IN THE NAME OF GOD

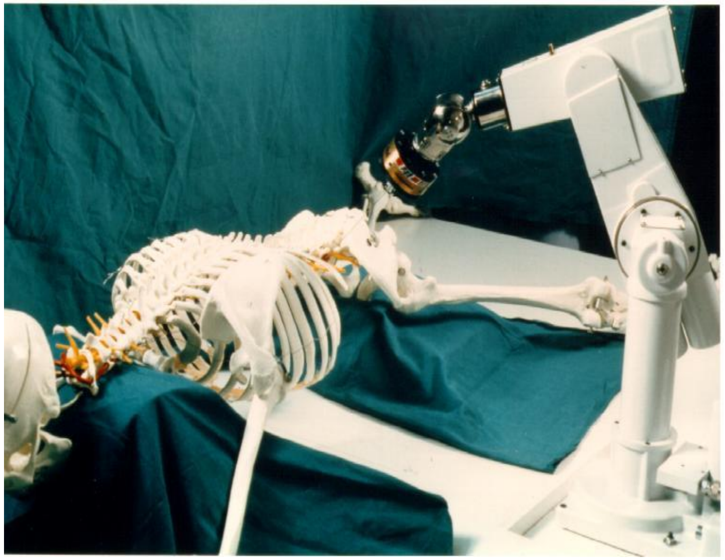
COMPUTATIONAL GEOMETRY

Chapter 13

Robot motion planning

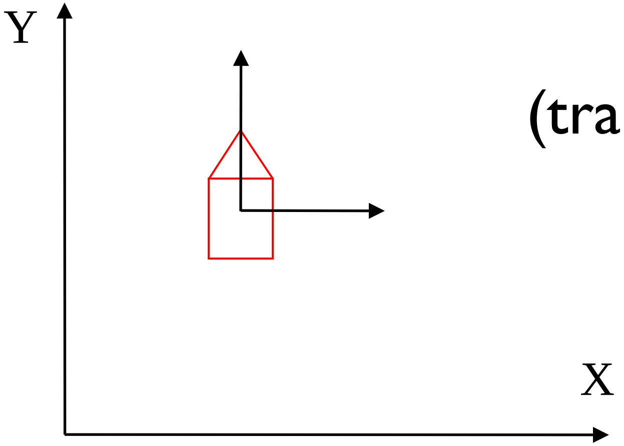
پیش گفتار

- هدف روباتیک طراحی روباتهای خودکار است به طوریکه فقط “هدف” مشخص شده باشد نه “رویه”
- برای این منظور روبات باید قادر به پیش بینی و تصمیم گیری درمورد حرکتش باشد
- و بازهم برای رسیدن به این مهم، روبات باید اطلاعاتی از محیط اطراف داشته باشد.

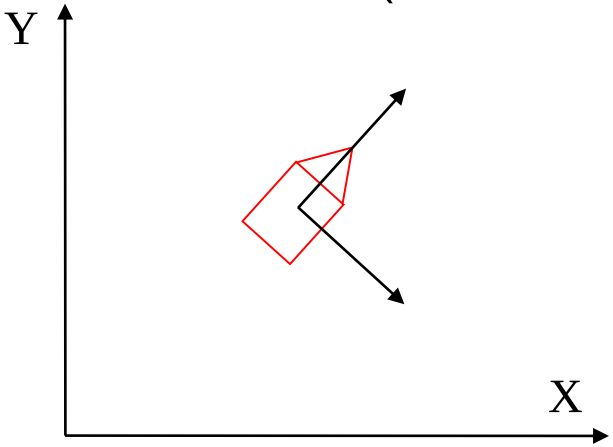


انواع روبات

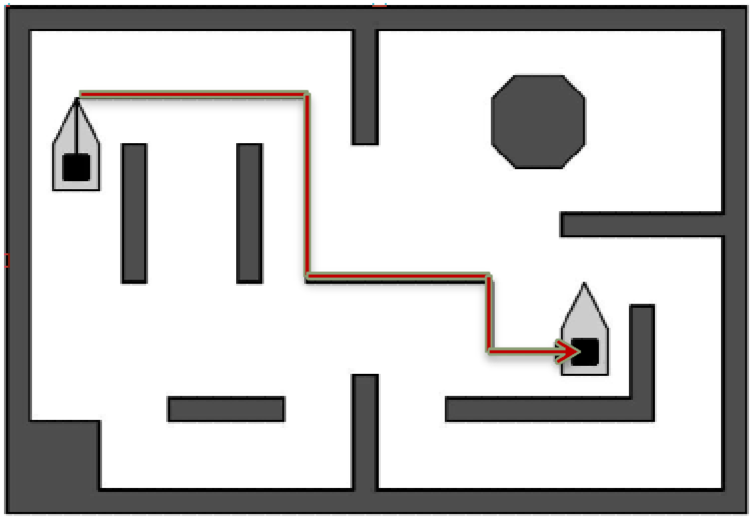
- روبات با حرکت انتقالی (translation robot)



- روبات با حرکت انتقالی و دورانی (rotation robot)



مسئله: یافتن مسیری برای رسیدن ازنقطه ای به نقطه دیگر،بدون برخورد با محیط اطراف



فرضیات:

- محیط ۲ بعدی است
- موانع چند ضلعی هستند
- روبات چندضلعی ساده است
- محیط static است
- روبات فقط قادر به حرکت انتقالی است

تعاریف

- فضای کار: محیط حرکت روبات،شامل مجموعه ای از موانع $S=\{p_1,p_2,\ldots,p_t\}$و روبات R
- نقطه مرجع:نقطه ای در فضای کار و نشان دهنده موقعیت روبات
- پارامترهای معرف روبات(فضای ۲بعدی)

  روبات با حرکت انتقالی: R(x,y)

  روبات با حرکت انتقالی و دورانی: R(x,y,ϕ)

  « ϕ زاویه دوران نسبت به خط افق-حول نقطه مرجع-در جهت پادساعتگرد»

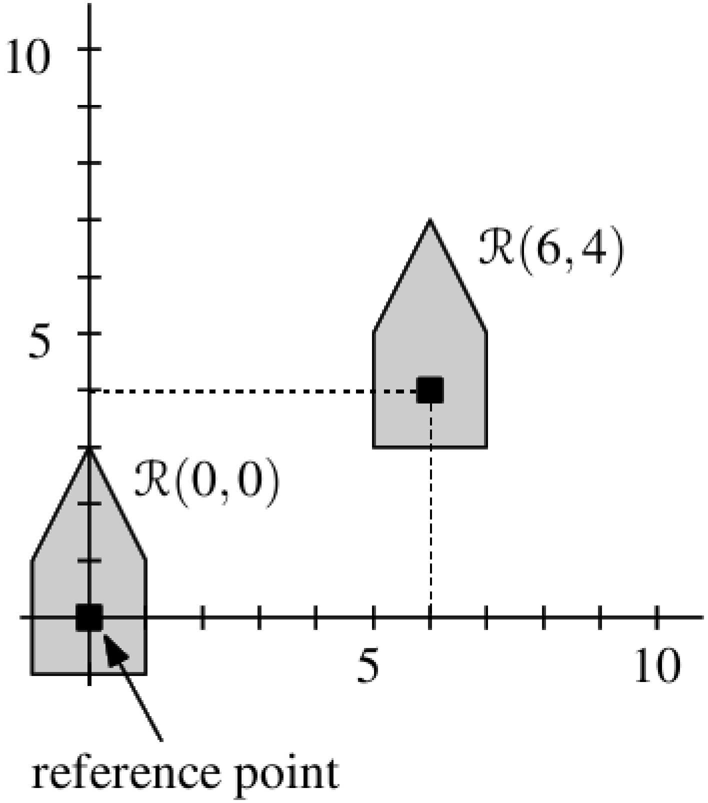
- فضای پارامتری روبات را،فضای پیکربندی آن گفته و باC(R) نشان می دهیم.

- درجه آزادی(DOF):تعداد پارامترهای مشخص کننده مکان و وضعیت روبات

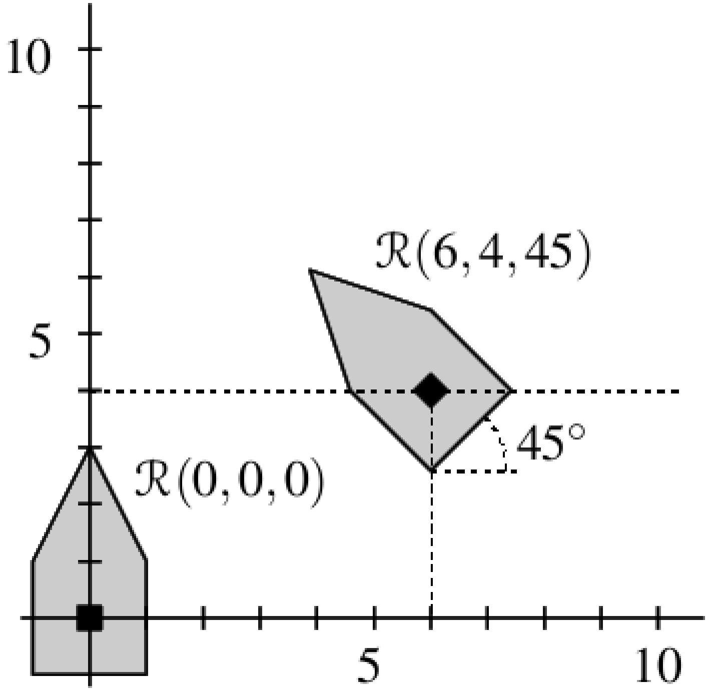
روبات در فضای ۲بعدی
- باحرکت انتقالی ← DOF=2
- با حرکت انتقالی و دورانی ← DOF=3

روبات در فضای ۳ بعدی
- با حرکت انتقالی ← DOF=3
- با حرکت انتقالی و دورانی ← DOF=6

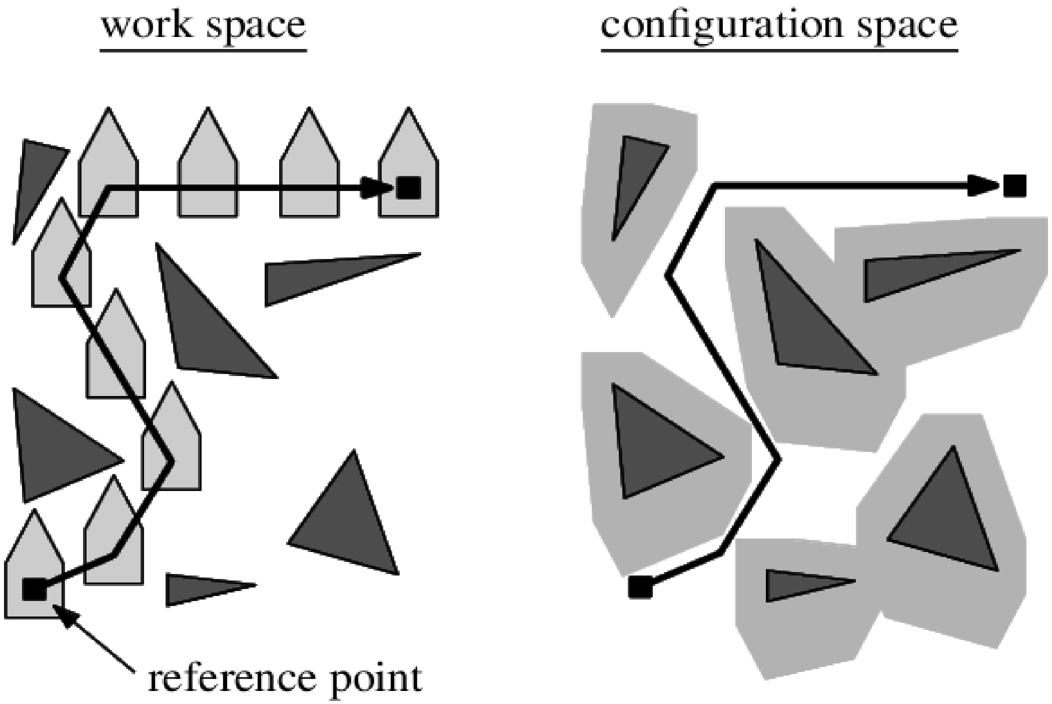
Translation robot


10
5
5
10
$\mathcal{R}(6,4)$
$\mathcal{R}(0,0)$
reference point

Rotation robot

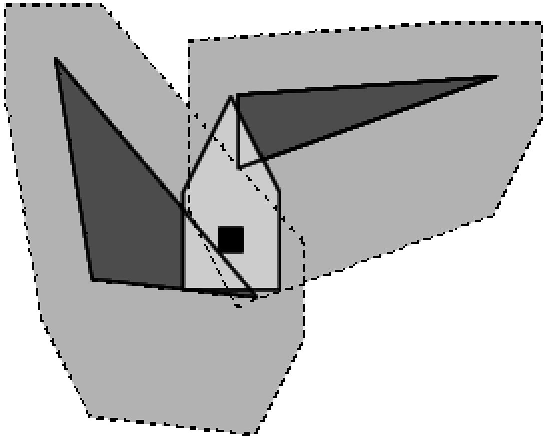

10
5
5
10
$\mathcal{R}(6,4,45)$
45°
$\mathcal{R}(0,0,0)$

- $C_{forb}(R,S)$: نقاطی از فضای پیکربندی که در آن نقاط روبات با یکی از موانع تقاطع دارد.
- $C_{free}(R,S)$: باقیمانده فضای پیکربندی


work space
configuration space
reference point

- یک مانع p_i بصورت مجموعه ای از نقاط (x,y) در فضای پیکربندی نگاشت داده میشود بطوریکه R(x,y) با p_i مشترک باشد.

  مجموعه حاصل را فضای پیکربندی موانع نامیده و با C-obstacles نشان می دهیم.
- C-obstacles ها ممکن است در زمانی که موانع در فضای کاری از هم جدا هستند، همپوشانی داشته باشند و این زمانی اتفاق می افتد که روبات در یک لحظه با بیش از یک مانع برخورد داشته باشد.



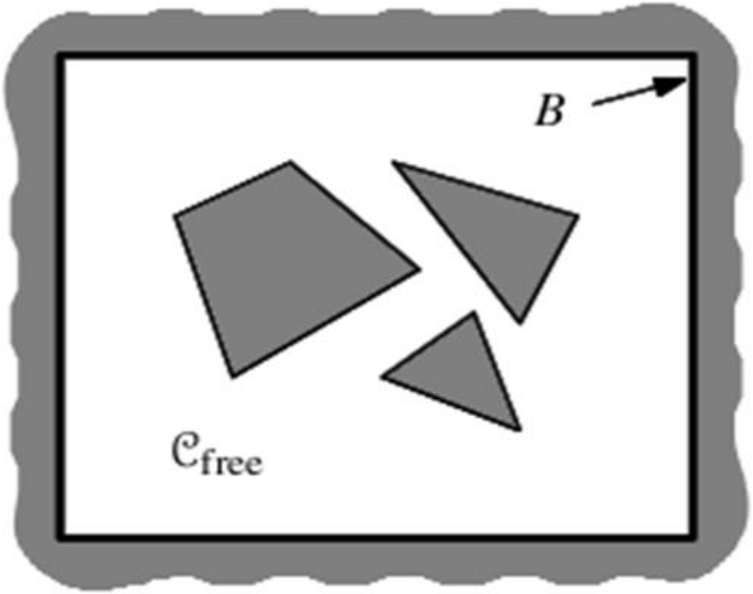
سوال:

- ✓ لغزیدن روبات روی مانع به معنای برخورد است؟ به عبارت دیگر، موانع را از لحاظ توپولوژیکی مجموعه هایی باز در نظر می گیریم یا بسته؟

جواب :موانع را باز در نظر میگیریم بدین معنی که روبات مجاز به حرکت و لغزیدن بر روی موانع است وچنین حرکتی،برخورد تلقی نمی شود.

روبات نقطه ای

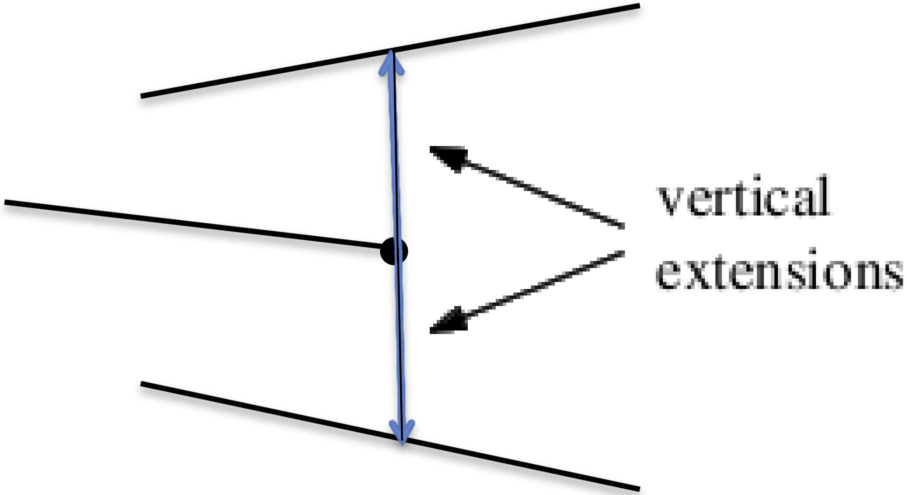
- برای درک بهتر مسئله ابتدا به جای روبات چندضلعی، روبات را بصورت یک نقطه فرض می کنیم(نقطه عطف روبات، خود روبات است)
- در این فضا، موانع چند ضلعی هایی مجزا هستند و n، تعداد کل رئوس موانع است.
- ازآنجاییکه فضای پیکربندی بی نهایت بزرگ است، بنابراین این فضا را با استفاده از یک مانع بزرگ(B) محدود میکنیم که دربرگیرنده همه موانع است و ناحیه غیر مجاز آن، خارج ازB قرار دارد

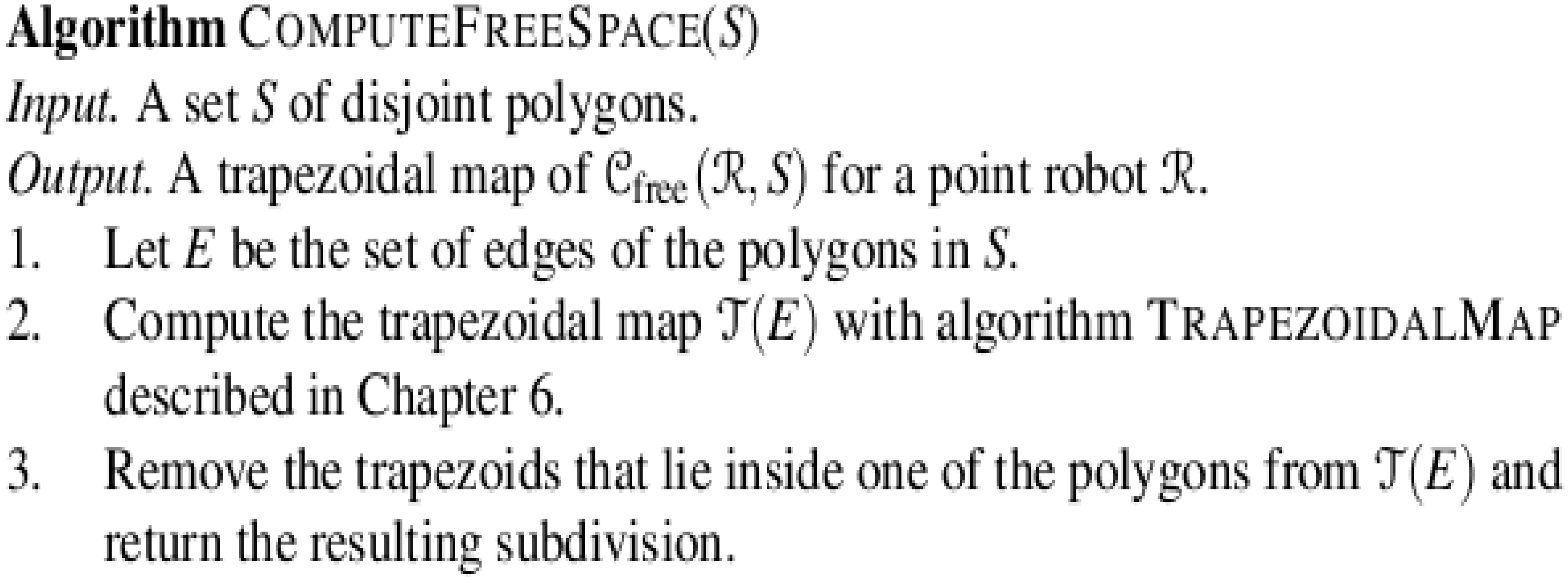

B
$\mathcal{C}_{\text{free}}$

- و بنابراین

$$C_{free} = B \setminus \bigcup_{i=1}^{t} P_i$$

- برای این کار نیاز به ساختمان داده ای است که فضای آزاد را ذخیره کند. این ساختمان داده را با استفاده از فصل ۶ (نقشه ذوزنقه ای) بدست می آوریم.


vertical
extensions

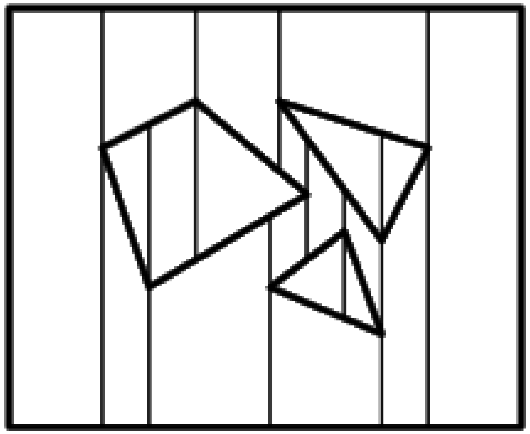


Algorithm COMPUTEFREESPACE(S)

Input. A set S of disjoint polygons.

Output. A trapezoidal map of $\mathcal{C}_{\text{free}}(\mathcal{R}, S)$ for a point robot $\mathcal{R}$.

```
Let $E$ be the set of edges of the polygons in $S$.
Compute the trapezoidal map $\mathcal{T}(E)$ with algorithm TRAPEZOIDALMAP
    described in Chapter 6.
Remove the trapezoids that lie inside one of the polygons from $\mathcal{T}(E)$ and
    return the resulting subdivision.
```

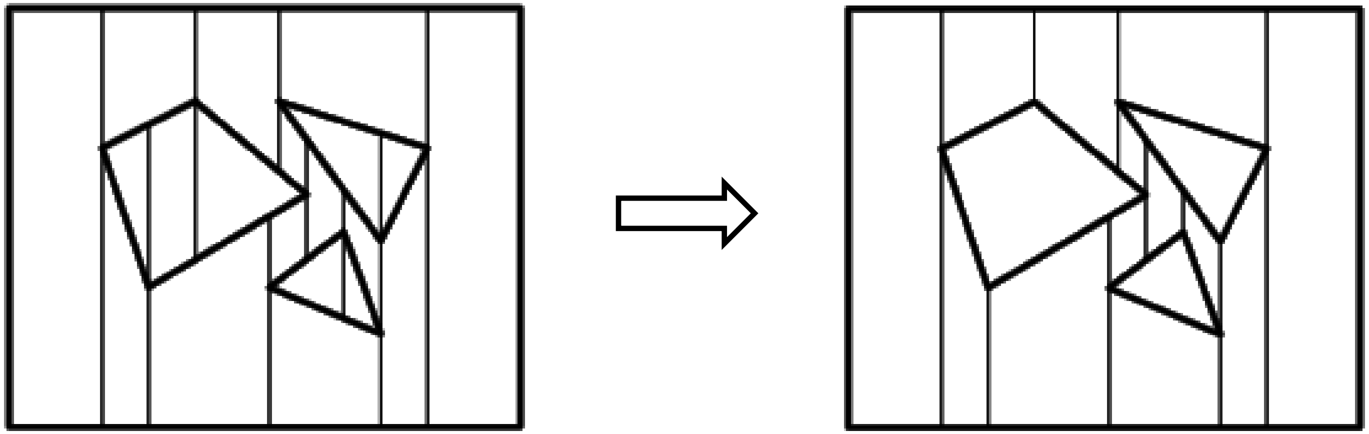


چگونه ذوزنقه های داخل حفره ها را بیابیم؟

نقشه ذوزنقه ای اطلاعات زیر را در اختیار ما میگذارد:

- یال Top هرکدام از ذوزنقه ها
- مانعی که این یال به آن تعلق دارد

اگر این یال، مانع را از بالا محدود کند آنگاه ذوزنقه ی مربوط به آن یال حذف میشود



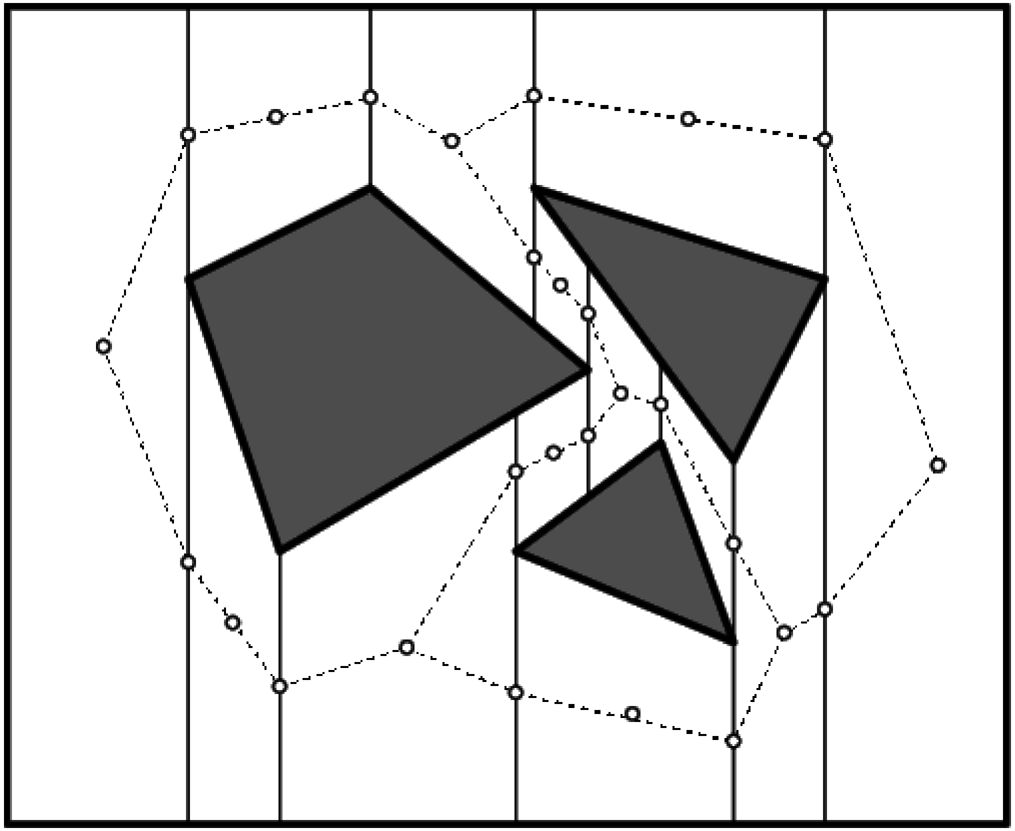
لم 13.1 زمان مورد انتظار ساخت یک نقشه ذوزنقه ای برای

“فضای پیکربندی آزاد”،”ربات نقطه ای” و “مجموعه ای از موانع چند ضلعی بدون برخورد و با n یال ”، O(n log n) است.

- زمان مورد انتظار برای ساخت نقشه ذوزنقه ای بدون حذف ذوزنقه های داخل موانع-با استفاده از الگوریتم رندوم: O(n log n)
- زمان حذف ذوزنقه های داخل موانع: O(n)

- نقشه ذوزنقه ای فضای آزاد را با $T(C_{free})$ نشان می دهیم.

- نقشه راه(g_{road}): یک گراف مسطح است که به فرم زیر ساخته میشود:
  - یک نود به ازای هر ذوزنقه و در مرکز آن قرار می دهیم.
  - یک نود به ازای هر امتداد عمودی و در میان آن در نظر میگیریم.
  - بین دو نود یک کمان وجود دارد اگر وتنها اگر یک نود در مرکز ذوزنقه و دیگری روی کناره ی همان ذوزنقه باشد.
  - کمان ها خطوط مستقیم هستند



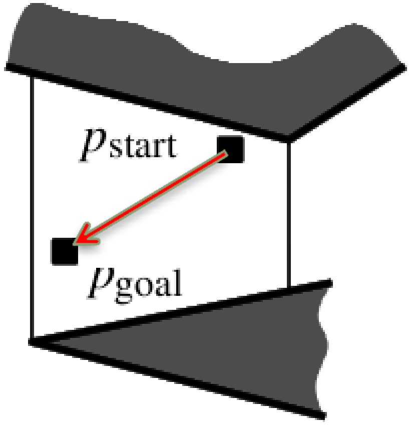
g_{road} با استفاده از ساختار DECL و خاصیت همسایگی دو ذوزنقه، در زمان O(n) قابل ساخت است.

- برنامه ریزی حرکت از p_{start} به p_{goal}

گام ۱- یافتن Δ_{start} و Δ_{goal} که به ترتیب ذوزنقه های حاوی p_{start} و p_{goal} هستند

گام ۲- دو حالت امکان پذیر است:

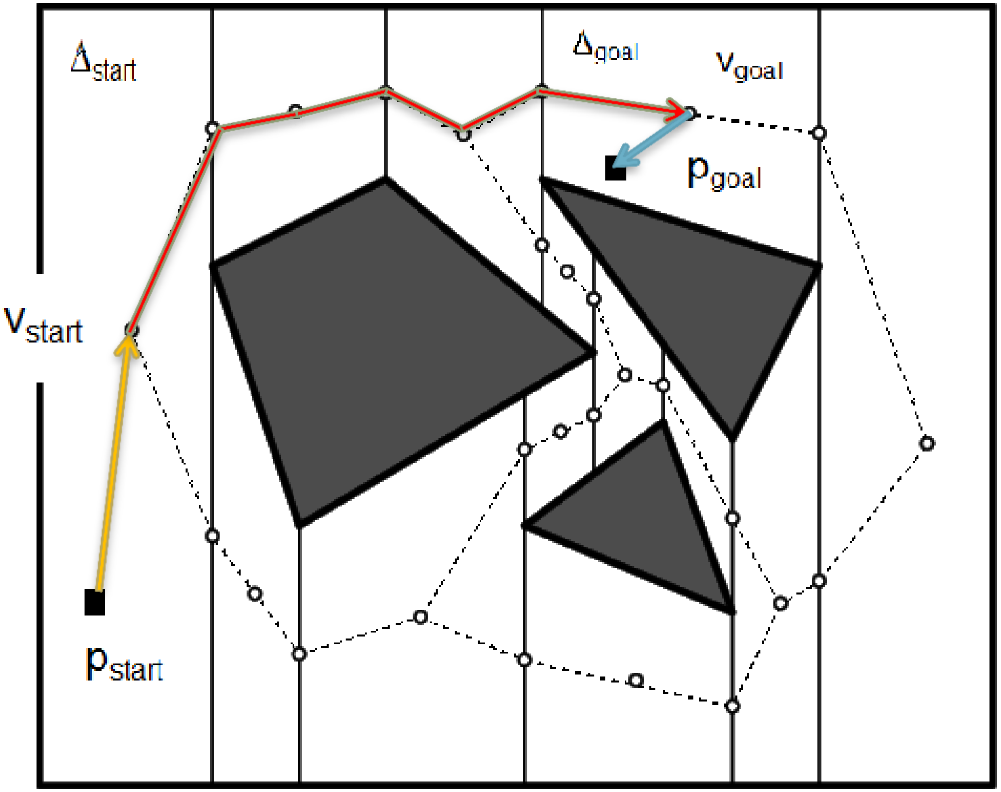
a) $\Delta_{start} = \Delta_{goal}$ ، در این حالت مسیر، یک خط مستقیم از p_{start} به p_{goal} خواهد بود.

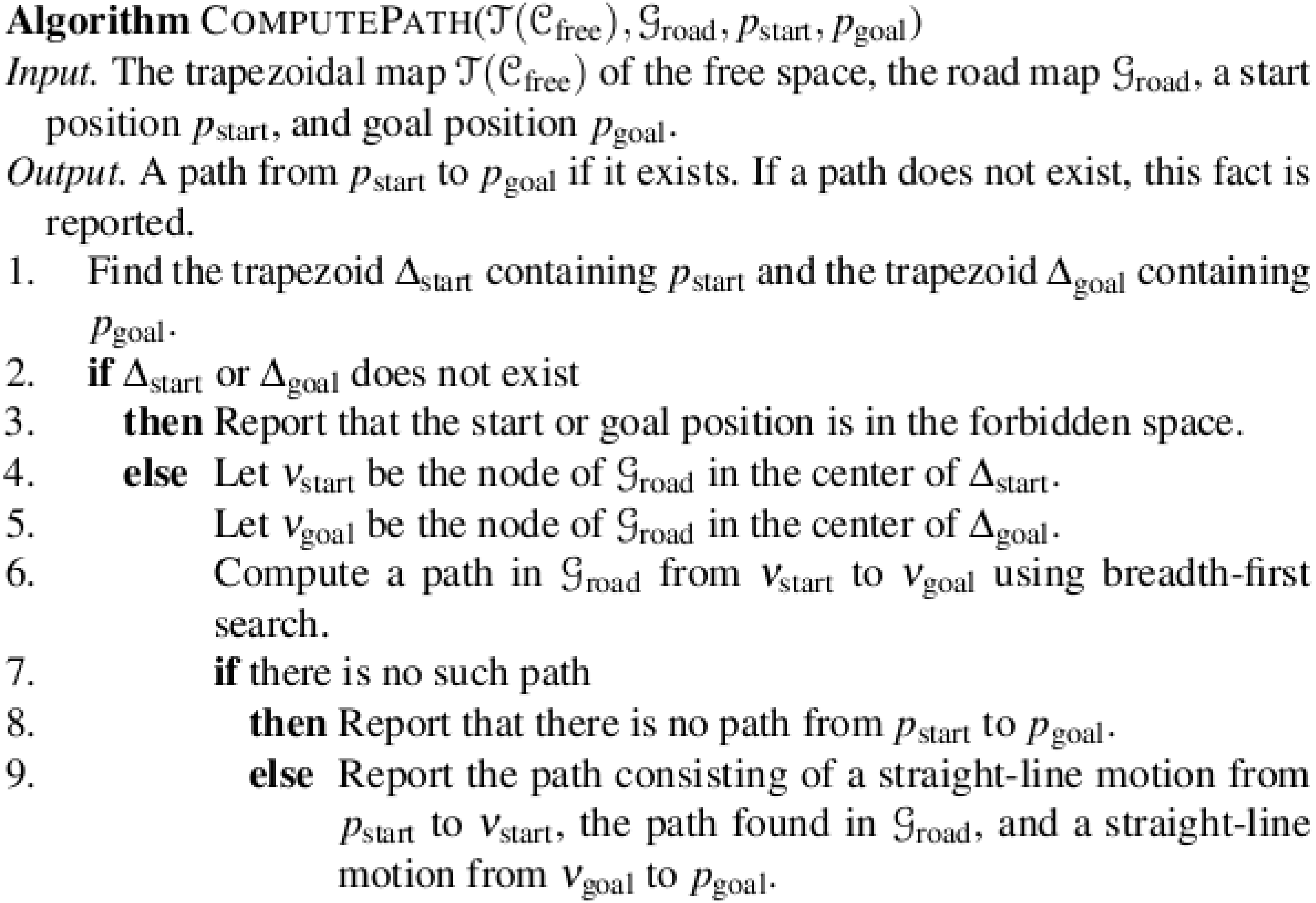


(b $\Delta_{goal} \neq \Delta_{start}$ ، در این حالت مسیر به ۳ قسمت تقسیم میشود:

((v_i نودی از g_{road} و در مرکز Δ_i باشد))

- یک خط مستقیم از p_{start} به v_{start}
- مسیری شامل کمانهای g_{road} و از v_{start} به v_{goal}
- یک خط مستقیم از v_{goal} به p_{goal}


Δ_{start}
Δ_{goal}
v_{goal}
p_{goal}
v_{start}
p_{start}



```
Algorithm ComputePath($\mathcal{T}(\mathcal{C}_{free}), \mathcal{G}_{road}, p_{start}, p_{goal}$)
Input. The trapezoidal map $\mathcal{T}(\mathcal{C}_{free})$ of the free space, the road map $\mathcal{G}_{road}$, a start
   position $p_{start}$, and goal position $p_{goal}$.
Output. A path from $p_{start}$ to $p_{goal}$ if it exists. If a path does not exist, this fact is
   reported.
1.  Find the trapezoid $\Delta_{start}$ containing $p_{start}$ and the trapezoid $\Delta_{goal}$ containing
    $p_{goal}$.
2.  if $\Delta_{start}$ or $\Delta_{goal}$ does not exist
3.    then Report that the start or goal position is in the forbidden space.
4.    else Let $v_{start}$ be the node of $\mathcal{G}_{road}$ in the center of $\Delta_{start}$.
5.         Let $v_{goal}$ be the node of $\mathcal{G}_{road}$ in the center of $\Delta_{goal}$.
6.         Compute a path in $\mathcal{G}_{road}$ from $v_{start}$ to $v_{goal}$ using breadth-first
           search.
7.         if there is no such path
8.           then Report that there is no path from $p_{start}$ to $p_{goal}$.
9.           else Report the path consisting of a straight-line motion from
                  $p_{start}$ to $v_{start}$, the path found in $\mathcal{G}_{road}$, and a straight-line
                  motion from $v_{goal}$ to $p_{goal}$.
```

درستی الگوریتم

- a) مسیر مشخص شده همواره بدون برخورد است؟

تمامی مسیرهای تعیین شده در داخل ذوزنقه ها قرار دارند و ذوزنقه در فضای آزاد ساخته شده اند.

- b) درصورت وجود یک مسیر بدون برخورد، حتما آن مسیر گزارش خواهد شد؟

p_{start} و p_{goal} دو نقطه که به ذوزنقه هایی از فضای آزاد تعلق دارند. پس قسمت اول و سوم مسیر حتما وجود دارد.

حکم: باید ثابت کنیم که مسیر از v_{start} به v_{goal} حتما گزارش میشود:

مسیر v_{start} به v_{goal} دنباله ای از ذوزنقه ها را قطع میکند

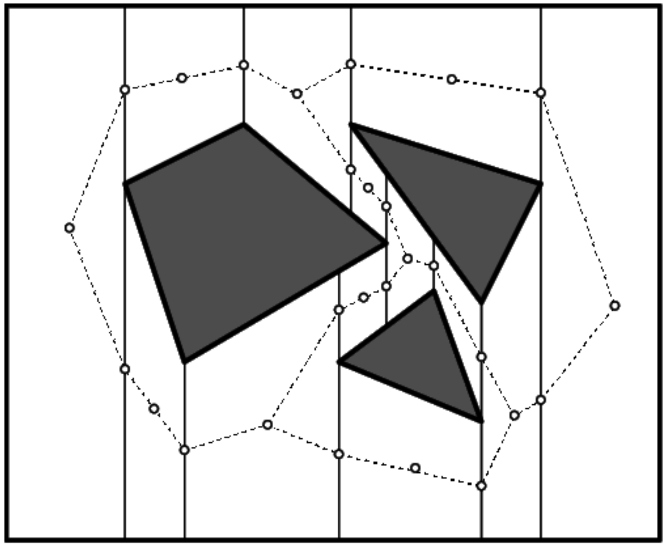
$\{\Delta_1,\Delta_2,\dots,\Delta_k\}$ بطوریکه $\Delta_1=\Delta_{start}$ و $\Delta_k=\Delta_{goal}$

در مسیر مشخص شده Δ_i و Δ_{i+1} دو ذوزنقه ملاقات شده متوالی باشند

بنابراین Δ_i و Δ_{i+1} دو ذوزنقه همسایه و دارای یک امتداد عمودی مشترک میباشند

در g_{road} نودهای v_i و v_{i+1} از طریق نود معادل این امتداد عمودی مشترک به هم متصل شده اند

بنابراین این مسیر گزارش میشود.



زمان اجرا

- زمان یافتن Δ_{start} و Δ_{goal} : O(log n) یا O(n)
- زمان پیمایش BFS : وابسته به اندازه گراف

تعداد نودها

تعداد ذوزنقه ها ≤ 3n+1 (لم 6.1)

+

تعداد امتدادهای عمودی ≤ 2n

تعداد کمان ها ≤ 4(3n+1)

پس زمان پیمایش BFS خطی است.

- زمان گزارش مسیر هم وابسته به تعداد نودها و کمان های آن است و خطی است

- **قضیه 13.1**

فرض کنید R روبات نقطه ای متحرک در میان مجموعه S از موانع با n یال باشد.میتوان S را در زمان O(n logn) طوری پردازش کرد مسیر بدون برخورد بین هر دو نقطه شروع و پایان را در صورت وجود در زمان O(n) محاسبه کرد.

پایان

زهره وکیل