

Моделирование колесных роботов

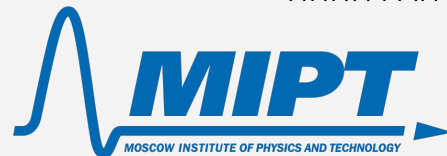
Лекция 6. Планирование пути и движения

Олег Шипитько

Москва, 2022



ИППИ РАН



СОДЕРЖАНИЕ ЛЕКЦИИ

1. Определение и постановка задачи
2. Пространство конфигураций
3. Клеточная декомпозиция (поиск в ширину, глубину, A^*)
4. Маршрутная декомпозиция (графы видимости, диаграмма Вороного, решетка состояний, RRT)
5. Метод потенциальных полей

ЧТО ТАКОЕ ПЛАНИРОВАНИЕ ПУТИ И ДВИЖЕНИЯ?

- ❑ **Планирование пути (path planning)** — нахождения свободного от коллизий перехода (пути) из некоторой начальной в некоторую конечную конфигурацию в некотором пространстве, содержащим препятствия.
- ❑ **Планирование движения (motion planning)** — планирование и исполнение команд приводящих к тому, что робот следует по запланированному пути.

ЧТО ТАКОЕ ПЛАНИРОВАНИЕ ПУТИ И ДВИЖЕНИЯ?

Планирование пути

“Через 3 километра поверните направо...”



Планирование движения

“Поверните руль на 20° градусов по часовой стрелке и нажмите педаль газа на 10%...”



ДРУГИЕ УПОТРЕБЛЯЕМЫЕ ТЕРМИНЫ

- ❑ **Глобальное планирование пути (global path planning, global path, route)** — имея начальную и конечную точку, а также карту, необходимо найти (оптимальный) путь. Часто при этом не учитываются кинематические ограничения робота.
- ❑ **Локальное планирование пути (local path planning, local path, trajectory)** — имея информацию, поступающую от сенсоров, необходимо найти участок пути, свободный от коллизий и как можно меньше отклоняющийся от ранее намеченного глобального пути.
- ❑ **Предотвращение столкновений (collision avoidance)** — тоже самое, что локальное планирование.

ГЛОБАЛЬНЫЙ ПУТЬ VS. ЛОКАЛЬНЫЙ ПУТЬ

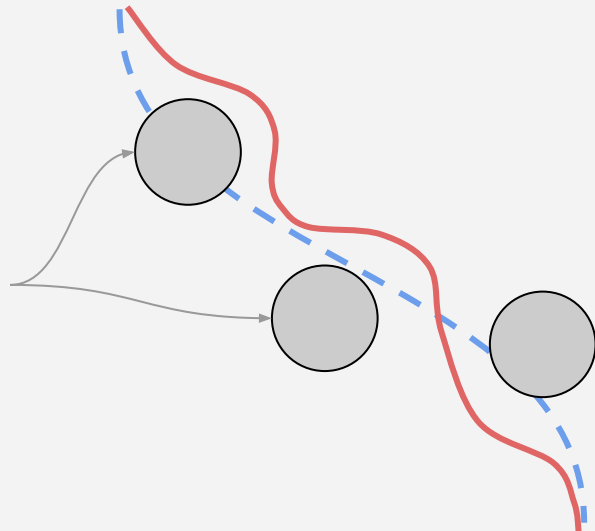
Старт



Финиш

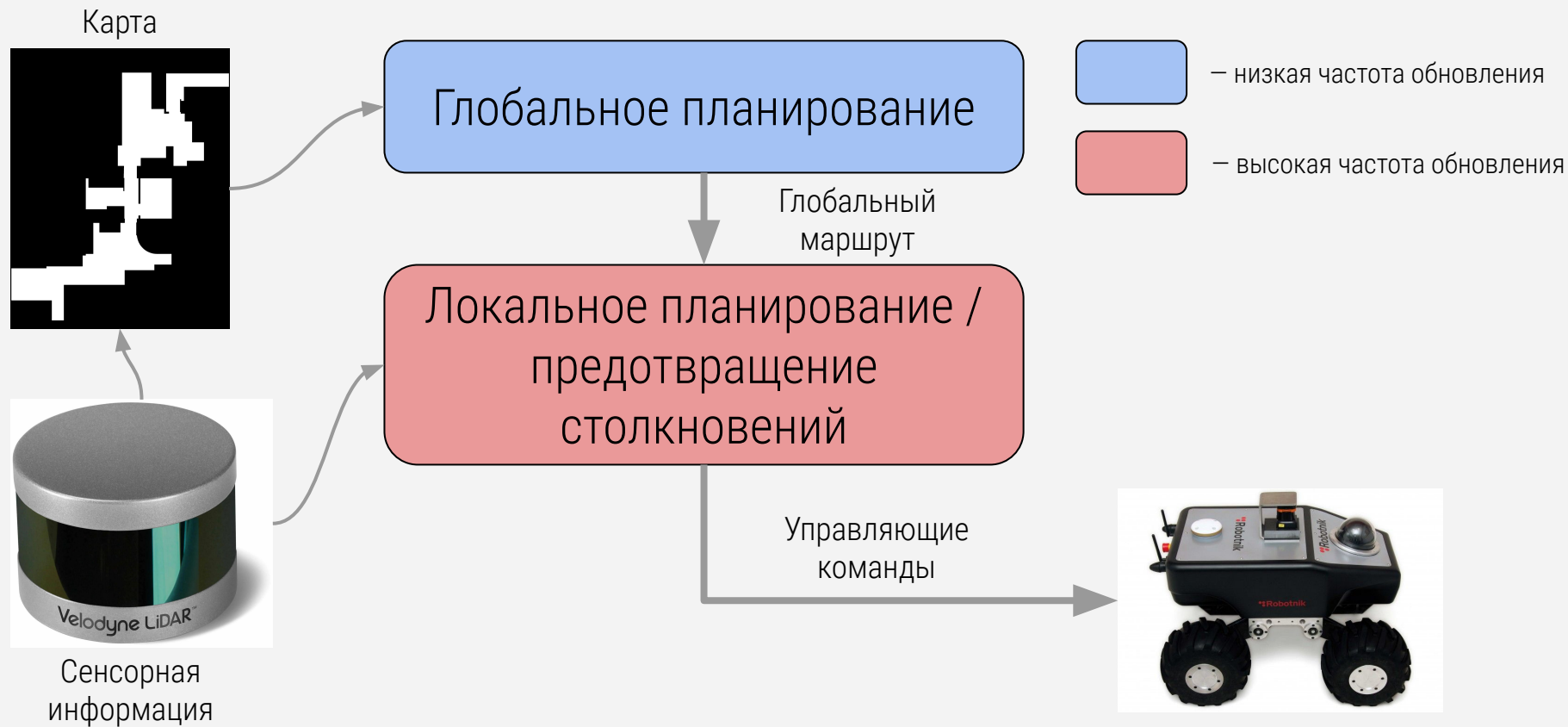
Старт

Динамически
детектируемые
препятствия



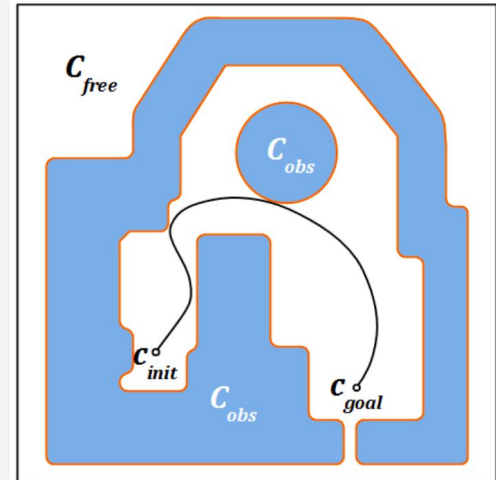
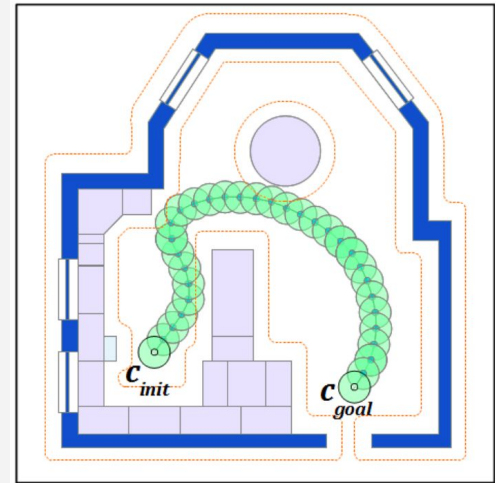
Финиш

ДВУХСЛОЙНАЯ АРХИТЕКТУРА ПЛАНИРОВАНИЯ



ПРОСТРАНСТВО КОНФИГУРАЦИЙ

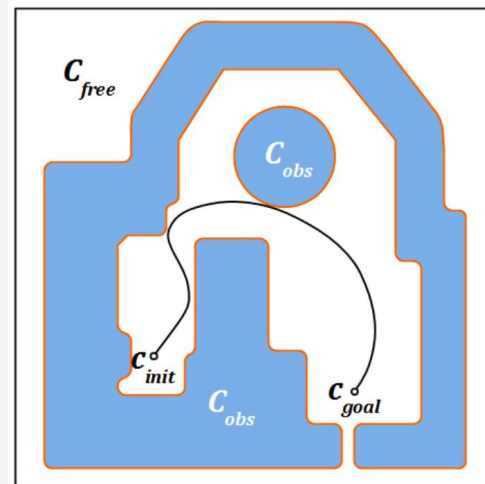
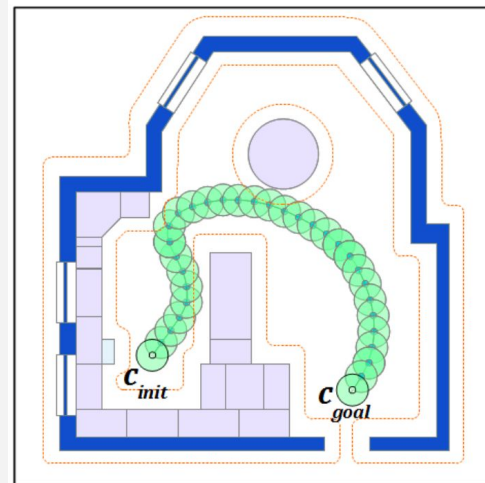
- ❑ Пусть q — вектор однозначно определяющий состояние робота
- ❑ Например, положение всех звеньев для манипулятора
- ❑ Для мобильного робота — положение в пространстве (x, y, θ)
- ❑ Тогда, пространство конфигураций C — пространство всех возможных состояний робота



ПРОСТРАНСТВО КОНФИГУРАЦИЙ

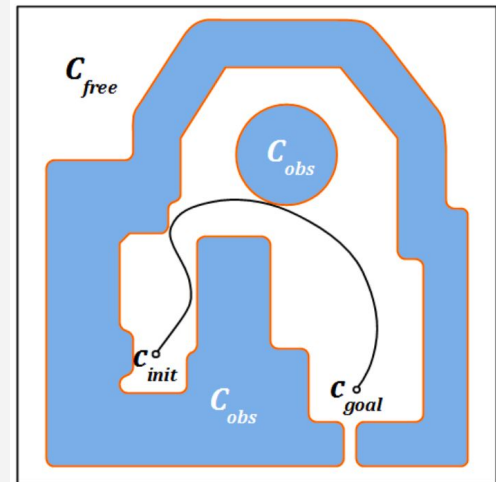
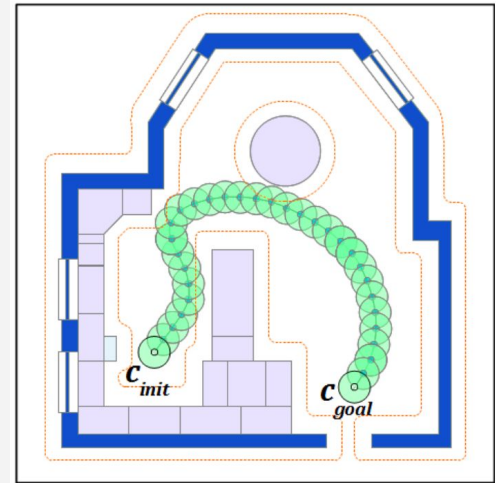
- ❑ Пусть $W = \mathbb{R}^m$ — рабочая область робота
(представление мира)
- ❑ $C_{obs} \in W$ — множество препятствий
- ❑ C_{init}, C_{goal} — начальная и конечная конфигурации
- ❑ Тогда:

$$C_{free} = \{q \in C \mid q \cap C_{obs} = \emptyset\}$$



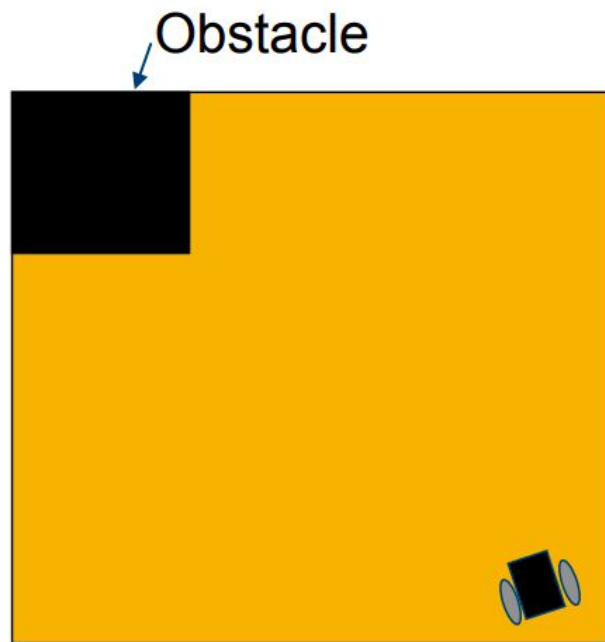
ПРОСТРАНСТВО КОНФИГУРАЦИЙ

- ❑ Тогда задача планирования пути сводится к поиску множества $T = \{C_{init} \dots C_{goal}\} \subset C_{free}$
- ❑ В такой постановке робот может рассматриваться как точка в C -пространстве

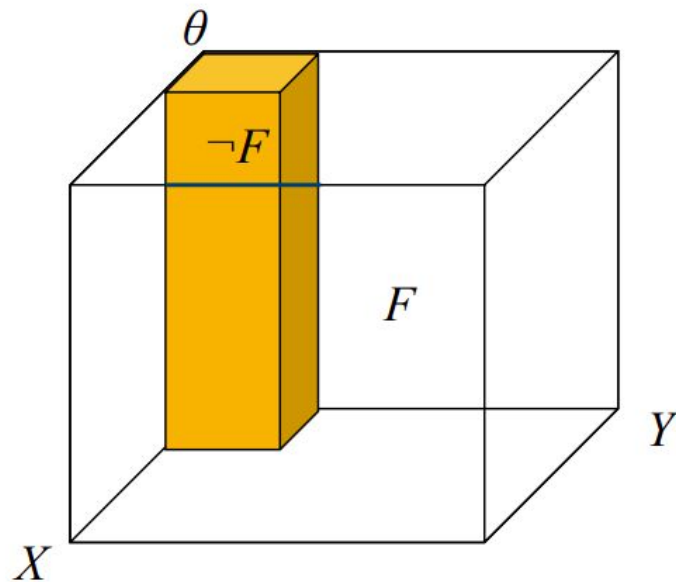


ПРОСТРАНСТВО КОНФИГУРАЦИЙ

Мобильный робот и статическое препятствие



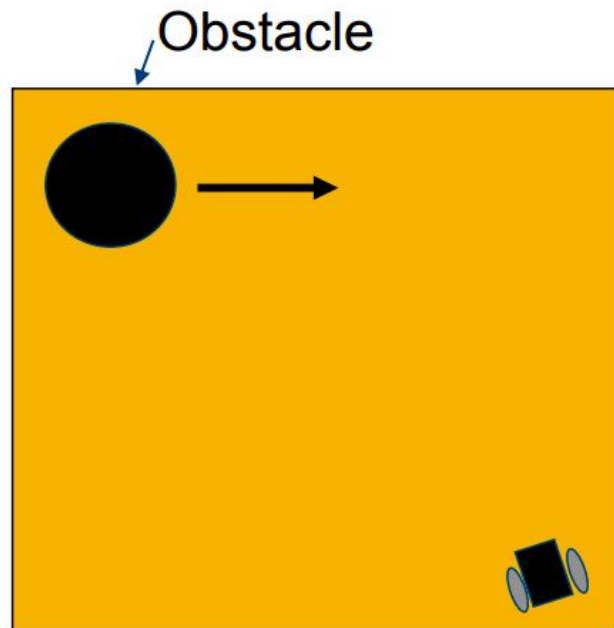
Workspace



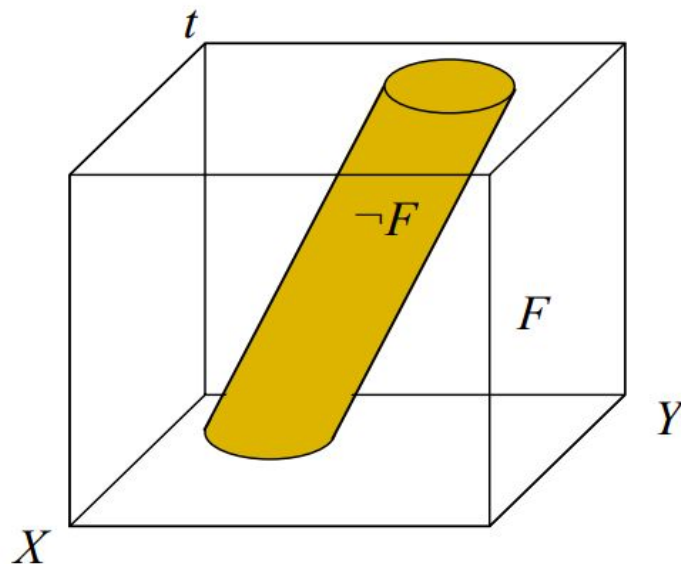
Configuration Space

ПРОСТРАНСТВО КОНФИГУРАЦИЙ

Мобильный робот и динамическое препятствие



Workspace



Configuration Space

ОБЩИЙ ПОДХОД К ПЛАНИРОВАНИЮ

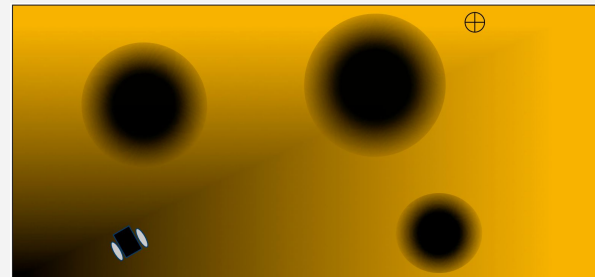
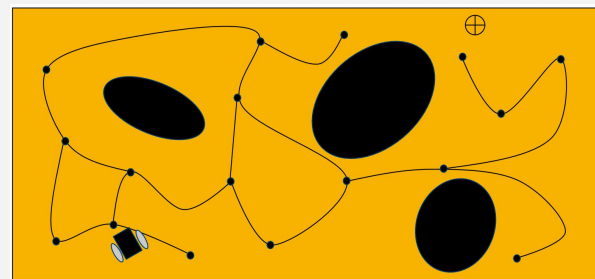
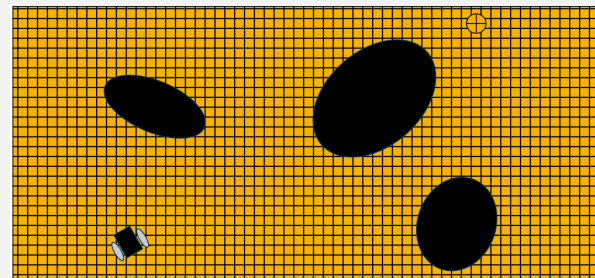
- ❑ Любой подход к планированию пути чаще всего сводится к следующему набору шагов:
 1. Определение пространства конфигураций C
 2. Декомпозиция (дискретизация) пространства C
 3. Поиск пути в дискретизированном пространстве
 - ❑ Поисковые алгоритмы
 - ❑ Алгоритмы на основе сэмплинга

ОПРЕДЕЛЕНИЕ ПРОСТРАНСТВА КОНФИГУРАЦИЙ

- ❑ Разные проблемы планирования могут иметь разные определения конфигурационного пространства. Для колесного робота возможно множество вариантов:
 - ❑ Для голономного робота (x,y)
 - ❑ Для неголономного робота (x,y,θ)
 - ❑ С учетом скоростей (x,y,θ,V,w)
 - ❑ ...

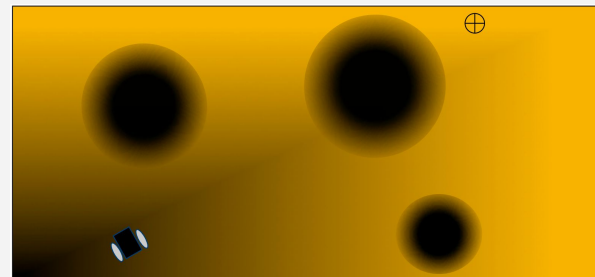
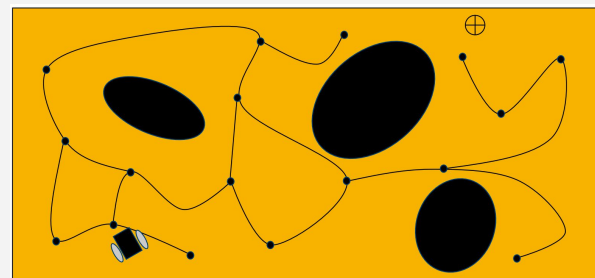
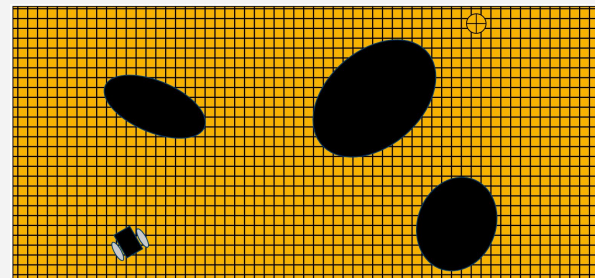
ДЕКОМПОЗИЦИЯ ПРОСТРАНСТВА КОНФИГУРАЦИЙ

- ❑ Типы декомпозиции пространства конфигураций C :
 - ❑ **Клеточная декомпозиция** (cell decomposition) – пространство разбивается на клетки. Граф связности строится на основе смежности клеток.
 - ❑ **Маршрутная декомпозиция** (roadmap decomposition) – представляет свободное пространство в виде связанных кривых (маршрутов).
 - ❑ **Потенциальное поле** (potential field) – определяет потенциальное поле над пространством конфигураций с минимумом в целевой точке и максимумами в препятствиях.



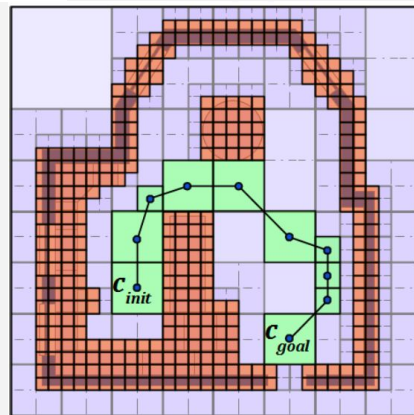
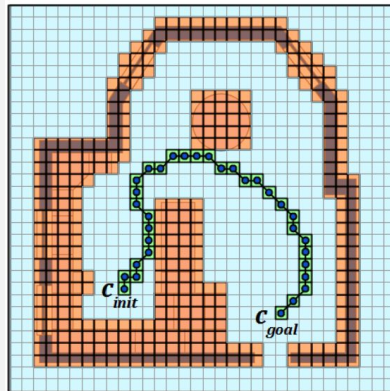
ДЕКОМПОЗИЦИЯ ПРОСТРАНСТВА КОНФИГУРАЦИЙ

- ❑ Типы декомпозиции пространства конфигураций C :
 - ❑ **Клеточная декомпозиция** (cell decomposition) – пространство разбивается на клетки. Граф связности строится на основе смежности клеток.
 - ❑ **Маршрутная декомпозиция** (roadmap decomposition) – представляет свободное пространство в виде связанных кривых (маршрутов)
 - ❑ **Потенциальное поле** (potential field) – определяет потенциальное поле над пространством конфигураций с минимумом в целевой точке и максимумами в препятствиях.



МЕТОДЫ НА ОСНОВЕ КЛЕТОЧНОЙ ДЕКОМПОЗИЦИИ

- ❑ **Все** пространство разбивается на регионы по тем или иным правилам
- ❑ Устанавливается смежность регионов
- ❑ Формируется граф связности
- ❑ Задача планирования сводится к задаче поиска пути в графе



НЕИНФОРМИРОВАННЫЙ ПОИСК VS. ИНФОРМИРОВАННЫЙ ПОИСК

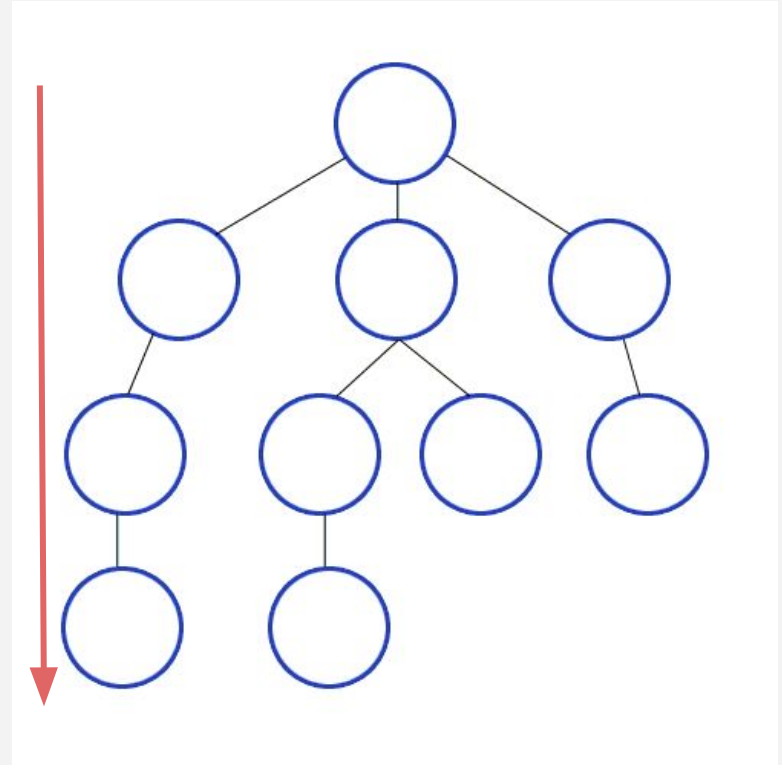
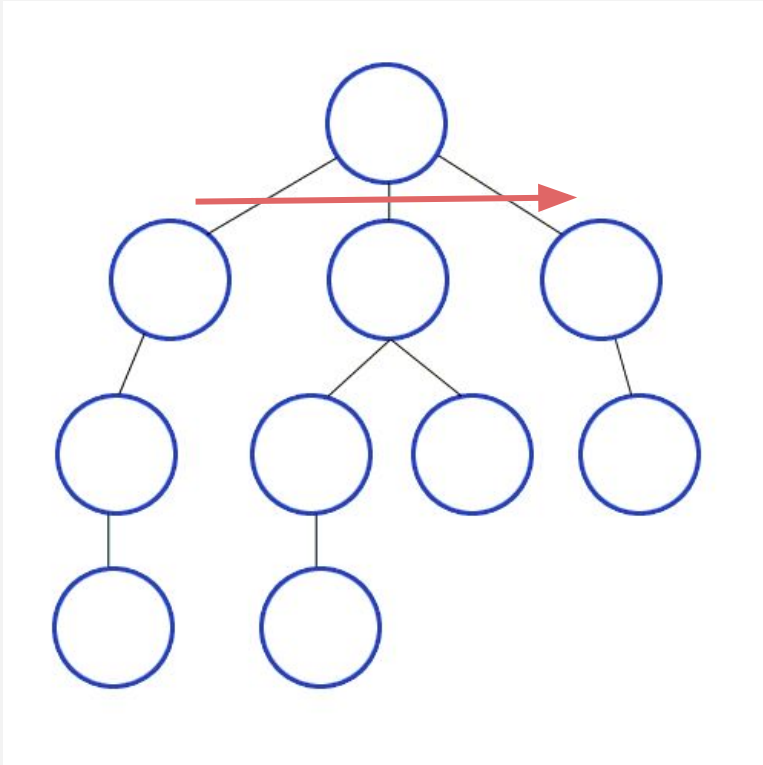
- ❑ **Неинформированный поиск (uninformed search, blind search, brute-force search)** — поиск пути в пространстве состояний при котором не используется вспомогательная информация. Единственное что можно делать — по-разному обследовать узлы графа.
- ❑ **Информированный поиск (informed search)** — когда доступна дополнительная информация (эвристика), позволяющая сказать, что один узел “более многообещающий”, чем другой.

НЕИНФОРМИРОВАННЫЙ ПОИСК

Алгоритмы:

- ❑ Поиск в ширину
- ❑ Поиск в глубину
- ❑ Двухнаправленный поиск
- ❑ Поиск с ограничением глубины
- ❑ Поиск по критерию стоимости
- ❑ ...

ПОИСК В ШИРИНУ VS. ПОИСК В ГЛУБИНУ



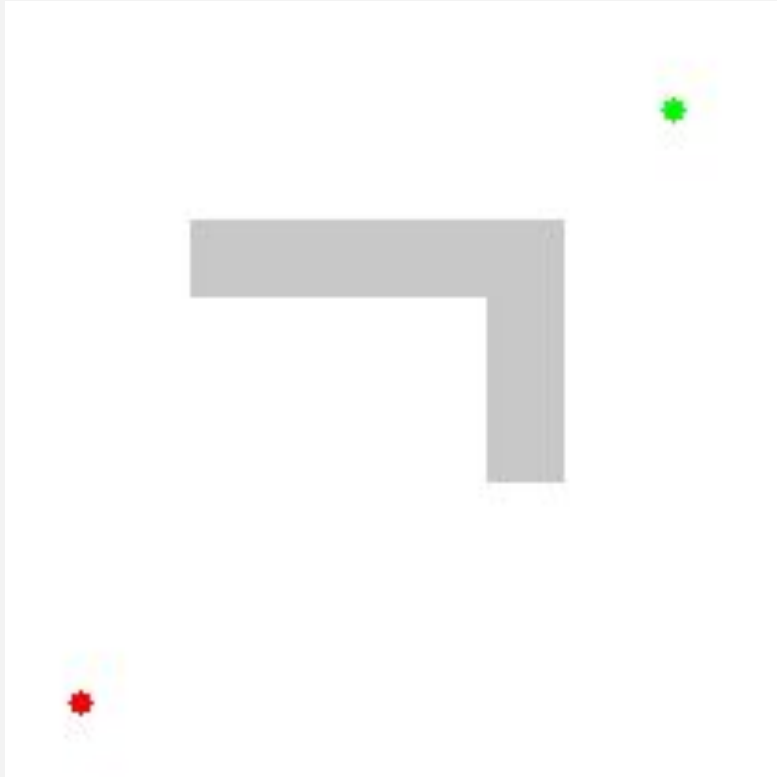
ПОИСК В ШИРИНУ

- ❑ **Поиск в ширину** является оптимальным алгоритмом неинформированного поиска, если все узлы (действия) имеют равную стоимость.
- ❑ **Поиск по критерию стоимости** обобщает идею поиска в ширину для случая, если узлы имеют разную стоимость.
- ❑ **Поиск по критерию стоимости** посещает узлы в порядке возрастания стоимости пути от корневого (стартового) узла (практически эквивалентен алгоритму Дейкстры).

A*

- ❑ Приоритет вершины для обхода определяется суммой:
 - ❑ $f(v) = g(v) + h(v)$,
 - ❑ где $g(v)$ — стоимость пути из начальной вершины в данную,
 - ❑ $h(v)$ — эвристическая оценка стоимости пути до целевой вершины.
- ❑ Таким образом, A* является обобщением алгоритма Дейкстры при $f(v) = g(v)$ и алгоритма жадного поиска при $f(v) = h(v)$.
- ❑ Алгоритм находит оптимальное решение при условии, что эвристическая оценка оптимистична: $h(v) \leq$ точной стоимости пути до целевой вершины
- ❑ В качестве эвристики часто используют евклидово расстояние до цели (L2 норму)

A*

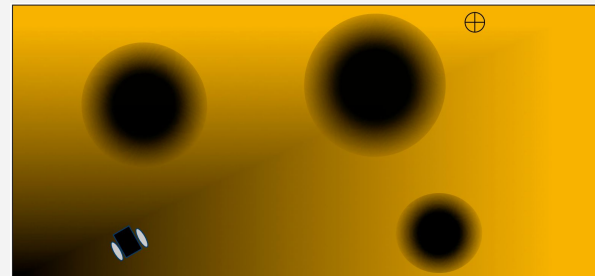
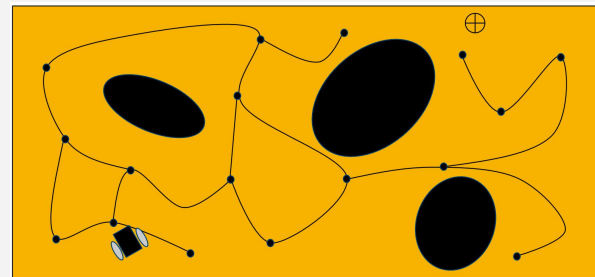
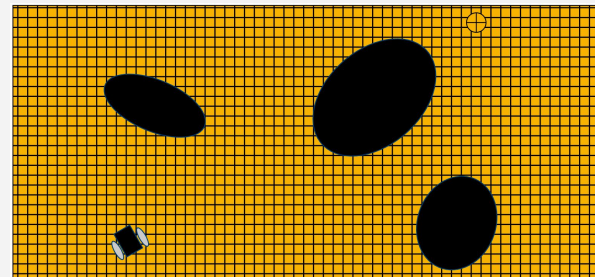


А* И ДРУЗЬЯ

- ❑ Семейство алгоритмов **D*** для учета динамической информации и перепланирования
- ❑ Семейство алгоритмов **Theta*** для планирования путей с произвольным направлением
- ❑ Семейство алгоритмов **Lifelong Planning A*** также используется для перепланирования с использованием предыдущих результатов при появлении новых препятствий в сцене
- ❑ ...

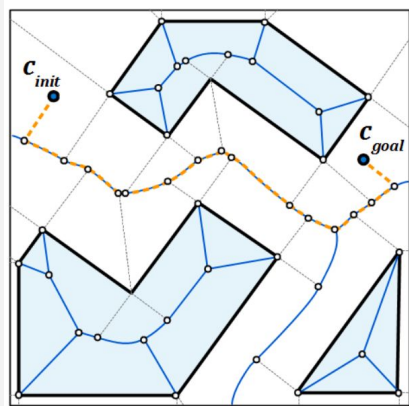
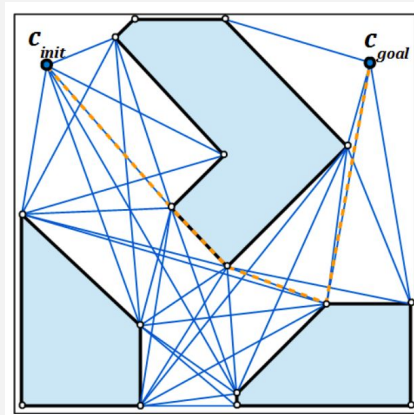
ДЕКОМПОЗИЦИЯ ПРОСТРАНСТВА КОНФИГУРАЦИЙ

- ❑ Типы декомпозиции пространства конфигураций C :
 - ❑ **Клеточная декомпозиция** (cell decomposition) – пространство разбивается на клетки. Граф связности строится на основе смежности клеток.
 - ❑ **Маршрутная декомпозиция** (roadmap decomposition) – представляет свободное пространство в виде связанных кривых (маршрутов)
 - ❑ **Потенциальное поле** (potential field) – определяет потенциальное поле над пространством конфигураций с минимумом в целевой точке и максимумами в препятствиях.



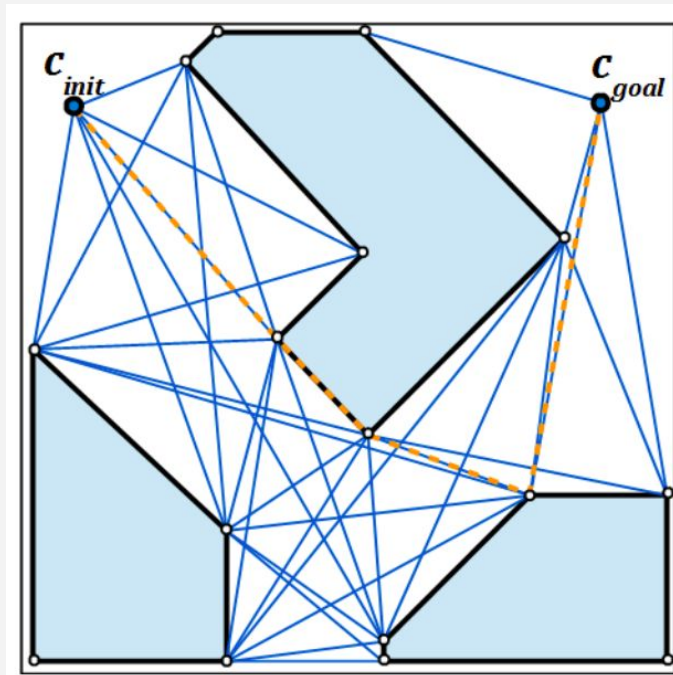
МЕТОДЫ НА ОСНОВЕ МАРШРУТНОЙ ДЕКОМПОЗИЦИИ

- ❑ **Свободное** пространство разбивается на регионы по тем или иным правилам
- ❑ Устанавливается смежность регионов
- ❑ Формируется граф связности
- ❑ Задача планирования сводится к задаче поиска пути в графе



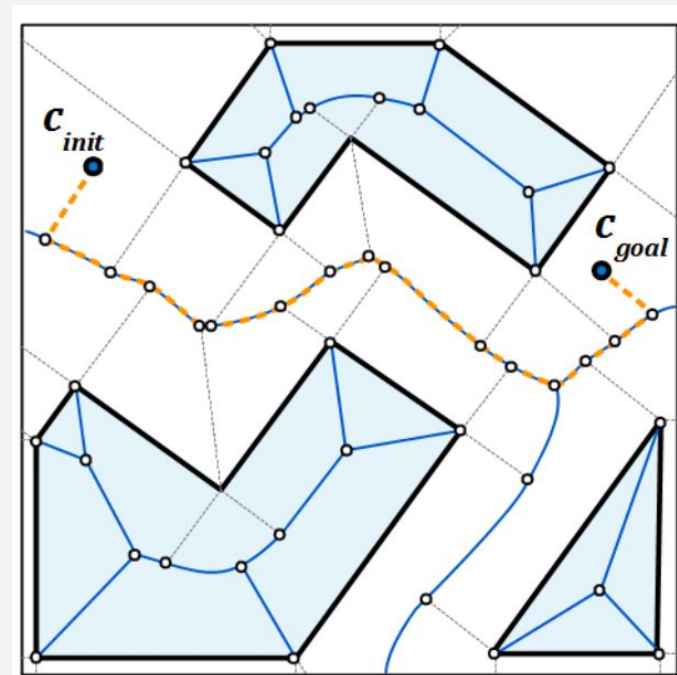
ГРАФЫ ВИДИМОСТИ (Visibility Graphs)

- ❑ В качестве узлов используются все вершины препятствий + начальная и конечная точка маршрута
- ❑ Две вершины соединяются ребрами если:
 - ❑ Одна вершина “видна” из другой
 - ❑ Полученное ребро не пересекает препятствий
- ❑ Сложность построения графа $O(n^2 \log n)$, где n — количество вершин



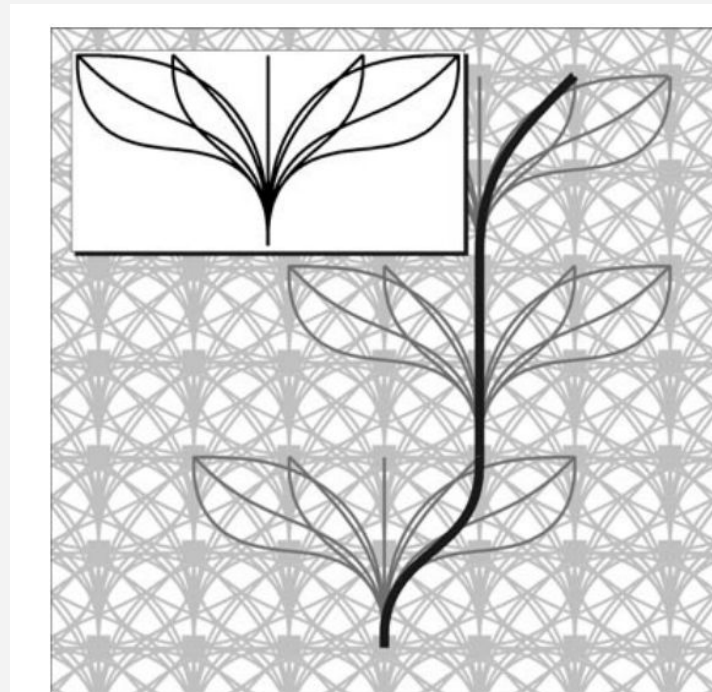
ДИАГРАММЫ ВОРОНОГО (Voronoi Diagram)

- ❑ Разбиение конечного множества точек при котором каждая область разбиения образует множество точек, более близких к одному из элементов множества , чем к любому другому элементу
- ❑ Маршруты построенные по диаграмме Вороного находятся на наибольшем удалении от препятствий
- ❑ Наивный алгоритм— $O(n^4)$. Лучший - $O(n \log n)$



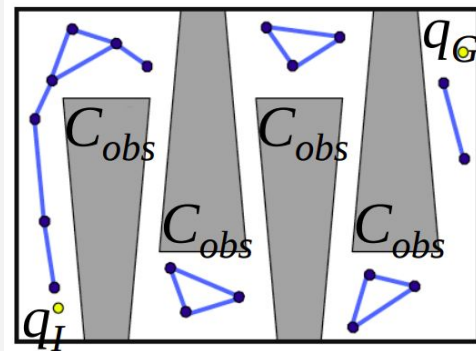
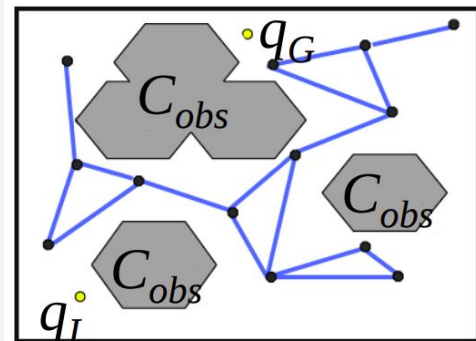
МЕТОД РЕШЕТКИ СОСТОЯНИЙ (State Lattice)

- ❑ Пространство дискретизируется таким образом, чтобы переходы между узлами были заведомо кинематически достижимыми
- ❑ Переходы строятся из заранее определенных примитивов движения
- ❑ Любой из существующих алгоритмов поиска в направленном графе может быть применен для поиска пути в полученной решетке состояний



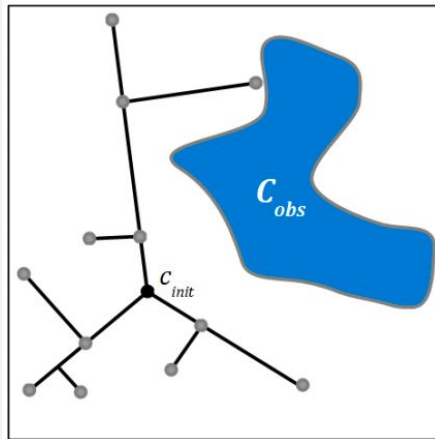
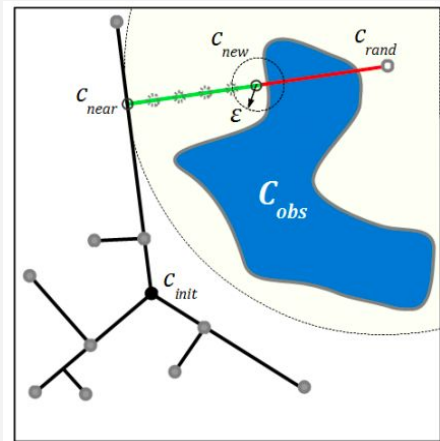
ВЕРОЯТНОСТНАЯ МАРШРУТНАЯ ДЕКОМПОЗИЦИЯ (Probabilistic Road Maps)

- ❑ Берутся случайные сэмплы из C_{free}
- ❑ Близкие сэмплы соединяются друг с другом ребром при условии отсутствия столкновения на ребре
- ❑ Близость узлов может определяться разными способами:
 - ❑ k-ближайшими соседями
 - ❑ Все узлы в фиксированном радиусе
 - ❑ ...
- ❑ Путь в полученном графе ищется любым поисковым алгоритмом



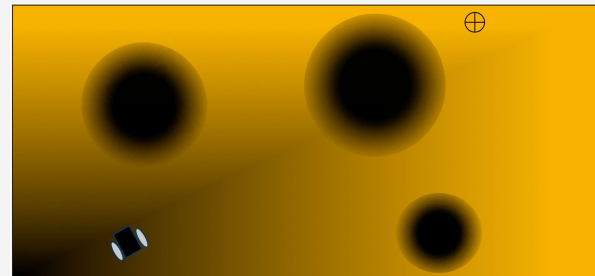
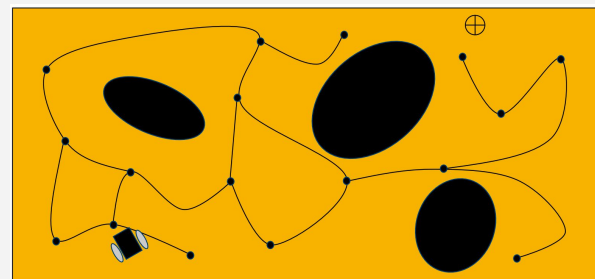
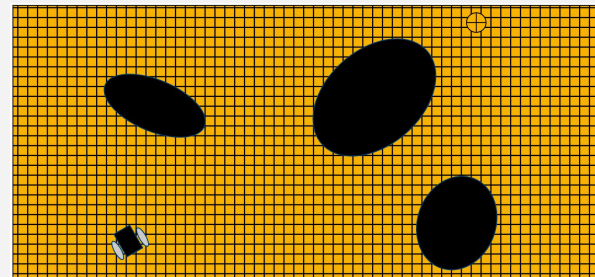
БЫСТРО ИССЛЕДУЮЩИЕ СЛУЧАЙНЫЕ ДЕРЕВЬЯ (Rapidly Exploring Random Trees)

- ❑ Основная идея – более эффективное исследование пространства
- ❑ Суть алгоритма:
 - ❑ Изначально дерево состоит из стартового узла C_{init}
 - ❑ Сгенерируем случайный узел $C_{rand} \in C_{free}$
 - ❑ Соединим ребром C_{rand} с ближайшим узлом дерева C_{near}
 - ❑ При этом, если ребро проходит через препятствие то оставим максимальную часть ребра (C_{near}, C_{new}) , не пересекающую препятствий
 - ❑ Каждую n -ую итерацию в качестве C_{rand} будем брать C_{goal} . Если удалось соединить C_{goal} с деревом – путь найден



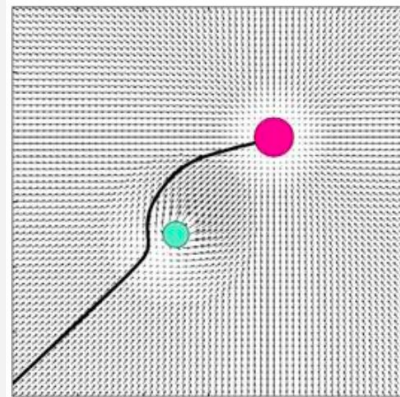
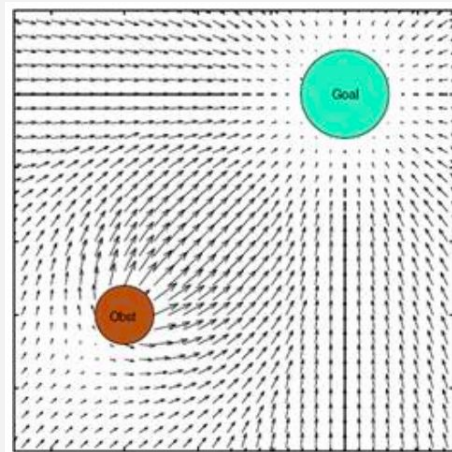
ДЕКОМПОЗИЦИЯ ПРОСТРАНСТВА КОНФИГУРАЦИЙ

- ❑ Типы декомпозиции пространства конфигураций C :
 - ❑ **Клеточная декомпозиция** (cell decomposition) – пространство разбивается на клетки. Граф связности строится на основе смежности клеток.
 - ❑ **Маршрутная декомпозиция** (roadmap decomposition) – представляет свободное пространство в виде связанных кривых (маршрутов)
 - ❑ **Потенциальное поле** (potential field) – определяет потенциальное поле над пространством конфигураций с минимумом в целевой точке и максимумами в препятствиях.



МЕТОДЫ НА ОСНОВЕ ПОТЕНЦИАЛЬНЫХ ПОЛЕЙ

- ❑ Основная идея – задание **навигационной** функции f , указывающей направление к целевой точке из любой текущей конфигурации робота
- ❑ $\nabla f(q)$ – указывает направление к целевой точке
- ❑ f задается таким образом, что препятствия отталкивают робота, а целевая точка притягивает



ЧТО МЫ УПУСТИЛИ?

- ❑ Огромное количество улучшений и расширений рассмотренных базовых методов
- ❑ Методы локального планирования на основе всевозможных кривых (полиномиальное планирование, клотоиды, Кривая Безье, пути Дубинса и т.д.)
- ❑ Оптимизационные методы планирования
- ❑ Методы на основе динамического программирования / Марковский процесс принятия решений

ДОПОЛНИТЕЛЬНЫЕ ИСТОЧНИКИ

1. [Казаков К. А., Семенов В. А. Обзор современных методов планирования движения](#)
2. [ARW – Lecture 03 Motion Planning. Chris Clark](#)
3. [Introduction to Mobile Robotics. Path and Motion Planning. Wolfram Burgard](#)
4. [Amit's A* Pages](#)
5. [PythonRobotics](#)

ИНФОРМАЦИЯ О ПРЕЗЕНТАЦИИ

Эта презентация была подготовлена Олегом Шипитько в рамках курса “Моделирование колесных роботов” кафедры когнитивных технологий Московского физико-технического института (МФТИ). Автор выражает благодарность, авторам, чьи материалы были использованы в презентации. В случае, если вы обнаружили в презентации свои материалы, свяжитесь со мной, для включения в список авторов заимствованных материалов.

This presentation was prepared by Oleg Shipitko as part of the “Mobile Robotics” course at the Department of Cognitive Technologies, Moscow Institute of Physics and Technology. The author is grateful to the authors whose materials were used in the presentation. If you find your materials in a presentation, contact me to be included in the list of contributing authors.