

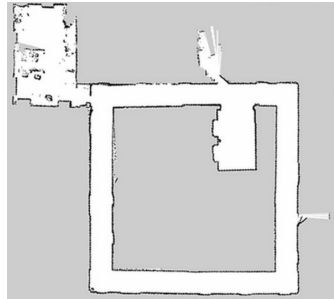
Разработка и анализ методов восстановления карты проходимости на основе показаний датчиков измерения расстояния

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 отличие от стереопары.

Постановка задачи восстановления карты проходимости

Дано:

- ▶ Картируемое окружение статично.
- ▶ Даны наблюдения датчиков:

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

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

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:

$$p(m|Z) = \prod_i p(m_i|Z) \quad (1)$$

- ▶ основанные на прямой модели сенсора:

$$p(Z|m) \quad (2)$$

Традиционный метод картирования на основе обратной модели сенсора

Допущения обратной модели:

- ▶ Клетки карты m_i - независимые случайные величины. Каждая ячейка карты m хранит вероятность занятости m_i , с учётом наблюдений Z :

$$p(m|Z) = \prod_i p(m_i|Z) \quad (3)$$

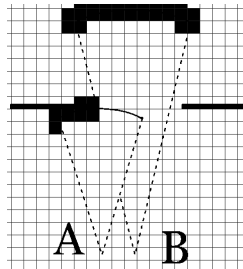
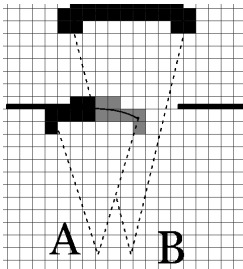
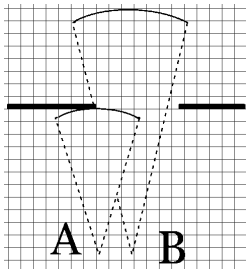


$$p(z_t|m, z_{1:t-1}) = p(z_t|m) \quad (4)$$

Значения в клетках обновляются по формуле:

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

Метод хотя и работает в режиме реального времени, однако когда допущения обратной модели невыполнены, 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: 1 - занятая, 0 - свободная.
- ▶ 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 $p(Z|m)$:

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

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

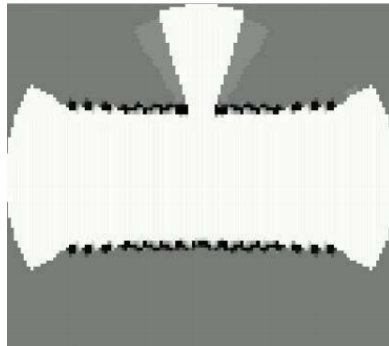
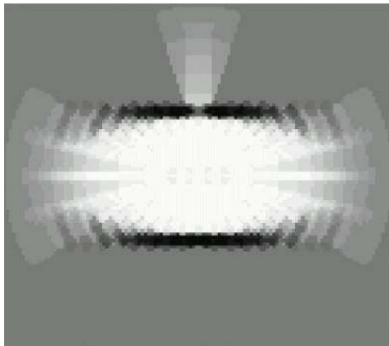


Рис. 1 : Результаты картирования двери, используя наблюдения сонаров. (а) - результаты алгоритма на основе обратной модели, (б) - на основе ЕМ-алгоритма с прямой моделью, предложенной в работе Себастьяна Труна.

Цель работы

Разработка методов восстановления карты проходимости, которые могут быть реализованы 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 основан на прямой модели сонара Труна, карта проходимости находится с помощью метода стохастического градиента;
- ▶ второй – на новой прямой модели сонара, с помощью которой задачу картирования удалось свести к задаче минимизации непрерывной невязки.

Описание прямой модели Труна

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 , $k \in \{0, .., K\}$

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

2. Если реализовался случай c_* , то

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

Окончательная формула прямой модели:

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

Картирование методом стохастического градиента

Вводится следующая функция от m при заданных наблюдениях Z :

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

$$\phi_{sonars}(m, Z) = p(z_1, \dots, z_T | m) = \prod_t p(z_t | m) \quad (11)$$

$$\phi_{occupancy}(m) = \sum_{m_i} w_o o(m_i) \quad (12)$$

$$\phi_{borders}(m) = \sum_{m_i} w_b n^2(m_i) \quad (13)$$

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

$$m^*(Z) = \operatorname{argmax}_m \left(\phi_{sonars}(m, Z) + \phi_{occupancy}(m) + \phi_{borders}(m) \right) \quad (14)$$

Оптимизационная задача (14) 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 $p_{randomfactor}$ сохраняется $o^*(m_i)$ и $\Phi^*(m, Z)$, $\phi_{sonars}^*(m, Z)$, $\phi_{occupancy}^*(m)$, $\phi_{borders}^*(m)$; иначе возвращаемся в предыдущее состояние.

Прямая модель для картирования 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.

$$\psi = \psi_{free} + \psi_{occ} \quad (15)$$

- ▶ $\psi_{free} \equiv \psi_{free}(X_{free})$ - значение зависит от суммарной эффективной занятости X_{free} внутри Ω_{free} . Если X_{free} больше некоторого порога, то значение штрафа достаточно велико, иначе - небольшое.
- ▶ $\psi_{occ} \equiv \psi_{occ}(X_{occ})$ - 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

$$X_{\Omega} = \sum_{x_i \in \Omega} x_i w_i \quad (16)$$

Функции штрафа - кусочно линейные функции вида:

$$\psi_{free}^t = \begin{cases} \alpha_{free} X, & X < X_{free}^t \\ \alpha_{free} X_{free}^t + \beta_{free}^t (X - X_{free}^t), & X \geq X_{free}^t \end{cases} \quad (17)$$

$$\psi_{occ}^t = \begin{cases} 1 - \beta_{occ}^t X, & X < X_{occ}^t \\ 1 - \beta_{occ}^t X_{occ}^t - \alpha_{occ} (X - X_{occ}^t), & X \geq X_{occ}^t \end{cases} \quad (18)$$

Картирование методом градиентного спуска

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

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

Таким образом, задача картирования сводится к минимизации непрерывного штрафа:

$$m^*(Z) = \underset{m}{\operatorname{argmin}} \left(\sum_{x_i} R^i(x_i) + \sum_{t: z_t \in Z} (\psi_{free}^t + \psi_{occ}^t) \right) \quad (20)$$

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

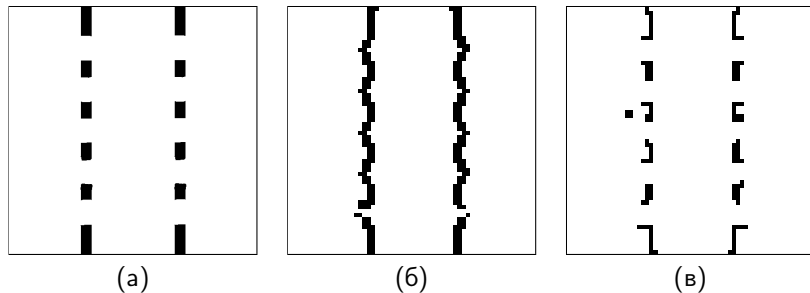
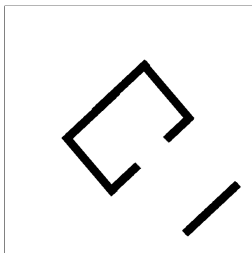


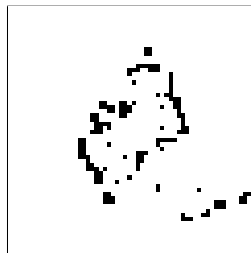
Рис. 2 : Результаты восстановления карты проходимости с помощью алгоритма картирования с обратной моделью (б) и методом стохастического градиента (в) на основе синтетических данных, собранных на территории (а).



(a)



(б)



(в)

Рис. 3 : Результаты восстановления карты проходимости с помощью алгоритма картирования с обратной моделью (б) и методом стохастического градиента (в) на основе реальных данных, собранных на территории (а).

Результаты картирования методом градиентного спуска

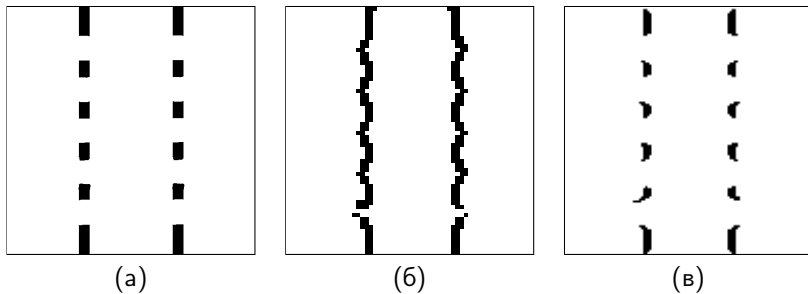
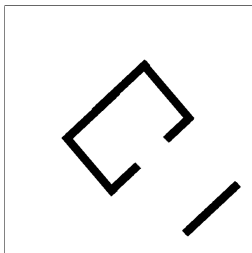


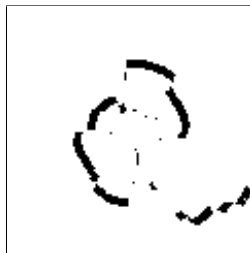
Рис. 4 : Результаты восстановления карты проходимости с помощью алгоритма картирования с обратной моделью (б) и методом градиентного спуска (в) на основе реальных данных, собранных на территории (а).



(a)



(б)



(в)

Рис. 5 : Результаты восстановления карты проходимости с помощью алгоритма картирования с обратной моделью (б) и методом градиентного спуска (в) на основе реальных данных, собранных на территории (а).

Заключение

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