A foglaltsági háló és más térképépítési stratégiák

Szabó Richárd *

Kivonat

A cikk áttekinti a robotikai navigációval kapcsolatos problémákat és bemutat néhány lehetséges térképépítési módszert, fókuszálva a kilencvenes években erősen elterjedt valószínűségi eljárásokra, így a Kálmán-szűrőre, a várhatóérték-maximalizálásra, és a foglaltsági hálóra.

Ezen belül részletesen tárgyalja a foglaltsági háló létrehozását és használatát a Webots nevű programozási környezetben. A szimuláció módosított Khepera típusú robotokkal működik, melyek szonar érzékelőiket felhasználva járják be a különféle kísérleti környezeteket.

Kulcsszavak: mobil robotok, szimuláció, metrikus/topologikus navigáció, kognitív térkép, foglaltsági háló.

1. Bevezetés

Tudományos előrejelzések alapján a robotika hasonlóan fontos részét fogja képezni hétköznapjainknak, mint ahogy az autó szerepe nőtt meg a 20. században. A fejlődést a tudósok és a mérnökök által létrehozott egyre fejlettebb robotok és robotirányító eljárások biztosítják.

A kutatások elengedhetetlen kellékei a robotszimulációs szoftverek. Ezek az eszközök lehetővé teszik a robot felépítésének tervezését és a robotot irányító algoritmus implementálását. Időigényes tanuló eljárások futtatása jóval egyszerűbb egy szimulátorban, míg a költséges robotot „csak” a finomhangolás elvégzésére kell használni ([1]).

Jelenleg a robotok terjedésének egyik fő akadálya a megbízható, gyors, tömegtermelésre alkalmas navigációs eljárás hiánya.

A navigációnak azért van kiemelt szerepe a robotok létrehozásában, mert ennek segítségével képes a mobil, intelligens jármű meghatározni saját és a számára fontos objektumok helyét a környezetében. Navigáció nélkül nem képzelhetőek el az olyan berendezések, mint az önjáró háztartási robotok, őrszemek vagy bolygókutató szondák.

2. A navigáció problémái

A navigáció során a robot saját és a számára fontos objektumok pozícióját igyekszik meghatározni. Ennek érdekében leggyakrabban egy belső térképet kezel. A feladat tehát kettős: egyszerre kell pozíciókat meghatározni és elhelyezni azokat a folyamatosan

*Eötvös Loránd Tudományegyetem, Általános Számítástudományi Tanszék, 1117 Budapest, Pázmány P. s. 1/D, and Tudománytörténeti és Tudományfilozófia Tanszék 1117 Budapest, Pázmány P. s. 1. e-mail: ricsinf.elte.hu

1. táblázat. Az odometria hibái

Rendszeres	**Rendszertelen**
– eltérő kerékméretek	– egyenetlen padló
– szabálytalan kerék	– kerekek csúszása
– szenzor mintavétele	– találkozás tárgyakkal
– szenzor felbontása	

alakuló térképen. Az irodalom a problémát *simultaneous localization and mapping* néven említi ([2]). A feladat tyúk-tojás jellege mellett egyéb problémák is jelentkeznek.

A robot mozgási utasításait és a szenzorok méréseit kísérő zaj rontja a navigáció minőségét. Ha a zaj független a robot helyétől és a cselekvés idejétől, akkor mind több begyűjtött adat előbb-utóbb konvergenciához vezet. Azonban a zaj legritkább esetben ilyen tulajdonságú, például a mérési hibák akkumulálódnak.

Egy másik probléma a leképezendő környezet sokdimenziós jellegéből adódik: egy szoba részletes, csupán kétdimenziós térképe is több ezer elemet tartalmazhat, ami a számítási igényt növeli.

Az adat-összerendelés problémája – mely talán az egyik legkomolyabb – rámutat arra a nehézségre, hogy miként lehet a térkép elemeit különböző időben és/vagy különböző nézőpontban összerendelni. Például amikor egy robot egy körfolyosó bejárása után visszaérkezik kiindulópontjára, miként veszi észre, hogy már járt ott.

A negyedik navigációs probléma a környezet változásában keresendő. A kísérleti terepek bejárása során létrehozott statikus térképek a valóságban nem állják meg a helyüket, amikor a robot körül emberek közlekednek, ajtók nyílnak és csukódnak, vagy amikor a robot átrendezett lakásba kerül.

A fentieken túlmenőleg egy jól működő navigációs eljárásnak valós időben kell kiadnia a mozgást szabályzó utasításokat, megbízhatónak kell lennie. A végső cél, hogy egy ilyen eljárás általános célú legyen, azaz tetszőleges környezetben működjön.

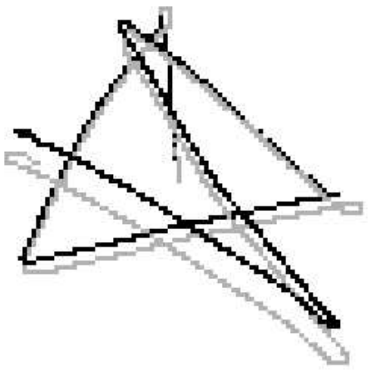
2.1. Lokalizáció

A navigáció alapvető nehézsége a mozgás és érzékelés során felmerülő zajból származik. Enélkül a robot — folyamatosan számontartva irányának és elmozdulásának változását — pontosan tudná saját és a tárgyak helyét a térképen. Ez az odometriának nevezett módszer egyszerű koordinátageometriával megoldhatná a legfőbb problémákat.

A folyamatosan jelen lévő rendszeres és rendszertelen zaj az idő előrehaladtával az odometriából nyert információt teljesen használhatatlanná torzítja ([3], [4]). A 1. táblázat a zajok különféle forrásait mutatja be.

A rendszeres hibák a robot megismerésével kompenzálhatóak, de a rendszertelenek ellen nincs védelem. Így hosszú távon, a pozícióbecslés hibajavítása nélkül divergál a robot, mint az a 1. ábrán is látható, ahol a kezdőpontban még együtt lévő valódi (fekete

szín) és számított (szürke szín) pozíció néhány száz lépés megtétele után viszonylag jelentős eltérést mutat.



1. ábra. Valós és számított útvonal eltérése

3. Navigációs módszerek osztályozásai

A navigáció összetett problémájára számtalan – részleges – megoldási módszer létezik. Ezeket az elképzeléseket többféleképpen lehet csoportosítani.

3.1. Világ- és robotközpontú navigáció

A navigáció egyik csoportosítási módja, hogy a térkép kinek a nézőpontjából készül. A világközpontú térkép egy globális koordináta-rendszerben ábrázolja a tárgyakat, míg a robotközpontú a felfedező robot szempontjából. Az előbbi térképet nehezebb előállítani, viszont a terep bejárása közben jobban használható, az utóbbi térkép könnyebben előáll, de a hasonló helyek megkülönböztetése nehezebb.

3.2. Metrikus és topologikus navigáció

A metrikus, geometrikus vagy háló-alapú navigáció a tárgyak számokkal mérhető térbeli tulajdonságaival foglalkozik, így távolságokkal, bezárt szögekkel és koordinátákkal. A generálódó térkép leginkább egy madártávlati képre hasonlít, a környezet egy méretarányos leképezése.

A topológiai navigáció a tárgyak egymáshoz való viszonyát tartja szem előtt. A fő kérdés itt az egyik pontból a másikba eljutás lehetősége. A topológiai típusú térképezés során a környezetről egy gráf készül, melynek csúcspontjai jól felismerhető tereptárgyak, élei pedig a tárgyak közötti közvetlen összeköttetést jelzik ([5], [6]).

A valóságban a metrikus és topologiai navigáció nem különül el élesen, a megoldások általában egy keveréket alkalmaznak. A 2. táblázat a két módszer előnyeit és hátrányait összegzi.

2. táblázat. Metrikus és topologikus navigáció

Metrikus	**Topologikus**
+ könnyen kezelhető	– nehezen kezelhető
+ zajra nem érzékeny	– zajra érzékeny
– nem alkalmazkodó	+ alkalmazkodó
– memóriaigényes	+ kevés memória elég
– a navigáció nehézkes	+ a navigáció könnyű
– pontos pozíció kell	+ becsült pozíció elég

3.3. Valószínűségi térképezés

A kilencvenes években komolyan elterjedő módszerek általában valószínűségszámítási alapokon nyugszanak. Ennek az lehet az egyik oka, hogy a zaj által keltett bizonytalanságokat egy monoton – azaz javításra nem alkalmas – térkép esetén nehéz kezelni. Ezen módszerek közös jellemzője, hogy mind a Bayes-tételen alapulnak ([7]), és maximum-likelihood térképet hoznak létre, azaz az adatok alapján legvalószínűbb térképet készítik el.

3.3.1. Kálmán-szűrő

A navigáción kívül is jól használható Kálmán-szűrő egy lineáris differenciaegyenletrendszer hatékony megoldási módja. A szűrő az időt diszkrét pillanatokra bontva kezeli, és rekurzívan próbálja meghatározni a zajjal rendelkező megoldást ([8]). A kitűzött feladat egyik felét az alábbi egyenlet írja le.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Itt x_k a közelítendő állapotleíró vektor a k időpillanatban, navigáció esetén a robot és a környező tárgyak pozíciója. x_k elsősorban a korábbi állapottól (x_{k-1}) függ, másodsorban a robotot irányító parancsoktól (u_{k-1}), valamint egy normális eloszlású zajtól (w_{k-1}).

A robot a környezet állapotát nem érzekeli közvetlenül, csupán szenzorain keresztül. Ezt a második egyenlet modellezi.

$$s_k = Hx_k + v_k$$

Itt s_k a robot által érzékelt, az adott állapotnak megfelelő környezet, melyet normális eloszlású zaj (v_k) terhel.

A Kálmán-szűrő a fent vázolt differenciaegyenlet-rendszert oldja meg az A, B, H mátrixok és a két normális eloszlású zaj paramétereinek ismeretében.

A Kálmán-szűrő előnyös tulajdonsága, hogy iteratív, azaz az előző számításokra alapulva lehet a következő becslést kalkulálni. Hátránya, hogy az adat-összerendelést nem tudja megoldani, mivel két, nem megkülönböztethető tereptárgy térképbe illesztése ütközik a normális eloszlású zaj feltétellel.

3.3.2. Várható érték maximalizálása

Ez a módszer szintén nem csak a navigáció témakörében ismert, hanem általános célú eszköz mintavételek alapján a rendszer ismeretlen paramétereinek meghatározására ([9], [10]). A várható érték maximalizálása két lépés nyugalmi állapotig tartó alternatív ismételéséből áll. Először – a navigációnál maradva – az adott térkép és a begyűjtött adatok alapján meg kell állapítani a robot pozíciójának várható értékét.

$$Q(m|m_k) = E_{m_k}[p(x, s|m_k)|s]$$

A másik lépésben a robot pozíciójából és a begyűjtött adatokból kell meghatározni a lehető legvalószínűbb térképet, meg kell keresni a térképek terében a maximális valószínűségűt.

$$m_{k+1} = argmax_m Q(m|m_k)$$

A várható érték maximalizálása hatásosabb megoldását nyújtja az adat-összerendelési problémának, mint a Kálmán-szűrő, valamint tetszőleges zajeloszlás mellett jól működik. A módszer komoly hátránya – ami miatt általában más eljárásokkal ötvözik –, hogy nem működik iteratív módon, vagyis az adatok alapján mindig elölről kell kezdeni a térkép építését. Ebből következően valódi robotban nem is alkalmazható nagy számításigénye miatt.

3.3.3. Foglaltsági háló

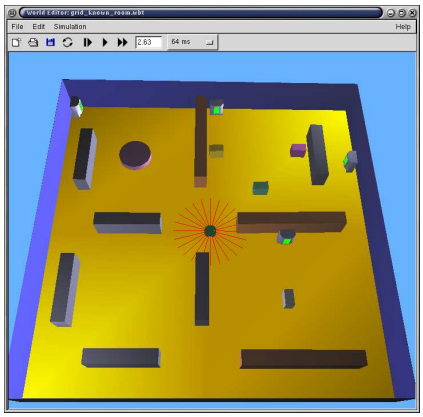
A foglaltsági háló a sík vagy a tér cellákra bontott parkettázása. Minden cella egy valószínűségi értéket tartalmaz, mely a tér adott részének foglaltságát reprezentálja. A módszer előnye a többivel szemben a viszonylag egyszerű implementálhatóság, és az iteratív térképépítés lehetősége. A hátrányok között érdemes megemlíteni, hogy a rendszerben fellépő zajnak időben függetlennek kell lennie, illetve, hogy a navigálás során ez a struktúra nehezen alkalmazható.

3.4. Egyéb jellemzések

A robotikai navigáció területe annyira szerteágazó, hogy még sokféle kategorizálását lehetne adni. Néhány érdekesebb csoportosítás: a robot szabadságfoka szerint (holonomikus, nem holonomikus), a navigációs elemek bonyolultsági szintjei szerint (alapviselkedések, elemi, taktikai, stratégiai navigáció - [11]) , a viselkedés biológiai megalapozottsága szerint (keresés, iránytartás, útvonalösszegzés, helyfelismerés - [12]).

4. Foglaltsági háló Webotsban

Kutatásunk célja egy foglaltsági hálót alkalmazó térképezés megvalósítása volt a Webots szimulációs környezetben ([13]). A robot a terep bejárása során egy felülnézeti valószínűségi térképet épít a néhány négyzetméteres környezetről ([14]).



2. ábra. A Webots szimulátor

A választott tenyérnyi Khepera robot infravörös érzékelőit 24 darab, 15 cm hatótávú szonarra cseréltük. Ezek a módosítások megkönnyítik a pontos térkép gyors létrehozását. A távolságszenzorok használata vizuális szenzorok helyett gyakran célszerű, mivel az érzetek dimenziószáma miatt a képfeldolgozás jóval összetettebb feladat, és valódi robotnál a kamerák felszerelése lényegesen drágább is.

A térkép építésének lényeges fázisai S. Thrun munkájához hasonlóan a következőek ([15]):

- szenzorinterpretálás
- időbeli integrálás
- pozícióbecslés
- globális hálóépítés
- felfedezés

4.1. Szenzorinterpretálás

A szenzorinterpretálás a foglaltsági háló készítésének első fázisa.

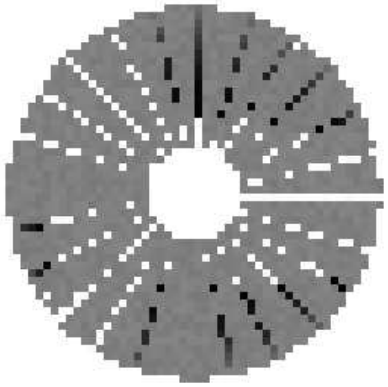
A 24 darab körkörösen elhelyezett szonar kellően sűrű információforrás ahhoz, hogy a robot környezetében a foglaltságot számítani lehessen. Ez a számítás skalárértékek áttranszformálása egy kétdimenziós valószínűségmátrixxá, vagyis

$$p(occ_{x,y}|s),$$

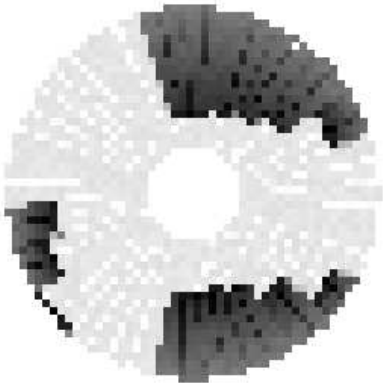
értékeket kell meghatározni, ahol s egy mérés az (x, y) cellán. Bár erre gyakran mesterséges neurális hálót alkalmaznak, az egyszeri kézi hangolás is elfogadott.

Minden szenzor által visszaadott érték arányos a legközelebbi tárgy távolságával az adott irányban. Ezért a robothoz a szenzorértéknél „közelebb" logikus alacsony valószínűséget (ε), a szenzorérték helyén magas valószínűséget ($1 - \varepsilon$) rendelni; majd ezután a bizonytalanságból adódóan az érték lecsökken a teljes információhiányig ($1/2$).

Mivel a cél bejárható útvonalak megtalálása, ezért hasznos a szonar sugárát mesterségesen kiszélesíteni a robot szélességére. Az approximációból adódó pontatlanságért bőven kárpótol a pontosabb lokális térkép létrehozásának lehetősége, amint az a 3. és a 4. ábrát összehasonlítva látható.



3. ábra. Egyszerű lokális térkép



4. ábra. Kiszélesített lokális térkép

4.2. Időbeli integrálás

A teljes foglaltsági háló a környezet bejárásával keletkezik. Ez annyit jelent, hogy sok mérést végez a robot egy adott cellára vonatkozóan is. Mivel a mérés körülményei (a robot helye és iránya, zaj, stb.) változnak, ezért különböző mérési értékeket kell egy helyre integrálni. Tehát az egyes mérésekből származó feltételes valószínűségekből ($p\left(occ_{x,y}|s^{(t)}\right)$) kell kiszámítani az összes mérésből adódó feltételes valószínűséget ($p\left(occ_{x,y}|s^{(1)}, s^{(2)}, \ldots, s^{(T)}\right)$). A számításhoz a Bayes-tételt ([7]) és a mérések egymástól való függetlenségének feltételét kell felhasználni. Vagyis $p\left(s^{(t)}|occ_{x,y}\right)$ független $p\left(s^{(t')}|occ_{x,y}\right)$-től, ha $t \neq t'$.

A fentiek alapján kijelenthető, hogy

$$\Pi(x, y, T) = p\left(occ_{x,y}|s^{(1)}, s^{(2)}, \ldots, s^{(T)}\right) =$$

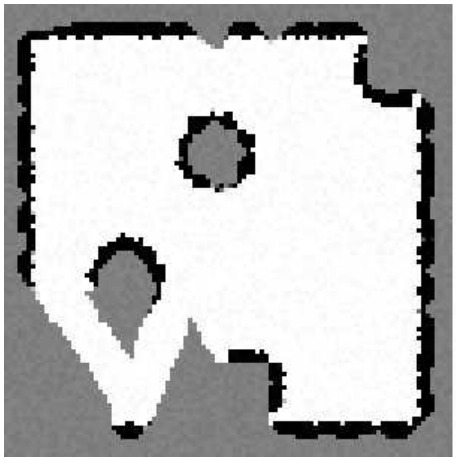
$$1 - \left(1 + \frac{p(occ_{x,y}|s^{(1)})}{1 - p(occ_{x,y}|s^{(1)})} \prod_{\tau=2}^{T} \frac{p(occ_{x,y}|s^{(\tau)})}{1 - p(occ_{x,y}|s^{(\tau)})} \frac{1 - p(occ_{x,y})}{p(occ_{x,y})}\right)^{-1} \quad (1)$$

ahol $p(occ_{x,y})$ a kezdeti valószínűség, általában $1/2$-re állítva.

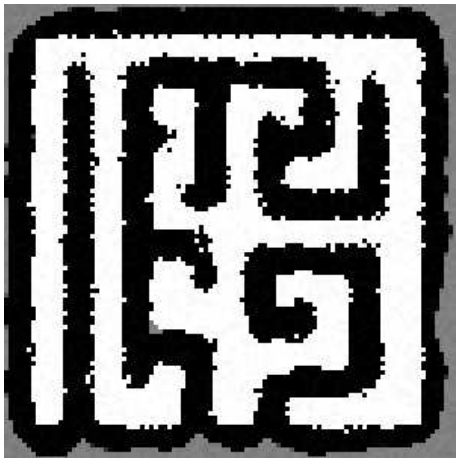
A (1) egyenlet fő következménye a mérések egyesítésének lehetősége, egymás utáni szorzások alkalmazásával.

4.3. Globális hálóépítés

Miközben a robot körbejár környezetében, a lokális információk a globális térképre kerülnek. Ez egyrészről polárkoordináták Descartes-koordinátákra transzformálását jelenti, másrészről a szenzorok által mért értékeket is egyesíteni kell. Az idő előrehaladtával a robot egyre nagyobb területet fedez fel, amint az nyílt terepen (5. ábra) és a labirintusban (6. ábra) is látható.



5. ábra. Nyílt terep foglaltsági hálója



6. ábra. Labirintus foglaltsági hálója

4.4. Felfedezés

Bár a robot az eddigi modulokkal felkészült a térképépítésre, szükség van valamilyen hajtóerőre, aminek hatására bejárja a teljes terepet, egyébként csupán véletlenszerűen mozogna.

Ennek érdekében egyfajta értékiterációt alkalmazó kereső eljárást valósítottunk meg ([16]). A választott eljárás minden egyes cellára kiszámítja a legrövidebb út költségét egy még nem bejárt cellához. A kalkulált értékek egy újonnan bevezetett költségmátrixba kerülnek.

1. Az eljárás első lépése az inicializálás. A nem bejárt cellák értéke logikusan 0, míg a bejártaké ∞.

$$V_{x,y} \leftarrow \begin{cases} 0 & \text{ha } (x,y) \text{ bejáratlan} \\ \infty & \text{ha } (x,y) \text{ bejárt} \end{cases}$$

2. A főciklus során minden bejárt cella értéke újraszámolódik. A valószínűleg foglalt cellák magas költségértéket kapnak. A többi cella értéke a környező cellák költségének és foglaltsági valószínűségének minimumából adódik. A δ tag az út hosszúságát bünteti.

   $\forall$ bejárt (x,y):

$$V_{x,y} \leftarrow \begin{cases} p\left(occ_{x+i,y+j}\right) & \text{ha } p\left(occ_{x+i,y+j}\right) > 1-\varepsilon \\ \min\limits_{\substack{i=-1,0,1 \\ j=-1,0,1}} \left\{V_{x+i,y+j} + p\left(occ_{x+i,y+j}\right) + \delta\right\} & \text{egyébként} \end{cases}$$

   Konvergencia után a V mátrix a kumulált költségeket tartalmazza.

3. Mivel egy terület felfedezése az egész költségmátrixra kihat, ezért azt folyamatosan újra kellene számolni. Ez azonban jelentősen lassítaná a program működését. Emiatt a mátrix nem csupán lokális, teljes újraszámolása csak bizonyos időközönként következik be.

A felfedezés irányát ezután lényegében a költségmátrix határozza meg. A robot igyekszik a kis környezetében található minimális érték felé mozogni, hisz abban az irányban érdemes nem bejárt cellák után nézni. A kis környezet segítségével az olyan cellák felé történő probálkozás elkerülhető, amivel a robot nincs közvetlen összeköttetésben.

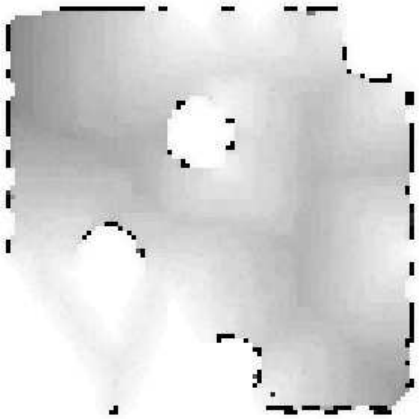
A robot aktuális orientációja szintén szerepet játszik a felfedezés irányának meghatározásában, ezzel biztosítható a folytonos, nem csapongó mozgás. Ezenkívül a környezet akadályait is figyelembe kell venni. Az egymást kiegészítő minimumkeresés és akadálykikerülés együtt egy elfogadható szintű navigációt tesz lehetővé.

A 7. és a 8. ábra a kialakult költségmátrixokat mutatja a bejárás egy adott pontján a 5. és a 6. ábrán bemutatott foglaltsági hálókhoz.

5. Eredmények

A kutatás során létrehoztunk egy kétdimenziós foglaltsági hálót, C nyelven, mely további navigációs feladatokra is alkalmazható a Webots szimulációs környezetben. A fent vázolt eljárás többféle kísérleti környezetben is működött, az 1 m^2-es nyílt terepel 8, a 2,25 m^2-es labirintussal, valamint az irodaszerű szobával 22, illetve 20 perc alatt végzett átlagosan.

A kísérlet során beigazolódott, hogy a metrikus navigáció eléggé számításigényes, ezért érdemes topológiai módszerrel ötvözni, vagyis a metrikus térkép alapján egy topologikus gráfot építeni, és az útvonaltervezést azon elvégezni.



7. ábra. Nyílt terep költségmátrixa

8. ábra. Labirintus költségmátrixa

A kísérletek egy további tapasztalata, hogy sok szempontból kifizetődő egy szimulátor használata. Nagyon egyszerű új kísérleti környezeteket létrehozni, módosítani a robot felépítését, ideértve szenzorainak számát, elhelyezkedését, modalitását és érzékenységét.

6. További lehetőségek

A kutatást több irányba is ki lehet terjeszteni.

- Más navigációs módszerek megvalósítása.
- Pozícióbecslés hibajavításának megvalósítása.
- Metrikus és topologikus navigálás kombinációja, kihasználva a metrikus előállításának egyszerűségét és a topologikus alkalmazásának hatékonyságát.
- Részben vagy teljesen dinamikus objektumok – ajtók, emberek – modellezése.
- Magasszintű feladatok elvégzése a navigációra alapozva.
- Feladatok megoldása a szabad ég alatt.
- Új érzékelők használata.

Köszönetnyilvánítás
Köszönet illeti témavezetőmet, Kampis Györgyöt támogatásáért és tanácsaiért. Köszönöm Salamon Andrásnak, hogy az anyag összeállítását segítette hasznos megjegyzéseivel.

Hivatkozások

[1] R. Szabó. *Mobil robotok szimulációja*. Eötvös Kiadó, 2001.

[2] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002. to appear.

[3] J. Borenstein and L. Feng. Correction of systematic odometry errors in mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 12, 1996.

[4] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning - sensors and techniques. *Journal of Robotic Systems*, 14(4):231–249, 1997.

[5] O. Trullier and J.-A. Meyer. Biomimetic navigation models and strategies in animats. *AI Communications*, 10(2):79–92, 1997.

[6] M. Franz and H. Mallot. Biomimetic robot navigation. *Robotics and Autonomous Systems*, 30:133–153, 2000.

[7] T. M. Mitchell. *Machine learning*. McGraw Hill, 1997.

[8] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina, Department of Computer Science, Chapel Hill, NC, USA, 2003.

[9] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots*, 31(5):1–25, 1998.

[10] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.

[11] B. Krieg-Bruckner. A taxonomy of spatial knowledge for navigation and its application to the bremen autonomous wheelchair, 1998.

[12] H. A. Mallot and M. O. Franz. Biological approaches to spatial representation - a survey. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Articial Intelligence (IJCAI-99)*. Morgan Kaufmann, 1999.

[13] Cyberbotics S.a r.l. *Webots 3.2. User Guide*, 2002.

[14] R. Szabó. Navigation of simulated mobile robots in the webots environment. *To appear in Periodica Politechnica*, 2002.

[15] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[16] A. Barto R. Sutton. *Reinforcement Learning: An Introduction*. Bradford Book, MIT Press, 1998.