

Multimodális pontfelhőregisztráció Hough tér alapú előillesztéssel

Nagy Balázs^{1,2}, Gálai Bence¹ és Benedek Csaba^{1,2}

¹ Gépi Érzékelés Kutatólaboratórium Magyar Tudományos Akadémia, Számítástechnikai és Automatizálási Kutatóintézet

Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar

² {vezetéknév.keresztnév}@sztaki.mta.hu

Absztrakt. Cikkünkben egy általános, multimodális pontfelhőregisztrációs módszert ismertetünk. Az eljárás robusztusan működik még az annyira különböző sűrűségű és karakterisztikájú pontfelhők esetén is, mint az önjáró járművek által készített ritka instant (I3D) és a mobil térképező rendszerek (MMS) által előállított sűrű és szabályos pontfelhők. Az algoritmus főbb lépései egy robusztus szegmentációs és objektumdetekciós eljárás, egy Hough térbeli objektum alapú transzformáció számítás és egy pontszintű regisztráció finomítás. A módszer kiértékelése során a Velodyne HDL64-E és VLP16 I3D szenzorok továbbá a Riegl VMX450 MMS által előállított pontfelhőket használtuk fel. Valós alkalmazási példák lehetnek önjáró járművek helymeghatározása az I3D és MMS adatok regisztrációjával vagy IMU mentes SLAM és változás detekció az I3D adatok alapján.

1. Bevezetés

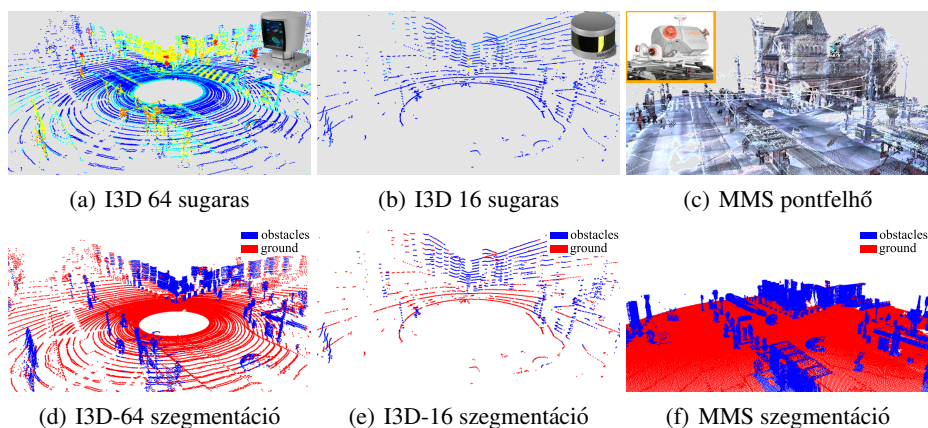
¹A járművekre rögzíthető mobil Lidar szenzorok egyre nagyobb szerepet játszanak számos alkalmazásban, annak köszönhetően, hogy nagy sebesség mellett a teljes környezetükről nagyon pontos 3D adatokat képesek szolgáltatni. Az automatikus pontfelhő regisztráció kulcsfontosságú lépés többek között az Szimultán Lokalizáció és Térképezés (SLAM) feladatában vagy éppen a különböző valós idejű terméggfigyelési alkalmazásoknál, ahol a GPS jel gyengeségéből adódóan nem mindig érhető el pontos pozíció információ. Azonban az egyes szenzorok az eltérő tervezési technológiák és a különböző felhasználási területek miatt nagyon különböző karakterisztikájú pontfelhőket állíthatnak elő [2], korlátozva ezzel az általános felhasználhatóságukat az egyes szabványos pontfelhőregisztrációs technikáknak [3, 4] vagy a specifikus szenzorra fejlesztett algoritmusoknak [5].

Az önjáró járművek (AV) navigálása és kontrollálása valós idejű adatgyűjtést és feldolgozást követel meg, hiszen mindig az adott környezet aktuális állapotától függően kell dönteniük. Ezért a pontfelhő időkereteket legalább 10-15 fps sebességgel kell rögzíteni, továbbá az egyes adatsomagok méretének korlátot szab az adott platformon

¹ A bemutatott módszer eredetileg az International Conference on Pattern Recognition 2016 nemzetközi konferencián, angol nyelven került közlésre [1].

valós időben feldolgozható adat mennyisége. Következésképpen az egyes I3D Lidar szenzorok által rögzített adatok általában alacsony sűrűségűek, mely a szenzortól való távolság függvényében gyorsan csökken, továbbá a pontfelhőben egyedi mintázatok jelenhetnek meg a szenzor karakterisztikájától függően, mint például a körkörös, gyűrűs mintázat a Velodyne szenzorok esetében (1(a) és 1(b) ábrák). Habár a 3D mérések nagyon pontosak (2cm-nél kisebb hiba) a szenzor lokális koordinárendszerében, a jármű globális pozícionálásában – a GPS jel vételének korlátjaiból adódóan – több méteres hiba is lehet a városi környezetben.

A városi útfelügyelet által használt újgenerációs geo-információs rendszerek rendkívül részletes 3D pontfelhő alapú város modelleket használnak az úthálózat karbantartásához, felügyeletéhez, és a különböző várostervező alkalmazásokhoz. A jelenlegi mobil térképező rendszerek (MMS), mint például a Riegl VMX450 (1(c) ábra) sűrű és nagyon pontos pontfelhőt állítanak elő homogén objektum felszínekkkel, a pontsűrűség közel lineáris változása mellett a szenzortól való távolság függvényében [2]. Az MMS adatai akár két-három nagyságrenddel is sűrűbbek lehetnek, mint az I3D típusú pontfelhők.



1. ábra: Felső sor: Három különböző Lidar szenzor által rögzített pontfelhő: Fővám tér, Budapest. Alsó sor: A 3.1. fejezetben bemutatott szegmentációs módszer eredménye.

A cikkben egy új, általános pontfelhőregisztrációs eljárást ajánlunk mobil lézerszkennelés alkalmazásaihoz. A fejlesztés során a hangsúlyt olyan jelenetekre helyeztük, ahol vagy mindkét pontfelhő ritkás a szenzor karakterisztikájából következően (I3D), vagy az egyik pontfelhőt az I3D a másikat pedig az MMS szenzor rögzítette, melyek így lényegesen eltérő karakterisztikát mutatnak.

1. *Multimodális pontfelhőregisztráció (I3D és MMS):* Az önjáró járműveknek nagy segítséget nyújthat a helymeghatározásban, az útvonal tervezésben és a magasabb szintű környezetértelmezésben egy pontos és részletes 3D térkép, amely globális,

georeferált koordinátarendszerben adott. Ehhez szükséges a ritkás I3D és a sűrű MMS pontfelhők gyors és precíz egymáshoz illesztése.

2. *IMU mentes SLAM ritkás I3D adatokon:* Megbízható külső navigációs (GPS) és IMU rendszer nélkül a SLAM algoritmus számos nehézségbe ütközik az I3D szenzor ritkás, inhomogén adatain.
3. *Különböző I3D szenzorok pontfelhőinek regisztrációja:* Az egyes önjáró járművek különböző típusú I3D szenzorokkal operálhatnak, így a multimodális adatok egymáshoz regisztrálása számos érdekes lehetőséget nyújt, mint például változás detekció, dinamikus környezet megfigyelés vagy helyzetmeghatározás az egyes járművek között. Amellett, hogy mind a két pontfelhő ritkás, további kihívást jelent, hogy a különböző típusú szenzorok nagyon eltérő pontsűrűség karakterisztikát mutathatnak a távolság függvényében.

Az algoritmus kiértékeléséhez három különböző szenzort használtunk: két I3D szenzor adatait (Velodyne HDL64-E 1(a) ábra és VLP16 1(b) ábra) és egy Riegl VMX450 (1(c) ábra) MMS szenzor pontfelhőit, melyeket valós városi utcai környezetben rögzítettünk. Ahogy az 1(a) és (b) ábra is demonstrálja, a HDL64-E és VLP16 mérések hasonló gyűrűs mintázatot mutatnak, azonban a VLP16 pontfelhője sokkal kevesebb pontot tartalmaz, hiszen csak 16 lézer diódát használ szemben a 64 lézeres HDL64-E szenzorral. Továbbá megfigyelhető, hogy a két szenzornak eltérő a vertikális felbontása is.

2. Szakirodalomban fellelhető módszerek

A szakirodalomban számos hatékony pontfelhőregisztrációs algoritmus található, többek között az Iterative Closest Point (ICP) [3] és a Normal Distribution Transform (NDT) [4], azonban ezek a módszerek legtöbbször nem működnek, ha a két pontfelhő kezdeti távolsága nagy, továbbá az eltérő pontfelhő karakterisztikák nagyon befolyásolhatják az illesztés eredményét [5]. Számos továbbfejlesztését publikálták az ICP algoritmusnak, amelyek vagy a szín információ felhasználásával [6] vagy a pont szomszédságból kinyert geometriai jellemzők segítségével [7] javítják tovább a módszert. Azonban ezek a megközelítések még tovább növelik az eredetileg is nagy számítási időt. A szakirodalomban fellelhetők olyan módszerek is, melyek nem módosítják az ICP alap működését, viszont trajektória információt használnak az ICP hibáinak korrekciójára [8]. [9] egy továbbfejlesztett ICP algoritmust ajánl, amely egy pontfelhő szegmentációs lépés után [10] a legközelebbi szomszéd keresést már csak a megfelelő szegmensek között végzi. Nem ICP alapú megoldások is találhatóak a szakirodalomban pontfelhőregisztrációra, mint például a [11], ahol kihasználják a többsugaras forgó szenzorok tulajdonságait (például Velodyne szenzor) a síkok detektálásához, majd valós időben regisztrálják a kinyert szegmenseket. Habár ez a megoldás valós idejű SLAM feldolgozást tesz lehetővé, azonban a legtöbb komplex, valós környezetben a pontos síkdetektálás feltétele nagy hátránya a módszereknek.

3. A javasolt pontfelhőregisztrációs algoritmus

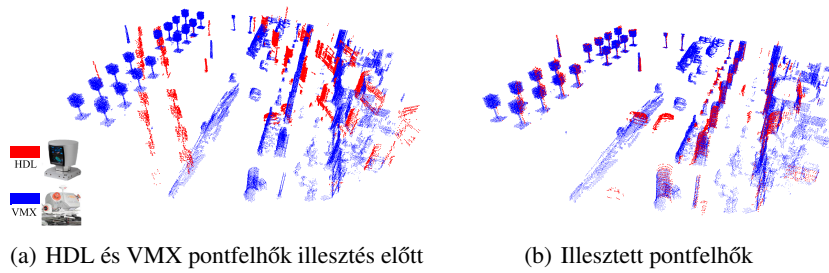
A következőkben bemutatunk egy új regisztrációs algoritmust, amely azonos karakterisztikájú pontfelhő párok mellett, akár egy ritkás I3D és egy sűrű MMS szenzortól származó pontfelhő egyeztetésénél is pontos eredményt garantál. A módszer azzal a feltételezéssel él, hogy a translációs különbség a két pontfelhő között 10 méter alatt van, köszönhetően a jelenlegi GPS technológia pontosságának, azonban az orientációs különbség továbbra is tetszőlegesen nagy lehet, ahogy az 2 i ábra is demonstrálja. Az algoritmusunk a következő négy fő lépésből áll: pontfelhő szegmentáció, absztrakt objektum detekció, objektum alapú előtranszformáció és egy pontos pont alapú regisztráció finomítás.

3.1. Pontfelhő szegmentáció

A többsugaras forgószenzoros Lidar szenzorok pontfelhőiben, főleg a talaj régióban megfigyelhető egy gyűrűs mintázat (1(a) és 1(b) ábrák), mely nagy kihívást jelent a hagyományos pontfelhőregisztrációs technikáknak. Megfigyeléseink egyrészt [5] azt mutatják, hogy az ICP és az NDT algoritmusok gyakran helytelenül illesztik egymáshoz a ritkás Velodyne időkereteket, ugyanis a koncentrikus gyűrű mintázatot regisztrálják egymáshoz, ahelyett, hogy az egyes strukturális elemek között találnák meg a megfelelő párosítást. Másrészt, a Velodyne és az MMS adatok illesztésénél a szegényes pont megfeleltetés az egyes szegmensek között szintén helytelen regisztrációhoz vezethet. A fenti tapasztalatokból következően első lépésként kiszűrjük a talaj régiót: egy a [5]-hez hasonló lokális talaj detektáló módszert használunk, amely nagy pontossággal képes kinyerni a talaj régiót, azokban az esetekben is, amikor a talaj felszíne nem tökéletes síkszerű. Első lépésben egy szabályos, fix cella méretű 2D rácsot feszítünk ki a horizontális $P_{z=0}$ síkra, ahol a z irány egybe esik a szenzor vertikális tengelyével. A pontfelhő minden p pontját hozzárendeljük a megfelelő cellához, amely megfelel a p pont $P_{z=0}$ projekciójának. A hozzárendelés után egy cella szintű pontfelhő szegmentációt hajtunk végre, ahol az egyes cellákat magasság alapján osztályozzuk, a hozzájuk rendelt pontok alapján. A szegmentáció előtt, azonban eltávolítjuk a zajos, kevés pontot tartalmazó cellákat, melyeket egy előre meghatározott szenzorspecifikus küszöbérték alapján szűrünk ki. A zajos régiók eltávolítása után az összes pontot talajpontként klasszifikáljuk egy adott cellában, ha a maximum és a minimum magasságú pont különbsége kisebb, mint egy magasság küszöbérték (25 cm), továbbá teljesül, hogy az átlagos magasság a cella szomszédságban nem nagyobb egy meghatározott értéknél. A talaj szeparáció eredményét az 1(d)-(f) ábra demonstrálja és megfigyelhető, hogy az algoritmus robusztusan képes kezelni a különböző típusú I3D és MMS pontfelhőket. Mivel az I3D szenzor karakterisztikájából következik, hogy az épületeknek és a magas objektumoknak csak az alsó részét látjuk, ezért ennek megfelelően az MMS pontfelhőknek levágtuk a talaj régióhoz képes 4 méternél magasabb részeit (1(f) ábra).

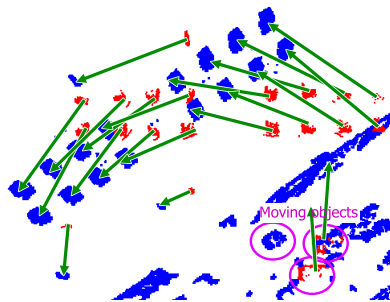
3.2. Absztrakt objektum detekció

Módszerünkkel szemben azt az elvárást tettük, hogy több méteres translációs és tetszőleges orientációbeli különbséget tudjon kompenzálni, ezért először objektum szinten

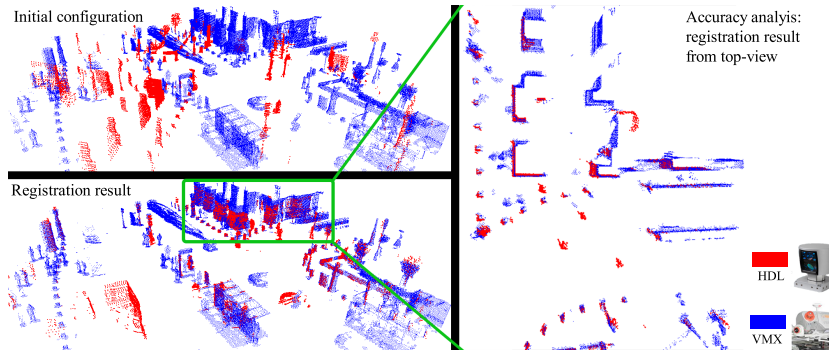


2. ábra: Velodyne HDL64E és Riegl VMX450 pontfelhők regisztrációja a bemutatott módszer alapján (Deák Ferenc tér, Budapest).

becsüljük meg a transzformációt a két pontfelhő között. A zaj és talaj régió eltávolítása után (kék színnel jelölve az 1(d)-(f) ábrákon), a következő feladat olyan elkülönülő ponthalmazok keresése, amelyek különböző utcai objektumokhoz tartozhatnak. A gyors objektum szeparáláshoz egy cella alapú módszert használtunk [12]. Egyszerűbb jelenetek esetében egy hagyományos cella alapú területelárasztó módszer felhasználásával is hatékonyan ki lehet nyerni az objektumokat. Azonban, összetettebb utcai jeleneteknél a számos takarás és nagyon közeli objektum esetén ez a módszer összeragaszthatja a közeli régiókat. Ennek a problémának a megoldására egy korábbi módszerünket használtuk [12], amely egy hierarchikus kétszintű model felhasználásával a komplexebb eseteket is képes robusztusan kezelni. A módszer először nagyobb objektumokat vagy azok csoportját nyeri ki az első cella felbontást használva, majd a második szinten finomítja a detekció eredményét. Ezt a módszert használva közel a kd-fa alapú módszerek pontosságával tudjuk szeparálni az egyes objektumokat, de a futási idő két vagy akár három nagyságrenddel is gyorsabb lehet a hagyományos fa alapú módszereknél.



3. ábra: Az objektum alapú illesztés demonstrációja az 1. algoritmus alapján.



4. ábra: Multimodális adatok (Velodyne HDL64 és Riegl VMX450) regisztrációja a bemutatott módszer alapján (Fővám tér, Budapest).

3.3. Objektum alapú transzformáció becslése

Tegyük fel, hogy az előző lépésben a két pontfelhő időkeretből kinyertük az egyes objektum halmazokat és a hozzájuk tartozó objektum középpontokat, melyek halmazát $C1$ és $C2$ -vel jelöljük. Hasonlóan a [13] minúcia alapú ujjlenyomat illesztő módszerhez, megkeressük a legjobb transzformációt a $C1$ és $C2$ halmazok között. Mivel mindkét időkereten lehetnek mozgó, rosszul detektált objektumok, ezért bármelyik objektum középpont halmaz tartalmazhat olyan pontokat, amelyek nem feleltethetők meg egyik középpontnak sem a másik halmaz mintáiból. Hasonlóan a [13] módszerhez a transzformáció paramétereinek a becsléséhez az általánosított Hough transzformációt használjuk. Először diszkrétizáljuk az összes megengedett transzformáció paraméterterét, majd minden egyes transzformáció esetén kiszámolunk egy jóság mértéket, és végül a legtöbb szavazattal rendelkező transzformációt választjuk eredményül.

A Lidar szenzorok az objektum pontjainak valós távolságát adják vissza metrikus formában, ezért a transzformációt egyszerűen előállíthatjuk translációk és rotációk összességéként. Megjegyezzük, hogy a városi utak néha tartalmaznak hirtelen emelkedő és lejtő részeket, de a mozgó szenzor orientációjának változása főleg a lokális pontfelhő z tengelye körül várható, addig a transláció az x és y irányokban figyelhető meg a horizontális $P_{z=0}$ sík mentén. Kihhasználjuk, hogy az objektum szintű transzformációnak csak az a célja, hogy egy megközelítő megoldást adjon az illesztésre, ezért leprojektáljuk a pontfelhőt a $P_{z=0}$ síkra és a képsíkon becsüljük meg a 2D translációt és a skalár orientációt 3 ábra. A fentiekből következően a keresett transzformáció a következő módon definiálható:

$$T_{dx,dy,\alpha} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

A transzformáció terét a (dx, dy, α) hármas írja le, ahol minden paraméter egy véges intervallumra van diszkrétizálva.

Az egyes transzformáció jelöltekhez tartozó jóság mértékeket egy akkumulátor tömbbe rendezzük, ahol az egyes $A[dx, dy, \alpha]$ értékekben számoljuk össze a megfelelő $T_{dx,dy,\alpha}$

transzformáció valószínűségét. Az A tömb iteratív módon feltölthető. Minden egyes objektum pár $(o1, o2)$ esetén, ahol $o1 = (x_1, y_1)$ egy $C1$ -beli, $o2$ pedig egy $C2$ -beli pont, meghatározzuk az összes lehetséges $T_{dx, dy, \alpha}$ transzformációt, ami $o1$ -ből $o2$ -be visz át, és növeljük az adott transzformáció valószínűségét az A akkumulátor tömb megfelelő indexén. Ezen a ponton kihasználjuk, hogy minden lehetséges rotációs értékhez α létezik egy egyedi translációs vektor $[dx, dy]^T$, vagyis $T_{dx, dy, \alpha}(o1) = o2$, melyet a következő képpen számíthatunk:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = o2 - \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} o1$$

A kapott dx és dy értékeket kvantálni kell a legközelebbi rekeszhez, megcímezve ezzel az A tömb növelendő elemét. Az illesztés teljes pszeudokódját az 1. algoritmus demonstrálja.

Algorithm 1 Pontfelhő illesztő algoritmus. Bemenetként két pontfelhőt vár, majd kiszámolja a transzformációt közöttük. $Rot(\alpha)$ a z tengely körüli forgatási mátrixot jelenti.

```

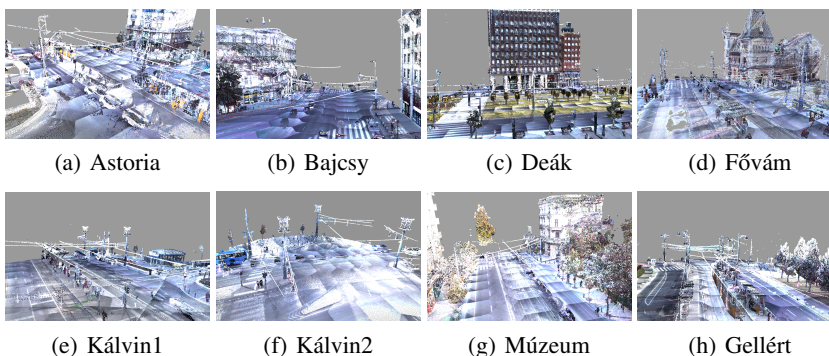
1: procedure SCANALIGNMENT( $F1, F2, T$ )
2:    $C1 \leftarrow ObjectDetect(F1)$ 
3:    $C2 \leftarrow ObjectDetect(F2)$ 
4:   Initialize 3D accumulator  $A$ 
5:   for all  $o1 \in C1$  do
6:     for all  $o2 \in C2$  do
7:       for  $\alpha \in [0, 359]$  do
8:          $o1' \leftarrow Rot(\alpha) * o1$ 
9:          $(dx, dy) \leftarrow o2 - o1'$ 
10:         $A[dx, dy, \alpha] \leftarrow A[dx, dy, \alpha] + 1$ 
11:       end for
12:     end for
13:   end for
14:    $\alpha, dx, dy \leftarrow FindMaximum(A)$ 
15:    $F1, T1 \leftarrow TransformCloud(F1, \alpha, dx, dy)$ 
16:    $F1, T2 \leftarrow NDT(F1, F2)$ 
17:    $T \leftarrow T2 * T1$ 
18: end procedure

```

3.4. Pont szintű regisztráció finomítás

Habár a bemutatott objektum alapú illesztés elég robusztusnak bizonyult a tesztek során, de a pontossága korlátozódik az adott síkon való transzformációra és rotációra, továbbá az objektum középpontok becslésének pontatlansága is limitálja az elérhető pontosságot.

A lézerszkennelési technológiák által használt, különféle speciális adatgyűjtési folyamatok hatására [5], a talaj nélküli, csak statikus objektumokat tartalmazó pontfelhőt hatékonyan lehet automatikusan illeszteni az NDT algoritmussal [4], feltéve, hogy a kezdeti transzformáció becslés elég jó minőségű, amit az objektum alapú előtranszformációnk biztosít. Ebből adódóan a bemutatott algoritmusunk első lépésben egymáshoz igazítja a pontfelhőket az optimális $T_{dx,dy,\alpha}$ transzformációval, mely számolását az 3.3. fejezetben láthattuk, majd egy NDT alapú traszformációval finomítjuk a regisztráció eredményét (az 1 algoritmus 16. sora).



5. ábra: A tesztek során használt MMS pontfelhőrészletek (adatforrás: Budapest Közút Zrt).

4. Kiértékelés

A bemutatott algoritmust a 1. fejezetben tárgyalt alkalmazások alapján értékeltük ki.

4.1. I3D és MMS típusú pontfelhők multimodális regisztrációja

Nyolc olyan MMS scenárión 5 értékeltük ki a bemutatott algoritmust, amelyek városi környezetben készültek a Riegl VMX450 szenzor felhasználásával, továbbá az adott régiókat beszkeneltük a HDL64-E és a VLP16 szenzorokkal is. Az 5. táblázat áttekintést ad a scenárió típusokról, továbbá felsorolja a kezdeti globális eltolás és orientációs eltéréseket az egyes I3D és MMS jelenetek között. Az algoritmus teljesítményének kiértékelésénél mind a két típusú Velodyne adatot illesztettük az MMS pontfelhőhöz. Az 2 és 4 ábrák kvalitatív megerősítések a Deák és a Fővám térről, még az 5. táblázat mutatja a pontos kvantitatív illesztés eredményeket. Mivel nem állt rendelkezésre referencia transzformáció, ezért kiszámoltuk az aszimmetrikus Hausdorff távolságot (MHD) a \mathcal{P}_{I3D} Velodyne és az \mathcal{P}_{MMS} MMS pontfelhők között:

$$\text{MHD}(\mathcal{P}_{I3D}, \mathcal{P}_{MMS}) = \frac{1}{\#\mathcal{P}_{I3D}} \sum_{p \in \mathcal{P}_{I3D}} \min_{q \in \mathcal{P}_{MMS}} \text{dist}(p, q)$$

ahol $\#\mathcal{P}$ jelöli a halmaz számosságát. Az 5. táblázat 5-7 oszlopai tartalmazzák a kapott kezdeti MHD értékeket, majd a Hough alapú illesztés utáni értékeket és végül az NDT alapú finomítás utáni értékeket. Az összes adathalmaz esetén megfigyelhető, hogy minden transzformációs lépés után jelentősen csökkent a távolság a két illesztendő szcenárió között. Azonban az abszolút MHD érték nem mutatja pontosan az algoritmus pontosságát, köszönhetően a szintéren megjelenő számos mozgó objektumnak, (különösen a nagy méretű villamosoknak és buszoknak), melyek hibásan növelik az átlagtávolságot. Ezért egy módosított mértéket használtunk a hiba mérésére, nevezzük átlagos pont távolságnak (MPD). Kiszámításához növekvő sorrendbe rendezzük az \mathcal{P}_{13D} pontfelhő pontjait $\min_q \text{dist}(p, q)$, majd az összes $p \in \mathcal{P}_{13D}$ estén a távolságok közép értékét vesszük. Az 5. táblázat 8-10 oszlopaiban láthatjuk, hogy az MPD értékek jelentősen csökkennek a regisztrációs folyamat alatt, és a nyolc jelenetből hét esetén 3cm alatt van az MPD hiba, melyet vizuálisan szintén validáltunk. Csak a Bajcsy teszt szcenárió esetében kaptunk helytelen regisztrációs eredményt, mind vizuális, mind a kvantitatív mérések során. Ebben a teszt esetben mind a két I3D szcenárió nagyszámú mozgó objektumot tartalmazott, beleértve nagy méretű buszokat is, melyek jelentős strukturális részeket takartak ki. Későbbi terveink közzé tartozik, egy hatékony időkeret eldobó algoritmus, mely kiszűri az olyan szituációkat, amelyek elrontják a regisztráció eredményét. Az 5. táblázat 11. (utolsó) oszlopa demonstrálja az egyes jelenetekhez tartozó futási időket (0.3-2.2 másodperc), mely felöleli a teljes regisztrációs folyamatot. Láthatjuk, hogy a módszer futási ideje közel van a valós idejű feldolgozáshoz.

4.2. IMU mentes SLAM ritkás I3D adatokon

A bemutatott algoritmust használva egy közös koordináarendszerbe tudunk regisztrálni egy adott pontfelhő szekvenciát, vagyis egy pontos 3D térkép hozható létre a környezetről, mindenféle külső pozicionáló eszköz (GPS, IMU) használata nélkül. A mérések során feltételeztük, hogy a Lidar szenzort mozgó jármű betartotta a városi sebességhatárokat, ezért egy kompakt ($50 \times 50 \times 360$) méretű A akkumulátor tömböt használtunk, mely mérete így lefed egy $[-25m, 25m]$ intervallumot a két egymást követő időkeret között. Ez a megkötés nagyon rövid számítási időt tesz lehetővé, akár $1m$ alatti pontosságú előillesztés mellett, mely távolságot már az NDT alapú finomítással hatékonyan tudjuk kezelni. A 6. és 7. ábrák egy sikeres regisztrációs eredményt demonstrálnak a Velodyne HDL64-E és VLP16-os szenzorok felhasználásával. A későbbiekben tervezünk egy nagy mértékű kvantitatív kiértékelést az algoritmus teljesítményéről SLAM feladatok esetén.

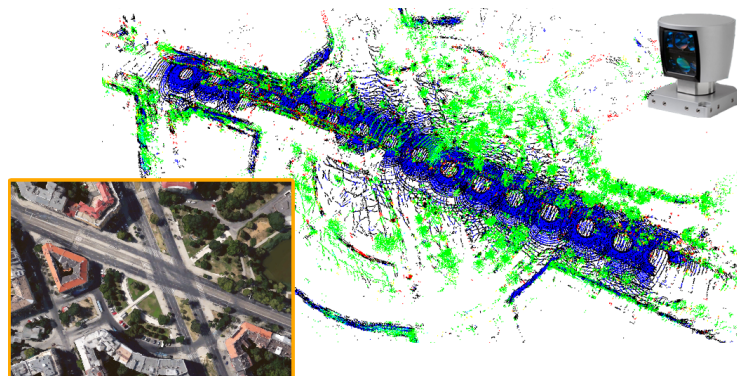
4.3. Különböző I3D szenzorok adatainak regisztrációja

Lehetőség szintjén megvizsgáltuk, és azt tapasztaltuk, hogy a bemutatott módszer nagy hatékonysággal képes egymáshoz illeszteni a Velodyne HDL64-E és VLP16 szenzorok adatait, melyet a 8) ábrán demonstrálunk. Ez a felfedezés lehetőséget biztosít a különböző I3D szenzorral felszerelt önjáró járművek hatékony adat cseréjéhez és összedolgozásához.

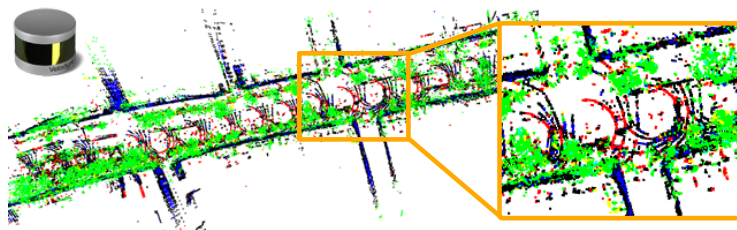
1. táblázat: I3D és MMS alapú pontfelhőregisztráció eredményei (Velodyne HDL64/VLP16 és Riegl VMX450)

Név	Helyszín típus	Szenzor	kezdeti, orient. kül.	MHD (m)			MPD (m)			Számít. idő
				Bemenet	Hough	Végső†	Bemenet	Hough	Végső	
Astoria	csomópont	HDL	2.2m, 62°	3.641	0.773	0.415	1.587	0.511	0.022	1.923
		VLP	2.2m, 99°	5.045	0.582	0.221	3.623	0.231	0.008	0.665
Bajcsy	főút	HDL	2.0m, 92°	5.657	11.441	10.105	1.177	2.702	4.539	0.992
		VLP	10.3m, 72°	6.971	20.115	17.796	4.179	17.319	14.341	0.329
Deák	út & tér	HDL	1.4m, 32°	3.638	0.717	0.338	1.516	0.345	0.004	1.960
		VLP	3.6m, 127°	7.348	0.870	0.911	5.502	0.143	0.101	0.769
Fővám	tér	HDL	2.0m, 134°	8.404	3.494	2.870	6.143	1.339	0.008	3.796
		VLP	0.1m, 20°	5.143	1.849	1.431	3.393	0.216	0.010	1.182
Kálvin 1. rész	út & tér	HDL	1.4m, 118°	9.891	0.774	0.205	5.808	0.469	0.005	1.159
		VLP	2.0m, 42°	11.427	7.016	8.178	5.007	0.752	0.014	0.573
Kálvin 2. rész	út & tér	HDL	5.8m, 104°	19.445	2.252	2.002	4.968	0.437	0.023	0.288
		VLP	6.1m, 56°	19.663	2.901	5.909	16.826	0.817	0.065	0.221
Múzeum körút		HDL	2.2m, 70°	14.911	3.358	1.373	12.354	1.315	0.009	2.574
		VLP	5.0m, 91°	6.970	2.489	3.412	1.477	0.312	0.018	1.403
Gellért tér		HDL	1.0m, 125°	3.180	0.949	1.046	1.238	0.224	0.014	1.045
		VLP	0.0m, 34°	5.241	2.438	1.574	4.037	1.173	0.029	0.852
Átlag értékek‡		HDL	2.3m, 92°	9.016	1.760	1.178	4.802	0.663	0.012	1.821
		VLP	3.7m, 68°	8.691	2.592	3.091	5.695	0.521	0.035	0.809

Hiba mérték: MHD: Módosított Hausdorff távolság, MPD: pontok távolságának mediánja.
†Hough+NDT algoritmus eredménye, ‡A Bajcsy jelenetet kizártuk az átlagolásból, a sikertelen regisztráció miatt.



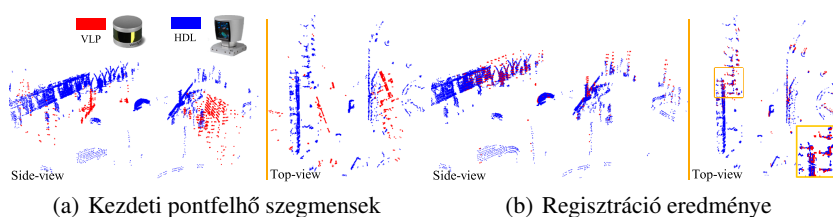
6. ábra: SLAM eredmények Velodyne HDL64 Kosztolányi tér, Budapest (1.2 millió pont, 80 időkeret, 3fps)



7. ábra: SLAM eredmények Velodyne VLP16 Bartók Béla út, Budapest (0.3 millió pont, 200 időkeret, 5fps).

5. Konklúzió és köszönetnyilvánítás

Egy új automatikus pontfelhőregisztrációs eljárást mutattunk be mobil lézerszenkerek méréseihez, amely hatékonyan képes kezelni a különböző karakterisztikájú pontfelhők illesztését. Az algoritmus egy hatékony szegmentációs lépés után egy objektum majd egy pont szintű transzformációt hajt végre az egyes pontfelhők között, mely akár 3cm alatti pontosságot is eredményezhet, közel valós feldolgozási idő mellett. A szerzők köszönetet mondanak A Budapest Közút Zrt-nek, a Riegl VMX450 MLS tesztadatok biztosításáért. A projektet a Magyar Tudományos Akadémia Bolyai János Kutatási Ösztöndíja, és a Nemzeti Kutatási, Fejlesztési és Innovációs Alap (NKFIA #K_120233) támogatta.



8. ábra: Velodyne VLP16 és HDL64 pontfelhőregisztráció eredménye.

Irodalom

1. Gálai, B., Nagy, B., Benedek, C.: Crossmodal point cloud registration in the hough space for mobile laser scanning data. In: International Conference on Pattern Recognition (ICPR), Cancun, Mexico (2016)
2. Behley, J., Steinlage, V., Cremers, A.: Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In: IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA (2012) 4391–4398
3. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision **13** (1994) 119–152
4. Magnusson, M.: The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection. PhD thesis, Örebro University (2009)
5. Józsa, O., Börcs, A., Benedek, C.: Towards 4D virtual city reconstruction from Lidar point cloud sequences. In: ISPRS Workshop on 3D Virtual City Modeling. Volume II-3/W1 of ISPRS Annals Photogram. Rem. Sens. and Spat. Inf. Sci., Regina, Canada (2013) 15–20
6. Men, H., Gebre, B., Pochiraju, K.: Color point cloud registration with 4D ICP algorithm. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. (2011) 1511–1516
7. Gressin, A., Mallet, C., David, N.: Improving 3D LIDAR Point Cloud Registration Using Optimal Neighborhood Knowledge. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (2012) 111–116

8. Gressin, A., Cannelle, B., Mallet, C., Papelard, J.P.: Trajectory-Based Registration of 3D LIDAR Point Clouds Acquired with a Mobile Mapping System. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* (2012) 117–122
9. Douillard, B., Quadros, A., Morton, P., Underwood, J.P., Deuge, M.D., Hugosson, S., Hallström, M., Bailey, T.: Scan segments matching for pairwise 3d alignment. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. (2012) 3033–3040
10. Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., Frenkel, A.: On the segmentation of 3D LIDAR point clouds. In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. (2011) 2798–2805
11. Grant, W.S., Voorhies, R.C., Itti, L.: Finding planes in LiDAR point clouds for real-time registration. In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. (2013) 4347–4354
12. Börzs, A., Nagy, B., Benedek, C.: Fast 3-D urban object detection on streaming point clouds. In: *Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving at ECCV'14*. Volume 8926 of LNCS. Springer, Zürich, Switzerland (2015) 628–639
13. Ratha, N.K., Karu, K., Chen, S., Jain, A.K.: A real-time matching system for large fingerprint databases. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **18** (1996) 799–813