

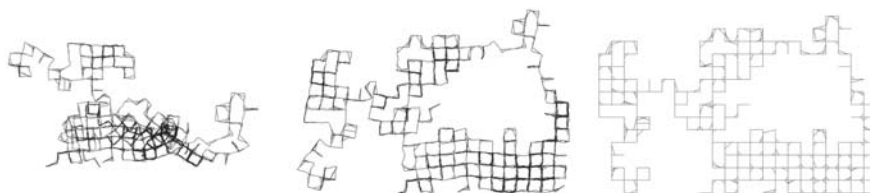
# Robusztus kezdeti pozíció-gráf előállítására hatékony szimultán helymeghatározási és térképezési rendszerekhez

Harsányi Károly, Kiss Attila, Majdik András, Szirányi Tamás

Gépi Érzékelés Kutatólaboratórium, MTA SZTAKI  
{harsanyika,kiss.attila,majdik,sziranyi}@sztaki.hu

**Kivonat** A robotikából ismert Szimultán Helymeghatározási és Térképezési rendszerek különböző alapeladatok megoldásával próbálnak feltérképezni egy adott területet. Ezt gyakran számos részfeladat szekvenciális megoldásával és feldolgozásával érik el. Az egyik ilyen részfeladat a kezdeti közelítő becslés meghatározása, amelyből valamilyen optimalizálási eljárással megpróbálnak eljutni a feltérképezett területet legjobban közelítő geometriai gráfhoz. Erre a közelítési eljárásra adunk egy újabb módszert, amelynél a sebesség és a robusztusság volt két alapvető költségfüggvény, melyekre optimalizáltunk. Kísérletekkel megmutattuk, hogy az eljárásunk futási ideje a hasonlóan robusztusra tervezett módszerektől eltérően független a mérési hibákból származó zaj méretétől és nagyságrendben a leggyorsabbak közé sorolható. Mégis robusztusabban jut el az eredeti struktúrát jól közelítő állapotba a belőle indított optimalizálási algoritmus úgy a mindenki által használt mesterséges teszt adathalmazokon, mint a valós életből vett mérési adatokon. Eredményeinket ebben a cikkben prezentáljuk.

## 1. Bevezető



1. ábra. A Gauss-Newton optimalizációs algoritmus eredményei különböző kezdeti becslésekből kiindulva. Balra: A *spanning tree* becslésből indított Gauss-Newton végeredménye, középen: az általunk prezentált algoritmusból indított Gauss-Newton végeredménye, jobbra: a közelíteni kívánt eredeti állapot.

Ez a cikk a pozíció-gráf (*pose graph*) alapú Szimultán Helymeghatározási és Térképezési (*Simultaneous Localization and Mapping*, továbbiakban SLAM)

rendszerekkel [1] és ezek kezdeti struktúrájának becslésvel (*initial guess*) foglalkozik.

Számos modern képalapú vizuális odometria (pl., SVO algoritmus [2]) és vizuális SLAM rendszer (pl., LSD-SLAM [3], ORB-SLAM [4] algoritmusok) a SLAM probléma gráf struktúrával való reprezentációjára épül. A gráf csúcspontjaiban a kulcsképek (*keyframe*) pozíciói találhatóak, a gráf élei pedig az ezek között mért elmozdulást jelölik. Az élek lehetnek a kamera képében - a mozgás által előidézett változás alapján - számolt elmozdulás (vizuális odometria), vagy a vizuális hasonlóság alapján felismert visszatérési pozíciók, azaz a körutak bezárása (*loop closure*). Nem kizárólag vizuális SLAM rendszerek esetében ezen mérések származhatnak elfordulást számláló haladásmérő (*wheel odometry encoder*), inerciális mérőegységekből (*IMU - Inertial Measurement Unit*) vagy más technikával előállított adatokból. A körutak bezárása történhet LiDAR letapogatással vagy pontfelhő megfeleltetéssel, és egyéb helyfelismerő algoritmusokkal is. A SLAM algoritmusok fő feladata az egymást követő adalékos mérések által felhalmozott hiba minimalizálása a különböző körutak bezárása által nyert megszorítások felhasználásával.

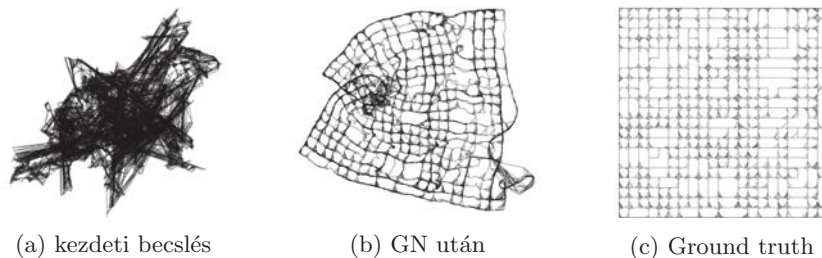
A korszerű pozíció-gráf optimalizációs eljárások eredményességét nagyban meghatározza a kezdeti becslés, amelyből az optimalizációt indítjuk (ld. 1. ábra). Ahhoz, hogy az optimalizáció megbízhatóan működjön a gráf struktúrájának jó - a valóshoz közeli - kezdeti becslésére van szükség. Enélkül a hiba minimalizálása során könnyedén elakadhatunk egy hamis lokális minimumban, valamint nagyon rossz kezdeti becslés esetén a hiba akár divergálhat is. Ha nem megfelelő kezdeti becslés miatt az optimalizációs algoritmus egy lokális minimumban akad el, az kritikus hibának tekintendő. Erre példa a 2. ábra. Ezen kívül egy jó kezdeti becslés nagyban csökkentheti az optimalizáció futási idejét is.

Továbbá fontos a jó kezdeti pozíció-gráf előállítása, ami elvezet az eredményes SLAM optimalizáláshoz, hiszen az automatikus navigáció - beleértve az autonóm járműveket és a pilóta nélküli földközeli repülő eszközöket is - egy dinamikus változó környezetben kell, hogy működjön. Ebben a környezetben a referenciaként megjelenő dolgok egy része ismerős lehet (hasonló valami korábbihoz egy közeli helyen), de lehetnek eddig nem látott objektumok is; ezért egy ilyen rendszer működése alapvetően folyamatos pozícionálást és helyzetfelismerést igényel. Emiatt szükségünk van egy olyan algoritmusra, amely a rendelkezésre álló - de folyamatosan újuló - adatokból gyorsan tud pozíció-gráfot előállítani. Ennek a folyamatnak az első kritikus lépése a kezdeti gráf előállítása.

Ugyanakkor az autonóm robotjárművek tájékozódásának segítése a környezeti modellekhez való folyamatos alkalmazkodást igénylik. Ahhoz, hogy a robotjárművek és a valós, élő szereplők közös forgalmában boldoguljunk, a modelleket folyamatosan frissíteni kell, amit a járművek maguk végeznek, miközben a leképező eszköz továbbhalad és másik jelenik meg. Ezekből az információdarabokból kell összeállítani a környezet folyamatos modelljét.

Napjainkban a legelterjedtebb pozíció-gráf SLAM keretrendszer a g2o [5], mely a Gauss-Newton [6] és Levenberg-Marquardt [7] algoritmusokra épül. A g2o-ban jelenleg két eljárás található a kezdeti becslés meghatározására. Ezek: (i)

az egyszerű odometria (*odometry*), amely az egymást követő adalékos méréseket használja kiindulási struktúraként; valamint (ii) a feszítőfa (*spanning tree*) módszer, amely minden mérési pontra meghatározza a legrövidebb mérés sorozatot az adott pont és egy kezdőpont között, majd ezen méréssorozatok alapján hozza létre a kiindulási struktúrát. A feszítőfa módszer egy szélességi keresésre épülő eljárás, amit először 1959-ben Dijkstra [8] és Moore [9] jegyeztek le.



2. ábra. (a): kezdeti becslés a City10k adathalmaz (0.3,0.3,0.3) zajjal terhelt példányára, (b): kezdeti becslésből indított 50 Gauss-Newton optimalizáció eredménye, (c): az eredeti állapot, amit közelíteni szeretnénk. Mivel a kezdeti becslés nem megfelelő, ezért a Gauss-Newton algoritmus lokális minimumban akad el.

Azonban zajos vagy sok mérést tartalmazó pozíció-gráf konfigurációk esetében mindkét eljárásra jellemzőek a korábban említett problémák: megnövekedhet a futási idő, a hibafüggvény lokális minimumba konvergálhat, vagy esetleg divergálhat.

A szakirodalomban több cikk is tárgyalja a rossz kezdeti becslésből eredő problémákat, illetve több eljárás létezik jó kezdeti becslés létehozására. Ilyen például a LAGO (*Linear Approximation for Graph Optimization*) [10] algoritmus, amely azonban csak kétdimenziós mérések esetén alkalmazható, valamint a TORO (*Tree-based Network Optimizer*) [11] és ennek általánosabb változata a CAUCHY [12] algoritmusok. Ezek közül a CAUCHY működik a legjobban. Nagy zajszint esetén is képes jó kezdőbecsléseket előállítani, azonban a módszer iteratív és nem-lineáris optimalizációt igényel, ami miatt futási ideje sokszor az optimalizáció többszöröse, ezért nem sorolható az egyszerű, heurisztikus algoritmusok közé. A LAGO, TORO és CAUCHY módszerek részletes összehasonlítása megtalálható a [12]-ben.

Ebben a cikkben a MASAT (*Multi-Ancestor Spatial Approximation Tree*) algoritmust mutatjuk be pozíció-gráfok kezdeti struktúrájának becslésére. A javasolt módszer előnye, hogy képes az *odometry* és *spanning tree* eljárásoknál lényegesen megbízhatóbb kezdeti becslést előállítani és futási ideje csak a gráf struktúrájától függ, vagyis független a méréseket terhelő zaj méretétől, ezért jóval gyorsabb, mint a TORO vagy a CAUCHY algoritmusok. A bevezetett algo-

ritmust részletesen összehasonlítjuk és kiértékeljük számos erre a célra előállított, leggyakrabban használt adathalmaz (*benchmark dataset*) segítségével.

## 2. Pozíció-gráf optimalizáció

A SLAM feladatban a mérési pontok pozícióját kell úgy meghatározni, hogy a köztük lévő mérések hibája minimális legyen. Ehhez definiálni akarunk egy  $G = (V, E, p)$  geometriai gráfot, melynek  $V$  csúcshalmazát az  $x_1, x_2, \dots, x_n$  pontok alkotják és az optimalizációhoz felhasznált mérési pontoknak felelnek meg.  $\forall i, j \in \{1, \dots, n\}, i \neq j$  indexpárra  $x_i x_j \in E$  ha a két mérési pont relatív elhelyezkedésére van valamilyen odometriából, vagy kör bezárásából származó mérésünk. Tehát minden  $x_i x_j \in E$  élhez tartozik

- egy  $z_{ij} = h(x_i, x_j) + \omega_{ij}$  mérés, ahol a  $h(x_i, x_j)$  az  $x_i$  és  $x_j$  pontok tényleges távolsága  $\omega_{ij}$  pedig a méréshez tartozó zaj
- egy  $\Omega_{ij}$  információs mátrix, amivel a mérés bizonytalanságát tudjuk kifejezni.

Az  $(x_i, x_j)$  mérés hibája tehát  $e_{ij} = (z_{ij} - h(x_i, x_j))^T \Omega_{ij} (z_{ij} - h(x_i, x_j))$ . Feladatunk, hogy minimalizáljuk az

$$F(x) = \sum_{e_{ij} \in E} e_{ij}^T \Omega_{ij} e_{ij} \quad (1)$$

függvényt, azaz a pozíció-gráf  $\chi^2$  hibáját. Tehát a csúcsok egy olyan  $p : V \rightarrow \mathbf{R}^d$  projekcióját keressük a  $d$  dimenziós euklideszi térbe, amire

$$p(V) = \operatorname{argmin}_{x \in V} \sum_{e_{ij} \in E} e_{ij}^T \Omega_{ij} e_{ij} \quad (2)$$

A feladat megoldására a legkorszerűbb és leggyorsabb módszerek a Levenberg-Marquardt [7] és a Gauss-Newton [6] algoritmusok. Ezek részletes leírása egy helyen is megtalálható egy a g2o keretrendszer bemutató cikkben [5]. Mindkét algoritmus futása akkor sikeres, ha a futtatást megelőzően rendelkezünk egy megfelelően megválasztott kezdőbecsléssel a gráf csúcsainak elhelyezkedésére. Ellenkező esetben az algoritmusok a hibafüggvény lokális minimumában akadhatnak el, vagy a hiba akár divergálhat is.

## 3. A javasolt módszer áttekintése

A korábban említett TORO és CAUCHY módszerek egészen jó kezdeti becsléseket adtak a feladatra, de meglehetősen nagy a számítási igényük a mi módszerünkhöz képest. A g2o-ban is megtalálható *odometry* és *spanning tree* kezdeti becslések pedig csak kis csúcsszámú gráfok és kis méretű zaj estén megbízhatóak. Célunk az volt, hogy ezeknél a módszereknél megbízhatóbb kezdeti becslésre

képes algoritmust hozunk létre a számítási idő és a műveletigény jelentős növekedése nélkül. Az algoritmusunk egy csúcs pozícióját a szomszédainak a "tippjéből" próbálja meghatározni. Formálisan ez azt jelenti, hogy a gráf  $x_i \in V$  csúcsain az  $x_1$  csúcsból indított szélességi bejárás alapján végighaladva megvizsgáljuk az  $N_i = \{\forall x_j \in V : x_i x_j \in E\}$  halmazt, ami tehát az  $x_i$  csúcs szomszédait tartalmazza. Minden  $x_j \in N_i$  csúcs, ami a szélességi bejárásban előbb következett az  $x_i$  csúcsnál ad egy  $p_{x_j}(x_i) = (p_1, \dots, p_d)$  pozíciót (ahol  $d$  a dimenziók számát jelöli), ahova szerinte el kell helyeznünk az  $x_i$  csúcsot a mért távolság és látószög adatok alapján. Ezután kiátlagoljuk ezeket a pozíciókat és megkapjuk az  $x_i$  csúcs új pozícióját. Ha minden csúcsot elért a szélességi bejárás az algoritmus leáll (ld. Algorithm 1).

---

**Algorithm 1: MASAT algoritmus**


---

```

1 MASAT ( $V, E, Z$ );
   Input : A nyers mérési adatok gráfja,  $v_i \in V$  csúcsok,  $e_{ij} \in E$  élek és  $z_{ij}$ 
           mérési eredmények minden  $e_{ij}$  élre
   Output: Egy kezdeti becslés a valós pozíció-gráfra
2 Kezdetben minden csúcsot "nem rögzített" címkével látunk el
3  $v_0$  pontot kiválasztjuk az origónak és rögzítjük.
4 Kiválasztjuk a  $v_0$  csúcs minden szomszédját, és elhelyezzük az indexeit a Q
   sorba.
5 while  $Q$  nem üres do
6    $w$  = az aktuálisan vizsgált csúcs indexe  $Q$ -ból.
7   for  $A$   $v_w$  csúcs minden  $v_i$  szomszédjára do
8     if  $v_i$  még nem volt kiválasztva then
9        $v_i$  csúcs indexét adjuk hozzá  $Q$ -hoz
10    else if  $v_i$  már rögzített then
11      a  $z_{iw}$  mérést vegyük hozzá a becslésünkhöz
12      Az eddigi becslésünket  $v_w$  pozíciójára módosítsuk az új mérési
           eredmény felhasználásával
13    end
14     $v_w$  csúcsot helyezzük el a fent meghatározott adatok alapján
15    rögzítsük le a  $v_w$  csúcsot.
16    Vegyük ki  $v_w$  csúcs indexét  $Q$ -ból.
17 end

```

---

Ahogy az algoritmusunk leírása is sejteti, az egyszerűsége törekedve és a nagy számításigényű műveletek nélkül, egy gyors algoritmust nyertünk, amely az esetek nagy részében nemcsak a mesterséges, hanem a valós adathalmazokon is jobban teljesít, mint a hasonló műveleti igényű algoritmusok. Ezt a kiértékelés részben számokkal is alátámasztjuk.

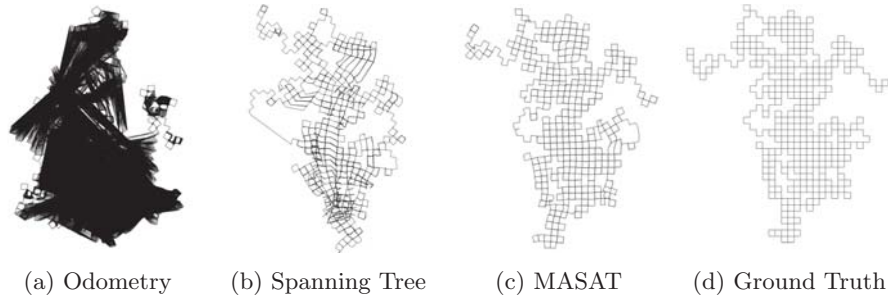
#### 4. Kiértékelés

A MASAT algoritmust az egyszerű, heurisztikus *odometry* és a *spanning tree* módszerekkel hasonlítottuk össze. A kiértékeléshez a g2o keretrendszert használtuk, és az ebben található (*gn\_var\_cholmod*) Gauss-Newton implementációt. A Gauss-Newton bemenetei a minimalizálni kívánt függvény - esetünkben a  $\chi^2$  hiba (1) - és egy kezdeti becslés a mérési pontok helyzetére (a 2. táblázatban a normalizált - vagy redukált -  $\chi^2$  hibát ( $\chi_\nu^2$ ) használjuk amelyet úgy kapunk, hogy a  $\chi^2$  hibát elosztjuk a rendszer szabadságfokával).

A teszteléshez három olyan adathalmazt használtunk, amelyeket a szakirodalom gyakran használ különböző módszerek értékelésére és összehasonlítására: a Manhattan3500, a Manhattan10000 [13], illetve a City10k adathalmazokat. Mindhárom adathalmazon az  $(x, y, \theta)$  méréseket - azaz a pozíció-gráf éleit - független, nulla várható értékű, normál eloszlású - vagyis  $N(0, \sigma_x)$ ,  $N(0, \sigma_y)$ ,  $N(0, \sigma_\theta)$  - zajjal terheltük, tehát az élek információs mátrixai:

$$\Omega = \begin{bmatrix} 1/\sigma_x^2 & 0 & 0 \\ 0 & 1/\sigma_y^2 & 0 \\ 0 & 0 & 1/\sigma_\theta^2 \end{bmatrix}$$

alakúak. Több zajszintet vizsgáltunk a zaj szórásának függvényében. Minden zajszintre minden adathalmazból 50 példányt generáltunk. Minden példányra létrehoztunk *odometry*, *spanning tree*, MASAT kezdőbecsléseket, majd maximum 50 Gauss-Newton iterációt futtattunk. Az adathalmazokhoz tartozó *ground truth*-ből - azaz a pontos, zajmentes adatokból - is indítottunk 50 Gauss-Newton iterációt, és az így kapott eredményeket tekintettük optimális megoldásnak (kontrollnak). A 3. ábrán látható, milyen kezdőbecslést készít a 3 összehasonlított eljárás a *ground truth*-hoz képest, a  $(0.0, 0.0, 0.1)$  meretű zajjal terhelt Manhattan10000 adathalmazon.



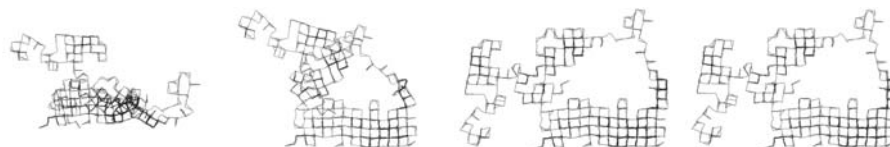
3. ábra. Különböző kezdőbecslések a Manhattan10000 adathalmazra  $(0.0, 0.0, 0.1)$ -es zaj mellett, illetve az adathalmazhoz tartozó *ground truth*.

Egy kísérletet akkor tekintettünk sikeresnek, ha két iterációs lépés között a  $\chi^2$  hiba változása a  $|10^{-6}|$  küszöb alá esett. Abban az esetben ha ez 50 iteráció alatt nem következett be, a kísérletet sikertelennek tekintettük. Egy-egy ilyen kísérlet végeredményét mutatják be a 4. és 5. ábrák a Manhattan3500, illetve Manhattan10000 adathalmazokon. A 6. ábra pedig azt mutatja meg, hogyan változtak a végeredmények a különböző kezdőbecslések esetén a City10k adathalmazon a zajszint függvényében.



(a) Odometry + GN (b) Span. tree + GN (c) MASAT + GN (d) Ground t. + GN

4. ábra. A Manhattan10000 adathalmaz egy  $(0.2, 0.2, 0.2)$ -es független zajjal terhelt változata, különböző kezdőbecslések és 50 Gauss Newton iteráció után.



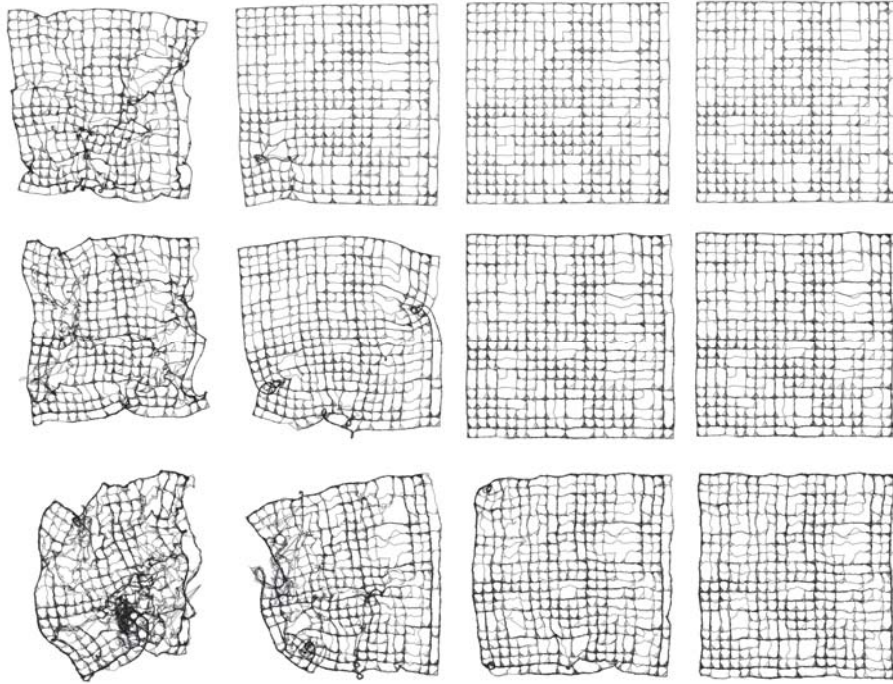
(a) Odometry + GN (b) Span. tree + GN (c) MASAT + GN (d) Ground t. + GN

5. ábra. A Manhattan3500 adathalmaz egy  $(0.2, 0.2, 0.2)$ -es független zajjal terhelt változata, különböző kezdőbecslések és 50 Gauss Newton iteráció után.

Mindhárom adathalmazra azt mértük, hogy a különböző zajszinteken a különböző kezdőbecslésekből kiindulva a Gauss-Newton futása milyen arányban sikeres. Ezeket a méréseket az 1. táblázat tartalmazza.

Sikeres kísérletek esetén összehasonlítottuk a  $|10^{-6}|$  küszöb eléréséhez szükséges Gauss-Newton iterációk számát, valamint a  $\chi^2$  hiba méretét. Az eredmények a 2. táblázatban találhatóak. Az iterációk száma a megállásig a futási sebességgel arányos. A hiba kontrolltól való eltérése azt mutatja, hogy a Gauss-Newton lokális minimumban állt meg.





(a) Odometry + GN (b) Span. tree + GN (c) MASAT + GN (d) Ground t. + GN

6. ábra. 50 Gauss-Newton futási eredményei a City10k adathalmazon, különböző kezdőbecslésekből kiindulva, különböző zajszintek mellett. A zaj szórása az első sorban  $(0.1, 0.1, 0.1)$ , a másodikban  $(0.2, 0.2, 0.2)$ , az utolsóban pedig  $(0.35, 0.35, 0.35)$ .

## 5. Konklúzió

Ahogy az 1. táblázatból is látszik, azt tapasztaltuk, hogy a vizsgált módszerek közül egyértelműen a MASAT-ból indított Gauss-Newton konvergált a legnagyobb arányban, minden zajszinten és adathalmazon. A 2. táblázat alapján kijelenthető, hogy sikeres kísérletek esetén a Gauss-Newton iterációk átlagos száma és az átlagos  $\chi^2$  hiba mérete MASAT algoritmussal volt a legközelebb a *ground truth*-ből indított kísérletek - vagyis az optimális megoldás - eredményeihez. Tehát a vizsgált módszerek közül a MASAT módszer minden szempontból jobban teljesített az *odometry* és *spanning tree* eljárásoknál. Az is megfigyelhető, hogy a módszer a sűrűbb gráfokon (Manhattan adathalmazok) működött a legpontosabban.

Az általunk kidolgozott eljárás biztos algoritmikus alapot ad olyan alkalmazásokban, ahol stabilá teheti az időnként előforduló nagy hibájú helyzet vagy elmozdulás-becslésekből adódó pozicionálási számításokat, illetve lehetővé teszi a



bizonytalan eredetű helyzetadatok vagy lazán kapcsolódó gráfok alapján történő pontosabb struktúra kialakítását.

Manhattan3500 convergence rates				
noise( $\sigma_x, \sigma_y, \sigma_\theta$ )	Odometry+GN	Sp. tree+GN	MASAT+GN	G. truth+GN
(0.1, 0.1, 0.1)	0.64	<b>1.0</b>	<b>1.0</b>	1.0
(0.15, 0.15, 0.15)	0.36	0.96	<b>1.0</b>	1.0
(0.2, 0.2, 0.2)	0.26	0.82	<b>0.96</b>	1.0
(0.25, 0.25, 0.25)	0.28	0.72	<b>0.94</b>	1.0
(0.3, 0.3, 0.3)	0.2	0.52	<b>0.8</b>	1.0
(0.35, 0.35, 0.35)	0.08	0.42	<b>0.7</b>	0.98
(0.15, 0.15, 0.3)	0.04	0.28	<b>0.76</b>	1.0
(0.3, 0.3, 0.15)	0.76	1.0	<b>1.0</b>	1.0
City10k convergence rates				
noise( $\sigma_x, \sigma_y, \sigma_\theta$ )	Odometry+GN	Sp. tree+GN	MASAT+GN	G. truth+GN
(0.1, 0.1, 0.1)	0.0	0.96	<b>1.0</b>	1.0
(0.15, 0.15, 0.15)	0.0	0.5	<b>1.0</b>	1.0
(0.2, 0.2, 0.2)	0.02	0.2	<b>0.96</b>	1.0
(0.25, 0.25, 0.25)	0.0	0.12	<b>0.66</b>	1.0
(0.3, 0.3, 0.3)	0.0	0.02	<b>0.34</b>	0.96
(0.35, 0.35, 0.35)	0.0	0.0	<b>0.18</b>	0.98
(0.15, 0.15, 0.3)	0.0	0.0	<b>0.22</b>	0.8
(0.3, 0.3, 0.15)	0.0	0.8	<b>1.0</b>	1.0
Manhattan10000 convergence rates				
noise( $\sigma_x, \sigma_y, \sigma_\theta$ )	Odometry+GN	Sp. tree+GN	MASAT+GN	G. truth+GN
(0.1, 0.1, 0.1)	0.22	0.96	<b>1.0</b>	1.0
(0.15, 0.15, 0.15)	0.1	0.78	<b>0.98</b>	1.0
(0.2, 0.2, 0.2)	0.08	0.46	<b>0.96</b>	1.0
(0.25, 0.25, 0.25)	0.02	0.18	<b>0.88</b>	1.0
(0.3, 0.3, 0.3)	0.0	0.1	<b>0.82</b>	1.0
(0.35, 0.35, 0.35)	0.0	0.02	<b>0.6</b>	0.94
(0.15, 0.15, 0.3)	0.0	0.02	<b>0.84</b>	1-0
(0.3, 0.3, 0.15)	0.3	0.84	<b>1.0</b>	1.0

1. táblázat. Sikeres kísérletek aránya különböző zajszintek és kezdeti becslések esetén a Manhattan3500, City10k és Manhattan10000 adathalmazokon. Odometry+GN: Odometry becslés + 50 Gauss-Newton, Sp. tree+GN: Spanning tree becslés + 50 Gauss-Newton, MASAT+GN: MASAT becslés + 50 Gauss-Newton, G. truth+GN: Ground truth + 50 Gauss-Newton.

Manhattan3500								
	Odometry+GN		Sp. tree+GN		MASAT+GN		G. truth+GN	
noise( $\sigma_x, \sigma_y, \sigma_\theta$ )	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$
(0.1, 0.1, 0.1)	30.19	7.13	7.76	1.02	<b>6.3</b>	<b>1.0</b>	4.48	1.0
(0.15, 0.15, 0.15)	33.11	5.8	13.02	1.24	<b>9.14</b>	<b>1.04</b>	5.18	1.0
(0.2, 0.2, 0.2)	37.54	4.68	21.44	1.39	<b>12.92</b>	<b>1.07</b>	5.66	1.0
(0.25, 0.25, 0.25)	37.71	3.9	25.44	1.6	<b>16.36</b>	<b>1.1</b>	6.34	0.99
(0.3, 0.3, 0.3)	42.9	3.6	32.08	1.74	<b>21.05</b>	<b>1.16</b>	6.98	0.99
(0.35, 0.35, 0.35)	44.5	3.2	35.33	1.82	<b>22.77</b>	<b>1.18</b>	7.89	0.99
(0.15, 0.15, 0.3)	46.0	2.36	36.07	1.42	<b>19.95</b>	<b>1.07</b>	7.32	1.0
(0.3, 0.3, 0.15)	26.68	5.87	9.68	1.25	<b>6.88</b>	<b>1.0</b>	5.28	1.0
City10k								
	Odometry+GN		Sp. tree+GN		MASAT+GN		G. truth+GN	
noise( $\sigma_x, \sigma_y, \sigma_\theta$ )	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$
(0.1, 0.1, 0.1)	—	—	8.75	1.01	<b>6.16</b>	<b>1.0</b>	4.0	1.0
(0.15, 0.15, 0.15)	—	—	18.16	1.15	<b>7.8</b>	<b>1.0</b>	4.1	1.0
(0.2, 0.2, 0.2)	30.0	7.66	32.5	1.24	<b>11.83</b>	<b>1.01</b>	5.02	1.0
(0.25, 0.25, 0.25)	—	—	26.5	1.29	<b>12.45</b>	<b>1.02</b>	5.02	1.0
(0.3, 0.3, 0.3)	—	—	45.0	1.46	<b>23.94</b>	<b>1.05</b>	5.81	0.99
(0.35, 0.35, 0.35)	—	—	—	—	<b>29.33</b>	<b>1.09</b>	6.73	0.99
(0.15, 0.15, 0.3)	—	—	—	—	<b>23.81</b>	<b>1.01</b>	6.05	0.99
(0.3, 0.3, 0.15)	39.8	16.29	13.92	1.17	<b>6.68</b>	<b>1.0</b>	4.1	1.0
Manhattan10000								
	Odometry+GN		Sp. tree+GN		MASAT+GN		G. truth+GN	
noise( $\sigma_x, \sigma_y, \sigma_\theta$ )	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$	Avg.It	Avg. $\chi_\nu^2$
(0.1, 0.1, 0.1)	31.09	1.49	11.79	1.01	<b>5.36</b>	<b>1.0</b>	4.42	1.0
(0.15, 0.15, 0.15)	33.8	1.27	19.51	1.02	<b>6.81</b>	<b>1.0</b>	4.8	1.0
(0.2, 0.2, 0.2)	41.0	1.22	28.35	1.04	<b>9.52</b>	<b>1.0</b>	5.2	1.0
(0.25, 0.25, 0.25)	47.0	1.15	24.44	1.03	<b>11.29</b>	<b>1.0</b>	5.46	1.0
(0.3, 0.3, 0.3)	—	—	34.6	1.04	<b>15.66</b>	<b>1.0</b>	6.7	1.0
(0.35, 0.35, 0.35)	—	—	45.0	1.05	<b>19.23</b>	<b>1.0</b>	7.17	1.0
(0.15, 0.15, 0.3)	—	—	48.0	1.03	<b>19.5</b>	<b>1.0</b>	6.02	1.0
(0.3, 0.3, 0.15)	36.06	1.41	14.83	1.02	<b>5.98</b>	<b>1.0</b>	4.82	1.0

2. táblázat. A sikeres kísérletek futási eredményei a Manhattan3500, City10k, és a Manhattan10000 adathalmazon, a zaj függvényében. Avg. it: az átlagos iteráció szám, Avg.  $\chi_\nu^2$ : az normalizált  $\chi^2$  hibák átlaga.

## Köszönetnyilvánítás

A cikkben prezentált munka a 120499-es számú OTKA pályázat keretein belül készült el.

## Hivatkozások

1. G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM", *IEEE Intelligent Transportation Systems Magazine*, 2(4):31-43, 2010.
2. C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry", *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
3. J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM", in *European Conference on Computer Vision (ECCV)*, 2014.
4. R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós "ORB-SLAM: A Versatile and Accurate Monocular SLAM System", *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.
5. R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, "g2o: A General Framework for Graph Optimization", *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
6. J. Nocedal, and S. Wright. "Numerical Optimization" Springer Science & Business Media, 2006.
7. K. Levenberg. "A method for the solution of certain non-linear problems in least squares" *Quarterly of Applied Mathematics* 2.2, 1944.
8. E. Dijkstra. "A note on two problems in connection with graphs" *Numerische Mathematik* Vol. 1., 269-271, 1959.
9. E. Moore. "The shortest path through a maze" *Bell Telephon System*, 1959.
10. L. Carlone, R. Aragues, J. Castellanos, and B. Bona, "A linear approximation for graph-based simultaneous localization and mapping" in *Proceedings of Robotics: Science and Systems*, 2011.
11. G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Proceedings of Robotics: Science and Systems*, 2007.
12. Hu, Gibson, Kasra Khosoussi, and Shoudong Huang. "Towards a reliable SLAM back-end." *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013.
13. E. Olson. "Robust and efficient robotic mapping" Ph. D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA, 2008.