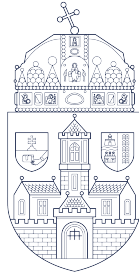


# Óbudai Egyetem

Doktori (PhD) értekezés  
tézisfüzete



Mobilrobot környezetének valósídejű térképezése

Somlyai László

Témavezető:

Dr. habil. Vámosy Zoltán

Alkalmazott Informatikai és Alkalmazott Matematikai  
Doktori Iskola

Budapest, 2024. szeptember

# Tartalomjegyzék

1	A kutatás előzményei	2
2	Célkitűzések	7
3	Vizsgálati módszerek	8
4	Új tudományos eredmények	14
5	Az eredmények hasznosítási lehetősége	27
6	Irodalmi hivatkozások listája	29

# 1 A kutatás előzményei

Napjainkban egyre elterjedtebbek az autonóm mobil robotok és az élet egyre több területén találkozhatunk velük. A feladatuktól függően lehet hogy ismerniük kell a környezetüket, észlelniük kell a felbukkanó akadályokat, meg kell tudniuk határozni a helyzetüket egy adott területen. Az egyes kutatások a robotok irányításához különféle típusú szenzorokat használnak fel, de a mobil robotok jelentős piaci térhódítása miatt már iparági szinten elterjedt szenzoros megoldásokról is beszélhetünk (Noda és Aizawa 2020), (Manikandababu és tsai. 2024).

A környezet érzékelésére az egyik elterjedt kutatási vonal a a gépi látás alapú technika. A jármű számára nem biztos, hogy rendelkezésre áll a környezetről térkép, így menet közben kell azt elkészíteni a bejárt területről szerzett információk alapján. Amikor a navigáció során a környezet megismerése és a saját helyzet meghatározása egyszerre történik, azt a navigációs eljárást nevezik Szimultán Lokalizáció és Térképmeghatározásnak, vagy a szakirodalomban alkalmazott rövidítéssel: SLAM rendszernek. A SLAM technika (Scaramuzza és Fraundorfer 2011b) célja, hogy a robot helyzetének és orientációjának folyamatos meghatározása mellett a környezetét is fel tudjuk térképezni.

A SLAM alapvető problémája az egyes mérésekből eredő bizonytalanságok kezelése, amelyek legfőképpen a szenzorok mérési pontatlanságának, a zajoknak és/vagy a technikai korlátoknak köszönhetők. Széles körben alkalmaznak különböző valószínűségi-modelleket ezeknek a bizonytalanságoknak a megfelelő közelítésére

és a hatékony becslésekhez. A folyamat általánosságban véve olyan adatokkal operál, amelyek szenzoros (például LIDAR) mérésekből adódnak [10]. A kutatásom során alkalmazott robot esetén a választott szenzor egy RGB-D kamera, amely színinformáció mellett az egyes képpontok, képsíktól mért távolsági adatát is szolgáltatja. A kutatás alapját képző RGBD-alapú kamerának köszönhetően, nem csak egyszerűen SLAM-ról beszélhetünk, hanem Visual Simultaneous Localization and Mapping-ről (VSLAM-ról).

A beltéri navigáció során nem lehet használni olyan abszolút helyzetmeghatározásra szolgáló rendszereket, mint például a GPS rendszer, ami nagymértékben megkönnyíti a nagyobb területen való helyzetmeghatározást. Azonban kiépíthető beltérben is használható abszolút helyzetmeghatározó rendszer, de ehhez külső szenzorok, vagy passzív referencia jelölők elhelyezése szükséges. A kutatásom elsősorban a beltéri helyzetmeghatározásra használható módszereket vizsgálja és mutat be egy saját, beltérben használható SLAM rendszert, aminek nincs szüksége külső szenzorokra, jeladókra. Az elmúlt időben egyre több gazdaságos és kis méretű szenzor érhető el gépi látást használó rendszerekhez, például a Xtion (Asus 2024), Intel RealSense (Intel 2024) és Kinect (Microsoft 2012) szenzor. Az egyszerűbb kivitelük ellenére elegendő pontossággal rendelkeznek egy kisebb méretű robot környezetének érzékeléséhez. A jármű környezetéről egy részletgazdag 3D térkép készíthető segítségükkel. Az RGB-D kamerákat, akárcsak a sztereókamerákat vagy a LIDAR-t igen sok kutatás használja, mint elsődleges szenzort.

Másfél évtizede foglalkozom mobil robotok fejlesztésével. Tevékenységem mindig kétirányú volt, megterveztem és elkészítettem a rendszerekhez szükséges elektronikai terveket, áramköröket, illetve kifejlesztettem nemcsak az ezek használatához szükséges programokat,

hanem magas szintű navigáló alkalmazásokat is terveztem és megvalósítottam. Az első ilyen mobilrobot 2008-ban a Magyarok a Marson című versenyen II. helyezést ért el. Továbbfejlesztett változata a 2009-es Országos Tudományos Diákköri Konferencia, Műszaki tudományok szekciójában első helyezésben részesült ("Mavridisz Vaszilizs, Somlyai László, Gál Béla: Lézerszkennerrel támogatott körbelátórendszer önjáró roboton"). Fő szerepem a hardvert működtető elektronika megtervezésében és az azt vezérlő firmware elkészítése volt.

A robotos rendszerünket lényegesen továbbfejlesztve mutattuk be a 2011-es OTDK Informatikatudományi szekcióban, ahol dolgozatunkat (Csaba György, Somlyai László: Gépi látáson alapuló akadályelkerülés) II. helyezéssel értékelték.

A rendszerünket egy távirányítós autóra építettük fel. Az autó eredeti vezérlő rendszere helyett két elektronikai kártyát terveztem, aminek a segítségével az autó számítógépről irányítható volt. Ez egy CAN bridge nevű kártya és egy motormeghajtó modul volt.

Az OTDK-n elért eredmény mellett a rendszerünket két magyar nyelvű előadáson [8], [7] és egy nemzetközi konferencián [9] is bemutatottuk.

A kisméretű robotok navigációja és vezérlése mellett dolgoztam számos másik projektben, így például egy kisméretű GPS által vezérelt robotrepülő kommunikációs protokolljának kifejlesztésében [14]. Terveztem merevszárnyú robotrepülőt vezérlő elektronikákat és készítettem egy antenaforgató rendszert robotrepülőkhez, ahol a GPS rendszer pontosságával megismerkedtem. Majd készítettem egy légköri paramétereket mérő adatgyűjtő rendszert [11], amit kis méretű robotrepülőn, vagy akár robotautón elhelyezhető.

Később a kutatási tevékenységem fokozatosan a navigáció területére és a navigációhoz szükséges adatok gyűjtésére irányult.

Így összehasonlítottuk a saját strukturált megvilágítást használó érzékelőnket, amit korábban fejlesztettünk ki, a Kinect szenzorral. Itt vizsgáltam a Kinect szenzor előnyeit és pontosságát [6].

A robotautó mozgásbecsléséhez később áttértem a Kinect szenzor használatára, hiszen ezzel nagyobb pontosság és részletgazdagabb térkép építhető fel. Ezért a korábban kifejlesztett lézershírnél felváltottam egy RGB-D szenzorral. A szenzort kalibráltam, majd elhelyeztem a saját robotautómra [12]. Az összeépített rendszer segítségével az Egyetem laborjaiban készítettem számos off-line adathalmazt a Kinect szenzorral, amit később az algoritmusaim teszteléséhez tudtam felhasználni.

Kezdeti próbálkozásként a szenzor mozgásának becslésére egy SVD felbontáson alapuló iteratív algoritmussal kísérleteztem. Néhány kiválasztott jellemző pont pár segítségével az SVD felbontást használva határoztam meg két egymás utáni pozíció közötti relatív elmozdulást [13]. Ennél az algoritmusnál a megvalósított rendszer még nagy futási idővel, és viszonylag nagy pontatlansággal is rendelkezett.

Az eredeti SVD felbontást használó algoritmusnak a továbbfejlesztésével jelentős javulást értem el a futási idő és pontosság tekintetében is. Ezt a munkámat ISVD algoritmusnak neveztem el és egy konferencián [2] és egy folyóiratban [3] is publikáltam.

A kezdeti keretrendszemet továbbfejlesztve elkészítettem egy alkalmazást, az ISVD algoritmusom teszteléséhez [1]. Ebben a keretrendszerben a saját laboratóriumi adathalmazokon kívül, két külső adathalmazt is vizsgáltam (TUM (J. Sturm és tsai. 2012), POZNAN (Schmidt és tsai. 2013)). A keretrendszer segítségével összehasonlítottam a saját munkám pontosságát más kutatások pontosságával.

A rendszer egy későbbi változata pontosabb becslést ad az elmozdulásra és a működése is robusztusabb a korábbi rendszernél, a továbbfejlesztett illesztési eljárásnak és a valós idejű Loop-closure

algoritmusnak köszönhetően [4]. Ennél a rendszernél nem egy korábbi képhez történt az illesztés, hanem állítható számú korábbi képhez illesztettem az utolsó mérést.

Végül a jelenlegi rendszerem pontosságát a több korábbi képnek már nem csak átlagolt, hanem súlyozott illesztésével értem el. Ezt az algoritmust ISLAM-nak neveztem el [5]. Itt vizsgáltam a saját rendszerem és számos más rendszer közötti pontosságot 20 online elérhető adathalmazt felhasználva. Vizsgáltam a rendszerem működőképességét statikus és dinamikus környezetben is, illetve a korábban kifejlesztett DIF szűrőmet is teszteltem.

## 2 Célkitűzések

A kutatás célja, egy mobil robotok navigációját segítő és a környezet háromdimenziós rekonstrukcióját létrehozó SLAM rendszer fejlesztése. A kutatás során megvizsgáltam a hasonló rendszereket és ezekből levont tapasztalatok alapján készítem el a saját SLAM megoldásomat. Az elkészült rendszert tesztelés során hasonlítom össze más munkák pontosságával.

Kutatási céljaim:

- Kifejleszteni egy SLAM eljárást, ami egy robotjármű mozgásbecslését, valamint ezzel egyidőben a munkakörnyezet modelljét megadja kizárólag egy RGB-D kamera képeit felhasználva.
- A Loop Closure probléma detektálása, az akkumulálódott hibák nagyságának csökkentésére.
- A hasonló robosztus rendszerekhez mérhető futási idő elérése, minimum 10Hz-es futási sebességgel.
- A hasonló rendszerek pontosságának megközelítése és túlszárnyalása, a szakirodalomban szokásos ATE (Absolute Trajectory Error) metrikával a publikusan elérhető adathalmazokon vizsgálva.



## 3 Vizsgálati módszerek

A kutatások irányát meghatározza, hogy milyen jellemzőpont detektort használnak, mivel azok a különböző kép-transzformációkra más és más módon reagálnak. Az egyszerűbb jellemződetektorok (FAST (Rosten és Drummond 2006)), futási ideje jóval kisebb, mint a bonyolultabb detektoroké (SIFT (Lowe 2004), PCASIFT és SURF (Bay és tsai. 2008)), de az egyes térbeli transzformációkra kevésbé invariánsak (A. S. Huang és tsai. 2017). A SIFT detektor adja az egyik legjobb eredményt több transzformációra és az esetleges képi elmosódásokra (Juan és Gwun 2009). A SIFT hátránya a mobil robotok kérdéskörében a magas feldolgozási idő. A valós idejű térképépítéshez elengedhetetlen feltétel a gyors jellemzőpont kereső algoritmus kiválasztása. Több cikkben is foglalkoznak az egyes detektorok tesztelésével, én is összehasonlítottam több detektort is, így például az ORB (Rublee és tsai. 2011), SIFT, SURF és FAST módszereket [3]. A robusztus működés és a gyorsaság alapján választottam közülük a vizsgálatok során.

A jellemzőpontok meghatározása mellett a megoldás kulcsképe, hogy az egymás után készített képeken meghatározott jellemzőpontokat, azokak térbeli pozícióját felhasználva, miként párosítjuk, azaz milyen transzformáció segítségével képezzük egymásba őket.

Az egyik tradicionális és gyakran alkalmazott technika pontfelhők illesztésére az Iterative Closest Points (ICP) algoritmus (Besl és McKay 1992), melynek fő lépései a következők: Adott két ponthalmazunk  $P$  és  $P'$ , és a  $P'$  ponthalmazhoz szeretnénk illeszteni az előző képből kinyert

$P$  ponthalmazt. A két ponthalmaz súlypontját meghatározzuk, és egy eltolás transzformációval az újabb ponthalmaz súlypontjába toljuk a korábbi.  $P$  minden pontjához megkeressük  $P'$  legközelebbi pontját, illetve  $P'$  minden pontjával is megcsináljuk ugyanezt. Ahol mindkét ponthalmazból vizsgálva ugyanaz a pontpárt kapjuk, meghatározzuk azt a transzformációt, ami a legkisebb hibával képezi  $P$ -t  $P'$ -be, majd iteráljuk a pontpárok megfeleltetését és az egymásba képzést. Az algoritmusnak számos változata van (Pomerleau, Colas és Siegwart 2015) és RGB-D szenzorok esetén is sokszor alkalmazták már (Henry és tsai. 2014; Seeger, Laboureaux és Häusler 2001).

Egy másik megoldási lehetőség SVD felbontást használ a probléma megoldására (Arun, Thomas S Huang és Steven D Blostein 1987) alapján a következő. A két ponthalmaz  $P = \{p_i\}$  és  $P' = \{p'_i\}$ ,  $i = 1, 2, \dots, N$ , ahol  $p_i$  és  $p'_i$   $3 \times 1$  méretű oszlopvektorok. A két ponthalmaz  $P$  és  $P'$  elemei között a következő kifejezés teremt kapcsolatot:

$$p'_i = Rp_i + T + N_i \quad (3.1)$$

ahol  $R$   $3 \times 3$ -as forgatómátrix,  $T$  az eltolásvektor ( $3 \times 1$ -es oszlopvektor) és  $N_i$  egy zajvektor. Feltételezzük a modellben, hogy a forgatás origón áthaladó tengely körül történik. Keressük  $R$  és  $T$  értékét, mely minimalizálja a következő kifejezést

$$E = \sum_{i=1}^N \|p'_i - (Rp_i + T)\|^2 \quad (3.2)$$

Arunék felhasználták kutatócsoportjuk korábbi, (T. Huang, S. Blostein és Margerum 1986)-ban publikált eredményét: Ha (3.2) legkisebb négyzetek módszer szerinti megoldása  $\hat{R}$  és  $\hat{T}$ , akkor a  $P = \{p'_i\}$  és  $P'' = \{p''_i\} = \{\hat{R}p_i = \hat{T}\}$  ponthalmazoknak ugyanaz a súlypontja, például

$$p' = p'', \quad (3.3)$$

ahol

$$p' = \frac{1}{N} \sum_{i=1}^N p'_i \quad (3.4)$$

$$p'' = \frac{1}{N} \sum_{i=1}^N p''_i = \hat{R}p + \hat{T} \quad (3.5)$$

$$p = \frac{1}{N} \sum_{i=1}^N p_i \quad (3.6)$$

Áttérve súlyponti koordinátákra,

$$q_i = p_i - p \quad (3.7)$$

$$q'_i = p'_i - p' \quad (3.8)$$

és a (3.2) kifejezésbe behelyettesítve, kapjuk

$$E = \sum_{i=1}^N \|p'_i - (Rp_i + T)\|^2 = \sum_{i=1}^N \|q'_i + p_i - (R(q_i + p) + T)\|^2 = \sum_{i=1}^N \|q'_i - Rq_i\|^2 \quad (3.9)$$

Ezért az eredeti legkisebb négyzetek szerinti probléma két részre redukálódik:

1. Az  $\hat{R}$  forgatómátrix meghatározása, amely minimalizálja  $E$  értékét a (3.9) kifejezésben.
2. Majd az elmozdulás értékének már nyilvánvaló kiszámítása:

$$\hat{T} = p' - \hat{R}p \quad (3.10)$$

A 1. részfeladat megoldása érdekében mindkét ponthalmaz esetében súlyponti koordinátarendszerre térünk át. Az így kapott értékekből egy kovariancia mátrixot konstruálunk, amelynek a következő lépésben elkészítjük az SVD felbontását  $H = U\Lambda V^T$ , majd kiszámoljuk  $X = VU^T$  mátrixot. Ha  $X$  sajátértéke,  $\det(x) = +1$ , akkor  $X$  a keresett forgatómátrix. Amennyiben a determináns értéke  $-1$ , akkor elfajuló esetet kell kezelni.

Umeyama (Umeyama 1991) cikkében továbbfejlesztette Arun et. al. fenti SVD megoldását. Egyrészt a transzformációt kibővítette skalárral való skálázással, azaz euklideszi transzformáció helyett általánosabb, hasonlósági transzformációt tárgyal, másrészt a pontvektorok dimenzióját sem kötötte meg 3D-re. Legfontosabb eredménye pedig, hogy mindig megadja zárt alakban a helyes transzformációs paramétereket, még akkor is, ha az adatok sérültek, zajjal terheltek.

Legyenek  $A$  és  $B$   $m \times n$  méretű mátrixok és  $R$   $m \times m$  forgatómátrix. Az  $AB^T$  SVD felbontása legyen  $UDV^T$ , ahol ( $UU^T = VV^T = I$ ,  $D = \text{diag}(d_i)$ ,  $d_1 \geq d_2 \geq \dots \geq d_m \geq 0$ ). Ekkor az  $\|A - (RB)\|^2$   $R$  forgatásra vonatkozó minimum értéke:

$$\min \|A - (RB)\|^2 = \|A\|^2 + \|B\|^2 - 2\text{tr}(DS) \quad (3.11)$$

ahol  $\text{tr}(DS)$  a mátrix főátlójában elhelyezkedő elemek összege és

$$S = \begin{cases} I & \text{ha } \det(AB^T) \geq 0 \\ \text{diag}(1, 1, \dots, 1, -1) & \text{ha } \det(AB^T) < 0 \end{cases} \quad (3.12)$$

Ha  $\text{rank}(AB^T) \geq m - 1$ , akkor az optimális  $R$  forgatómátrix, amely

a minimális értéket szolgáltatja:

$$R = USV^T \quad (3.13)$$

ahol  $S$  értékét a (3.13) kifejezésben a következő módon kell megválasztani:

$$S = \begin{cases} I & \text{ha } \det(U)\det(V)=1 \\ \text{diag}(1, 1, \dots, 1, -1) & \text{ha } \det(U)\det(V)=-1 \end{cases} \quad (3.14)$$

és ekkor  $\det(AB^T) = 0$ , valamint  $\text{rank}(AB^T) = m - 1$ . Mivel az SVD alapú módszer az irodalom alapján gyorsabb megoldást szolgáltat, ezért saját rendszeremben is később ilyen irányba léptem.

A sajátomhoz hasonló SLAM rendszerket dolgoztak ki Scaramuzza-ék (Scaramuzza és Fraundorfer 2011a), Whelan és kollégái (Whelan, Kaess és tsai. 2015) (Whelan, Salas-Moreno és tsai. 2016), Qiang csapata (Q. Liu és tsai. 2016), Endres-ék (Endres, Hess, Engelhard és tsai. 2012) (Endres, Hess, Jurgen Sturm és tsai. 2014), Stücklerék (Stuckler és Behnke 2014), Henryék csapata (Henry és tsai. 2010) (Henry és tsai. 2012), Hachiuma-ék (Hachiuma, Pirchheim és Schmalstieg 2019), Yangdong-ék (Y. Liu, Gao és hu 2019) és Rongsong-ék (Gou és tsai. 2022). Valamint ilyen rendszer a Co-fusion (Rünz és Agapito 2017) és a Static Fusion (Scona és tsai. 2018). A felsorolt munkákat vettem a kutatásom alapjául és a saját rendszeremet elsősorban ezekkel a munkákkal hasonlítottam össze minőségi jellemzőkben.

A legtöbb hasonló kutatásban a becsült elmozdulás pontosságának a megállapítására gyakran használják az Absolute Trajectory Error (ATE) mérőszámot (Jurgen Sturm és tsai. 2011). Az ATE érték megadja, hogy a teljes becsült útvonal és a valós pozíció között mekkora eltérés

tapasztalható. Az elmozdulás becslés esetén, az egyes illesztésekkor keletkezett kis hibák összeadódnak, ezért ha hiba keletkezett egy illesztésnél, az már a többin is megjelenik. Az ATE értékeket a becslt és a valós út eltérésére méterben vizsgáltam. Az egyik publikációmban [5] az ATE értékek Root Mean Square Error összegét hasonlítottam össze hasonló munkák eredményeivel, valamint a saját algoritmusom különböző paraméterei esetén is vizsgáltam az ATE értéket.

A Relative Pose Error (RPE) mérőszám (Jürgen Sturm és tsai. 2011) két egymásután készült kép közötti relatív elmozdulást vizsgálja csak. A RPE az aktuális illesztés hibáját veszi alapul. A jelenlegi és az azt megelőző valós és becslt elmozdulás különbségéből számolható.

A saját adathalmazok elemzésénél a következő módon vizsgáltam a pontosságot. A kezdő pozíció és az utolsó pozíció hasonló helyzetben készült képeket tartalmazott. Az útvonal bejárása után a teljes akkumulálódott hibát vizsgáltam.

# 4 Új tudományos eredmények

## 1. tézis: SVD alapú iteratív SLAM eljárást fejlesztettem RGB-D kamerák által szolgáltatott pontfelhők illesztésére.

Kifejlesztettem egy SVD-n alapuló iteratív illesztési eljárást (ISVD), amely az illesztésre szánt pontfelhők jellemzőpontjaiból több lépésben távolítja el a kiugró értékeket (1. algoritmus). A kidolgozott módszer minden iterációban megvizsgálja a jellemző pontpárok relatív hibáját, a nagy hibával jellemezhető pontpárokat eltávolítja az illesztendő halmazából, majd fokozatosan csökkenti a megengedett hibaértéket a következő párosítási iterációhoz. Az iterációk végén a legjobb illesztést biztosító pontpárokból kerül meghatározásra az illesztést biztosító transzformáció (becsült elfordulás és elmozdulás).

A tézist alátámasztó saját publikációk: [1], [2], [3].

### Magyarázat

A kifejlesztett algoritmus (ISVD) egy RGB-D kamera használatán alapú háromdimenziós rekonstrukciós és térképépítési eljárás. A rendszer valós időben képes megbecsülni a jármű (mobil robot) mozgását, valamint folyamatosan frissíti a navigáció során a detektált környezet térképét. A működés során csak a szenzor színes kamera képe és pixelszintű mélységi információja kerül felhasználásra.

---

**1. algoritmus** Pontfelhők illesztése SVD módszerrel iteratívan.

---

**Input:**  $D, S$  az aktuális és az azt megelőző jellemzőpontfelhő

**Input:**  $maxIt$  az iterációk maximális száma

**Input:**  $e_{start}$  kezdeti tolerálható hiba

**Input:**  $e_{min}$  elérendő hibanagyság

**Input:**  $T_0$  kezdeti becslés a transzformációra, vagy ennek hiányában egységmátrix

**Output:**  $T$  az illesztésből becsült orientációváltozás és elmozdulás homogén-koordinátás transzformációs mátrix formájában

$$T = T_0$$

$$e_n = e_{start}$$

**for**  $j = 0 \rightarrow maxIt$  **do**

$$T = SVD(D, S, T)$$

**for**  $i = 1 \rightarrow |D|$  **do**

▷  $|D|$  az illesztésben szereplő

jellemzőpontpárok száma

$$S' = T * S$$

▷  $s'_i \in S'$  és  $d_i \in D$  vizsgálandó jellemzőpontpár

**if**  $\sqrt{(d_{ix} - s'_{ix})^2 + (d_{iy} - s'_{iy})^2 + (d_{iz} - s'_{iz})^2} > e_n$  **then**

REMOVE( $d_i, s_i$ )

**if**  $e_{min} > e_n$  **then**

end algorithm

**else**

$$e_n = e_n/2$$

---



Az algoritmus olyan háromdimenziós pontfelhőkre támaszkodik, amik között átfedés található, valamint a pontfelhőkhöz kamerakép rendelhető. Az ilyen pontfelhők egymáshoz illeszthetőek, meghatározható ez alapján a szenzor relatív elmozdulása és orientációja. Az elmozdulás becsléséhez az algoritmus a környezet háromdimenziós pontfelhőiben, az egymás után vett képeken az összetartozó jellemzőpontok párjait keresi meg. A különböző időpontokban készült háromdimenziós jellemzőpontok közti transzformáció meghatározása egy SVD algoritmuson alapuló módszerrel történik. Az illesztés többlépcsős iteráció során minimalizálja az illesztési hibát és törli a hibásan detektált pontpárokat (1. algoritmus).

## Gyakorlati alkalmazás

Az algoritmusomat az Egyetemen készített saját adatsorokon teszteltem. A kapott eredményekben mind a becsült átlagos elfordulás, mind a becsült átlagos elmozdulás érték kisebbnek bizonyultak, mint ha a szakirodalomban hagyományos ICP módszert alkalmaznám (4.1. táblázat).

4.1. táblázat. Az ICP és az SVD módszerrel meghatározott hiba nagysága egy mérésnél.

Módszer	$x$ [mm]	$y$ [mm]	$z$ [mm]	$\phi$ [rad]	$\theta$ [rad]	$\psi$ [rad]	Összhiba [mm]	Összhiba [rad]
ICP	-183,3	617,4	-231,1	-0,0365	0,0019	-0,1124	343,8	0,05
SVD	-99,2	380,3	-264,7	-0,0336	0,0058	-0,0404	<b>248</b>	<b>0,026</b>

## **2. tézis: Több egymás után rögzített RGB-D pontfelhő illesztésének minőségét figyelembe vevő eljárást fejlesztettem, amely a becsült elmozdulások pontosságát javítja.**

Az SVD-n alapuló iteratív illesztési eljárást (ISVD) úgy fejlesztettem tovább, hogy a módszer több legutóbbi illesztésre szánt pontfelhő jellemzőpontjaiból határozza meg iteratívan az illesztést biztosító translációt és orientációt (2. algoritmus). Az egyes illesztések pontosságára, minőségére vonatkozóan paramétert határoztam meg, és a végső elmozdulás becslésébe ezen paraméterek súlyozásával épül be a részeredmény.

A tézist alátámasztó saját publikáció: [5].

### **Magyarázat**

A kifejlesztett algoritmus (ISLAM) egymás után készített háromdimenziós pontfelhőkre épül, és feltételezi, hogy közöttük átfedés van. A pontfelhők között illesztést biztosító transzformációt az ISVD algoritmus segítségével határozza meg a módszer több szálon párhuzamosan futtatva az egyes illesztéseket. Az illesztés két lépcsőben valósul meg. Először a két színes képen határoz meg jellemző pontokat, majd pedig ezeket illeszti úgy, hogy a kapott homográfia transzformáció a megvizsgált szituációk közül a legtöbb illeszkedő pontpárt biztosítsa. A második részben az egyes illesztések hibájából az illesztés minőségére vonatkozóan jellemzőt határoz meg módszerem, majd pedig ezekkel a súlyokkal javítja az elmozdulás becslését (2. algoritmus).

Az algoritmus jelentősen javuló pontossági értékeket szolgáltat, ami minden esetben hasonló nagyságú, mint a szakirodalmi módszerek,

---

## 2. algoritmus Többlépéses illesztési eljárás (ISLAM) pontfelhők közötti transzformáció meghatározására.

---

**Input:**  $MP$  az illesztésben résztvevő előző pontfelhők száma

**Input:**  $X_n, X_{n-1}, \dots, X_{n-MP}$  az  $n$ . és az azt megelőző színezett pontfelhők

**Input:**  $F_n, F_{n-1}, \dots, F_{n-MP}$  jellemzőpontok halmaza az  $n$ . és az azt megelőző képeken

**Input:**  $maxIt$  az iterációk maximális száma

**Input:**  $minError$  a hiba legkisebb értéke

**Input:**  $minPoint$  a pontpárok minimális száma

**Input:**  $\alpha, \beta$  az akceptálható hibát meghatározó konstansok

**Input:**  $T_0$  kezdeti becslés a transzformációra, vagy ennek hiányában egységmátrix

**Output:**  $Tw_n$  az egyes illesztésekből súlyozottan becsült elmozdulást tartalmazó transzformáció

**Output:**  $W_{n-f}$  az egyes illesztések súlya

```
1: for  $f = 1 \rightarrow MP$  do ▷ Korábbi állapotokhoz illesztés külön szálakon
2:    $D_{n-f}, S_{n-f} = match(X_n, F_n, X_{n-f}, F_{n-f})$  ▷ A színes kamerakép
   jellemző leíró vektorait felhasználva megadja a két kép összetartozó pontpárjait és ebből
   meghatározza a 3D összetartozó pontpárokat.
3:    $Tw_{n-f} = T_0$ 
4:   for  $j = 1 \rightarrow maxIt$  do
5:      $Tw_{n-f} = SVD(D_{n-f}, S_{n-f}, Tw_{n-f})$ 
6:      $th = \alpha * (j)^\beta$ 
7:
8:      $S' = Tw_{n-f} * S_{n-f}$ 
9:      $D = D_{n-f}$ 
10:     $e = 0$ ;
11:    ▷  $s'_i \in S'$  és  $d_i \in D$  vizsgálendő pontpárok
12:    for  $i = 1 \rightarrow |D|$  do ▷  $|D|$  az illesztésben szereplő pontpárok száma
13:       $e = e + |d_{ix} - s'_{ix}| + |d_{iy} - s'_{iy}| + |d_{iz} - s'_{iz}|$ 
14:       $W_{n-f} = 1/(e/|D|)$ 
15:
16:      for  $i = 1 \rightarrow |D|$  do
17:        if  $\sqrt[2]{(d_{ix} - s'_{ix})^2 + (d_{iy} - s'_{iy})^2 + (d_{iz} - s'_{iz})^2} > th$  then
18:          remove( $d_i, s_i$ )
19:      if  $th < minError$  OR  $|D| < minPoint$  then
20:        end iteration ▷ A szálak bevárása és utána az elmozdulások súlyozott kiértékelése
21:  $t = 0$ 
22:  $w = 0$ 
23: for  $f = 1 \rightarrow MP$  do
24:    $t_{n-f} = Tw_{n-f}$  mátrix 4. oszlopa ▷ az elmozdulás vektor
25:    $t = t + t_{n-f} * W_{n-f}$ 
26:    $w = w + W_{n-f}$ 
27:  $Tw_n$  4. oszlopa =  $t/w$ 
```

---

4.2. táblázat. A saját algoritmus és a hasonló rendszerek Absolut Trajectory Error (ATE RMSE) értékei, fr1, fr2 és fr3 adathalmazokon vizsgálva, statikus környezetben. Minden érték méterben van megadva.

Adathalmaz	Saját jobb	E-RGBD SLAM <sup>(1)</sup>	Whelan <sup>(2)</sup>	Liu <sup>(3)</sup>	RGBD SLAM <sup>(4)</sup>	MRSMap <sup>5</sup>
fr1/rpy	<b>0,025</b>		0,028			0,027* <sup>(2)</sup>
fr1/xyz	<b>0,013</b>	0,014	0,017	0,013	0,014* <sup>(3)</sup>	0,013* <sup>(2)</sup>
fr1/360	<b>0,056</b>	0,075				
fr1/desk	0,026	<b>0,022</b>	0,037	0,064	0,026	0,043* <sup>(2)</sup>
fr1/room	0,14		0,075		0,087	<b>0,069*</b> <sup>(2)</sup>
fr1/desk2	0,049	<b>0,034</b>	0,071			0,049* <sup>(2)</sup>
fr1/plant	0,034	0,068	0,047			<b>0,026*</b> <sup>(2)</sup>
fr2/xyz	0,0155			0,015	<b>0,008*</b> <sup>(3)</sup>	
fr2/desk lex]	0,12	0,090	<b>0,034</b>		0,057	0,052* <sup>(2)</sup>
fr2 / flowerbouquet	<b>0,1202</b>	0,137			0,131* <sup>(1)</sup>	
fr2 / metallic sphere	<b>0,7785</b>	0,914			1,099* <sup>(1)</sup>	
fr2 / pioner slam	0,36	<b>0,349</b>			0,367* <sup>(1)</sup>	
fr2/pioner slam 2	1,1754	0,4			<b>0,381*</b> <sup>(1)</sup>	
fr2/pioner slam 3	0,4558	<b>0,341</b>			0,511* <sup>(1)</sup>	
fr3/long	0,11			<b>0,028</b>	0,032* <sup>(3)</sup>	

<sup>(1)</sup> (Guclu és Can 2017)

<sup>(2)</sup> (Whelan, Kaess és tsai. 2015)

<sup>(3)</sup> (Q. Liu és tsai. 2016)

<sup>(4)</sup> (Endres, Hess, Jurgen Sturm és tsai. 2014)

<sup>(5)</sup> (Henry és tsai. 2012)

bizonyos teszthalmazoknál pedig meg is haladta azok jellemzőit (4.2. táblázat).

## Gyakorlati alkalmazás

Az algoritmusomat a TUM (Schmidt és tsai. 2013) adathalmazon teszteltem. A kapott eredményeket összehasonlítottam 20 adathalmazon, ahol voltak statikus és dinamikusan változó környezetek

is. A becsült átlagos elmozdulás érték kisebbnek bizonyultak a következő benchmark adathalmazokon: *fr1/rpy*, *fr1/xyz*, *fr1/360*, *fr2/flowerbouquet*, *fr2/metallicsphere*.

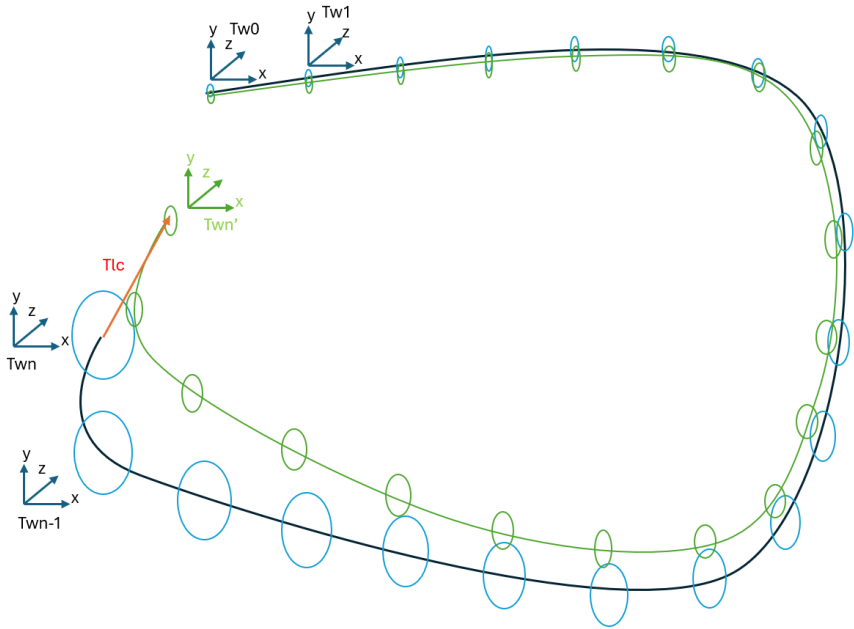
### **3. tézis: RGB-D pontfelhők szűrésével és zárt hurok (LC) detektálására kifejlesztett módszerrel teljes rendszert fejlesztettem guruló robotok SLAM problémájának megoldására.**

Rendszerem Front-end modulja saját zárthurok detektáló (LC, vagy Loop Closure) algoritmust futtat, ami folyamatosan figyeli az aktuális kép és a kulcsképkockák közti egyezéseket. Ha egyezést talál a rendszer egy korábbi kulcsképpel, akkor az aktuális pozíció pontossága tovább javítható a korábbi kulcskép pozícióját felhasználva (4.1. ábra). Amennyiben az RGBD szenzor méréseit feltételes átlagoló szűrővel javítjuk (DIF szűrő), akkor a SLAM rendszerem minősége összevethető eredményt szolgáltat több hasonló rendszer jellemzőivel (4.2. ábra).

A tézist alátámasztó saját publikációk: [4], [5].

#### **Magyarázat**

A LC eljárás az aktuális kép jellemző leíróit a korábbi kulcsképek jellemző leíróival összehasonlítva határozza meg az egyezést. Ha egyezést talál egy korábbi képpel, akkor a többlépéses illesztéssel (2. algoritmus) megpróbál relatív elmozdulást meghatározni a kulcskép és az aktuális kép között. A rendszer nem vizsgál meg minden aktuális képet, csak akkor fut le a Loop-closure algoritmus, ha az adott kép megfelelő élességgel rendelkezik. Az élesség megállapítása a korábbi 20 kép jellemző leíróinak alapján történik. Ha a legutóbbi képeken talált jellemző leíró pontok számának a mediánértékénél 1,2-szer több az aktuális kép jellemző leírók száma, akkor ez a kép kevésbé elmosódott és részletgazdagabb, mint a többi (4.3. ábra). Az LC algoritmus futási ideje jelentősen csökken, mert kis számú kulcsképet kell megvizsgálni



4.1. ábra. Zárt hurok detektálása. Feketével a becsült útvonal, pirossal a detektált eltérés nagysága és zöld színnel a LC algoritmus futtatása utáni utvonal. Az  $n$ . állapotban sikerült detektálni azt a részt, amit a szenzor a  $0$ . állapotban látott, így pontosítható a megtett út minden közbenső helyzete. Az ellipszisek az akkumulálódott hibák nagyságát szimbolizálják.

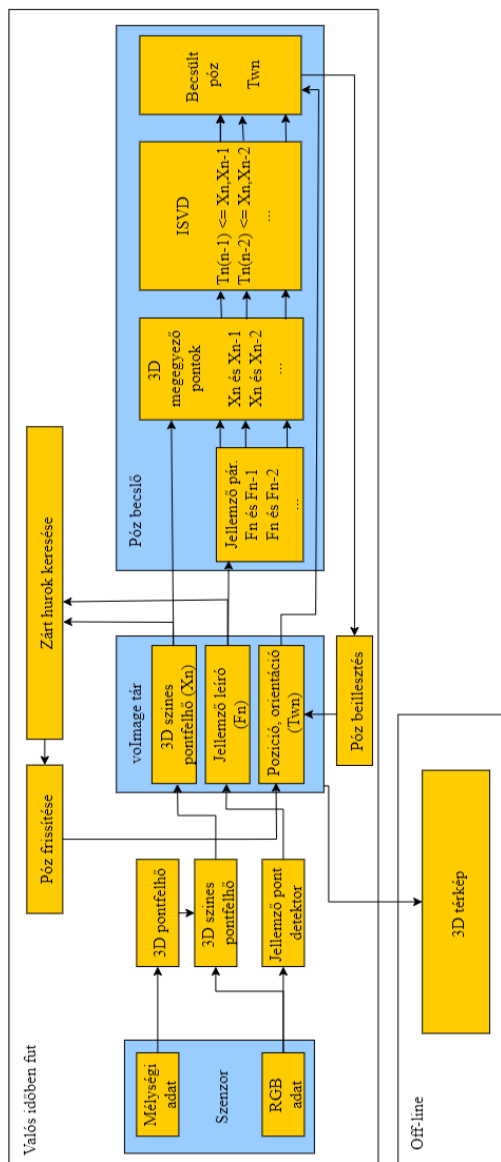
a többlépéses illesztéssel, aminek nem elhanyagolható a futási ideje a jellemző párok keresése, illetve a többlépéses illesztés miatt.

Ha a rendszert az RGBD szenzor mérészejának csökkentésére kifejlesztett feltételes átlagoló szűrőt alkalmazza (DIF), akkor guruló robotok navigációjára alkalmas SLAM rendszerem pontossága tovább javul.

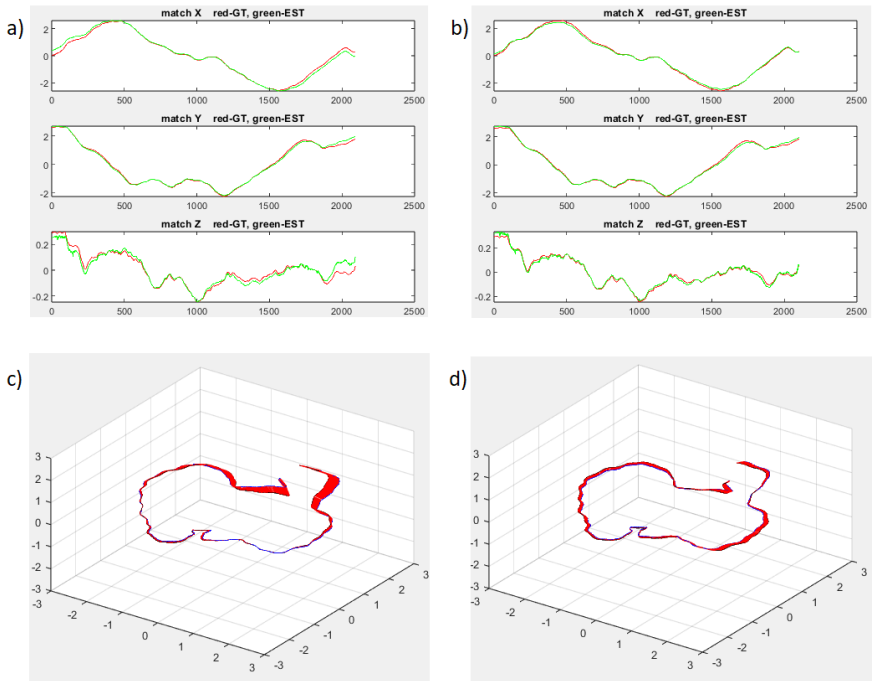
### **Gyakorlati alkalmazás**

A LC algoritmus egyetlen kivétellel tovább javította a SLAM megoldáom pontosságát. A vizsgált 22 darab adathalmazból 13 esetben pontosabb lett az eredmény, amikor a DIF szűrővel és a zárt hurok detektáló eljárással együtt teszteltem SLAM rendszerem (4.3. táblázat).





4.2. ábra. SLAM rendszer felépítése



4.3. ábra. Loop Closure algoritmus eredménye. Az *FB3/long\_office\_household* adathalmazon ISVD algoritmus  $MP=4$  és DIF szűrő mellett a 2100. képkockáig, ahol az első nagy zárt hurok itt található. Az a) képen a LC előtti, a b) képen az LC utáni eredmény látható, piros színnel a valós, zöld színnel a becült x, y és z koordináta irányban a történt elmozdulás (méterben) a képkockák függvényében (vízszintes tengely). Valamint a c) képen a LC előtti, a d) képen az LC utáni eredmény látható, feketével a valós, kékkel a becült útvonal és pirossal az abszolút hiba látható. Az a) és b) képen az y tengelyen méterben, az x tengelyen a képkocka sorszámja látható. A c) és d) képen a tengelyek méterben vannak megadva.

4.3. táblázat. ISVD algoritmus pontosságának vizsgálata. A táblázatban az ATE értékek méterben vannak megadva. Vizsgáltam az ISVD illesztési algoritmust önmagában és Loop Closure algoritmussal illetve DIF szűrővel kiegészítve is. Az algoritmus futtatásakor az MP paraméter 4 volt minden esetben. A legjobb eredményeket félkövérrel jelöltem a táblázatban.

Adathalmaz	ISVD	ISVD+LC	ISVD+LC+DIF
fr1/rpy	0,0466	<b>0,026</b>	0,032
fr1/xyz	0,0232	<b>0,014</b>	0,0148
fr1/360	0,1028	<b>0,0683</b>	0,0757
fr1/desk	0,0455	0,03	<b>0,0298</b>
fr1/room	0,2727	0,17551	<b>0,1403</b>
fr1/desk2	0,0847	<b>0,0534</b>	0,0535
fr1/plant	0,0517	<b>0,032</b>	0,0338
fr2/xyz	0,0776	0,0506	<b>0,0155</b>
fr2/desk	0,3431	0,217	<b>0,1039</b>
fr2 pioner 360	0,5301	0,4069	<b>0,3877</b>
fr2/pioner slam	0,5569	0,4421	<b>0,3571</b>
fr2/pioner slam 2	1,2	<b>1,1754</b>	1,311
fr2/pioner slam 3	0,471	<b>0,4558</b>	0,9211
fr1 teddy	0,1414	0,1432	<b>0,1383</b>
fr3 long	0,4502	0,2267	<b>0,11</b>
fr2/flowerbouquet	0,1688	0,1682	<b>0,1202</b>
fr2/metallic_sphere	0,7879	<b>0,7785</b>	0,8273
fr3/walking static	0,0391	0,0268	<b>0,0252</b>
fr3 walking xyz	0,2696	0,1184	<b>0,09</b>
fr3/walking halfsphere	0,4286	<b>0,1044</b>	0,1063
fr3/sitting xyz	0,0716	0,0468	<b>0,0324</b>
fr3/sitting halfsphere	0,0716	0,0524	<b>0,0499</b>

## 5 Az eredmények hasznosítási lehetősége

Az általam megalkotott SLAM rendszer és a kidolgozott eljárásaim segítségével számos adathalmazon végeztem méréseket a rendszer pontosságával kapcsolatban. Megvizsgáltam több off-line elérhető adathalmaz esetében, hogy elegendő pontosságot tudok-e elérni egy kisméretű robotautó irányításához.

A mérések alapján a megalkotott eljárásaim (ISVD alapú SLAM rendszer és a Loop-closure algoritmus) alkalmazhatóak kisméretű civil, katonai, üzleti vagy oktatási célú robotrobotautók, fedélzeti autonóm irányítási rendszerek fejlesztése során. Továbbá a rekonstrukciós eljárásom segítségével, egy ismeretlen terület pontfelhője előállítható (5.1. ábra).



5.1. ábra. Az egyik saját adatsoron futtatott eredmény. Az egyetem laborjában készült 310 képsorból keletkezett 3D modell.

# 6 Irodalmi hivatkozások listája

## Referenciák

- Arun, K Somani, Thomas S Huang és Steven D Blostein (1987). „Least-squares fitting of two 3-D point sets”. *IEEE Transactions on pattern analysis and machine intelligence* 5, 698–700. old.
- Asus (2024). *Asus Xiton*. URL: <http://xtionprolive.com/asus-3d-depth-camera/asus-xtion2> (elérés dátuma 2024. 08. 01.).
- Bay, Herbert, Andreas Ess, Tinne Tuytelaars és Luc Van Gool (2008). „Speeded-up robust features (SURF)”. *Computer vision and image understanding* 110.3, 346–359. old.
- Besl, P.J. és Neil D. McKay (1992). „A method for registration of 3-D shapes”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2, 239–256. old. DOI: [10.1109/34.121791](https://doi.org/10.1109/34.121791).
- Endres, Felix, Jurgen Hess, Nikolas Engelhard, Jurgen Sturm, Daniel Cremers és Wolfram Burgard (2012. máj.). „An evaluation of the RGB-D SLAM system”. *Proceedings - IEEE International Conference on Robotics and Automation*, 1691–1696. old. DOI: [10.1109/ICRA.2012.6225199](https://doi.org/10.1109/ICRA.2012.6225199).
- Endres, Felix, Jurgen Hess, Jurgen Sturm, Daniel Cremers és Wolfram Burgard (2014. febr.). „3-D mapping with an RGB-D camera”. *Robotics, IEEE Transactions on* 30, 177–187. old. DOI: [10.1109/TR0.2013.2279412](https://doi.org/10.1109/TR0.2013.2279412).

- Gou, Rongsong, Guangzhu Chen, Chengliang Yan, Xin Pu, Yuanyuan Wu és Yuan Tang (2022. nov.). „Three-dimensional dynamic uncertainty semantic SLAM method for a production workshop”. *Engineering Applications of Artificial Intelligence* 116. DOI: [10.1016/j.engappai.2022.105325](https://doi.org/10.1016/j.engappai.2022.105325).
- Guclu, Oguzhan és Ahmet Can (2017). „Fast and effective loop closure detection to improve SLAM performance”. *J. Intell. Robot. Syst.*, 1–23. old.
- Hachiuma, R, C Pirchheim és D Schmalstieg (2019). „DetectFusion: Detecting and segmenting both known and unknown dynamic objects in real-time SLAM”. *arXiv preprint arXiv:1907.09127*.
- Henry, Peter, Michael Krainin, Evan Herbst, Xiaofeng Ren és Dieter Fox (2010). „RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments”. In *Proc. of the Intl. Symp. on Experimental Robotics (ISER)*.
- (2012. ápr.). „RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments”. *International Journal of Robotic Research - IJRR* 31, 647–663. old. DOI: [10.1177/0278364911434148](https://doi.org/10.1177/0278364911434148).
- (2014). „RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments”. *Experimental robotics*. Springer, 477–491. old.
- Huang, Albert S, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox és Nicholas Roy (2017). „Visual odometry and mapping for autonomous flight using an RGB-D camera”. *Robotics Research*. Springer, 235–252. old.
- Huang, TS, SD Blostein és EA Margerum (1986). „Least-squares estimation of motion parameters from 3-D point correspondences”.

- Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 10. köt. IEEE Computer Soc. Press Washington DC, 112–115. old.
- Intel (2024). *RealSense*. URL: <https://www.intelrealsense.com/> (elérés dátuma 2024. 08. 01.).
- Juan, Luo és Oubong Gwun (2009). „A comparison of SIFT, PCA-SIFT and SURF”. 3. köt. 4, 143–152. old.
- Liu, Qiang, Ruihao Li, Huosheng Hu és Dongbing Gu (2016. szept.). „Building semantic maps for blind people to navigate at home”. *2016 8th Computer Science and Electronic Engineering (CEECE)*, 12–17. old. DOI: [10.1109/CEECE.2016.7835881](https://doi.org/10.1109/CEECE.2016.7835881).
- Liu, Yangdong, Wei Gao és Zhanyi hu (2019. febr.). „3D Scanning of High Dynamic Scenes Using an RGB-D Sensor and an IMU on a Mobile Device”. *IEEE Access* PP. DOI: [10.1109/ACCESS.2019.2900740](https://doi.org/10.1109/ACCESS.2019.2900740).
- Lowe, David G (2004). „Distinctive image features from scale-invariant keypoints”. *International journal of computer vision* 60.2, 91–110. old.
- Manikandababu, C.S., R Vijayakrishna, R Venugopal és Y Vishnuprasad (2024). „Development of an Autonomous Solar Grass Cutting Robot with a Path Memorizing Mechanism”. *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 1–7. old. DOI: [10.1109/ACCAI61061.2024.10602185](https://doi.org/10.1109/ACCAI61061.2024.10602185).
- Microsoft (2012). „Microsoft Kinect SDK”. <http://www.microsoft.com/en-us/kinectforwindows/> (Visited: 2012. jan. 26.)
- Noda, Kazutoshi és Hidenobu Aizawa (2020). „Indoor environmental monitoring system using a robot vacuum cleaner”. *Sensors and Materials* 32.3, 1133–1140. old.
- Pomerleau, François, Francis Colas és Roland Siegwart (2015). „A Review of Point Cloud Registration Algorithms for Mobile



- Robotics”. *Foundations and Trends® in Robotics* 4.1, 1–104. old. ISSN: 1935-8253. DOI: [10.1561/23000000035](https://doi.org/10.1561/23000000035).
- Rosten, Edward és Tom Drummond (2006). „Machine learning for high-speed corner detection”. *European conference on computer vision*. Springer, 430–443. old.
- Rublee, Ethan, Vincent Rabaud, Kurt Konolige és Gary Bradski (2011). „ORB: An efficient alternative to SIFT or SURF”. *2011 International conference on computer vision*. Ieee, 2564–2571. old.
- Rünz, Martin és Lourdes Agapito (2017). „Co-Fusion: Real-time Segmentation, Tracking and Fusion of Multiple Objects”. In *IEEE International Conference on Robotics and Automation*, 4471–4478. old.
- Scaramuzza, Davide és Friedrich Fraundorfer (2011a). „Part I the first 30 years and fundamentals”. *IEEE Robotics & Automation Magazine* 18.4, 80–92. old.
- (2011b). „Visual odometry [tutorial]”. *IEEE robotics & automation magazine* 18.4, 80–92. old.
- Schmidt, Adam, Michał Fularz, Marek Kraft, Andrzej Kasinski és Michał Nowicki (2013). „An Indoor RGB-D Dataset for the Evaluation of Robot Navigation Algorithms”. *Prof. of Int. Conf. on Advances Concepts for Intelligent Vision Systems, Lecture Notes in Computer Science*, 321–329. old.
- Scona, R., M. Jaimez, Y. R. Petillot, M. Fallon és D. Cremers (2018). „StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments”. In *IEEE International Conference on Robotics and Automation*.
- Seeger, S, X Laboureux és G Häusler (2001). „An accelerated ICP-algorithm”. *Lehrstuhl für Optik, Annual Report*, 32. old.

- Stuckler, J. és S. Behnke (2014). „Multi-resolution surfel maps “ for efficient dense 3d modeling and tracking”. *Journal of Visual Communication and Image Representation*, 137–147. old.
- Sturm, J., N. Engelhard, F. Endres, W. Burgard és D. Cremers (2012). „A benchmark for the evaluation of RGB-D SLAM systems”. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 573–580. old.
- Sturm, Jürgen, Stéphane Magnenat, Nikolas Engelhard, François Pomerleau, Francis Colas, Daniel Cremers, Roland Siegwart és Wolfram Burgard (2011. jan.). „Towards a benchmark for RGB-D SLAM evaluation”. *Proc. of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at Robotics: Science and Systems Conf.(RSS)*.
- Umeyama, Shinji (1991). „Least-squares estimation of transformation parameters between two point patterns”. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13.04, 376–380. old.
- Whelan, Thomas, Michael Kaess, Hordur Johannsson, Maurice Fallon, John Leonard és John McDonald (2015). „Real-time large-scale dense RGB-D SLAM with volumetric fusion”. *The International Journal of Robotics Research* 34, 598–626. old. DOI: [10 . 1177 / 0278364914551008](https://doi.org/10.1177/0278364914551008).
- Whelan, Thomas, Renato Salas-Moreno, Ben Glocker, Andrew Davison és Stefan Leutenegger (2016. szept.). „ElasticFusion: Real-time dense SLAM and light source estimation”. *The International Journal of Robotics Research* 35, 1697–1716. old. DOI: [10.1177/ 0278364916669237](https://doi.org/10.1177/0278364916669237).

## Saját publikációk amik a tézisekre vonatkoznak

- 1 Somlyai, L., Gy. Csaba és Z. Vámosy (2018). „Benchmark system for novel 3D SLAM algorithms”. *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 131–136. old.
- 2 Somlyai, L. és Z. Vámosy (2015). „SLAM algorithm for mobile robot localization with RGB-D camera”. *Fluids, Heat and Mass Transfer, Mechanical and Civil Engineering: Proceedings of the 11th International Conference on Fluid Mechanics (FLUIDS '15)*, 89–94. old.
- 3 — (2016). „Vision-based mobile robot localisation and mapping system with Kinect”. *International Journal of Systems Applications - Engineering and Development* 10, 241–246. old. ISSN: 2074-1308.
- 4 Somlyai, László és Zoltán Vámosy (2022). „ISVD-Based Advanced Simultaneous Localization and Mapping (SLAM) Algorithm for Mobile Robots”. *MACHINES, IF: 2.6 Q2* 10, 519. old. DOI: [10.3390/machines10070519](https://doi.org/10.3390/machines10070519).
- 5 — (2024). „Improved RGB-D Camera-based SLAM System for Mobil Robots”. *Acta Polytechnica Hungarica, IF: Q2* 21.8, 107–124. old. DOI: [10.12700/APH.21.8.2024.8.6](https://doi.org/10.12700/APH.21.8.2024.8.6).

## Saját publikációk, amik nem vonatkoznak a tézisekre

- 6 Csaba, G., L. Somlyai és Z. Vámosy (2012). „Differences between Kinect and structured lighting sensor in robot navigation”. *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 85–90. old.

- 7 Csaba, György, László Somlyai és Zoltán Vámosy (2011). „Mobil robot navigációja strukturális megvilágítás valamint Kinect szenzor használatával”. *Informatika a felsőoktatásban 2011 konferencia Debrecen, Magyarország*, 835–842. old.
- 8 Csaba, György, László Somlyai és Zoltán Vámosy (2011a). „Mobil robot fejlesztése strukturált megvilágításon alapuló szenzorok felhasználásával”. *RobotNap 2011 – Robotika Szakosztály Évkönyve, CD ISSN: 2060-5943, Magyarország*.
- 9 — (2011b). „Mobile robot navigation in unknown environment using structured light”. *LINDI 2011 - 3rd IEEE International Symposium on Logistics and Industrial Informatics, Proceedings*, 249–254. old. DOI: [10.1109/LINDI.2011.6031158](https://doi.org/10.1109/LINDI.2011.6031158).
- 10 Csaba, György, László Somlyai és Zoltán Imre Vámosy (2018). „Mobil Robot Navigation Using 2D LIDAR”. *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI): Dedicated to the Memory of Pioneer of Robotics Antal (Tony) K. Bejczy*, 143–148. old. DOI: [10.1109/SAMI.2018.8324002](https://doi.org/10.1109/SAMI.2018.8324002).
- 11 Somlyai, L., A. Turóczy és A. Molnár (2012). „Atmospheric Analyser for Mobile Robots”. *13th IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2012)*, 181–185. old.
- 12 Somlyai, L. és Z. Vámosy (2012). „Map Building with RGB-D Camera for Mobil Robot”. *IEEE 16th International Conference on Intelligent Engineering Systems 2012 (INES 2012)*, 489–493. old.
- 13 Somlyai, László (2013). „Mobil robot localization using RGB-D camera”. *2013 IEEE 9th International Conference on Computational Cybernetics (ICCC)*, 131–136. old. DOI: [10.1109/ICCCyb.2013.6617575](https://doi.org/10.1109/ICCCyb.2013.6617575).

- 14 Stojcsics, D. és L. Somlyai (2010). „Improvement methods of short range and low bandwidth communication for small range UAVs”. *IEEE 8th International Symposium on Intelligent Systems and Informatics*, 93–97. old. DOI: [10.1109/SISY.2010.5647224](https://doi.org/10.1109/SISY.2010.5647224).