is independent of the outputs from other sensors given the estimated map $\hat{\mathbf{m}}_t$, which means

$$P(Z_t|\hat{\mathbf{m}}_t) = P(z_t^1, ..., z_t^K|\hat{\mathbf{m}}_t) = \prod_{k=0}^{K} P(z_t^k|\hat{\mathbf{m}}_t) \tag{6}$$

where K denotes the number of sonar sensors mounted on the robot.

B. Expectation for Measurement

This subsection we try to estimate $P(\hat{\mathbf{m}}_t|Z_{1-t}, X_{1-t})$. Equation (4) transforms it into $P(\hat{\mathbf{m}}_t|Z_t, \mathbf{m}_{\hat{t-1}})$ where pose of the robot is neglected for simplicity.

According to Bayes' Rule, we have

$$P(\hat{\mathbf{m}}_t|Z_t, \mathbf{m}_{\hat{t-1}}) = \frac{P(Z_t|\hat{\mathbf{m}}_t, \mathbf{m}_{\hat{t-1}})P(\hat{\mathbf{m}}_t|\mathbf{m}_{\hat{t-1}})}{P(Z_t|\mathbf{m}_{\hat{t-1}})} \tag{7}$$

and it is worth noting that

1) $P(Z_t|\mathbf{m}_{\hat{t-1}})$ is constant with respect to $\hat{\mathbf{m}}_t$ so it is reasonable to substitute it with a constant.
2) It would be difficult and not necessary to analytically calculate the term $P(\hat{\mathbf{m}}_t|\mathbf{m}_{\hat{t-1}})$, but such term implies that, at each iteration, maps $\hat{\mathbf{m}}_t$ should not change a lot. To keep the map from getting abrupt changes, $\mathbf{m}_{\hat{t-1}}$ would be the starting point of out gradient ascent at time t and the gradient would be clipped from getting too large.
3) $P(Z_t|\hat{\mathbf{m}}_t, \mathbf{m}_{\hat{t-1}})$ is relevant to the forward sensor model and will be analysed in details below.

From (5),(6) we obtain

$$P(Z_t|\hat{\mathbf{m}}_t, \mathbf{m}_{\hat{t-1}}) = \prod_{k=0}^{K} P(z_t^k|\hat{\mathbf{m}}_t) \tag{8}$$

$P(z_t^k|\hat{\mathbf{m}}_t)$ refers to the forward sensor model of the k th sonar sensor on the mobile robot.

Let k_N be the number of points covered by the k th sonar sensor. These points are sorted based on their likelihood to reflect a sonar beam $p^k_{hit,(x_i,y_i)}$ as stated in section II and i is the index of these sorted points. Furthermore, let $d^k_{(x_i,y_i)}$ denotes the distance from point (x_i, y_i) to the k th sensor. $r^k_{(x_i,y_i)}$ denotes the probability of point (x_i, y_i) being occupied that can be retrieved from $\hat{\mathbf{m}}$.

The probability of a sonar beam is reflected on the i th point in the list, $P(hit, i|\hat{\mathbf{m}}_t)$, can be calculated by

$$P(unhit, i|\hat{\mathbf{m}}_t) = \begin{cases} 1 & i = 1 \\ \prod_{\ell=1}^{i-1}(1 - r^k_{x_\ell,y_\ell} \times p^k_{hit,(x_\ell,y_\ell)}) & i > 1 \end{cases} \tag{9}$$

$$P(hit, i|\hat{\mathbf{m}}_t) = P(unhit, i|\hat{\mathbf{m}}_t) \times r^k_{x_i,y_i} \times p^k_{hit,(x_i,y_i)} \tag{10}$$

where $P(unhit, i|\hat{m}_t)$ denotes the probability that all points before the i th point are not significantly reflecting a sonar beam.

The probability of the k th sensor not detecting any obstacles is estimated by

$$P(z_t^k, free|\hat{\mathbf{m}}_t) = \begin{cases} 0 & z_t^k != z_{max} \\ P(unhit, k_N + 1|\hat{\mathbf{m}}_t) & z_t^k = z_{max} \end{cases} \tag{11}$$

According to the forward sensor model described in section II, the complete forward sensor model should be the integration of the hit case and the un-hit case.

The analysis above yields

$$P(z_t^k|\hat{\mathbf{m}}_t) = \eta_k \sum_{j=1}^{k_N} P(hit, j|\hat{\mathbf{m}}_t) \times e^{-\frac{(z_t^k - d_j)^2}{2\delta^2}} + P(z_t^k, free|\hat{\mathbf{m}}_t) \tag{12}$$

Equation (9 - 12) is sufficient to perform the calculation of $P(z_t^k|\hat{\mathbf{m}}_t)$ under the assumptions made in this paper. η_k denotes the coefficients of Gaussian distribution $\eta_k = \frac{1}{\sqrt{2\pi\delta^2}}$.

To give a brief summary, equation (12) suggests that we iterate through all grid cells covered by a sonar sensor, calculate the likelihood of each grid cell inducing the measurements and obtain the expected likelihood of this measurement given the current map.

C. Calculating the gradient

The major difficulty of calculating the gradient lies in the fact that each grid cells will affect the calculation of the points behind it.

Equation (13 - 15) gradually decompose the gradient into basic blocks calculated in the forward sensor model.

$$\frac{\partial P(z_t^k|\hat{\mathbf{m}}_t)}{\partial r^k_{x_i,y_i}} = \sum_{j=i}^{k_N} \frac{\partial P(hit, j|\hat{\mathbf{m}}_t)}{\partial r^k_{x_i,y_i}} \eta_k^j + \frac{\partial P(z_t^k, free|\hat{\mathbf{m}}_t)}{\partial r^k_{x_i,y_i}}$$

$$\eta_k^j = \eta_k \times e^{-\frac{(z_t^k - d_j)^2}{2\delta^2}} \tag{13}$$

$$\frac{\partial P(hit, j|\hat{\mathbf{m}}_t)}{\partial r^k_{x_i,y_i}} = \begin{cases} P(unhit, i|\hat{\mathbf{m}}_t) \times p^k_{hit,(x_i,y_i)} & j = i \\ P(hit, j|\hat{\mathbf{m}}_t) \times \frac{-p^k_{hit,(x_i,y_i)}}{1 - p^k_{hit,(x_i,y_i)} r^k_{x_i,y_i}} & j > i \end{cases} \tag{14}$$

$$\frac{\partial P(z_t^k, free|\hat{\mathbf{m}}_t)}{\partial r^k_{x_i,y_i}} = \begin{cases} 0 & z_t^k < z_{max} \\ P(unhit, k_N + 1|\hat{\mathbf{m}}_t) \times \frac{-p^k_{hit,(x_i,y_i)}}{1 - p^k_{hit,(x_i,y_i)} r^k_{x_i,y_i}} & z_t^k = z_{max} \end{cases} \tag{15}$$

Moreover, in order to keep $0 <= r^k_{x_i,y_i} <= 1$, we further define the notation $a^k_{x_i,y_i}$, so that

$$r^k_{x_i,y_i} = sigmoid(a^k_{x_i,y_i}) = \frac{1}{1 + e^{-a^k_{x_i,y_i}}} \tag{16}$$

With chain rule, the derivative w.r.t. $a^k_{x_i,y_i}$ can be written as

$$\frac{\partial P(hit, j|\hat{\mathbf{m}}_t)}{\partial a^k_{x_i,y_i}} = \frac{\partial P(hit, j|\hat{\mathbf{m}}_t)}{\partial r^k_{x_i,y_i}} r^k_{x_i,y_i}(1 - r^k_{x_i,y_i}) \quad (17)$$

Equation (6) shows that $P(Z_t|\hat{\mathbf{m}}_t)$ is the product of the probability of each individual measurement $P(z^k_t|\hat{\mathbf{m}}_t)$. There are also many points covered by more than one sonar sensor. As a result, the gradient w.r.t. a point (x, y) should be calculated analytically with chain rule and be furthered simplified as

$$\frac{\partial P(Z_t|\hat{\mathbf{m}}_t)}{\partial a_{x,y}} = \sum_{\ell=1}^{K} \left[\prod_{j!=\ell} P(z^j_t|\hat{\mathbf{m}}_t) \times \frac{\partial P(z^\ell_t|\hat{\mathbf{m}}_t)}{\partial a_{x,y}} \right] \quad (18)$$

$$= \sum_{\ell=1}^{K} \left[\frac{P(Z_t|\hat{\mathbf{m}}_t)}{P(z^\ell_t|\hat{\mathbf{m}}_t)} \frac{\partial P(z^\ell_t|\hat{\mathbf{m}}_t)}{\partial a_{x,y}} \right] \quad (19)$$

$$= \sum_{\ell=1}^{K} \left[\alpha \frac{1}{P(z^\ell_t|\hat{\mathbf{m}}_t)} \frac{\partial P(z^\ell_t|\hat{\mathbf{m}}_t)}{\partial a_{x,y}} \right] \quad (20)$$

where α is used to denote the learning rate.

D. Algorithm Review

The entire algorithm is summarized as Algorithm 1. The program integrates the calculation of expectation and gradients and makes the entire process more compact. Furthermore, the program calculate the variables $P(unhit, i|\hat{\mathbf{m}}_t)$ recursively in the form of P_{unhit}. The overall asymptotic time complexity of processing a measurement of a sensor is $\mathcal{O}(N^2)$ where N is the number of points covered by a sensor.

In practice, the program manages most of the variables in single-precision floating-point format instead of double-precision floating-point one, which significantly boosts the operational speed of the program while the maps produced are basically unchanged.

Algorithm 1 Mapping via Gradient Ascent

Input: :
z^k_t : measurement from the k th sensor at time t;
$a^k_{x,y}$:map cells covered by the k th sensor estimated last time;
Procedure: :
$P \leftarrow 0; P_{unhit} \leftarrow 1; \frac{dP}{da^k_{x_i,y_i}} \leftarrow 0; \frac{dP_{unhit}}{da^k_{x_i,y_i}} \leftarrow 0;$
for $i = 1;\ i <= k_N;\ i++$ **do**
  set $N(z^k_t|d_i) = \frac{1}{\sqrt{2\pi\delta^2}} e^{\frac{(z^k_t - d_i)^2}{\delta^2}}$
  set $r^k_{x_i,y_i} = sigmoid(a^k_{x_i,y_i})$
  $P+ = N(z^k_t|d_i) P_{unhit} \times p^k_{hit,(x_i,y_i)} r^k_{x_i,y_i}$
  $\frac{dP}{da^k_{x_i,y_i}} = N(z^k_t|d_i) P_{unhit} \times p^k_{hit,(x_i,y_i)} r^k_{x_i,y_i}(1 - r^k_{x_i,y_i})$
  for $j = 1; j < i; j++$ **do**
    $\frac{dP}{da^k_{x_i,y_i}} += N(z^k_t|d_i) r^k_{x_i,y_i} p^k_{hit,(x_i,y_i)} \frac{dP_{unhit}}{da^k_{x_j,y_j}}$
  end for
  $\frac{dP_{unhit}}{da^k_{x_i,y_i}} = P_{unhit}(-p^k_{hit,(x_i,y_i)}) r^k_{x_i,y_i}(1 - r^k_{x_i,y_i})$
  $P_{unhit} = P_{unhit}(1 - p^k_{hit,(x_i,y_i)} r^k_{x_i,y_i})$
  for $j = 1; j < i; j++$ **do**
    $\frac{dP_{unhit}}{da^k_{x_j,y_j}} = \frac{dP_{unhit}}{da^k_{x_j,y_j}}(1 - p^k_{hit,(x_i,y_i)} r^k_{x_i,y_i})$
  end for
end for
if $z^k_t \geq z_{max}$ **then**
  $P+ = P_{unhit}$
  for $i = 1;\ i <= k_N;\ i++$ **do**
    $\frac{dP}{da^k_{x_i,y_i}} += \frac{dP_{unhit}}{da^k_{x_i,y_i}}$
  end for
end if
// Update map
for $i = 1;\ i <= k_N;\ i++$ **do**
  $gradient = clip(\frac{1}{P} \frac{dP}{da^k_{x_i,y_i}}, max_{gradient})$
  $a^k_{x_i,y_i} = a^k_{x_i,y_i} + \alpha \times gradient$
end for

IV. Experiments and Results

Simulations are conducted to verify the algorithm proposed in section IV.

In a simulation, a mobile robot mounted with sonar sensors is programmed to move along a fixed route in a synthetic map. The readings of sonar sensors are randomly sampled from the forward sensor model at each time steps. Concurrently, the algorithm we proposed and one based on traditional inverse sensor model are both tested. Both programs would try to reproduce the original map based on those limited measurements. There are some more details in experiment implementation to optimize the performance of both of the algorithms and ensure the plausibility of the experiment.

First, the forward sensor model used in simulation differs from the one in the algorithm in the parameter of $p^k_{hit,(x,y)}$ and the variance of Gaussian noise δ. This setting challenges the robustness of both algorithms. For example, we make $\delta = 0.05$ in the mapping nodes while $\delta = 0.1$ in simulation node.

Second, each mapping program implements a threshold to transform its outputs into binary maps. Any cells with occupancy value larger than the threshold would be rendered as obstacles. The binary maps are further filtered with morphological operations like eroding and dilating. These operations are computer graphics operations that filter out scattered misjudged cells or expand detected obstacles.

Third, hyper-parameters like learning rate, thresholds, and morphological operations are manually tuned to minimize the mean square error (MSE) between the reconstructed map and the original map. It is worth noting that MSE can hardly indicate the similarity between two matrices but it could be simple and efficient hints to quantify the tuning process.

Some parameters critical to the experiments including the learning rate, the grid resolution of the map, the horizontal angle of the sonar sensors are presented in Table 1. We learned from the experiments that the execution time and mapping performance of the algorithm would be significantly influenced by different combinations of different parameters.

Parameter	Value	Parameter	Value
learning rate	30	sensor lowest reading	0.1m
grid resolution	0.1m	sensor highest reading	3m
threshold	0.7	horizontal angle	90°
size of the map	70m * 70m		

TABLE I: Parameter table shows parts of the hyperparameter used in the experiments.

The distribution of sonar sensors are shown by the overhead view of the robot in Figure 3.

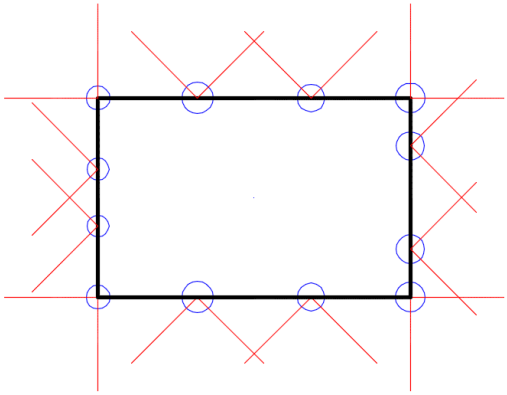


Fig. 3: Sensors' placement used in the experiments. Black lines represent the boundary of the robot. Blue circles correspond to the placement of the sonar sensors. Red lines indicate the FOV of these sensors.

Subfigures (a-c) in Figure 4 show the original map, the map reconstructed by our method, and the map generated by traditional algorithm [5] . Both of the mapping programs are optimized for this map by tuning hyper-parameters.

Freezing the hyper-parameters in both programs, we test our proposed method and the traditional algorithm on another slightly more difficult map to further validate their performances. Subfigures (d-f) in Figure 4 show the outcome of this map. Both of the mapping programs remain unchanged after fine-tuned for the first map.

It can be observed from the second experiment that some parts of the maps are unobservable and some corners can only observed in only one or two time steps. Parts circled in red, as stated in section II would introduce sensor conflicts in many mapping algorithms based on inverse sensor models.

Experiments shows that the algorithm proposed in this paper can produce maps with high accuracy online as well as solve the problem of sensor conflicts. The traditional produce misjudged, scattered obstacles even with optimized filters and it fails to detect the doorway circled in red. However, the method we propose sometimes seems to do mapping more aggressively, leaving occupied cells free until fully convinced.

V. Discussion and Reasoning

As shown in section III and IV, the algorithm is grounded with solid mathematical bases and we could use the fine-tuned algorithm to do mapping and static obstacle detection with rather satisfying accuracy. The program can handle many sonar sensors with proper efficiency and performance. During the experiments in section IV, the program takes 30 milliseconds to perform an iteration of gradient ascent, updating the map based on measurements from 12 sensors on a single thread of i7-8750H (Intel, The United State of America).

It should be worth noting that the mapping algorithm proposed is based on several non-trivial assumptions as stated in section III. They limit the algorithm's potential to achieve global optimum, but also simplify the maths, makes the algorithm more robust and flexible.

The assumption that sensors are independent of each other given $\hat{\mathbf{m}}_t$ avoids the need to directly model the interference of different sensors. However, since a sensor could still alter the occupancy value and thus affect the other sensors at next iteration, the problem of sensor conflicts can still be solved with this moderately complicated forward sensor model under such independent assumption.

The assumption that all measurements before should be integrated into a simple occupancy grid map and the premise that we are not analytically solving for $P(\hat{\mathbf{m}}_t|\hat{\mathbf{m}_{t-1}})$ are strong claims that significantly reduce the difficulty of the maths. Consequently, the learning rate is manipulated to balance the weight between current measurements and the past ones based on the application scene. Aggressive learning rate tends to enable the robot to recognize obstacles faster and even adapt to some dynamic obstacles, but sometimes it also over-optimizes the maps for the current measurement which leads to sub-optimal estimation of static environments, while moderate learning rate could be optimized for static environments.

The forward sensor model used in this paper has considered the case of dealing with multiple obstacles. It also performs neat expectation estimation given maps with continuous values, which is an attempt to generalize the work done in [8] and [11]. This particular model manages to maintain the balance between accuracy and optimization difficulty.

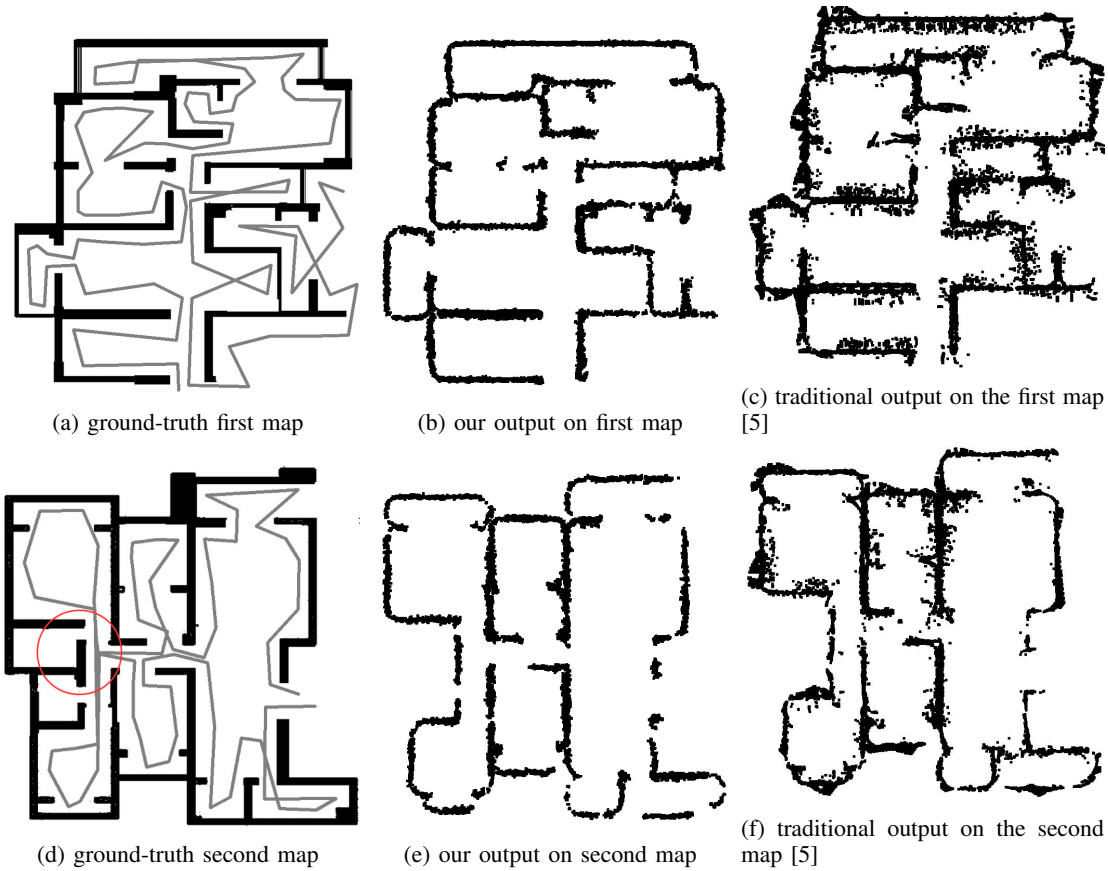
VI. Conclusion

The paper investigated deeply into the problem of mapping with sonar sensors, introduced a generalized forward sensor model for occupancy grids with continuous value, proposed an algorithm that utilizes both the sensor model and gradient ascent method, and conducted simulations to verify the algorithm.

We presented several assumptions to make the problem manageable and we explored the derivation of such EM framework including the calculation of expectation and gradients based on the forward sensor model. Then we compared our algorithm with a traditional method to understand both the advantage and disadvantage of our procedure.

The workflow we proposed can reproduce maps in real time with high accuracy, overcomes the problem of sensor conflicts. Fine tuning the parameters can make a program specifically optimized for a particular set of problems.

There are still a number of limitations in our method. The optimization method we adopted is a basic one and it



(a) ground-truth first map (b) our output on first map (c) traditional output on the first map [5]

(d) ground-truth second map (e) our output on second map (f) traditional output on the second map [5]

Fig. 4: **Experimental result**. (a),(d) shows the original map and the route of the robot. (b),(e) and (c),(f) show the maps rendered by our algorithm and the traditional one respectively. Figure (a-c) depict the map both program optimize on while figure (d-f) depict the test image.

is possible to further optimize the gradient ascent method we implemented. Furthermore, the forward sensor model we proposed does not consider the height of an obstacle. This would sometimes affect the performance of the algorithm in the form of modeling error.

ACKNOWLEDGMENT

This work is supported by Shenzhen Science, Technology and Innovation Comission (SZSTI) JCYJ 20170818153518789 and the National Science Foundation of China No. 61603376, awarded to Dr. Lujia Wang. This paper is also supported by Shenzhen Science, Technology and Innovation Commission (SZSTI) JCYJ 20160428154842603 and the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017 award to Ming Liu.

REFERENCES

[1] S. Thrun, *Probabilistic Robotics*. Mit Press, 2005.

[2] O. Wijk and H. I. Christensen, "Triangulation-based fusion of sonar data with application in robot pose tracking," *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 740–752, Dec 2000.

[3] H. Ismail and B. Balachandran, "Algorithm fusion for feature extraction and map construction from sonar data," *IEEE Sensors Journal*, vol. 15, pp. 6460–6471, Nov 2015.

[4] D. Bank, "A novel ultrasonic sensing system for autonomous mobile systems," in *SENSORS, 2002 IEEE*, vol. 2, pp. 1671–1676 vol.2, June 2002.

[5] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, pp. 46–57, June 1989.

[6] E. Kaufman, T. Lee, Z. Ai, and I. S. Moskowitz, "Bayesian occupancy grid mapping via an exact inverse sensor model," in *2016 American Control Conference (ACC)*, pp. 5709–5715, July 2016.

[7] S. Thrun, "Learning occupancy grids with forward models," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 3, pp. 1676–1681 vol.3, Oct 2001.

[8] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, pp. 111–127, Sep 2003.

[9] K. Lee, I. H. Suh, S. Oh, and W. K. Chung, "Conflict evaluation method for grid maps using sonar sensors," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2908–2914, Sep 2008.

[10] K. Konolige, "Improved occupancy grids for map building," *Autonomous Robots*, vol. 4, pp. 351–367, Oct 1997.

[11] E. A. Shvets, D. A. Shepelev, and D. P. Nikolaev, "Occupancy grid mapping with the use of a forward sonar model by gradient descent," *Journal of Communications Technology and Electronics*, vol. 61, pp. 1474–1480, Dec 2016.