



Bunková dekompozícia prostredia v mobilnej robotike

František Duchoň, Ladislav Jurišica

Abstrakt

Článok sa zaoberá analýzou súčasného stavu možností dekompozície prostredia na bunky pre potreby reprezentácie prostredia v mobilnej robotike. V článku sú predstavené metodológie exaktnej bunkovej dekompozície a približnej bunkovej dekompozície. Tieto metodológie tvorby reprezentácie prostredia umožňujú robot inteligentne navigovať v už spoznanom prostredí.

Kľúčové slová: exaktná bunková dekompozícia, približná bunková dekompozícia

Úvod

Pre riešenie úloh vyskytujúcich sa v mobilnej robotike je kľúčovou otázkou poznanie prostredia. Robot spoznáva prostredie, v ktorom vykonáva svoju činnosť pomocou snímačov. Dôležitým aspektom mobilných robotických systémov je nielen reagovať na podnety z prostredia (teda navigovať sa reaktívne), ale si aj pre potreby inteligentnej navigácie ukladať informácie o už spoznanom prostredí. Práve otázku ako ukladať informácie o spoznanom prostredí rieši bunková dekompozícia prostredia.

1 Bunková dekompozícia [1]

Ideou bunkovej dekompozície prostredia je rozdelenie priestoru reprezentovaného geometrickou mapou na geometrické úseky, resp. bunky, ktoré sú voľné a bunky, ktoré sú obsadené. Základný algoritmus dekompozície priestoru na bunky a nájdenia dráhy v tomto priestore môže byť rozdelený do týchto krokov:

1. Rozdeľ priestor na jednoduché spojené úseky, nazývajú sa bunky.
2. Rozhodni, ktoré bunky sú voľné a jednotlivé príslušné bunky spoj do grafu spojov.
3. Nájdi bunky, v ktorých sa nachádza štartovacia a cieľová pozícia a nájdi dráhu v grafe spojov tak, aby si spojil tieto bunky.
4. Zo sekvencie buniek nájdených vhodným algoritmom, vypočítaj dráhu, napríklad predpisom stredných bodov jednotlivých buniek alebo predpisom sledovania obrysov v jednotlivých bunkách.

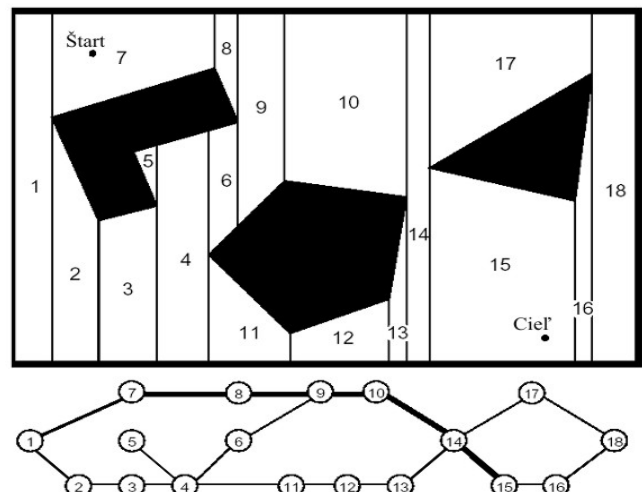
Dôležitým aspektom bunkovej dekompozície je umiestnenie hraníc medzi jednotlivými bunkami. Ak sú hranice súčasťou štruktúry prostredia, čiže dekompozícia je bezstratová, potom ide o exaktnú bunkovú dekompozíciu. Ak je dekompozícia na bunky len priblížením aktuálnej mapy, ide o približnú bunkovú dekompozíciu.

2 Exaktná bunková dekompozícia [2]

Ako už bolo spomenuté, výsledkom delenia prostredia pomocou exaktnej bunkovej dekompozície sú bunky, ktorých

hranice sú tvorené geometrickými vlastnosťami prostredia. Každá vytvorená bunka je potom alebo voľná, alebo obsadená. Pri plánovaní dráhy robota teda nezáleží na umiestnení robota v konkrétnej bunke, ale na tom, či je možné prejsť z jednej bunky do druhej. Nevýhodou tohto prístupu je veľká výpočtová náročnosť pri plánovaní a dekompozícii v prípade štruktúrovaného prostredia s veľkým množstvom a komplexnosťou objektov. Naopak aj pri prostredí s veľkými rozmermi, ale s malým množstvom objektov je vhodné použiť práve túto dekompozíciu.

Medzi metódy exaktnej bunkovej dekompozície patrí *lichobežníková dekompozícia*. Táto dekompozícia sa tvorí pre každý vrchol v_i z geometrickej reprezentácie prostredia. Z vrcholov sa tvoria tzv. horné a dolné vertikálne predĺženia. Tieto predĺženia majú ukončenia na hrane prekážok. Susednosť buniek je určená zdieľaním spoločnej hranice. Na základe grafu susednosti buniek je potom možné vyhľadávať cestu medzi štartom a cieľom. Najskôr je však potrebné určiť, ktorá bunka obsahuje štart a cieľ, a potom je možné určiť sekvenciu buniek, ktorými treba prejsť. Pričom tento prístup pre robot neurčuje množinu súradníc, ktorými má robot prejsť. Keďže lichobežník je konvexná množina, medzi dvoma ľubovoľnými bodmi lichobežníka možno určiť voľnú cestu. Preto možno robotu predpísať cestu stredmi jednotlivých predĺžení alebo princípom napnutého vlákna, teda bodmi na prieniku prekážok a predĺžení.



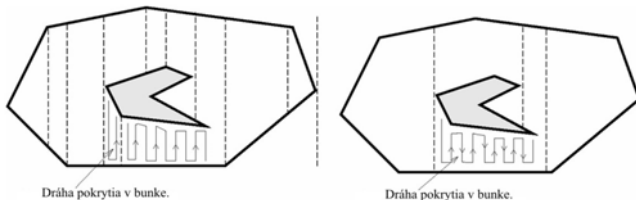
Obr. 1 Lichobežníková dekompozícia [3]

Fig. 1 Trapezoidal decomposition



V prípade úloh mobilnej robotiky, ako je prejsť celú plochu prostredia (napr. vysávanie mobilných robotov), je využitie lichobežníkovej dekompozície nevýhodné. Táto dekompozícia definuje príliš veľa buniek a robot musí začať prechádzať každú bunku od jej hranice, pričom niektoré bunky by bolo logické spojiť do väčšej a skrátiť tak na dráhu robota. Z tohto dôvodu boli zadefinované iné exaktné bunkové dekompozície.

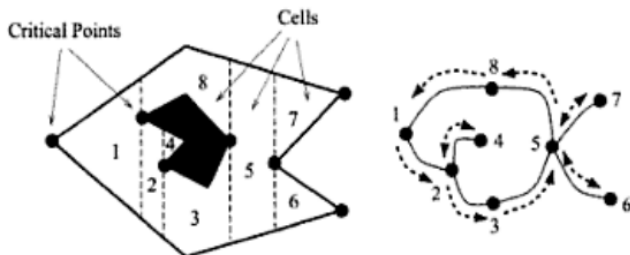
Zrkadlová dekompozícia určuje predĺženia z vektorov len v kritických bodoch. Tie sú definované tak, že je možné z kritického bodu vytvoriť horné aj dolné vertikálne predĺženie.



Obr. 2 Porovnanie lichobežníkovej a zrkadlovej dekompozície [2]

Fig. 2 Trapezoidal and mirror decomposition

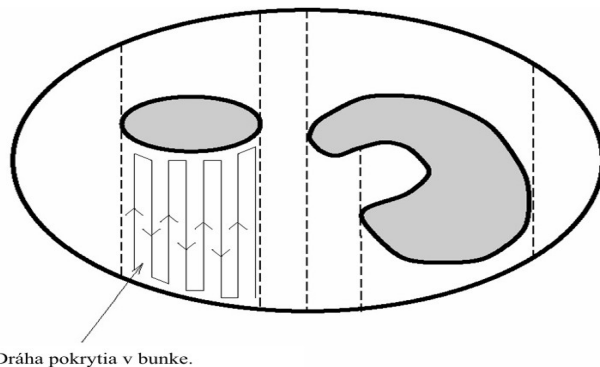
V prípade aplikácie pre prechod robota celým prostredím je potrebné určiť graf susednosti jednotlivých buniek. Na tomto grafe je potom potrebné určiť prechod robota bunkami tak, aby do každej vstúpil minimálne raz. Dráha v jednotlivých bunkách pozostáva z opakovaných sekvencií rovných čiar oddelených šírkou robota a krátkych segmentov spájajúcich konce týchto čiar. Obvykle tieto spájajúce segmenty ohraničujú hranice prostredia.



Obr. 3 Zrkadlová dekompozícia a odpovedajúci graf susednosti [2]

Fig. 3 Mirror decomposition and corresponding adjacent graph

Inou bunkovou dekompozíciou je *Morseova dekompozícia*. Vychádza z princípu obrysových metód, ktoré rozdeľujú prostredie pomocou rezov. Morseova dekompozícia je exaktná bunková dekompozícia, ktorej bunky sú spojené komponenty voľného priestoru a sú vytvorené na základe kritických bodov v priestore. Tieto body sú totožné s kritickými bodmi pri zrkadlovej dekompozícii.



Obr. 4 Morseová dekompozícia v nepolygonálnej reprezentácii prostredia [2]

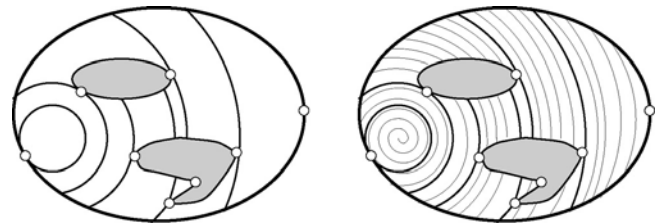
Fig. 4 Morse decomposition in not polygonal representation of environment

Morseova teória zaručuje, že medzi rezmi zostáva topológia prostredia rovnaká (t.j. nemení sa počet bodov na prieniku obrysových kriviek a priamky tvoriacej smer rezu). Pri úlohách pokrytia celého priestoru robotom je potom jednoduché poslať robot vykonávať pohyb medzi dva kritické body, teda do bunky. Pohyb v bunke potom pozostáva z pohybu pozdĺž rezu a z pohybu medzi hranicami prekážok. Pohyb pozdĺž rezu je ukončený, keď robot narazí na prekážku. Pohyb popri prekážke je odškálovaný o šírku robota (teda funkcia rezu je zväčšená o šírku robota). Výhodou použitia Morseovej dekompozície je možnosť použiť variabilný tvar rezov.

Prostredie je možné rozdeliť aj pomocou funkcie. Príkladom takejto funkcie je:

$$h(x, y) = \sqrt{x^2 + y^2}, \quad (1)$$

ktorá rozdeľuje prostredie na sústredné kružnice. Polomer kružníc sa zväčšuje práve na kritických bodoch, teda v bodoch dotyku kružníc s prekážkou. Pre úlohy pokrytia celého prostredia robotom však nie je možné vykonávať pohyb v bunkách ako v predchádzajúcich prípadoch, keďže bunky neoddeľujú jednoduché čiarové segmenty. V tomto prípade robot vykonáva špirálovitý pohyb tak, že opisuje kružnicu, ktorej polomer pri opätovnom dosiahnutí štartu zväčší o šírku robota. Ak robot narazí na prekážku, obchádza ju pozdĺž hranice dovtedy, kým robot nedosiahne zväčšenie polomeru o šírku robota a potom pokračuje opäť opisovaním kružnice s väčším polomerom.



Obr. 5 Exaktná bunková dekompozícia pomocou funkcie $h(x, y) = \sqrt{x^2 + y^2}$ a jej príslušné pokrytie prostredia [2]

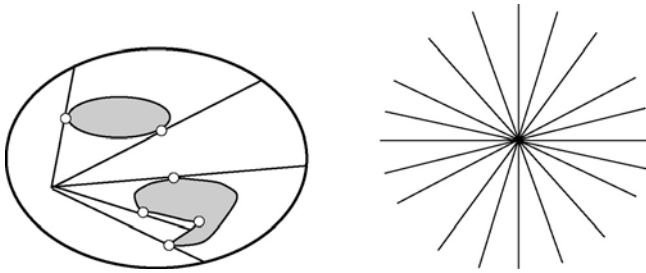
Fig. 5 Exact cell decomposition with function

$$h(x, y) = \sqrt{x^2 + y^2}$$

Inou funkciou rozdelenia prostredia rezmí je funkcia:

$$h(x, y) = \tan\left(\frac{y}{x}\right) \quad (2)$$

Táto funkcia rozdelí prostredie na čiary kolmé k sústredným kružniciam. Použitie rozdelenia prostredia na takéto bunky potom vedie na hustejšie pokrytie prostredia robotom v mieste stredu tejto funkcie.



Obr. 6 Exaktná bunková dekompozícia pomocou funkcie $h(x, y) = \tan\left(\frac{y}{x}\right)$ a jej príslušné pokrytie prostredia [2]

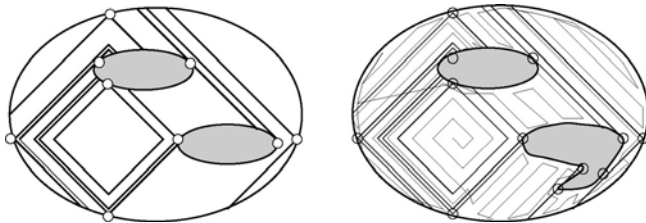
Fig. 6 Exact cell decomposition with function

$$h(x, y) = \tan\left(\frac{y}{x}\right)$$

Funkcia rezov:

$$h(x, y) = |x| + |y| \quad (3)$$

rozdelení prostredia na rotované štvorce alebo diamanty. Pokrytie prostredia robotom potom nie je vykonávané v jednotlivých bunkách, ale špirálou medzi oblasťami obmedzenými kritickými bodmi.

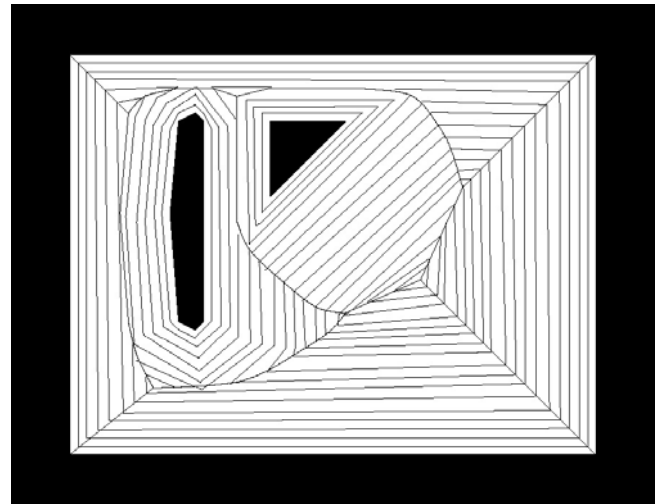


Obr. 7 Exaktná bunková dekompozícia pomocou funkcie $h(x, y) = |x| + |y|$ a jej príslušné pokrytie prostredia [2]

Fig. 7 Exact cell decomposition with function

$$h(x, y) = |x| + |y|$$

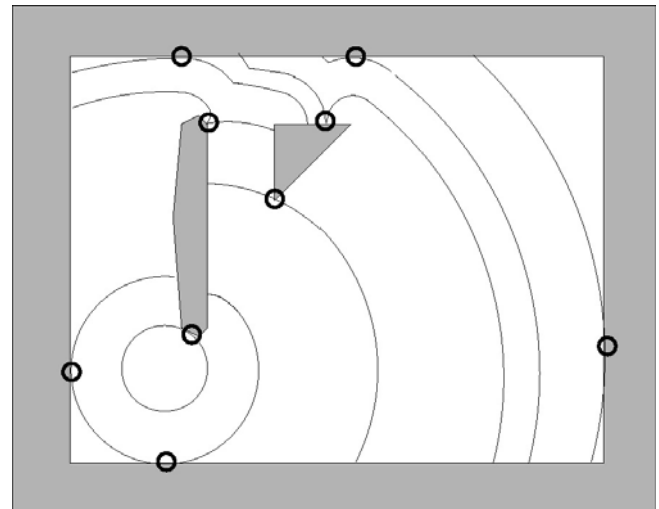
Na exaktnú bunkovú dekompozíciu možno využiť aj algoritmus *brushfire*, resp. *Voronoi diagram*. Vytvorením Voronoiho diagramu pomocou *brushfire* algoritmu možno vytvoriť jednotlivé bunky prostredia. Pre pokrytie tohto prostredia robotom je potom potrebné vytvoriť graf susednosti jednotlivých buniek a naplánovať pokrytie tak, aby každá bunka bola prejdaná aspoň raz. Pohyb potom pozostáva z troch možností: pohyb pozdĺž rezu, pohyb kolmo na rez a pohyb pozdĺž hranice bunky. Z definície Voronoiho diagramu možno jednoducho dodržať pohyb pozdĺž rezu tak, že sa robot pohybuje v rovnakej vzdialenosti od najbližšej prekážky. Robot sa pohybuje pozdĺž prekážky kým nenarazí na štartovací bod alebo na bod, kde je vzdialenosť k dvom prekážkam rovnaká. Táto bunková dekompozícia je vhodná pre roboty s veľkou chybou v odometrickom systéme a zároveň veľkým meracím rozsahom snímačov robota.



Obr. 8 Exaktná bunková dekompozícia pomocou *brushfire* algoritmu a jej príslušné pokrytie prostredia [2]

Fig. 8 Exact cell decomposition with brushfire algorithm

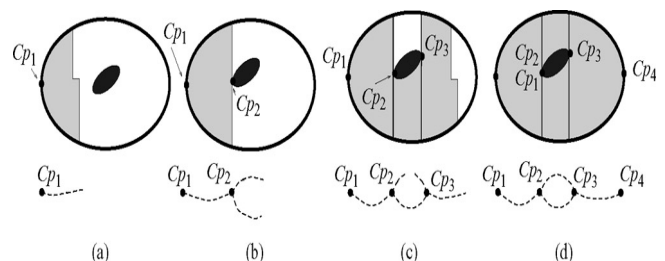
Podobne možno zdefinovať aj exaktnú bunkovú dekompozíciu pomocou *wavefront algorithm*. Kritické body sa objavujú v miestach kde sa vlna *wavefront* algoritmu stáva dotyčnicou k prekážkam.



Obr. 9 Exaktná bunková dekompozícia pomocou *wavefront* algoritmu [2]

Fig. 9 Exact cell decomposition with wavefront algorithm

Exaktnú bunkovú dekompozíciu prostredia možno získať aj pri pohybe robota v neznámom prostredí. Ide o metódu nazvanú exaktná bunková dekompozícia pomocou pokrytia snímačmi. Pri tejto metóde sa konštruuje duálny graf, tzv. Reebov graf. Vrcholmi Reebovho grafu sú kritické body a hrany tvoria spojenia medzi týmito kritickými bodmi. Princíp tvorenia grafu možno vysvetliť na obrázku (Obr. 10).



Obr. 10 Exaktná bunková dekompozícia pomocou pokrytia snímačmi [2]

Fig. 10 Exact cell decomposition with sensor coverage

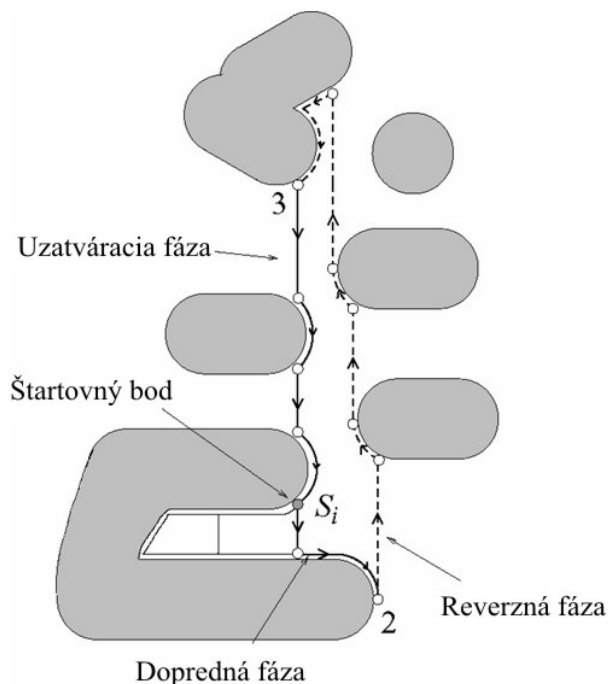


Robot začína pokrývať priestor v kritickom bode Cp_1 . Ak robot pokryje celú bunku medzi dvoma kritickými bodmi Cp_1 a Cp_2 , vytvorí odpovedajúce hrany k ďalším vrcholom. Keďže je prostredie neznáme, robot vyberie ľubovoľnú hranu. Keď robot dosiahne vrchol Cp_3 , bunka medzi Cp_2 a Cp_3 je pokrytá. V bode Cp_3 sa robot rozhoduje pokryť bunku vpravo od tohto bodu. Ak narazí na bod Cp_4 vracia sa na Cp_3 a pokryje bunku nahor od tohto bodu. Potom sa vráti až k bodu Cp_2 a rozhoduje, že boli pokryté všetky kritické body v prostredí.

Dôležitou otázkou je ako detegovať kritické body a ako zabezpečiť pokrytie všetkých kritických bodov. Kritické body možno detegovať pomocou snímačov ako extrémny v prostredí, čiže detegovať normály prekážok na smer pohybu robota, resp. detegovať globálne minimá v snímaných údajoch. Na zabezpečenie detegovania všetkých kritických bodov je potrebné, aby popri pokrývaní bunky robotom boli detegované ďalšie kritické body. Toto možno zabezpečiť pomocou algoritmu (Obr. 11) pozostávajúceho z troch krokov:

1. Dopredná fáza: Robot sleduje rez, kým nenarazí na prekážku. Potom robot sleduje hranicu prekážky, kým sa od pôvodného rezu nevzdiali o šírku robota alebo nenarazí na kritický bod.
2. Reverzná fáza: Robot vykonáva jedno alebo viac kôl (definované na šírku robota) v opačnom smere spolu s opačným sledovaním hraníc prekážok. Každá takáto operácia končí ak narazí na kritický bod alebo celkový pohyb v tejto fáze prekročí šírku robota.
3. Uzatváracia fáza: Robot vykonáva jedno alebo viac kôl v smere rezu spolu so sledovaním hraníc prekážok. Táto fáza končí v prípade narazenia na štartovací bod.

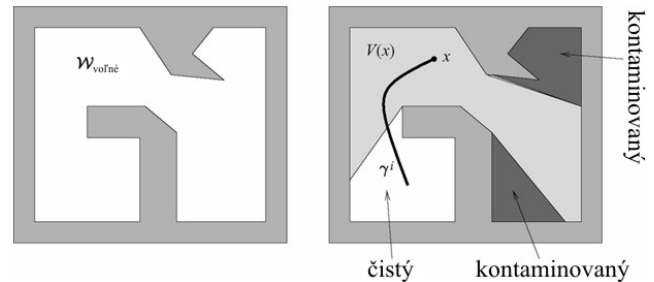
Tento algoritmus zaručuje, že robot narazí na susedný kritický bod, ak taký existuje.



Obr. 11 Algoritmus zaručujúci nájdenie susedného kritického bodu [2]

Fig. 11 Algorithm guarantee found of adjacent critical point

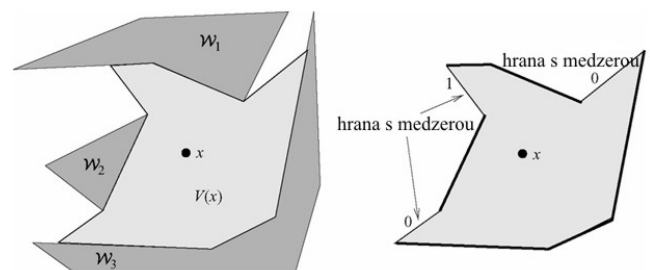
Ďalšou možnou metódou využiteľnou na tvorbu exaktnej bunkovej dekompozície prostredia počas pohybu robota v neznámom prostredí je *metóda viditeľnej dekompozície*. Táto metóda funguje na princípe viditeľnosti časti prostredia, v ktorom sa robot pohybuje. Metóda rozdelenia prostredia na bunky pomocou metódy viditeľnej kompozície je často aplikovaná na problematiku prenasledovaný - prenasledovateľ. Z tohto dôvodu je prostredie rozdelené na regióny kontaminované, teda ešte nepreskúmané (resp. kde sa môže nachádzať prenasledovaný), regióny zjavné, teda preskúmané a na regióny znovu kontaminované, čiže so zmenou v prostredí (resp. kde sa prenasledovaný mohol vrátiť aj keď región už bol preskúmaný) (Obr. 12).



Obr. 12 Definovanie typov regiónov viditeľnej dekompozície [2]

Fig. 12 Definition of region types

Vytvorením oblasti viditeľnosti z nejakého miesta v prostredí vzniknú v prostredí dva typy hrán. Jeden typ hrán charakterizuje obrysy prekážok. Druhý typ hrán charakterizuje náhlu zmenu prekážky, resp. veľký skok v meranej vzdialenosti a teda predstavuje medzeru medzi dvoma prekážkami (hrana s medzerou). Tieto medzery bývajú označené nulou, ak sa dotýkajú zjavného regiónu a jednotkou ak sa dotýkajú kontaminovaného regiónu (Obr. 13).



Obr. 13 Označenie dvoch typov hrán [2]

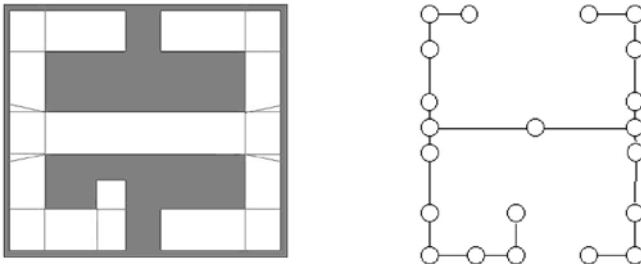
Fig. 13 Definition of edges types

Nech $B(x)$ predstavuje binárny vektor označení medzier regiónu so stredom v x . Pár $(x, B(x))$ potom predstavuje informačný stav a množina všetkých možných informačných stavov predstavuje informačný priestor. Spojená množina $v \subset Q_{free}$ je konzervatívna ak pre všetky $x \in v$ je $B(x)$ rovnaké. Z konzervatívnych oblastí potom možno vytvoriť graf susednosti jednotlivých buniek. Na vytvorenie takejto dekompozície v prostredí reprezentovanom polygónmi je potrebné predĺžiť každý konvexný vrchol čiarou v smere hrany prekážky tak, aby dosiahol prekážku. Ak sú dva vrcholy navzájom viditeľné, treba z nich vytvoriť dve čiary, každú z jedného vrcholu orientovanú v smere spojnic vrcholu smerom od opačného vrcholu. Inak povedané, ak platí pre vrchol v_i z prekážky O_i a pre vrchol v_j z prekážky O_j :

$$\lambda v_i + (1 - \lambda) v_j \in Q_{free} \quad \forall \lambda \in (-\varepsilon; 1 + \varepsilon), \quad (4)$$

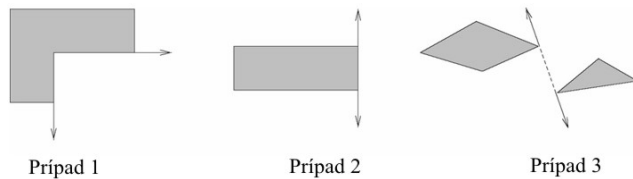


pre nejaké $\varepsilon > 0$, potom predĺž čiaru z v_i smerom od v_j a naopak, kým čiara nepretne prekážku. Takto vznikne bunková dekompozícia, kde každá bunka je konzervatívny región.



Obr. 14 Dekompozícia prostredia na konzervatívne regióny a odpovedajúci graf susednosti [2]

Fig. 14 Environment decomposition on conservative regions and corresponding adjacent graph



Obr. 15 Tri možnosti predĺžení pri tvorbe konzervatívnych regiónov [2]

Fig. 15 Three cases of extensions in creation of conservative regions

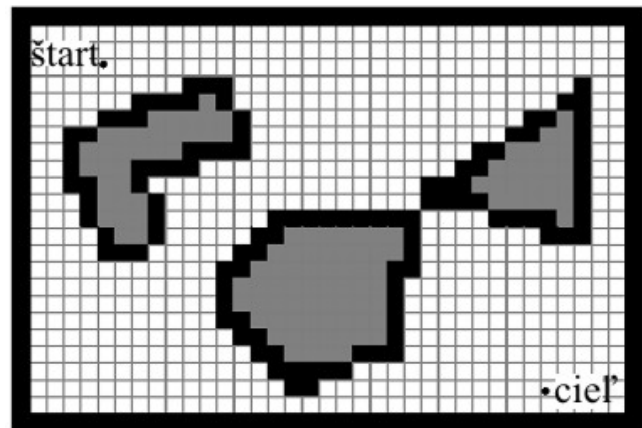
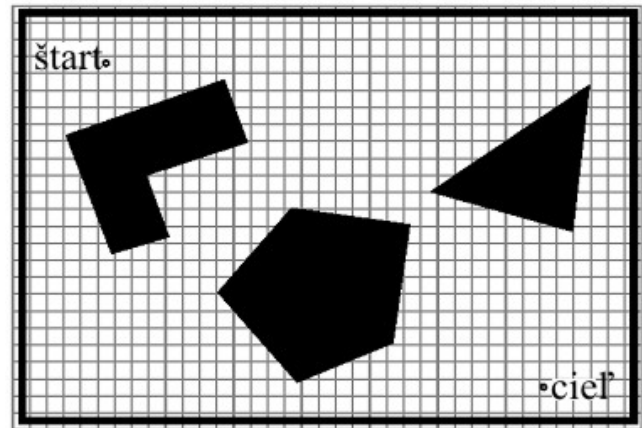
Takáto dekompozícia však nie je vhodná na riešenie problému prenasledovaný - prenasledovateľ. Je potrebné vytvoriť informačný graf:

1. Ak zmizne medzera, nerob nič.
2. Ak sa objaví medzera, priradi jej hodnotu 0.
3. Ak sa dve alebo viac medzier spoja do jednej a jedna z nich má hodnotu 1, priradi hodnotu 1 aj novej medzere.
4. Ak sa jedna medzera rozdelí do viacerých medzier, každej z nich priradi pôvodnú hodnotu.

Pre každú konzervatívnu bunku je potrebné priradiť možné označenie viditeľných medzier. Hrany medzi jednotlivými priradeniami potom určujú cestu prechodu prostredím. Pre pravú hornú bunku existuje len jedna medzera, ktorá môže byť označená jednotkou alebo nulou. To isté platí aj pre pravú spodnú bunku. Avšak prechod medzi týmito bunkami je obmedzený v zmysle prehľadávania priestoru. Inými slovami, ak pre hornú bunku $B(x) = 1$, potom aj pre spodnú bunku musí byť $B(x) = 1$. Prehľadávanie na takomto grafe zaručí pokrytie celého priestoru.

3 Približná bunková dekompozícia [3] [4] [5]

Populárnejšou a častejšie využívanou ako exaktná bunková dekompozícia je približná bunková dekompozícia, teda vytvorenie globálnej metrickej mapy. Je to dané najmä tým, že sa v robotike často využíva mriežková reprezentácia prostredia [6]. Najčastejšie sa používa mriežka s pevnou veľkosťou bunky. Veľkosť tejto bunky nezávisí od objektov v prostredí, je volená, preto sa môže stať, že niektoré vlastnosti prostredia sa v tejto mriežkovej reprezentácii neobjavia.



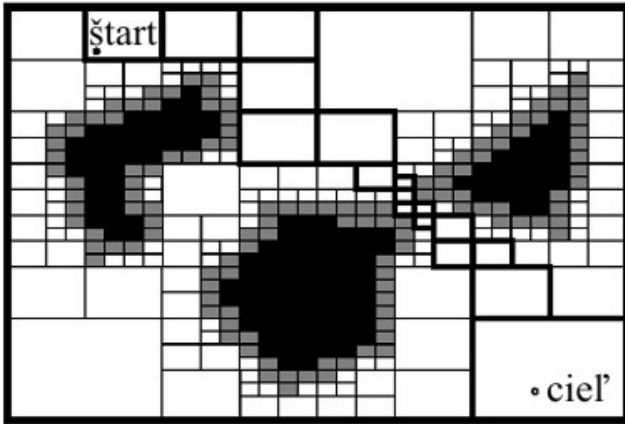
Obr. 16 Približná bunková dekompozícia [3]

Fig. 16 Approximate cell decomposition

Veľkou výhodou tohto prístupu je však nenáročný plánovanie dráhy mobilného robota. Algoritmus jednoducho ohodnotí bunky spätne od definovaného cieľa tak, že ku každej bunke priradí vzdialenosť od cieľa. Proces pokračuje, kým nie je dosiahnutá štartovacia pozícia. Je potom jednoduché odhadnúť vzdialenosť robota od cieľa, ako aj navrhnúť dráhu prechodu prostredím. Tá je určená vždy susednými bunkami pri uplatnení pravidla, že susedná bunka je vždy bližšie k cieľu. Takto definovaným algoritmom je každá bunka robotom "navštívená" len raz. Výpočtová náročnosť nezávisí od počtu objektov v prostredí ani od ich komplexnosti.

Nevýhodou použitia dekompozície prostredia pomocou približnej bunkovej dekompozície sú veľké nároky na pamäť. Pre geometrický veľký prostredie musí byť uložená celá táto mriežka reprezentujúca prostredie, aj keď sa v prostredí nachádza len malý počet objektov. Aj z tohto dôvodu sa používajú bunky premenlivej veľkosti.

Najčastejšie používanou metodológiou je *kvadrantový strom* [7][8]. Tento kvadrantový strom je vlastne rekurzívnu mriežkou, kde každý uzol má štyri odvodené uzly. Rekurzívna mriežka sa generuje na celej globálnej metrickej mape. V prvom kroku je do algoritmu vložená bunka veľkosti celej mapy. Algoritmus generovania kvadrantového stromu otestuje, či nejaký objekt zasahuje do tejto bunky. Ak áno, táto bunka je rozdelená na štyri časti, ktoré sú potom rovnakým spôsobom testované. Algoritmus generovania kvadrantového stromu pokračuje dovtedy, kým nie sú všetky uzly rozdelené na voľné a obsadené alebo keď veľkosť najmenších odvodených uzlov je menšia alebo rovná veľkosti robota.



Obr.17 Dekompozícia prostredia kvadrantovým stromom [3]

Fig. 17 Quadtree decomposition

PodĎakovanie

Tento článok vznikol pri riešení projektu VEGA 1/0690/09 a KEGA 3/7307/09.

Literatúra

- [1] NIEMULLER T., WIDYADHARMA S.: Artificial Intelligence - An Introduction to Robotics. In: proseminar Artificial Intelligence, 2003.
- [2] 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.
- [3] SIEGWART R., NOURBAKHSI I.R.: Introduction to Autonomous Mobile Robots. Massachusetts Institute of Technology, 2004. ISBN-13 978-0-262-19502-7.

[4] MURPHY R. R.: Introduction to AI Robotics. Massachusetts Institute of Technology, 2000. ISBN 0-262-13383-0.

[5] PATNAIK S.: Robot Cognition and Navigation (An Experiment with Mobile Robots). Springer-Verlag Berlin Heidelberg, 2007. ISSN 1611-2482.

[6] HANZEL J.: Mapovanie prostredia na báze meraní ultrazvukom. Bratislava, STU v Bratislave FEI, 2007, 138 s. + Autoref. 2008, 34 s.

[7] VÖRÖS J.: A strategy for repetitive neighbor finding in images represented by quadrees. Pattern Recognition Letters, Vol. 18 (1997), s. 955-962, ISSN 0167-8655

[8] VÖRÖS J.: Quadtree-Based Representation of Grid-Oriented Data. Image and Vision Computing, Vol. 24 (2006), s. 263-270. ISSN 0262-8856.

Abstract

This paper deals with analysis of present knowledge of environmental cell decomposition for purpose of environment representation in mobile robotics. In paper are described methods of exact cell decomposition and approximate cell decomposition. These methods allow intelligent navigation of robot in described environment.

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

Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3
812 19 Bratislava

ladislav.jurisica@stuba.sk, frantisek.duchon@stuba.sk