METODA RETRAKCJI – DIAGRAM VORONOI'a

Metoda Retrakcji należy do grupy metod Mapy Dróg. Polega na zdefiniowaniu ciągłego odwzorowania wolnej przestrzeni C_{free} na 1-wymiarową sieć krzywych R położoną w C_{free}.

1. Podstawy formalne

Retrakcja jest klasycznym pojęciem w topologii.

Def. Niech X będzie przestrzenią topologiczną. Niech $Y \subseteq X$.
Odwzorowanie $\rho : X \to Y$, które jest ciągłe, i którego ograniczenie do Y jest odwzorowaniem identycznym ($\rho(y) = y, \ \forall y \in Y$), jest nazywane ***retrakcją X w Y.***

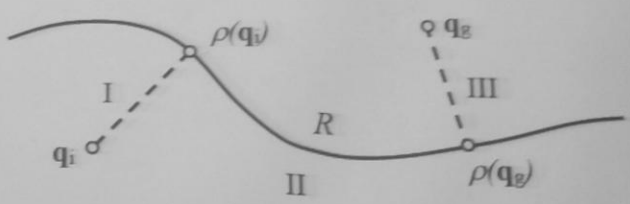
Def. Niech ρ będzie retrakcją przestrzeni topologicznej X w Y.
Mówimy, że ρ ***zachowuje spójność*** X wtedy i tylko wtedy gdy $\forall x \in X$, x i $\rho(x)$ należą do tego samego, połączonego ścieżką składnika X.

Niech ρ będzie zachowującą spójność retrakcją przestrzeni C_{free} na 1-wymiarowy podzbiór $R \subset C_{free}$ (siatkę krzywych). Następujące twierdzenie redukuje planowanie ruchu w C_{free} do planowania w R:

Tw. Niech $\rho : C_{free} \to R$, gdzie $R \subset C_{free}$ jest siatką 1-wymiarowych krzywych, będzie zachowująca spójność retrakcją.
Pomiędzy dwoma konfiguracjami $\mathbf{q}_i$, $\mathbf{q}_g \in C_{free}$ istnieje ścieżka, w. i t. w., gdy istnieje ścieżka pomiędzy ich odwzorowaniami $\rho(\mathbf{q}_i)$, $\rho(\mathbf{q}_g) \in R$.

Dowód: Jeżeli istnieje ścieżka w R pomiędzy $\rho(\mathbf{q}_i)$ a $\rho(\mathbf{q}_g)$, to istnieje ścieżka składająca się z trzech odcinków:

I. $(\mathbf{q}_i, \rho(\mathbf{q}_i))$,
II. $(\rho(\mathbf{q}_i), \rho(\mathbf{q}_g))$,
III. $(\rho(\mathbf{q}_g), \mathbf{q}_g)$.



Odcinki I i III istnieją ponieważ przekształcenie ρ zachowuje spójność C_{free}.

Efektywność planowania ścieżki zależy od:
- wyboru odwzorowania ρ,
- algorytmów konstruowania grafowej reprezentacji sieci krzywych R,
- obliczania odwzorowania $\rho(\mathbf{q}_i)$ i $\rho(\mathbf{q}_g)$,
- generowania ścieżek $(\mathbf{q}_i, \rho(\mathbf{q}_i))$ i $(\rho(\mathbf{q}_g), \mathbf{q}_g)$,

2. Wieloboczna przestrzeń konfiguracyjna 2D

Dla przestrzeni $C = \boldsymbol{R}^2$, ograniczonej wielobokami, efektywną retrakcją jest jej diagram Voronoi'a. Ten diagram ma interesująca własność maksymalizacji odległości pomiędzy robotem a przeszkodami.

Def. Niech β będzie obrysem (otoczeniem) przestrzeni swobodnej C_{free}.
Prześwit (*clearance*) w punkcie $\mathbf{q} \in C_{free}$ jest funkcją:

$$clear(\mathbf{q}) = \min_{p \in \beta} \| \mathbf{q} - p \|,$$

gdzie: $\|.\|$-metryka Euklidesowa.

Def. Niech będzie dany zbiór:

$$near(\mathbf{q}) = \{ p \in \beta / \| \mathbf{q} - p \| = clear(\mathbf{q}) \},$$

punktów obrysu β przestrzeni swobodnej, spełniających funkcję prześwitu (względem pewnej konfiguracji **q**).

diagram Voronoi'a. Ten diagram ma interesująca własność maksymalizacji odległości pomiędzy robotem a przeszkodami.

Def. Niech β będzie obrysem (otoczeniem) przestrzeni swobodnej C_{free}.
Prześwit (*clearance*) w punkcie $\mathbf{q} \in C_{free}$ jest funkcją:

$$clear(\mathbf{q}) = \min_{p \in \beta} \| \mathbf{q} - p \|,$$

gdzie: $\|.\|$-metryka Euklidesowa.

Def. Niech będzie dany zbiór:

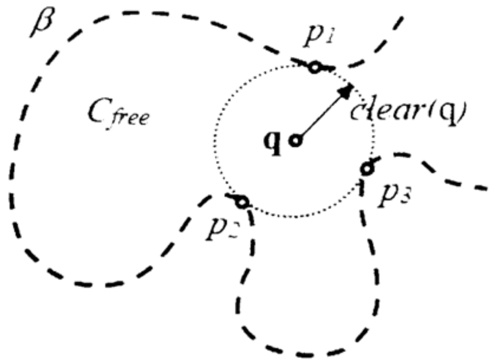
$$near(\mathbf{q}) = \{ p \in \beta / \| \mathbf{q} - p \| = clear(\mathbf{q}) \},$$

punktów obrysu β przestrzeni swobodnej, spełniających funkcję prześwitu (względem pewnej konfiguracji **q**).

Diagramem Voronoi'a przestrzeni C_{free} nazywamy zbiór konfiguracji:

$$Vor(C_{free}) = \{\ \mathbf{q} \in C_{free} \quad card\ (near(\mathbf{q})) > 1\ \}.$$

gdzie: *card E* – oznacza liczność (*cardinality*) zbioru *E*.

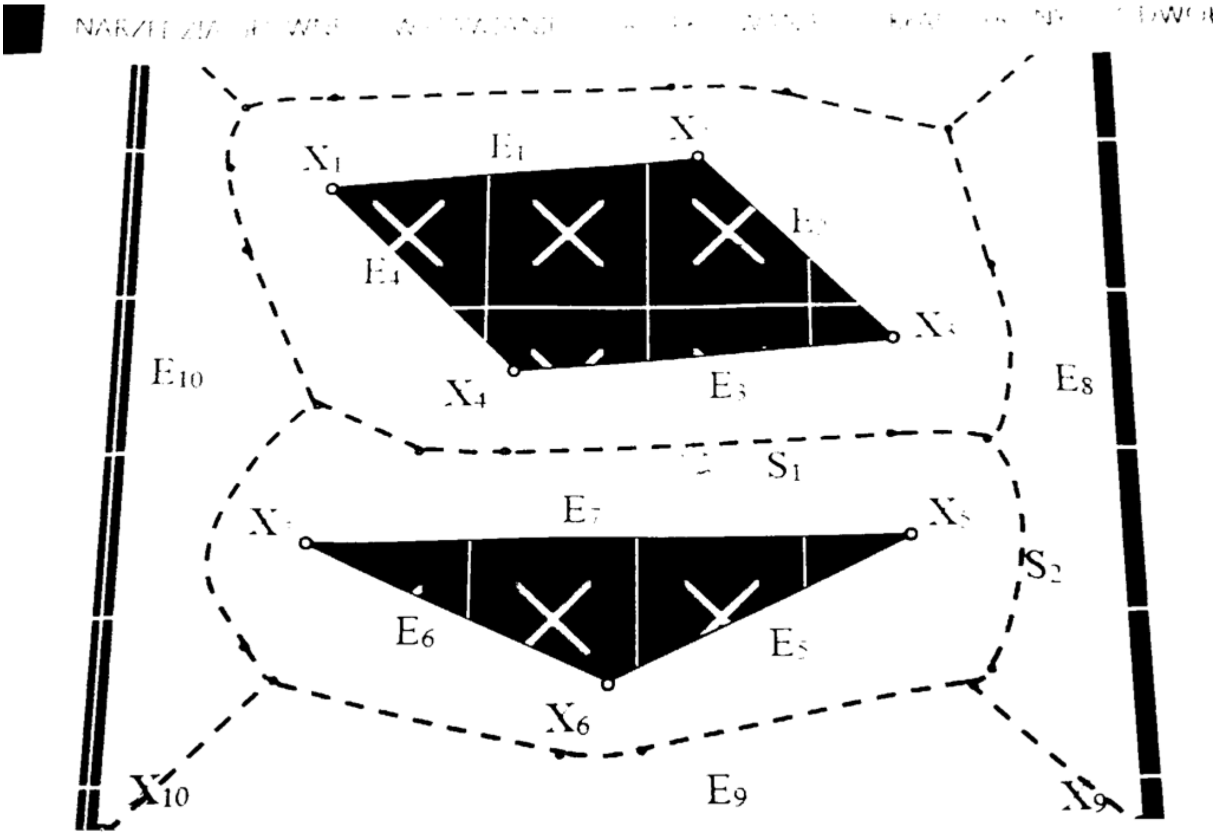


$\{\ p_1, p_2, p_3\ \} = near(\mathbf{q})$

Diagram Voronoi'a jest zbiorem konfiguracji zachowujących minimalny dystans do więcej niż jednego punktu otoczenia β przestrzeni C_{free}.

3. Konstrukcja diagramu

Gdy przestrzeń swobodna C_{free} ***jest ograniczona wielobokami***, to diagram Voronoi'a składa się z odcinków prostych i paraboli.



Każdy ***odcinek prostej*** jest zbiorem konfiguracji położonych najbliżej pary „krawędź – krawędź" (E-E) lub „wierzchołek – wierzchołek" (X-X). Para taka określa równanie prostej. Przykładowo, odcinek prostej S_1, jest zbiorem konfiguracji położonych najbliżej pary krawędzi (E_3, E_7): $S_1 \leftarrow (E_3, E_7)$.

Każdy ***odcinek paraboli*** jest zbiorem konfiguracji położonych najbliżej pary „krawędź – wierzchołek" (E-X). Para taka określa równanie paraboli. Przykładowo, odcinek paraboli S_2 jest zbiorem konfiguracji położonych najbliżej pary krawędź – wierzchołek (E_8, X_5): $S_2 \leftarrow (E_8, X_5)$).

Złożoność obliczeniowa

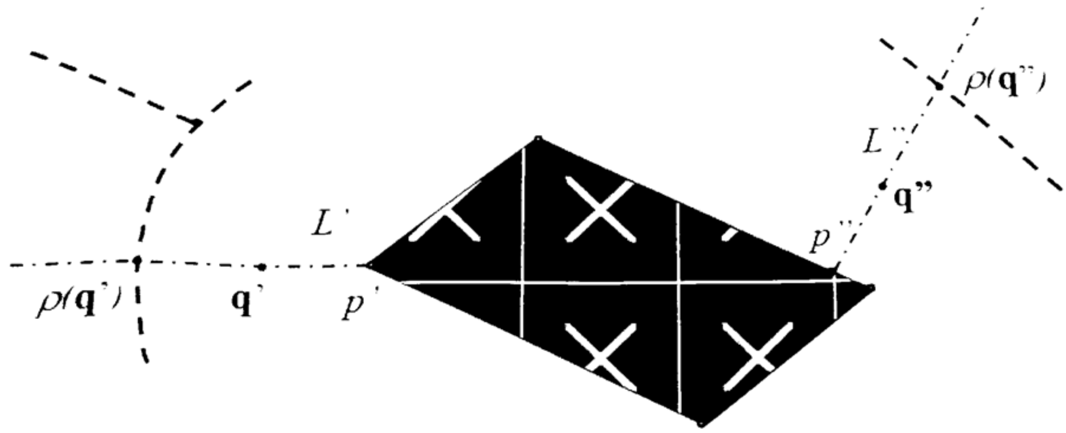
Jeżeli obrys β przestrzeni swobodnej C_{free} ma „*n*" wierzchołków, to naiwny algorytm tworzy diagram Voronoi'a w czasie $O(n^4)$ – rozważając $O(n^2)$ par (E-E), (X-X), (E-X), i obliczając przecięcia odpowiadających im równo-odległych krzywych.

Można łatwo wykazać, że całkowita liczba łuków $Vor(C_{free})$ wynosi $O(n)$, a więc istnieje algorytm, który oblicza $Vor(C_{free})$ w czasie $O(n^2)$.

Istnieją także algorytmy o złożoności $O(n \log^2 n)$ opracowane przez Lee i Drysdale'a oraz przez Leven'a i Sharir'a, a także algorytmy optymalne $O(n \log n)$opracowane przez Kirkpatrick'a oraz przez Yap'a a także przez Fortune'a.

4. Konstrukcja odwzorowania $\rho(\mathbf{q})$

Rozważmy konfigurację swobodną **q**. spoza diagramu Voronoi'a ($\mathbf{q} \notin Vor(C_{free})$). oraz najbliższy dla niej punkt obrysu $p \in \beta$. czyli taki że: $\| \mathbf{q} - p \| = clear(\mathbf{q})$.
Punkt p jest albo pojedynczym wierzchołkiem obrysu ($p = X_i \in \beta$) albo elementem otwartej krawędzi obrysu ($p \in E_j \in \beta$).
Rozważmy pół-prostą L wychodzącą z p i przechodząca przez **q**.
Odcinek łączący punkt p z najbliższym przecięciem prostej L z diagramem $Vor(C_{free})$, przebiega po „max. spadku" funkcji prześwitu $clear(\mathbf{q})$, tj. dla każdego **q'** położonego na tym odcinku, wektor $\overline{\nabla}\, clear(\mathbf{q}')$ wskazuje wzdłuż L.



Pierwsze przecięcie L z diagramem $Vor(C_{free})$. oznaczone przez $\rho(\mathbf{q})$. jest „***obrazem konfiguracji* q *na diagramie*** $\boldsymbol{Vor(C_{free})}$".

Poza $\rho(\mathbf{q})$. funkcja prześwitu $clear(\mathbf{q})$ maleje lub zmienia się kierunek jej gradientu.
Funkcję ρ można rozszerzyć na całą C_{free}. kładąc: $\rho(\mathbf{q}) = \mathbf{q}$ dla $\forall \mathbf{q} \in Vor(C_{free})$.

Tw.

Aplikacja odwzorowania ρ: $C_{free} \rightarrow Vor(C_{free})$. jest ciągła.

A więc ρ **jest retrakcją** C_{free} na diagram $Vor(C_{free})$. Ze względu na sposób **konstrukcji zachowuje ona spójność** C_{free}.

5. Algorytm planowania ścieżki

Metoda planowania ścieżki oparta na retrakcji ρ wykorzystuje diagram Voronoi'a jako mapę dróg R. Algorytm jest następujący:

1. Oblicz diagram $Vor(C_{free})$.
2. Oblicz punkty $\rho(\mathbf{q}_i)$, $\rho(\mathbf{q}_g) \in Vor(C_{free})$ i określ łuki $Vor(C_{free})$ zawierające te punkty.
3. Szukaj w $Vor(C_{free})$ sekwencji łuków $A_1, \ldots A_p$, takiej że $\rho(\mathbf{q}_i) \in A_1$, $\rho(\mathbf{q}_g) \in A_p$, oraz $\forall i \in [1, p-1]$ łuki A_i i A_{i+1} mają wspólny wierzchołek.
4. Gdy poszukiwanie zakończy się sukcesem, podaj $\rho(\mathbf{q}_i)$ i $\rho(\mathbf{q}_g)$ oraz sekwencję łączących je łuków, w przeciwnym przypadku podaj brak drogi.

Krok 1 wymaga czasu O(n log n).

W kroku 2 trzeba najpierw obliczyć dystans konfiguracji $\mathbf{q}_i$ (odpowiednio $\mathbf{q}_g$) do każdej krawędzi E_s i każdego wierzchołka X_r obrysu β, a potem wyznaczyć punkt $p_i \in \beta$ (odpowiednio $p_g \in \beta$), dla którego ten dystans jest najmniejszy, następnie określić wszystkie przecięcia promienia wychodzącego z p_i (odpowiednio p_g) i przechodzącego przez $\mathbf{q}_i$ (odpowiednio $\mathbf{q}_g$) z diagramem $Vor(C_{free})$ oraz wybrać przecięcie najbliższe p_i (odpowiednio p_g). Ponieważ w $Vor(C_{free})$ istnieje O(n) łuków te obliczenia wymagają O(n) czasu.

Poszukiwanie z kroku **3** również wymaga O(n) czasu.

Tak więc całkowita złożoność czasowa metody wynosi O(n log n). Najbardziej kosztowny jest krok **1**, lecz zależy on jedynie od przestrzeni C_{free} a nie od położenia konfiguracji $\mathbf{q}_i$ i $\mathbf{q}_g$. Więc planowanie innych ścieżek w tej samej przestrzeni pomija krok **1** i zajmuje O(n) czasu. Dodatkowo istnieje liniowa metoda uaktualniania diagramu gdy kilka przeszkód ulegnie zmianie.