

Globálna navigácia mobilných robotov na báze geometrickej mapy

František Duchoň, Ladislav Jurišica

Abstrakt

Článok analyzuje súčasný stav problematiky navigácie mobilných robotov na základe známej geometrickej reprezentácie prostredia. V článku sú predstavené základné metódy plánovania cesty robota v známom prostredí a zároveň sú analyzované ich výhody a nedostatky.

Kľúčové slová: geometrická mapa, graf viditeľnosti, Voronoiov diagram, tangent graf, polygónové mapy, metóda napnutého vlákna, metóda potenciálových polí

Úvod

Napriek existujúcim vhodnejším reprezentáciám prostredia pre mobilné roboty z hľadiska ukladacieho priestoru a štruktúry zápisu, sa v mobilnej robotike uplatňujú aj geometrické reprezentácie prostredia, obvykle tvorené polygónmi. Napriek náročnosti uchovávať takúto reprezentáciu prostredia sa v mobilnej robotike uplatňujú z rôznych hľadísk optimálne metódy hľadania najkratších ciest v takýchto mapách [6][7]. Medzi takéto metódy možno zaradiť graf viditeľnosti, všeobecný Voronoiov diagram, tangent graf, metódu napnutého vlákna alebo metódu potenciálových polí.

1. Graf viditeľnosti [1] [2]

Spojením všetkých párov vrcholov prekážok v prostredí (resp. v mape prostredia), ktoré sa navzájom "vidia" vznikne graf, na ktorom sa dá plánovať globálna dráha. Graf obsahuje aj počiatočnú a cieľovú pozíciu robota. Čiary spájajúce jednotlivé vrcholy prekážok obvykle reprezentujú najkratšiu cestu medzi týmito vrcholmi. Úlohou robota je potom už len nájsť najkratšiu cestu medzi počiatočnou a cieľovou pozíciou v takto definovanom grafe. Na to možno využiť algoritmy A*, Dijkstra (viď AT&P Journal, Globálna navigácia mobilných robotov na báze topologickej mapy) alebo jednoduché prehľadávanie na základe euklidovskej vzdialenosti.

Hrany grafu viditeľnosti e_{ij} sú rovné čiary, ktoré spájajú dva "viditeľné" vrcholy v_i a v_j :

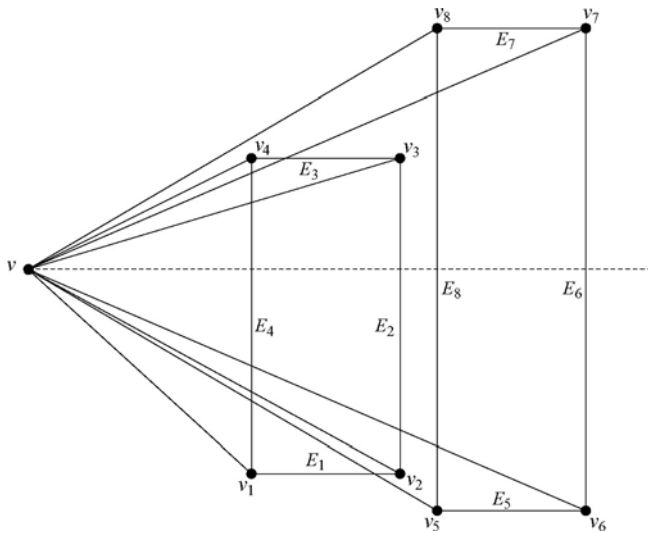
$$e_{ij} \neq 0 \Leftrightarrow s \cdot v_i + (1-s) \cdot v_j \in S_{free} \quad \forall s \in \{0,1\}, \quad (1)$$

kde S_{free} predstavuje voľný priestor v prostredí. Nech množina $V = \{v_1, \dots, v_n\}$ je množina vrcholov prekážok v konfiguračnom priestore zahŕňajúca aj štartovný a cieľový bod robota. Zostrojenie grafu viditeľnosti spočíva v určení všetkých ostatných viditeľných vrcholov prekážok pre vrchol $v \in V$. Logicky sa ponúka možnosť otestovať všetky dvojice vrcholov, či nepretínajú prekážky. Je teda potrebné otestovať $O(n)$ prienikov, pretože existuje $O(n)$ hrán od prekážok. Takže je tu $O(n)$ možných hrán z vrcholu v ,

preto je potrebné otestovať $O(n^2)$ vrcholov viditeľných pre vrchol v . To je nutné vykonať pre všetky $v \in V$, čiže výpočtová náročnosť sa zvýši až na $O(n^3)$. Existuje však aj efektívnejšia metóda ako určiť viditeľné vrcholy pre vrchol v . Táto metóda sa nazýva plane sweep algoritmus. Vstupom do tohto algoritmu je množina vrcholov $\{v_i\}$ a vrchol v . Výstupom je podmnožina z $\{v_i\}$ viditeľná z vrcholu v . Algoritmus spočíva v týchto krokoch:

1. Pre každý vrchol v_i sa vypočíta uhol α_i medzi horizontálnou osou (obvykle os x) a čiarou $\overline{vv_i}$.
2. Vytvorí sa zoznam vrcholov \mathcal{E} , obsahujúci α_i vo vzostupnom poradí.
3. Vytvorí sa aktívny zoznam S obsahujúci hrany, ktoré pretínajú polpriamku v kladnom smere osi x vychádzajúcej z v .
4. Všetky α_i je potrebné otestovať. Ak je v_i viditeľné z v , potom čiaru $\overline{vv_i}$ je potrebné pridať do grafu viditeľnosti. Ak je v_i začiatkom hrany E , ktorá nie je v S , potom je potrebné vložiť túto hranu do množiny S . Ak je v_i koncom hrany v S , potom je potrebné túto hranu vymazať zo zoznamu S .

Príklad takéhoto algoritmu možno ukázať na obrázku (Obr. 1).



Obr. 1 Príklad prostredia pre vytvorenie grafu viditeľnosti [1]

Fig. 1 Example of environment used for visibility graph creation

Najskôr sú vypočítané uhly k jednotlivým vrcholom a inicializované množiny $\mathcal{E} = \{\alpha_3, \alpha_7, \alpha_4, \alpha_8, \alpha_1, \alpha_5, \alpha_2, \alpha_6\}$ a $S = \{E_4, E_2, E_8, E_6\}$. Ďalší postup znázorňuje tabuľka (Tab. 1). Treba si uvedomiť definovanie začiatku a konca hrany. Začiatok hrany je definovaný menším uhlom od horizontálnej osi vytvorenej z vrcholu v , naopak koniec hrany má väčší uhol od tejto horizontálnej osi oproti začiatku hrany. Viditeľnosť vrcholu v_i z v sa testuje len vzhľadom na množinu S . Takýmto spôsobom sa výpočtová náročnosť algoritmu znižuje na $O(n^2 \cdot \log n)$. Ak sú náhodou tri vrcholy kolineárne (ležia na jednej priamke), jeden vrchol sa mierne posunie a pokračuje sa vyššie opísaným algoritmom.

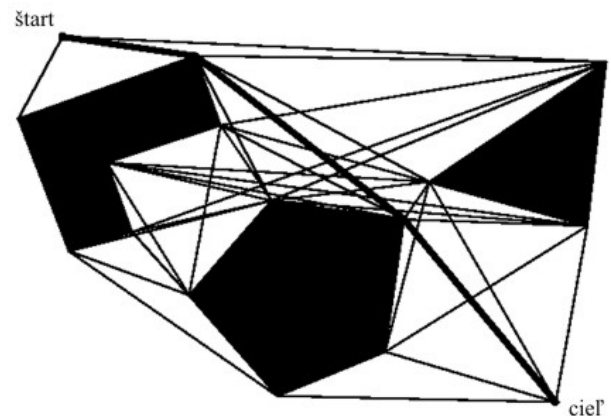
Vrchol	Množina S	Akcia
Inicializácia	$\{E_4, E_2, E_8, E_6\}$	Zoradenie množiny \mathcal{E}
α_3	$\{E_4, E_3, E_8, E_6\}$	Vymaž E_2 , Vlož E_3
α_7	$\{E_4, E_3, E_8, E_7\}$	Vymaž E_6 , Vlož E_7
α_4	$\{E_8, E_7\}$	Vymaž E_3 , Vymaž E_4 , Pridaj $\overline{vv_4}$ do grafu
α_8	$\{\}$	Vymaž E_7 , Vymaž E_8 , Pridaj $\overline{vv_8}$ do grafu
α_1	$\{E_1, E_4\}$	Vlož E_4 , Vlož E_1 , Pridaj $\overline{vv_1}$ do grafu

α_5	$\{E_4, E_1, E_8, E_5\}$	Vlož E_8 , Vlož E_5
α_2	$\{E_4, E_2, E_8, E_5\}$	Vymaž E_1 , Vlož E_2
α_6	$\{E_4, E_2, E_8, E_6\}$	Vymaž E_5 , Vlož E_6

Tab. 1 Postup výpočtu pomocou plane sweep algoritmu [1]

Fig. 1 Progress of computation with plane sweep algorithm

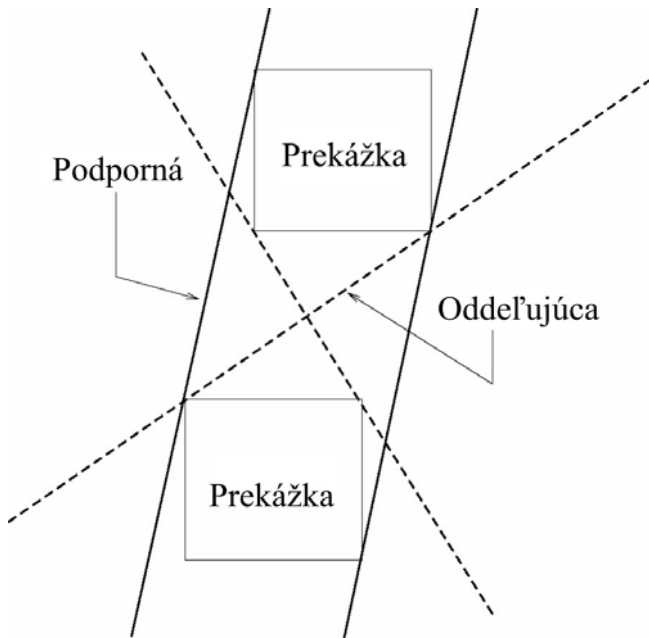
Napriek jednoduchej implementácii, má táto metóda dva hlavné nedostatky. Veľkosť reprezentácie prostredia a teda aj počet vrcholov a ciest v grafe sa zvyšuje s množstvom prekážok v prostredí. Metóda je teda rýchla a efektívna v prostredí s menším počtom prekážok, avšak pri väčšom počte prekážok je v porovnaní s inými metódami pomalá a neefektívna. Druhou nevýhodou je, že nájdená dráha v grafe viditeľnosti predpisuje robotu, čo najviac sa priblížiť k prekážkam. Napriek tomu, že nájdená dráha je optimálna, podmienky pre bezpečnosť robota ako aj jeho prostredia optimálne nastavené nie sú. Obvyklým riešením je definovanie rozšírenia prekážok o vzdialenosť väčšiu ako je polomer robota, čo má však často za následok obetovanie optimálnosti nájdenej dráhy.



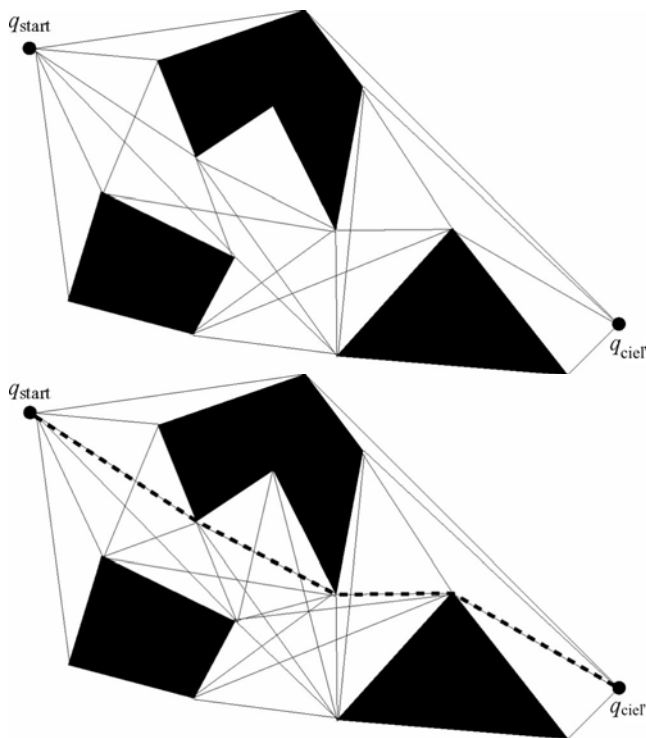
Obr. 2 Graf viditeľnosti [2]

Fig. 2 Visibility graph

Z prvej nevýhody vyplýva, že graf viditeľnosti (Obr. 2) obsahuje mnoho nadbytočných vrcholov. Použitím podporných a oddeľujúcich čiar možno tento počet zredukovať (Obr. 3). Podporná čiara je dotyčnica k dvom prekážkam, pričom obe prekážky sa nachádzajú len v jednej polrovine vymedzenej čiarou. Oddeľujúca čiara je tak isto dotyčnica k dvom prekážkam, avšak prekážky sa nachádzajú v opačných polrovinách vymedzených čiarou. Redukovaný graf (Obr. 4) viditeľnosti je zostrojený z práve takýchto podporných a oddeľujúcich čiar. Uzly grafu viditeľnosti, ktoré neležia na takýchto čiarach sú odstránené. Redukovaný graf viditeľnosti možno zostrojiť aj pre konkávne typy prekážok. Uplatňuje sa tu definícia lokálnej konvexnosti.



Obr. 3 Príklady podporných a oddeľujúcich čiar [1]
Fig. 3 Examples of supporting and segregating lines



Obr. 4 Graf viditeľnosti a redukovaný graf viditeľnosti [1]
Fig. 4 Visibility graph and reduced visibility graph

2. Voronoiov diagram [1] [2] [3] [4]

Voronoiov diagram je cestná mapa, ktorá maximalizuje vzdialenosť medzi robotom a prekážkami v mape. Plánovanie dráhy mobilného robota pomocou Voronoiovoho diagramu vychádza z geometrickej mapy prostredia. Pre každý bod patriaci do množiny voľných bodov (teda nie prekážky) je vypočítaná vzdialenosť k najbližšej prekážke. Voronoiov diagram je definovaný na množinách bodov nazývaných základne. Voronoiov región je potom množina bodov najbližších k danej základni. Voronoiov diagram je množina bodov ekvidištancných k dvom základniam. Pri plánovaní

cesty Voronoioovým diagramom môžeme prekážky pokladať za základne. Prekážky nie sú však opísané množinou bodov, preto definíciu Voronoiovoho regiónu je potrebné rozšíriť na definíciu všeobecného Voronoiovoho diagramu. Nech V_i je množina bodov najbližších k prekážke Q_i :

$$V_i = \{q \in Q_{free} \mid d_i(q) \leq d_h(q) \quad \forall h \neq i\}, \quad (2)$$

kde $d_i(q)$ je vzdialenosť k prekážke Q_i z polohy q , teda:

$$d_i(q) = \min_{c \in O_i} d(q, c) \quad (3)$$

Základným stavebným kameňom všeobecného Voronoiovoho diagramu (VVD, v angl. GVD - Generalised Voronoi Diagram) je množina ekvidištancných bodov ku dvom množinám Q_i a Q_j , pre ktoré platí:

$$d_i(q) - d_j(q) = 0 \quad (4)$$

Dvojekvidištancná plocha je potom definovaná ako (Obr. 5 a):

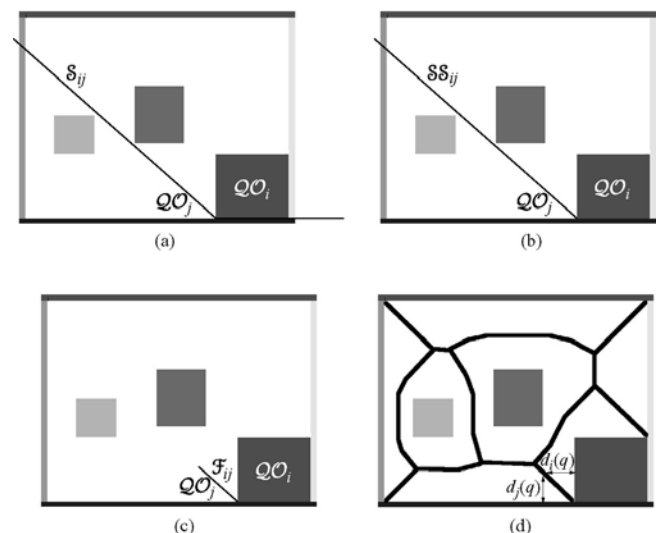
$$S_{ij} = \{q \in Q_{ij} \mid (d_i(q) - d_j(q) = 0)\} \quad (5)$$

Takáto plocha rozdeľuje prekážky v priestore a môžeme ju obmedziť na množinu bodov ekvidištancných k prekážkam Q_i a Q_j , ktoré sú zároveň najbližšie k obom týmto prekážkam. Takáto obmedzená štruktúra sa nazýva dvojekvidištancný zlom (Obr. 5 c):

$$V_{ij} = \{q \in Q_{ij} \mid d_i(q) \leq d_j(q) \quad \forall h\}^2 \quad (6)$$

Spojením dvoch takýchto dvojekvidištancných zlomov vznikne VVD (Obr. 5d):

$$VVD = \bigcup_i \bigcup_j V_{ij} \quad (7)$$

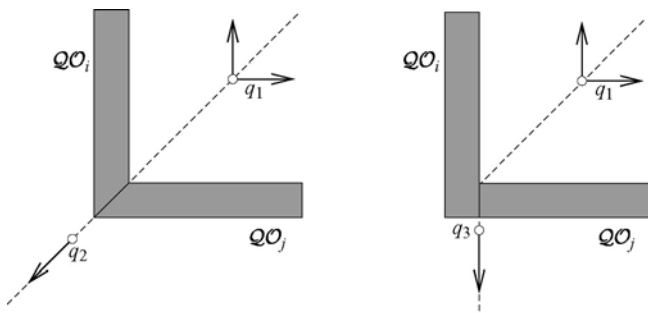


Obr. 5 a.) Množina S_{ij} ekvidištancných bodov k dvom prekážkam b.) Množina SS_{ij} ekvidištancných bodov s jednoznačným gradientom c.) Množina V_{ij} ekvidištancných bodov k dvom prekážkam d.) Voronoiov diagram [1]

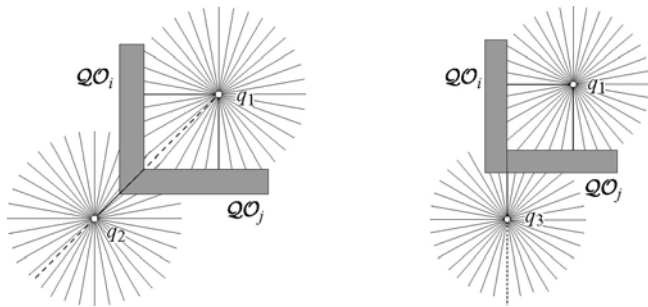
Fig. 5 a.) Set S_{ij} of equidistant points to both obstacles
b.) Set SS_{ij} of equidistant points with univocal gradient c.) Set V_{ij} of equidistant points to both obstacles d.) Voronoi diagram

V dvojrozmernom priestore potom V_{ij} nazývame hrany VVD, ktoré sa končia alebo na dotykových bodoch (množina ekvidistančných bodov k trom alebo viacerým prekážkam), alebo na hraničných bodoch (množina bodov, ktorých vzdialenosť k prekážke je nulová).

Používanie funkcie vzdialenosti d_i (jednoznačné určenie vzdialenosti) predpokladá výskyt len konvexných prekážok v priestore, čo je v mnohých prípadoch nereálne. Je teda potrebné konkávne prekážky rozdeliť na viacero konvexných. Na Obr. 6 možno vidieť rozdelenie konkávnej prekážky na dve konvexné dvoma rôznymi spôsobmi. Oba sú správne, avšak každá z dekompozícií vytvára odlišné množiny ekvidistančných bodov S_{ij} k prekážkam O_i a O_j .



Obr. 6 Konkávna prekážka a jej možnosti rozdelenia [1]
Fig. 6 Concave obstacle and their possibilities of partition



Obr. 7 Umiestnenie robota v troch bodoch. V bode q_2 a q_3 neexistuje lokálne minimum. [1]

Fig. 7 Placement of robot in three positions. In position q_2 and q_3 doesn't exist local minimum.

Pre gradienty platí:

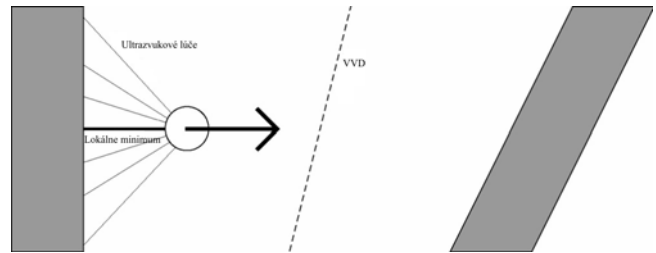
$$\begin{aligned} \nabla d_i(q_1) &\neq \nabla d_j(q_1) \\ \nabla d_i(q_2) &= \nabla d_j(q_2) \\ \nabla d_i(q_3) &= \nabla d_j(q_3) \end{aligned} \quad (8)$$

Je teda potrebné eliminovať časť dvojekvidistančnej plochy s nejednoznačným vektorom gradientu (Obr. 7). To je možné urobiť zadefinovaním dvojekvidistančnej surjektívnej plochy (Obr. 5b):

$$SS_{ij} \{q \in Q_{ij} \mid \nabla d_i(q) \neq \nabla d_j(q)\} \quad (9)$$

Voronoi diagram možno zostrojiť na základe troch typov reprezentácií prostredia: na základe informácií zo snímačov, na základe polygónovej mapy prostredia a na základe mriežkovej (metrickej) reprezentácie prostredia.

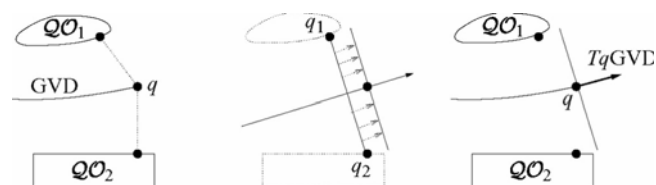
VVD môže byť skonštruovaný priamo z informácií zo snímačov. Používajúc senzory vzdialenosti je možné robot naviesť na dotykový alebo hraničný bod. Robot sa dostane na štruktúru VVD tak, že sa pohybuje smerom od najbližšej prekážky dovtedy, kým nie je v rovnakej vzdialenosti od dvoch prekážok. Túto charakteristiku možno využiť aj pri reaktívnej navigácii (Obr. 8).



Obr. 8 Spôsob nájdenia štruktúry VVD pri reaktívnej navigácii [1]

Fig. 8 The way of Voronoi diagram construction at reactive navigation

Ak robot narazí na nový, ešte nepreskúmaný, dotykový bod, zapamätá si, z akého smeru na tento dotykový bod narazil. Potom identifikuje všetky nové hrany VVD vychádzajúce z tohto dotykového bodu. Z dotykového bodu potom robot skúma hranu VVD kým nenašetrá ďalší dotykový alebo hraničný bod. Ak robot narazí na ďalší dotykový bod, celý proces sa rekurzívne opakuje. Ak robot narazí na už preskúmaný dotykový bod, robot dokončil cyklus vo VVD grafe a pokračuje s ešte nepreskúmaným dotykovým bodom. Ak robot narazí na hraničný bod, jednoducho sa otočí a vráti sa k dotykovému bodu s nepreskúmanými hranami VVD. V prípade, že všetky hrany sú už preskúmané, tvorba VVD je ukončená. Ak sa robot dostane na VVD, je potrebné, aby "stopoval" VVD, teda aby platilo $d_i(q) = d_j(q)$. Dotyčnica je potom definovaná ako $\nabla d_i(q) - \nabla d_j(q) = 0$. To sa zhoduje s prechodom čiarou spájajúcou dotykové body najbližších prekážok a generovaním vektora pohybu kolmého na túto čiaru (Obr. 9) Dotykové body VVD sú potom detegované na základe náhlych zmien informácií o vzdialenosti k jednej z najbližších prekážok. Takýmto sledovaním VVD možno spresniť aj odometrickú informáciu o polohe robota.

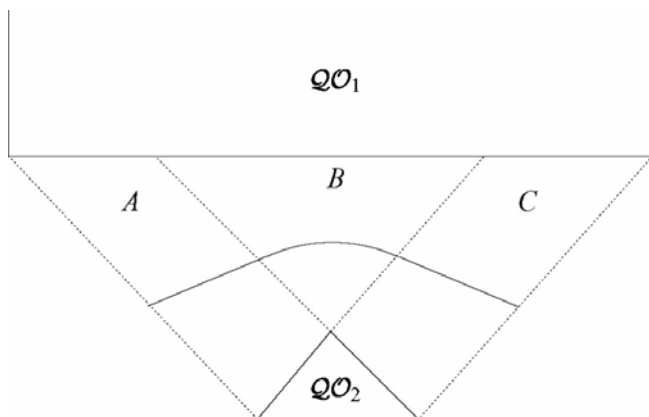


Obr. 9 Smer pohybu po VVD. Smer je vždy kolmý na čiaru, ktorá spája dva najbližšie body dvoch najbližších prekážok [1]

Fig. 9 Direction of robot on GVD. Direction is always orthogonal on line, which connects the two nearest points of the nearest obstacles.

Vytvorenie VVD na polygónovej mape je jednoduché (Obr. 10). Táto mapa definuje prekážky pomocou vrcholov a hrán. Množina ekvidistančných bodov k dvom hranám je čiara. Množina ekvidistančných bodov k dvom vrcholom je čiara. Množina ekvidistančných bodov medzi vrcholom a hranou je

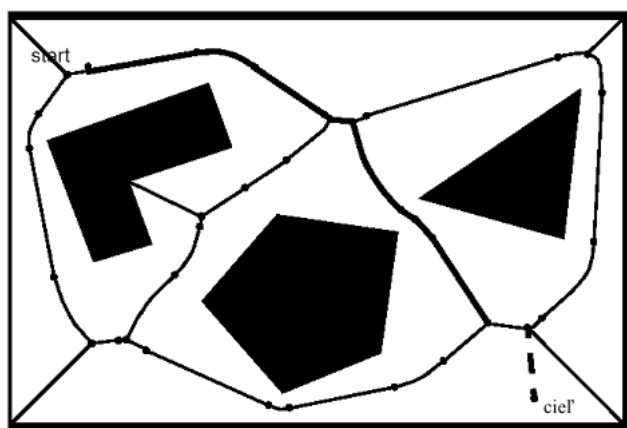
parabola. Priestor polygórovej mapy je teda potrebné rozdeliť na časti s odpovedajúcimi dvojicami.



Obr. 10 VVD medzi dvoma polygórovými prekážkami [1]
Fig. 10 GVD between two polygonal obstacles.

Na určenie VVD v mriežkovej reprezentácii sa používa Brushfire algoritmus (viď. AT&P Journal, Globálna navigácia mobilných robotov na báze metrickej mapy). Pri tejto metóde má každá bunka definovanú vzdialenosť akú "vlna už prešla". Práve v bunkách, kde sa vlna "láme" sú rovnaké vzdialenosti k dvom najbližším prekážkam. Tieto miesta sú potom súčasťou VVD. Lom vlny môžeme definovať ako miesta s aktuálne šírenou hodnotou nedotýkajúce sa žiadnej bunky s nulovou hodnotou. Iným spôsobom detegovania buniek VVD je pamätanie si, od ktorej prekážky sa vlna v danej bunke šíri. Ak sa algoritmus pokúsi zapísať pre jednu bunku dve rôzne hodnoty, potom táto bunka je súčasťou VVD.

Algoritmy, ktoré prehľadávajú cestu vo Voronoiovom diagrame, sú podobné ako pri metóde graf viditeľnosti. Najdéná dráha vo Voronoiovom diagrame nie je optimálna z hľadiska prejdenej dĺžky, je však najbezpečnejšia z hľadiska kolízií. Obmedzením pre využitie Voronoivho diagramu je aj druh použitých snímačov na lokalizáciu robota v prostredí. Keďže táto metóda predpisuje najväčšiu vzdialenosť robota od prekážok, akékoľvek senzory s krátkym dosahom použité pre lokalizáciu môžu v tomto procese zlyhávať.



Obr. 11 Voronoiov diagram [2]

Fig. 11 Voronoi diagram

3. Tangent graf [5]

Táto metóda bola vyvinutá kvôli nedostatkom grafu viditeľnosti:

- Graf viditeľnosti potrebuje $O(N^2)$ pamäte, kde N predstavuje počet vrcholov.
- Graf viditeľnosti je obmedzený na prekážky polygonálneho tvaru, zatiaľ čo v skutočnosti prekážky majú rôzne zakrivenia.
- Ak sa zmení nejaký parameter pri výpočte grafu viditeľnosti (napríklad počet prekážok alebo zmena rozmeru robota) je potrebné celý graf určiť nanovo.

Na riešenie týchto nedostatkov bol zavedený tangent graf. Pri určovaní tangent grafu bol zavedený pojem lokálne najkratšia cesta, ktorý bol zavedený kvôli hľadaniu cesty medzi zakrivenými hranami prekážok. Táto cesta predstavuje najkratšiu cestu na zvolenom malom regióne. Tangent graf sa podobá grafu viditeľnosti, avšak vrcholy tohto grafu zodpovedajú dotykovým bodom s prekážkami v zmysle lokálnych najkratších ciest a jeho hrany reprezentujú po pohyb robota voľné dotyčnice alebo konvexné dotykové segmenty medzi vrcholmi. Pomocou tangent grafu je možné hľadať cestu nielen medzi prekážkami reprezentovanými polygónmami, ale aj medzi prekážkami, ktoré majú zakrivené hrany. Výpočtová náročnosť tangent grafu je $O(K^2)$, kde K predstavuje počet konvexných segmentov na hraniciach prekážok.

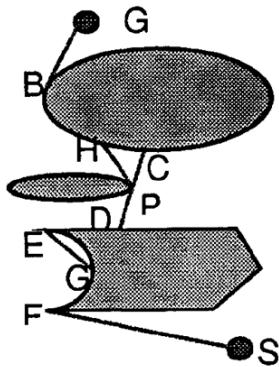
Cesta P je lokálne najkratšia cesta ak na malom ε okolí nie je možné nájsť kratšiu bezkolíznu cestu ako P . Je teda možné konštatovať, že ak neexistuje žiadna lokálne najkratšia cesta v mape, potom neexistuje bezkolízna cesta pre pohyb robota a naopak.

Ak sa čiara dotýka hranice prekážky v bode p , ale nepretína žiaden malý región z prekážky v blízkosti bodu p , potom táto čiara je dotyčnica k prekážke a bod p je nazývaný dotykový bod. Navyše ak je táto čiara dotyčnicou naraz k dvom prekážkam, nazýva sa všeobecná dotyčnica. Ak sa dotykové body dotyčnice k obom alebo viacerým prekážkam nachádzajú na jednej strane dotyčnice ide o vonkajšiu všeobecnú dotyčnicu, inak o vnútornú všeobecnú dotyčnicu.

Ak spojnica dvoch bodov z hranice prekážky definovaných na malom pozitívnom γ okolí bodu p z hranice prekážky je vo vnútri prekážky, potom bod p je konvexný, inak je konkávny. Ak sú všetky body zo segmentu hranice prekážky C konvexné, potom aj segment C je konvexný. Ak sú všetky body tohto segmentu konkávne, potom aj segment C je konkávny.

Pre robot je potom lokálne najkratšia cesta zostavená z konvexných hraničných segmentov a zo všeobecných dotyčnic konvexných hraničných segmentov. Prvú časť tohto tvrdenia možno ukázať na oblúku \tilde{EF} . Predpokladajme, že oblúk \tilde{EF} (Obr. 12) je súčasťou lokálne najkratšej cesty. Na malom ε okolí oblúka \tilde{EF} je možné nájsť bod G , kde priamy segment EG je bezkolízny. Platí, že $|EG| + |\tilde{GF}| < |\tilde{EF}|$. Druhú časť tvrdenia možno ukázať na

segmentoch CD a \tilde{BC} . Predpokladajme, že segment CD je súčasťou lokálne najkratšej cesty prechádzajúcej do segmentu \tilde{BC} . Segment CD nie je dotyčnicou v bode C , ale môžeme povedať, že je dotyčnicou k iným hraničným segmentom. Najbližším dotykovým bodom k bodu C zo segmentu CD je bod P . Potom môžeme podobne na ε okolí bodu C nájsť bod H na hranici \tilde{BC} , pre ktorý platí, že segment HP je bezkolízny a $|HP| < |\tilde{HC}| + |CP|$.

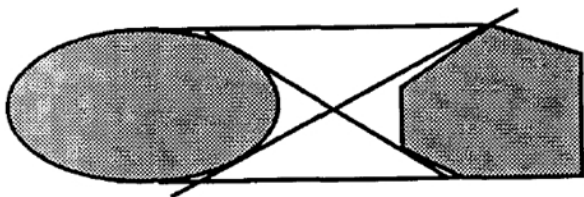


Obr. 12 Vzťahy medzi lokálne najkratšími cestami, hranicami a segmentmi spájajúcimi body na hraničiach [5]

Fig. 12 Relations between locally shortest paths, borders and segments linking point on borders

Tangent graf je potom definovaný ako množina (V, L) , kde vrchol $v \in V$ reprezentuje dotykový bod na hranici prekážky a hrana $l \in L$ odpovedá bezkolíznej všeobecnej dotyčnici alebo konvexnému hraničnému segmentu medzi dvoma bodmi dotyku. Počet dotyčníc potom určuje veľkosť tangent grafu.

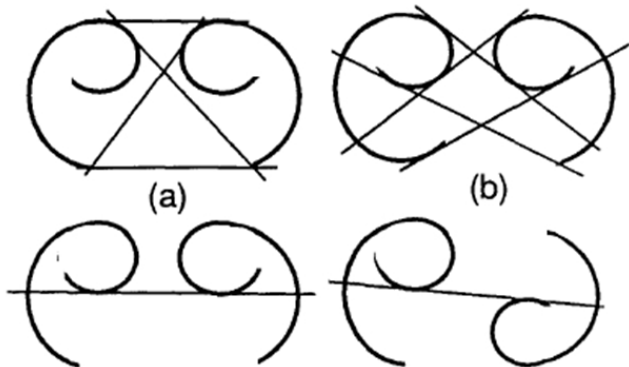
Pre dva konvexné objekty existujú dve vnútorné a dve vonkajšie všeobecné dotyčnice. Dokázať to možno tak, že čiara rozdeľuje rovinu na dve pol roviny (Obr. 13). Ak je táto čiara dotyčnicou k obojm regiónom, regióny sa musia nachádzať v opačných pol rovinách. Z toho vyplýva, že pre dva konvexné objekty môžu existovať len dve vnútorné a dve vonkajšie všeobecné dotyčnice.



Obr. 13 Dotyčnice medzi dvoma konvexnými objektmi [5]

Fig. 13 Tangents between two convex obstacles

Existuje najviac 9 bezkolíznych dotyčníc medzi dvoma špirálami. Existujú tri typy bezkolíznych dotyčníc medzi dvoma špirálami. Prvý typ nikdy nepretne špirály (Obr. 14 a). Druhý typ pretne hociktorú zo špirál len raz (Obr. 14 b). Tretí typ pretne obe špirály (Obr. 14 c, d).

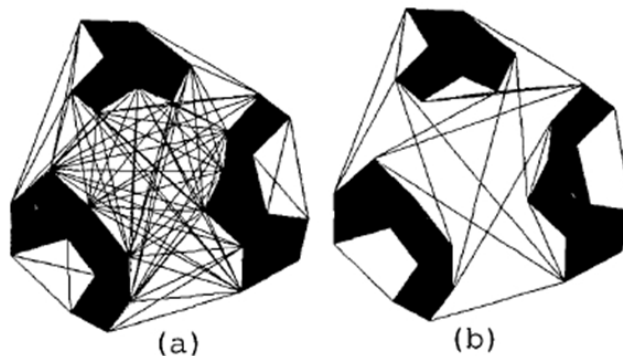


Obr. 14 Dotyčnice medzi dvoma špirálami [5]

Fig. 14 Tangents between two spirals

Z tvrdení o maximálnom počte dotyčníc medzi dvoma konvexnými objektmi a dvoma špirálami možno odvodiť veľkosť tangent grafu. Ak je prostredie definované tangent grafom,

cestu pre robot zo štartovacej do cieľovej pozície možno nájsť metódami, ktoré prehľadávajú grafy, napr. A* alebo Dijkstra algoritmus.

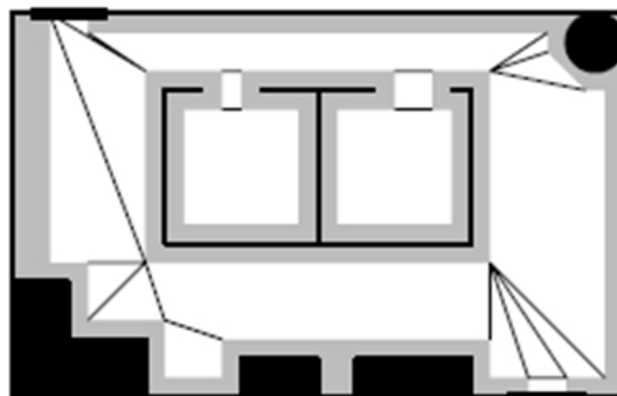


Obr. 15 Porovnanie grafu viditeľnosti (a) a tangent grafu (b) [5]

Fig. 15 Comparison of visibility graph (a) and tangent graph (b).

4. Polygónové mapy [3]

Táto metóda vytvára na základe geometrickej mapy prostredia polygónovú reprezentáciu tohto prostredia (Obr. 16). Táto reprezentácia prostredia transformuje voľný priestor do konvexných polygónov.

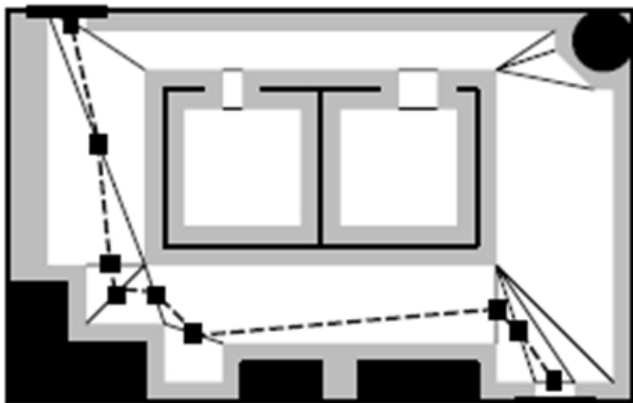


Obr. 16 Rozdelenie priestoru na konvexné polygóny [3]

Fig. 16 Partition of environment on convex polygons

Samotné polygóny v týchto mapách majú jednu dôležitú vlastnosť. Ak robot štartuje na obvode niektorého z polygónov a potrebuje sa dostať na iný bod z obvodu tohto polygónu, nikdy neprejde za okraj polygónu. Polygón teda reprezentuje bezpečné pásmo prechodu pre robot. Plánovanie dráhy na základe takejto mapy je potom otázkou výberu správneho poradia polygónov, ktorými treba prejsť. Konštrukcia takejto mapy je položená na základe globálnej geometrickej mapy. Najskôr sú prekážky "obalené" veľkosťou robota. Je to najmä z bezpečnostného dôvodu. Druhým aspektom je možnosť reprezentovať robot bodom. Ďalej je potrebné v globálnej metrickej mape nájsť význačné vlastnosti prostredia z hľadiska navigácie, napr. rohy, dvere, hranice objektov apod. Na základe týchto vlastností je možné potom vygenerovať samotné polygóny. Aj keď sa môže zdať, že plánovanie na takejto mape je už jednoduché, nie je celkom tomu tak. Niektoré čiary tvoriace hranice polygónu nemusia byť napojené na ďalší polygón (napríklad ak sú súčasťou prekážky), takže pre plánovanie ich nie je možné brať do úvahy. Iné čiary môžu byť príliš dlhé, čo spôsobuje rôzne celkové dráhy pri plánovaní. Keďže je ťažké diskretizovať čiarové segmenty, ponúka sa viacero riešení. Jedným

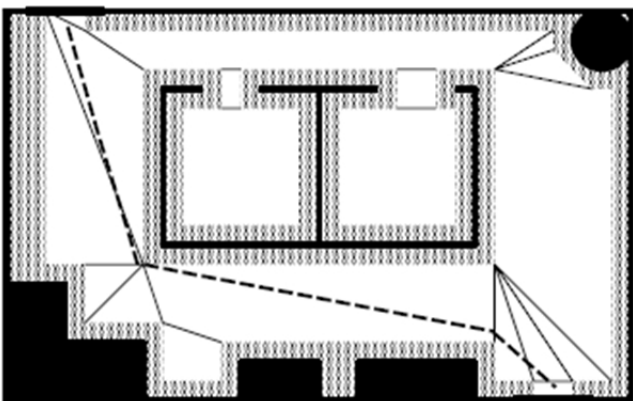
z nich je najst' stredné body čiar polygónov (Obr. 17), ktoré susedia s ďalším polygónom. Tak vznikne graf, na ktorom je pomocou grafových metód (Dijkstra, A* apod.) možné naplánovať dráhu.



Obr. 17 Nájdenie cesty pomocou stredných bodov čiar [3]

Fig. 17 Path searching with middle points of lines

Ako je vidieť, plánovanie dráhy na takto vytvorenej mape bude vždy nejakým spôsobom bez hladkého priebehu dráhy. Z tohto dôvodu Chuck Thorpe navrhol riešenie nazývané metóda napnutého vlákna (Obr. 18). Navrhnutá dráha robota je vytvorená akoby napnutým vláknom, ktoré spája oba konce prechodu polygónom. Takto navrhnutý algoritmus vyhledá priebeh dráhy bez narušenia bezpečnosti prechodu polygónmi.



Obr. 18 Nájdenie cesty pomocou napnutého vlákna [3]

Fig. 18 Path searching with tightening rope

Nevýhodami použitia polygónových máp je veľká výpočtová náročnosť, samotné využitie vlastností prostredia z globálnej mapy namiesto využitia snímaných údajov a otázka aktualizácie takejto mapy.

5. Metóda potenciálových polí [2] [4]

Táto metóda vytvára pole alebo gradient na globálnej geometrickej mape prostredia. Metóda potenciálového poľa považuje robot za bod, ktorý je ovplyvnený umelým potenciálovým polom $U(p)$. Robot sa v prostredí pohybuje pomocou sledovania tohto poľa. Cieľ je definovaný ako minimum v priestore a správa sa ako príťažlivá sila, zatiaľ čo prekážky majú predpísané správanie v podobe odpudivých síl. Superpozíciou všetkých síl aplikovaných na robot vznikne umelé potenciálové pole, ktoré plynule dovedie robot do cieľovej pozície bez kontaktu s prekážkami.

Predpokladajme, že robot je v dvojrozmernom priestore definovaný ako bod s orientáciou, teda pomocou (x, y, θ) .

Ak zavedieme diferenciálovú potenciálovú funkciu $U(p)$, môžeme potom zdefinovať aj umelú silu $F(p)$ pôsobiacu na pozícii $p = (x, y)$:

$$F(p) = -\nabla U(p), \quad (10)$$

kde $\nabla U(p)$ označuje gradient vektora U na pozícii p :

$$\nabla U(p) = \nabla U = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{bmatrix} \quad (11)$$

Potenciálové pole pôsobiace na robot je potom vypočítané ako suma príťažlivého poľa od cieľa a odpudzujúcich polí od prekážok:

$$U(p) = U_{att}(p) + U_{rep}(p) \quad (12)$$

Podobne môžeme definovať aj sily pôsobiace na robot:

$$F(p) = F_{att}(p) + F_{rep}(p) = -\nabla U_{att}(p) - \nabla U_{rep}(p) \quad (13)$$

Ďalším krokom v definovaní umelého potenciálového poľa je definícia príťažlivého a odpudivého potenciálu. Príťažlivý potenciál môže byť definovaný napríklad takto:

$$U_{att}(p) = \frac{1}{2} k_{att} \rho_{goal}^2(p), \quad (14)$$

kde k_{att} je kladný súčiniteľ a súčiniteľ $\rho_{goal}(p)$ je definovaný ako Euklidovská vzdialenosť $\|p - p_{goal}\|$. Podobne možno definovať aj príťažlivú silu F_{att} :

$$\begin{aligned} F_{att}(p) &= -\nabla U_{att}(p) = -k_{att} \rho_{goal}(p) \nabla \rho_{goal}(p) = \\ &= -k_{att} (p - p_{goal}) \end{aligned} \quad (15)$$

Táto sila konverguje k 0, ak sa robot blíži k cieľu.

Hlavnou myšlienkou definície odpudivého potenciálu je vygenerovať odpudivé sily od všetkých známych prekážok. Tento potenciál, resp. sila, by mal byť veľmi veľký v blízkosti prekážok, nemal by však ovplyvňovať pohyb robota vo veľkej vzdialenosti od týchto prekážok. Príkladom takéhoto odpudivého potenciálu môže byť:

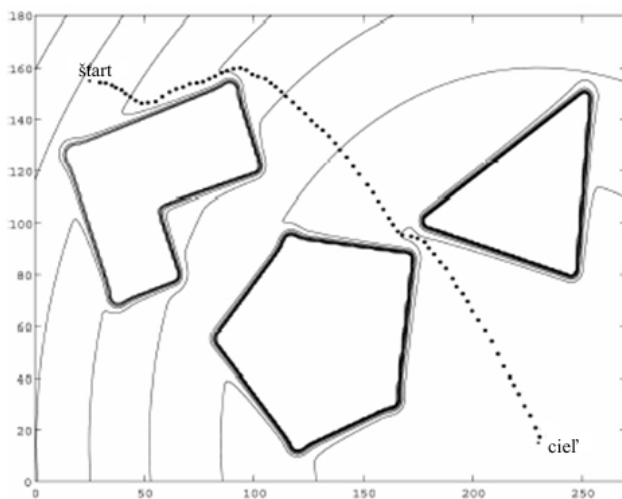
$$U_{rep}(p) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{\rho(p)} - \frac{1}{\rho_0} \right)^2 & \rho(p) \leq \rho(0), \\ 0 & \rho(p) > \rho(0) \end{cases} \quad (16)$$

kde k_{rep} je opäť súčiniteľ, $\rho(p)$ je minimálna kolmá vzdialenosť od pozície robota k objektu a ρ_0 je vzdialenosť vplyvu objektu na dráhu robota. Obdobne je možné definovať aj odpudivú silu F_{rep} :

$$F_{\text{req}}(p) = -\nabla U_{\text{rep}}(p) = \begin{cases} k_{\text{rep}} \left(\frac{1}{\rho(p)} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2(p)} \frac{p - p_{\text{prek}}}{\rho(p)} & \rho(p) \leq \rho(0) \\ 0 & \rho(p) > \rho(0) \end{cases} \quad (17)$$

kde p_{prek} je pozícia prekážky, ktorá vplyva na dráhu robota.

Výslednú silu môžeme definovať ako súčet príťažlivej a odpudivých síl. Pri ideálnych podmienkach možno nastaviť vektor rýchlosti robota proporčne vzhľadom na vektor výslednej sily a robot by mal hladko smerovať do cieľa. Avšak tento prístup je obmedzený vzhľadom na existenciu lokálnych minimí vyplývajúcich z tvaru a veľkosti prekážok. Ďalší problém sa objavuje pri výskyte konkávných prekážok. Pri takejto prekážke sa objaví niekoľko minimálnych vzdialeností $\rho(p)$, čo vedie na osciláciu robota medzi dvoma najbližšími bodmi k objektu.



Obr. 19 Prechod robota prostredím pomocou metódy umelého potenciálového poľa s vykreslením ekvipotenciálových čiar [2]

Fig. 19 Path of robot in environment calculated by artificial potential field and with portrayal of equipotential lines

PodĎakovanie

Tento článok vznikol pri riešení projektu VMSP-P-0004-09.

Literatúra

- [1] CHOSET H., LYNCH K. M., HUTCHINSON S., KANTOR G., BURGARD W., KAVRAKI L. E., THRUN S.: Principles of Robot Motion (Theory, Algorithms and Implementations). Massachusetts Institute of Technology, 2005. ISN 0-262-03327-5.
- [2] SIEGWART R., NOURBAKHSI I.R.: Introduction to Autonomous Mobile Robots. Massachusetts Institute of Technology, 2004. ISBN-13 978-0-262-19502-7.
- [3] MURPHY R. R.: Introduction to AI Robotics. Massachusetts Institute of Technology, 2000. ISBN 0-262-13383-0.
- [4] NIEMULLER T., WIDYADHARMA S.: Artificial Intelligence - An Introduction to Robotics. In: proseminar Artificial Intelligence, 2003.
- [5] LIU Y.-H., ARIMOTO S.: Proposal of Tangent Graph and Extended Tangent Graph for Path Planning of Mobile Robots. Proceedings of the 1991 IEEE International Conference on Robotics and Automation Sacramento, California, April 1991, p. 312-317.
- [6] HARGAŠ L., HRIANKA M., KONIAR D.: Image Processing and Analysis. A Practical Approach – Text Book (CD-ROM), Žilinská univerzita v Žiline 2008, ISBN 978-80-8070-962-4.
- [7] SMRČEK J., BOBOVSKÝ Z., FRIGA P.: Inovačné prístupy riešenia. Strojárstvo, Roč. 13, č. 3 (2009), s. 62-63.

Abstract

The paper analysis current state of mobile robots navigation on basis of known geometrical representation of environment. In paper are presented basic methods of path planning in known environment and there are presented advantages and disadvantages of mentioned methods.

prof. Ing. Ladislav Jurišica, PhD.,
Ing. František Duchoň, PhD.

Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3
812 19 Bratislava
ladislav.juristica@stuba.sk, frantisek.duchon@stuba.sk