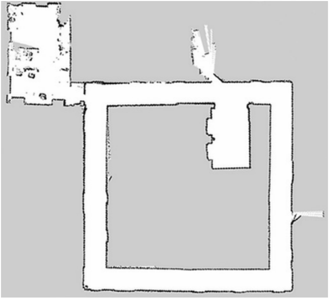
Разработка и анализ методов восстановления карты проходимости на основе показаний датчиков измерения расстояния

Денис Шепелев

студент группы 073а
ФУПМ МФТИ

Карта проходимости

- Карта проходимости - сетка, состоящая из квадратных клеток одинакового размера.
- Клетка карты - некоторая область пространства, содержащая информацию о наличии препятствия в соответствующей этой клетке территории.
- Такие карты используются в мобильной робототехнике для навигационных задач.



Датчики измерения расстояния

- Сонары
- Лидары
- Стереокамеры

Сонары выбраны в качестве основного датчика, так как они значительно дешевле и доступнее лидаров, и могут использоваться при любом освещении, в отличии от стереопары.

Постановка задачи восстановления карты проходимости

Дано:

- Картируемое окружение статично.
- Даны наблюдения датчиков:

$$Z = \{z_1, ..., z_T\}$$

$$z_t = (x_t, y_t, \varphi_t, r_t)$$

Цель:

- Восстановить карту проходимости m на основе наблюдений датчиков Z.

Существующие методы картирования

На данный момент можно выделить два семейства методов восстановления карты проходимости

- основанные на обратной модели сенсора:

$$p(m|Z) \tag{1}$$

- основанные на прямой модели сенсора:

$$p(Z|m) \tag{2}$$

Картирование на основе обратной модели сенсора

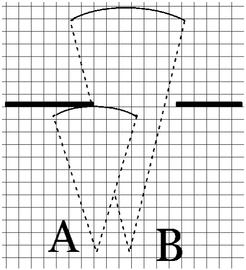
Допущения метода:

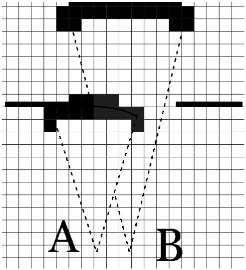
- Клетки карты m_i - независимые случайные величины. Каждая ячейка карты m хранит вероятность занятости m_i, с учётом наблюдений Z:
$$p(m|Z) = \prod_i p(m_i|Z) \tag{3}$$
-
$$p(z_t|m, z_{1,t-1}) = p(z_t|m) \tag{4}$$

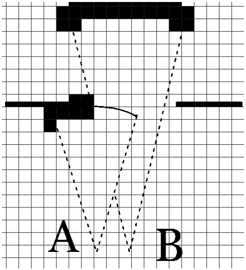
Значения в клетках обновляются по формуле:

$$\frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})} = \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)} \frac{p(m_i|z_{1..t-1}, x_{1:t-1})}{1 - p(m_i|z_{1..t-1}, x_{1:t-1})} \frac{1 - p(m_i)}{p(m_i)} \tag{5}$$

Метод и работает в режиме реального времени, однако когда допущения описанного метода неверны, карта проходимости, построенная с его помощью, может содержать грубые ошибки:







Картирование на основе прямой модели сенсора

Основная идея методов, основанных на прямой модели:

- Каждая клетка может принимать только два значения занятости: 0 - свободная от препятствий ячейка, 1 - занята каким либо препятствием.
- Прямая модель $p(Z|m)$ показывает, на сколько хорошо некоторая карта m объясняет показания сонаров Z.
- Таким образом, задача картирования сводится к поиску такой карты m^*, которая максимизирует значение $p(Z|m)$:

$$m^*(Z) = \underset{m}{\operatorname{argmax}}\, p(Z|m) \tag{6}$$

Проблема большинства таких алгоритмов - невозможность имплементации для работы в режиме реального времени.

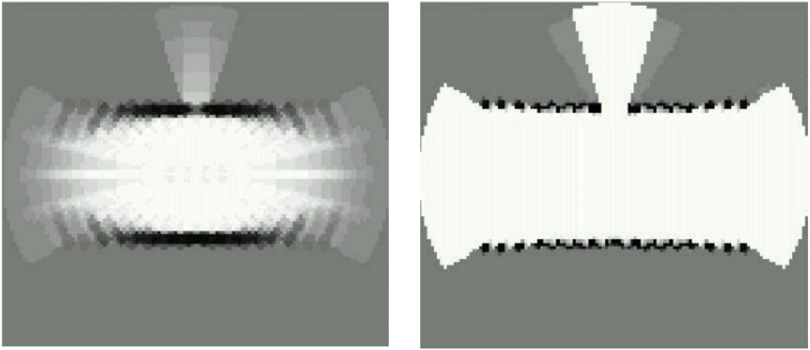


Рис. 1 : Результаты картирования двери, используя наблюдения сонаров. (а) - результаты алгоритма на основе обратной модели, (б) - на основе EM-алгоритма с прямой моделью, предложенной в работе Себастьяна Труна.

Цель работы

Разработка методов восстановления карты проходимости, используя наблюдения сонаров, которые могут быть имплементированы для работы в режиме реального времени, и картирующие лучше традиционного метода, основанного на обратной модели.

Предложенные в работе методы картирования

В данной работе предложены два новых метода построения карты проходимости:

- Первый метод основан на прямой модели, карта проходимости находится с помощью метода стохастического градиента.
- Во втором методе предложена новая прямая модель сонара, с помощью которой задачу картирования можно свести к задаче минимизации непрерывной невязки.

Описание прямой модели, использованной в алгоритме картирования методом стохастического градиента

Пусть внутри области видимости сонара находятся K препятствий, отсортированных в порядке возрастания дистанции d_k. Через $\{c_*, c_0, c_{1,K}\}$ будем обозначать множество различных сценариев работы сонара, через c_* - случайный выброс, c_0 - измерение R_{max}.

1. Пусть реализовался случай, когда измерение было порождено событием c_k, $k \in \{0, .., K\}$

$$p(z|m, c_k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z-d_k)^2}{\sigma^2}} \tag{7}$$

2. Если реализовался случай c_*, то

$$p(z|m, c_*) = \frac{1}{R_{max}} \tag{8}$$

Окончательная формула для описанной прямой модели:

$$\begin{aligned} p(z|m) &= \frac{1}{R_{max}} p_{rand} \\ &+ \sum_{i \in \{1,\dots,K\}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z-d_k)^2}{\sigma^2}} (1-p_{rand})(1-p_{hit})^{k-1} p_{hit} \\ &+ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(z-R_{max})^2}{\sigma^2}} (1-p_{rand})(1-p_{hit})^{K} \end{aligned} \tag{9}$$

Картирование методом стохастического градиента

Вводится следующая функция от m при заданных наблюдениях Z:

$$\Phi(m, Z) = \phi_{sonars}(m, Z) + \phi_{occupancy}(m) + \phi_{borders}(m), \qquad (10)$$

Задача картирования сводится к поиску карты m^*, которая максимизирует 10:

$$m^*(Z) = \underset{m}{\operatorname{argmax}} \left(\phi_{sonars}(m, Z) + \phi_{occupancy}(m) + \phi_{borders}(m) \right) \qquad (11)$$

Рассмотрим составляющие части функции $\Phi(m, Z)$:

1. ϕ_{sonars} - распределение наблюдений сонаров z_t при заданной карте проходимости m.

$$\phi_{sonars}(m, Z) = p(z_1, ..., z_T|m) = \prod_t p(z_t|m) \tag{12}$$

2. $\phi_{occupancy}(m)$ отвечает за априорные знания о проходимости карты

$$\phi_{occupancy}(m) = \sum_{m_i} w_o o(m_i) \tag{13}$$

   При $w_o < 0$ карта m штрафуется за каждую занятую клетку m_i.

3. $\phi_{borders}(m)$ представляет собой следующую сумму

$$\phi_{borders}(m) = \sum_{m_i} w_b n^2(m_i) \tag{14}$$

   где $n(m_i)$ - число таких соседей клетки m_i, состояние которых не совпадает с состоянием m_i. При $w_b < 0$ клетка штрафуется за каждого соседа, состояние которого не совпадает со состоянием клетки.

Оптимизационная задача (11) решается методом стохастического градиента. Каждый оптимизационный шаг состоит из следующих действий:

1. Случайным образом выбирается клетка m_i и значение проходимости $o(m_i)$ инвертируется

$$o^*(m_i) = 1 - o(m_i)$$

2. Для нового состояния клетки m_i пересчитываются $\phi_{sonars}(m, Z)$, $\phi_{occupancy}(m)$ и $\phi_{borders}(m)$.
3. При $\Phi^*(m, Z) > \Phi(m, Z)$ или с вероятностью p_{random_factor} сохраняется $o^*(m_i)$ и $\Phi^*(m, Z)$, $\phi^*_{sonars}(m, Z)$, $\phi^*_{occupancy}(m)$, $\phi^*_{borders}(m)$; иначе возвращаемся в предыдущее состояние.

Картирование методом градиентного спуска

Представим клетки карты проходимости m как переменные x_i, которые принимают значения от 0 до 1 и характеризуют степень занятости ячеек.

Каждое наблюдение сонара порождает два множества клеток:

- $\Omega_{free}(m, z)$ - суммарная занятость клеток внутри этого множества, должно быть не больше некоторого порога.
- $\Omega_{occ}(m, z)$ - суммарная занятость клеток внутри этого множества, должно быть не меньше некоторого порога.

Описание прямой модели, использованной в алгоритме картирования методом градиентного спуска

$$\psi = \psi_{free} + \psi_{occ} \tag{15}$$

- $\psi_{free} = \psi_{free}(X_{free})$ - значение зависит от суммарной эффективной занятости X_{free} внутри Ω_{free}. Если X_{free} больше некоторого порога, то значение штрафа достаточно велико, иначе - небольшое.
- $\psi_{occ} = \psi_{occ}(X_{occ})$ - значение зависит от суммарной эффективной занятости X_{occ} внутри Ω_{occ}. Если X_{occ} меньше некоторого порога, то значение штрафа достаточно велико, иначе - небольшое.
- Эффективная занятость множества клеток Ω считается следующим образом

$$X_{\Omega} = \sum_{x_i \in \Omega} x_i w_i \tag{16}$$

Функции штрафа - кусочно линейные функции вида:

$$\psi^t_{free} = \begin{cases} \alpha_{free} X, & X < X^t_{free} \\ \alpha_{free} X^t_{free} + \beta^t_{free}(X - X^t_{free}), & X \geq X^t_{free} \end{cases} \tag{17}$$

$$\psi^t_{occ} = \begin{cases} 1 - \beta^t_{occ} X, & X < X^t_{occ} \\ 1 - \beta^t_{occ} X^t_{occ} - \alpha_{occ}(X - X^t_{occ}), & X \geq X^t_{occ} \end{cases} \tag{18}$$

Картирование методом градиентного спуска

Для каждой клетки вводится кусочно линейная регуляризация $R^i(x_i)$:

$$R^i(x_i) = \begin{cases} c_1(x_i - 1), & x_i > 1 \\ c_2 x_i, & x_i \in [0, 0.5] \\ c_2(1 - x_i), & x_i \in [0.5, 1] \\ -c_1 x_i, & x_i < 0 \end{cases} \tag{19}$$

Таким образом, задача картирования сводится к минимизации непрерывного штрафа:

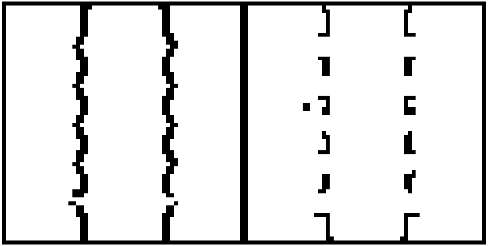
$$m^*(Z) = \underset{m}{\operatorname{argmin}} \left(\sum_{x_{\boldsymbol{i}}} R^i(x_i) + \sum_{t : z_{\boldsymbol{t}} \in Z} (\psi^t_{free} + \psi^t_{occ}) \right) \tag{20}$$

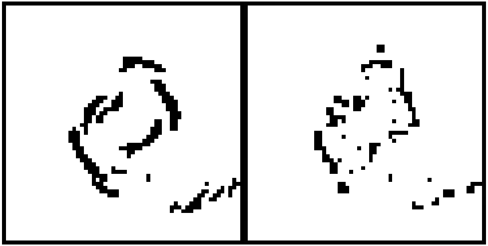
Режим реального времени

Чтобы предложенные методы могли работать в режиме реального времени, нужно ограничить число наблюдений. Для этого можно воспользоваться следующими способами:

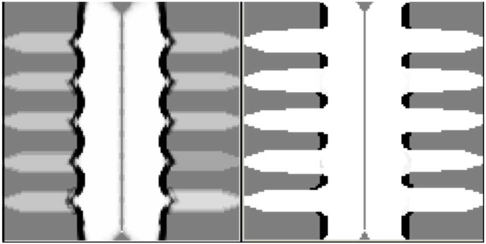
- Использовать скользящее окно и рассматривать последние N_{max} наблюдений.
- В каждом узле трехмерной сетки (x_i, y_i, φ_i) хранится список наблюдений $z_t = (x_t, y_t, \varphi_t, r_t)$, которые принадлежат соответствующей части пространства: $x_t \in [x_i - \Delta_x, x_i + \Delta_x]$, $y_t \in [y_i - \Delta_y, y_i + \Delta_y]$ и $\varphi_t \in [\varphi_i - \Delta_\varphi, \varphi_i + \Delta_\varphi]$. Тогда в каждом узле можно хранить N_{max} последних измерений или некоторым образом фильтрованные наблюдения. При этом масштаб этой сетки, может не совпадать с масштабом карты проходимости.

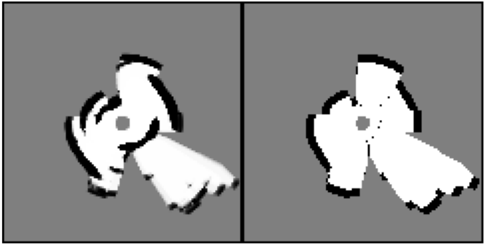
Результаты картирования методом стохастического градиента





Результаты картирования методом градиентного спуска





Заключение

- В результате работы исследованы различные метода восстановления карты проходимости с помощью наблюдений сонаров. Оказалось, что традиционные методы восстанавливают карты с грубыми ошибками, другие подходы невозможно использовать в режиме реального времени.
- В работе разработаны два новых метода картирования, предложена новая прямая модель сонара.
- Предложенные методы картируют лучше традиционного метода и могут быть использованы в режиме реального времени.