

# Valós idejű pontfelhőillesztés és járműlokalizáció nagy felbontású 3D térképen

Zováthi Örkény<sup>1,2</sup>, Nagy Balázs<sup>1,3</sup> és Benedek Csaba<sup>1,3</sup>

<sup>1</sup> Gépi Érzékelés Kutatólaboratórium, Magyar Tudományos Akadémia, Számítástechnikai és Automatizálási Kutatóintézet

<sup>2</sup> Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki és Informatikai Kar

<sup>3</sup> Pázmány Péter Katolikus Egyetem, Információs Technológiai és Bionikai Kar

{vezetéknév.keresztnév}@sztaki.mta.hu

**Absztrakt.** Cikkünkben egy LiDAR alapú, valós idejű, önjáró járművek (SDV) számára is pontos helymeghatározási eljárást ismertetünk, melynek során a LiDAR szenzor ritka pontfelhőjét a jármű kezdeti GPS pozíciójából egy nagy felbontású, a környezetből mobil lézershíradó (MLS) segítségével kinyert 3D térképhez regisztráljuk. Az algoritmus első lépésében robusztus objektumkinyerést, majd az objektumok kulcspontjai, illetve az MLS térkép egyéb szemantikus információi alapján egy illesztési transzformációt végzünk. A módszert egy nagyváros forgalmas, jelentős GPS hibájú belvárosi útszakaszain értékeltük ki. Tesztjeinkben különböző kulcspontválasztási stratégiákat is alkalmaztunk, melyek mind azt igazolták, hogy a javasolt módszer egy nagyságrenddel megnöveli a pontosságot a jelenleg ismert legjobb megoldáshoz képest.

## 1. Bevezetés

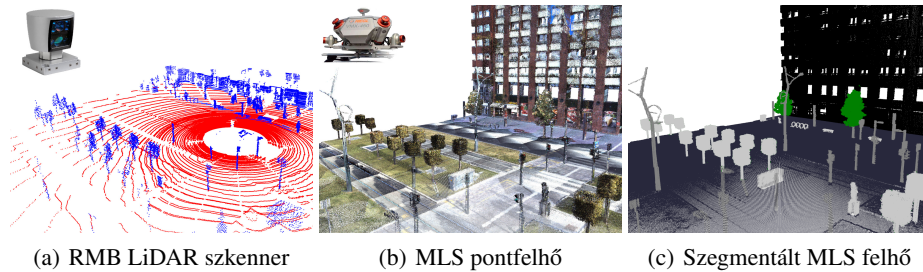
<sup>1</sup> Az önjáró járművek (SDV) nagy mértékben hozzájárulhatnak az úti balesetek számának csökkenéséhez és a zsúfolt utak hatékony forgalomeloszlásához. Mindez annak köszönhető, hogy fejlett, intelligens szenzorai segítségével az eredeti, személy- vagy teherszállító funkciójuk mellett a környezet megfigyelésére, feltérképezésére, felügyeletére, illetve változás detektálására is alkalmasak [2], [3]. Az így kapott szenzoradatok felhasználó algoritmusok többek között előre tudják jelezni a forgalmi dugókat, vagy automatikusan értesíthetik a közösséget szokatlan eseményekről, például balesetek vagy rendőri intézkedések.

Az említett autonóm funkciókhoz elengedhetetlen a precíz, robusztus lokalizáció és a környezet alapos feltérképezése. Ugyan a GPS alapú helymeghatározás általában megfelel az emberi sofőrök számára, pontossága nem elegendő egy önvezető jármű navigálásához. Ehelyett azonban az egzakt pozíciót és orientációt kiszámolhatjuk a jármű saját szenzoradatainak egy elérhető, nagy felbontású (HD) 3D várostérképhez [4,5] történő regisztrálásával.

<sup>1</sup> A bemutatott módszer a 4. fejezet kivételével eredetileg a European Conference on Computer Vision 2018 nemzetközi konferencia Road Scene Understanding and Autonomous Driving workshopján, angol nyelven került közzésre[1].

A többsugaras forgó (RMB) LiDAR lézerszkennerek [6,7] működése során közvetlenül méri a környezet térbeli információit, melyekről valós időben készít egy, a saját lokális koordináta-rendszerében elérhető pontfelhőt. Ennek középpontja maga az érzékelő, amit a széles látótérhez a jármű tetején célszerű elhelyezni. A szenzor további előnye, hogy aktív technológiaként különböző megvilágítási és időjárési körülmények között is megbízhatóan működik, hátránya viszont, hogy a leképzett pontfelhő meglehetősen ritka és inhomogén, emellett a forgó szkennelési folyamat jellegéből adódóan az objektumok elmosódhatnak, illetve ki is takarhatják egymást. Ilyen feltételek mellett a robusztus objektumdetekció és osztályozás nehézségekbe ütközhet [6]. A jármű globális helymeghatározásához a LiDAR pontfelhőt a szenzor lokális koordináta-rendszeréből az aktuális GPS pozíció alapján világkoordináta-rendszerbe kell transzformálni. Városi környezetben azonban a navigációs jel vételi korlátai miatt a GPS alapú pozícióbecslés gyakran csak 2-10 méter pontosságú, ami egy önvezető autó számára nem elegendő.

A naprakész HD térképek nagy fejlődésen mentek keresztül az utóbbi időben. A mobil lézerszkennerek (MLS) napjainkban már rövid idő alatt egy nagyon sűrű és részletes 3D térképet alkotnak városokról, igaz, ennek *offline* szűrése, szemantikus címkézése és a nyers adatok *valós idejű* kiaknázása még jelenleg is egy kihívást jelentő feladat [8].



**1. ábra.** Többsugaras forgó Velodyne HDL64E LiDAR szenzorral és Riegl VMX450 mobil lézerszkennerek (MLS) rendszerrel felvett pontfelhők egy belvárosi útszakaszon. A szegmentált felhő osztályainak színek: fekete: *épület*, sötétszürke: *talaj*, szürke: *magas oszlop*, világosszürke: *utcai bútorzat*, zöld: *lombkorona*

Cikkünkben a Velodyne HDL64E RMB LiDAR szenzor mérési adatait és a Riegl VMX450 MLS rendszert használtuk fel a jármű precíz és robusztus helymeghatározásához. A Velodyne szenzort eredetileg autonóm robotok és járművek valós idejű érzékelésének támogatására tervezték, amihez egy relatív ritka ( $60-100 \times 10^3$  pont/időkeret), 10-15 fps sebességű pontfelhő-adatfolyamot biztosít. A szenzor térbeli pontossága körülbelül 1-2 cm a lokális koordináta-rendszerében, a pontsűrűség azonban jelentősen csökken az érzékelőtől való távolság függvényében, és a forgó szkennelési eljárás eredményeképp a pontok eloszlása tipikusan körgyűrűs mintákat mutat (1(a) ábra).

A Riegl VMX450 MLS rendszert várostérképezési, tervezési, illetve útfelügyeleti alkalmazásokhoz használják. A rendszer két Riegl lézerszkennerből, egy jól kalibrált kamera platformból és egy nagy teljesítményű Globális Navigációs Szatellit Rendszerből (GNSS) áll, amiket felhasználva egy rendkívül sűrű, relatív egységes eloszlású és globálisan pár centiméteres pontosságú pontfelhőt állít elő (1(b) ábra).

A jármű precíz helymeghatározásához a ritkás Velodyne adatot a nagy sűrűségű MLS pontfelhőhöz, mint HD térképhez regisztráljuk a kezdeti GPS alapú pozícióbecslés alapján. Ugyan a pontfelhőregisztráció már egy mélyen tárgyalt és feltárt téma, 3D mérések illesztése ilyen típusú sűrűségkarakterisztikák mellett mégis egy kihívást jelentő feladat. Megközelítésünkben erre egy pontos és robusztus, objektum alapú illesztési algoritmust ismertetünk a LiDAR pontfelhő és az MLS HD térkép között.

## 2. Szakirodalomban fellelhető pontfelhőregisztrációs módszerek

A szakirodalomban számos hatékony pontfelhőregisztrációs eljárás található, melyek közül a legtöbbet idézett a Normal Distribution Transform (NDT) [9] és az Iterative Closest Point (ICP) [10]. Az ICP algoritmusnak számos változatát és továbbfejlesztését publikálták: [11] a pont szomszédságból kinyert geometriai jellemzők segítségével, [12] szín információ felhasználásával, míg [13] időbeli követéssel javítja tovább a módszert. Ezek a megközelítések azonban érzékenyek az eltérő sűrűségkarakterisztikákra, különösen a Velodyne szenzor tipikus körgyűrűs mintázata vezetheti félre a regisztrációs eljárást. Ezen kívül minden ICP alapú megközelítésnek kritikus feltétele a pontfelhők pontos előillesztése, ami a gyakorlatban, GPS alapú kezdeti pozícióban a néhány méteres hibataromány miatt nem teljesül.

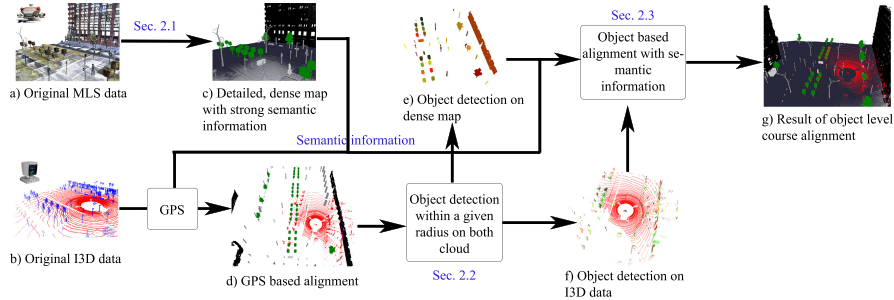
Más módszerek a pontfelhők átfogóbb megfeleltetésére fókuszálnak: [14] és [15] lokális jellemzők kiterjesztésével keresnek globális megfelelőket, melyek szintén használhatóak az ICP eljárás előillesztéseként. Nagy hátrányuk azonban, hogy ezek az algoritmusok már kis pontfelhő részek esetén is nagy számítási teljesítményt igényelnek, így valós idejű feltérképezésre, SLAM (Szimultán Lokalizáció és Térképezés) feldolgozásra és helymeghatározásra hatékonyan nem alkalmazhatóak.

Egy másik ICP alapú megközelítés [7] pont-pont párosítások helyett objektum szinten keres megfeleltetést szegmensek között. Ez a módszer hatékonyan illeszti össze a különböző Velodyne pontfelhőket, bár egy-egy párosítás kiszámolása továbbra is 3-15 másodpercig tart. Különböző modalitású és sűrűségkarakterisztikájú pontfelhőregisztrációra nagyon kevés szakirodalmi hivatkozás található.

Korábbi módszerünkben [16] egy absztrakt objektumdetekció után a becsült objektumközpontokat használtunk fel, melyekkel először egy hozzávetőleges pontfelhőillesztést, majd egy NDT alapú, pontszintű finomítást alkalmaztunk. Az eljárás hátrányai, hogy különböző típusú objektumok hibát okozhatnak a megfeleltetések keresésekor, az illesztés finomhangolását végző NDT lépések pedig jelentősen megnövelik a számítási igényt.

### 3. A javasolt algoritmus

Cikkünkben egy valós idejű, robusztus, objektum szintű illesztési módszert ismertetünk a ritka LiDAR mérések és sűrű MLS pontfelhők között. Az algoritmus felépítése és lépései a 2. ábrán láthatóak.



2. ábra. A javasolt algoritmus felépítése.

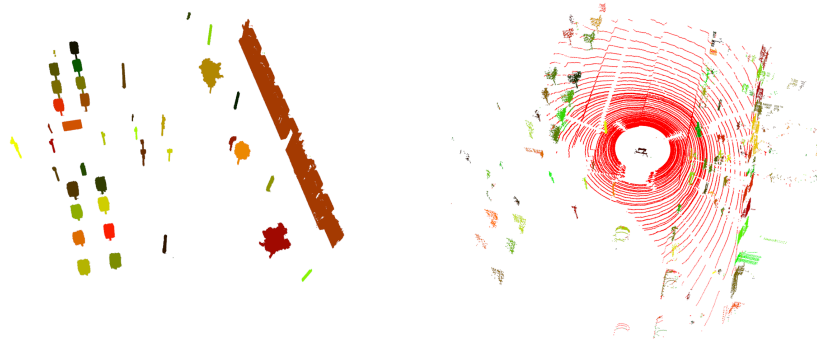
Az eljárás előkészítéséhez szemantikus szegmentáljuk az MLS pontfelhőt, és a pontokat, illetve a hozzájuk tartozó címkéket nagyfelbontású térképként eltároljuk a későbbi, valós idejű felhasználásra. Ezt követően kiszámoljuk az optimális pontfelhő-illesztési algoritmust minúcia alapú ujjlenyomat illesztő módszer [16,17] segítségével. Ennek előnye, hogy akkor is találhatunk robusztus transzformációkat két pontfelhő között, ha azok jellegükben nagy mértékben különböznek. Az eddigi megközelítésekkel ellentétben azonban az objektumközéppontok figyelembevétele helyett – ami csak hozzávetőleges megfeleltetéseket eredményez [16] – egy új kulcspontválasztási technikát is javasolunk, ami a precíz illesztést a nagy számítás igénylő pontszintű finomhangolási lépések nélkül valósítja meg, és stabil eredményre vezet eltérő, vagy csak részben látható objektumok esetén is. A hatékonyság növeléséhez a potenciális objektumokkal szemben további szemantikus megszorításokat is alkalmazunk a két pontfelhőn.

#### 3.1. A referencia HD térkép létrehozása

A Riegl VMX450 rendszer rövid idő alatt egy széles látóterű, részletes és sűrű MLS pontfelhőt alkot kültéri környezetekről. Ez a rögzített pontfelhő földrajzi adatokat is tartalmaz, így megfelelő szemantikus szegmentációval közvetlenül is felhasználhatjuk 3D nagyfelbontású (HD) térképként. A billió nagyságrendű pontok kézi címkézése ugyan meglehetősen forrásigényes feladat, de különböző deep learning alapú pontfelhőosztályozási megközelítések, például a PointNet++ [18] segítségével automatizálhatjuk a folyamatot. Ezen felül városban felvett MLS adatok esetén számos speciális kihívással is szembe kell néznünk – például a felvételkor végzett független mozgások eredményeként létrejövő fantom effektus [19] –, amit az általános pontfelhőszegmentációs algoritmusok nem képesek lekezelni. Ehhez kiterjesztettünk egy részben MLS adatszűrésre

fejlesztett 3D konvolúciós neurális hálózatot [19], és az objektumokat az alábbi osztályokba soroltuk: *talaj*, *épület*, *fantom*, *jármű*, *gyalogos*, *növényzet* (bokrok és lombkoronák), *magas oszlop* (utcai táblák és fatörzsek) és *utcai bútorzat* (eddig nem említett egyéb utcai objektumok, pad, szemetes, alacsony oszlop). Az 1(c) ábra demonstrálja az eljárás eredményét, melyen a dinamikus osztályokat (*jármű*, *gyalogos* és *fantom*) már eltávolítottuk.

A már szegmentált MLS HD térképen az egyes objektumokat – például jelzőtáblák – a megfelelő szegmentált osztályokból euklideszi klaszterezéssel, [20] offline módon kaphatjuk meg. Mivel a regisztrációs folyamathoz az MLS térkép objektumait fix, statikus pontokként egy háttérmodellben akarjuk használni, a dinamikus (*jármű*, *gyalogos* és *fantom*), az időfüggő (*növényzet*), illetve széles tartományokat lefedő (*talaj* és *épület*) osztályokat figyelmen kívül kell hagynunk. Ezen osztályok elhagyásával a megfeleltetéseket csupán a statikus (*magas oszlop* és *utcai bútorzat*) osztályok között keressük, melyek a különböző szituációkban állandó és tömör alakban jelennek meg, így valóban alkalmasak az illesztés meghatározására.



(a) Fix, statikus objektumok a HD térképen (b) Detektált objektumok a LiDAR pontfelhőn

**3. ábra.** Az illesztés meghatározásához használt objektumok. A színek a következők: (a) a HD térkép statikus objektumai különböző színekkel jelölve (b) piros: talaj/út, egyéb színek: különböző detektált objektumok a LiDAR pontfelhőn

### 3.2. Valós idejű objektumdetekció többsugaras forgó LiDAR pontfelhőn

A javasolt objektum alapú pontfelhőillesztéshez a LiDAR pontfelhőn végzett precíz és robusztus objektumdetekció is elengedhetetlen lépés. Ez a feladat azonban a pontfelhő alacsony és inhomogén sűrűségkarakteristikája, az objektumok deformálódása és a valós idejűség követelménye miatt meglehetősen komplex.

Megközelítésünk első lépésében a szenzor pontjainak azon részét tartjuk meg, melyek egy  $r = 30\text{m}$  sugarú körön belül helyezkednek el az érzékelő forgástengelyéhez

képest. Ennek oka, hogy a szenzor sűrűségkarakterisztikája miatt a távoli területek már túl ritkák megbízható információk szolgáltatásához. A távolság-paraméter értékét ( $r$ ) a Velodyne HDL64 szenzorhoz optimalizáltuk.

A következő lépésben a talajpontokat távolítjuk el, ami két szempontból is fontos. Egyrésztől, a LiDAR szenzor körgyűrűs mintázata nagy részben a talaj területeken jelenik meg, ami félrevezetheti a regisztrációs algoritmust, másrésztől a lépés megkönnyíti azoknak az objektumoknak az elválasztását is, melyek a pontfelhőben talajterületeken keresztül nőttek össze.

A szakirodalomban erre előszeretettel alkalmazzák a robusztus síkbecslésen – például RANSAC – alapuló talajmodelleket, ám bár ezek kevésbé hatékonyak az útrészek közti jelentős magasságbeli különbségek esetén (például emelkedő és lejtő útszakaszok). Ehelyett [6] alapján egy cella alapú, lokálisan adaptív terepmodellezési eljárást javasolunk. Ehhez először egy szabályos 2-D rácsot feszítünk ki 0.2m oldalszélességgel (rács távolság) – az érték [6] alapján városi környezetre optimalizált – a LiDAR pontfelhő lokális euklideszi koordináta-rendszerben vett  $P_z$  síkjára. Ezután minden egyes ponthoz hozzárendeljük azt a cellát, ami a  $P_z$  síkra vett vetületét tartalmazza. Egy cellát akkor jelölünk talaj vagy útpontnak, ha a hozzátartozó maximum és minimum pontok magasságainak különbsége kisebb, mint 10cm – ez a feltétel egy cellán belül  $26^\circ$ -os talajferdeséget enged meg. Ezt követően a magasság meghatározásához a talajnak jelölt cellákban kiszámoljuk a tartalmazott pontok  $z$ -koordinátáinak átlagát. A kiugró értékek – például autók sík teteje – eltávolításához egy medián szűrőt alkalmazunk a szomszédos talaj cellák felhasználásával. A fennmaradó nem-talaj cellák – melyek feltehetően térbeli objektumokat is tartalmaznak – esetén a lokális talajmagasság értékét ( $z_0$ ) a szomszédos talaj cellák interpolációjával számoljuk, és az összes  $z_p$  magasságú pontot, melyre  $z_p - z_0 > \tau$  (ahol  $\tau = 10\text{cm}$ ) teljesül, nem-talaj pontnak tekintjük.

A talaj eltávolítása után klaszterezzük a megfelelő *nem-talaj* pontokat, amivel elválasztjuk egymástól az objektumjelölteket. A folyamatot egy 2D cella térképen régió növesztéses algoritmussal implementáltuk, az üres cellákat használva leállási feltételnek. Ugyan ezáltal néhány határos objektum egybeolvadhat a rács véges felbontása miatt, azonban az offline HD térképpel szemben itt a számítási sebességnek teljesítenie kell a valós idejűség követelményét, és ez a 2D objektumdetekciós megközelítés két nagyságrenddel gyorsabbnak bizonyult a hagyományos kd-fa alapú 3D euklideszi klaszterezési algoritmusnál.

A 3(b) ábra alapján az objektumdetekció kvalitatívan is elemezhető. A térbeli objektumok – járművek, oszlopok és fák – a legtöbb esetben önálló klaszterként jelennek meg, míg a hosszú épületrészek kisebb falszegmensekre esnek szét. Mindamelllett a LiDAR pontfelhő korábban részletezett korlátai miatt nem egy egzakt klasszifikációt végzünk, és a javasolt pontfelhőillesztéshez sem használjuk fel az összes objektumot.

### 3.3. Objektum alapú illesztés

Ebben a fejezetben célunk egy olyan optimális geometriai transzformáció meghatározása, mellyel már a LiDAR mérések ritka pontfelhőit az MLS rendszerrel felvett HD térképhez regisztrálhatjuk. Ehhez először a jármű GPS alapú pozícióbecslését ( $p_0$ ) használjuk fel – ez lesz a jármű pontfelhőjének középpontja a HD térkép globális

koordináta-rendszerében –, majd annak érdekében, hogy a két illesztendő pontfelhő határai közel azonosak legyenek, a jelenlegi  $p_0$  pozíció 30 méteres környezetét kivágjuk az MLS pontfelhőből.

Kihasználva azt, hogy a LiDAR szenzorok közvetlen, cm pontosságú információkat tartalmaznak a 3D euklideszi térben, a két pontfelhő közti térbeli transzformáció megadható egy homogén transzformációként translációs és rotációs komponensekkel. Ebben egyrészt keressük azt a 3D translációs vektort ( $dx$ ,  $dy$  and  $dz$ ), ami lényegében a GPS szenzor eredeti pozícionálási hibájának felel meg, másrészt, az MLS pontfelhőből az út lokális normálvektorát is jól becsülhetjük analitikusan, a keresési tartományon belül sík útszakaszokat feltételezve. Ennek alapján a regisztráció során csupán a jármű tetejéhez viszonyított  $\alpha$  forgási komponenst kell meghatározunk. Összegezve, egy optimális transzformációt modellezünk a két pontfelhő között az alábbi homogén mátrix segítségével:

$$T_{dx,dy,dz,\alpha} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & dx \\ -\sin \alpha & \cos \alpha & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

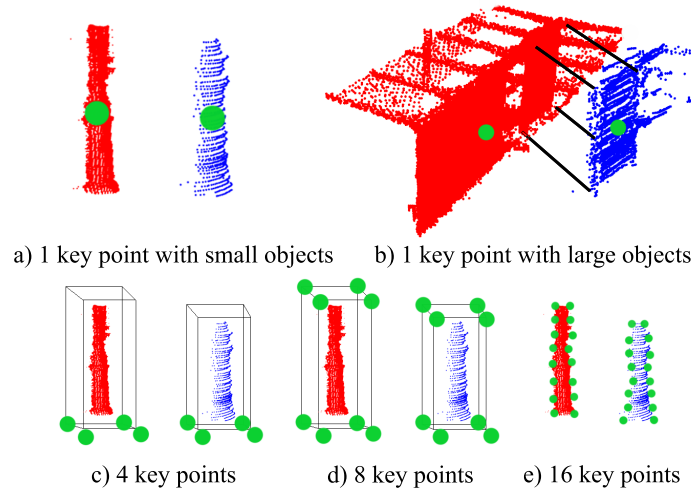
A paramétertér korlátozásának érdekében mindkét irányban maximum  $45^\circ$  forgásszöveget engedélyezünk, mivel a GPS adatból a hozzávetőleges orientáció már a rendelkezésünkre áll. Hasonló megfontolásból a  $dx$  és  $dy$  paraméterekhez  $\pm 12\text{m}$ , míg a függőleges translációhoz  $\pm 2\text{m}$  maximális bizonytalanságot feltételezünk.

A következő lépés a transzformációt meghatározó algoritmus. A javasolt regisztrációs technikával ahelyett, hogy a nyers pontfelhőt illesztenénk, különböző kulcspontokat párosítunk, melyek a HD térkép *fix, statikus objektumaiból* lettek kinyerve (3.1. fejezet). További szemantikus megszorításokat alkalmazva, csak azokat a kulcspontokat próbáljuk meg párosítani, melyek a LiDAR pontfelhőn is a megfelelő, statikus objektumokhoz tartoznak. Ennek meghatározásához az algoritmus hátralevő része három lépésből áll: i) *kulcspontválasztás*, ii) *kompatibilitási feltételek meghatározása a LiDAR szenzorral megfigyelt és fix objektumok között*, iii) *optimális transzformáció számolása* a kompatibilis kulcspontok alapján.

**Kulcspontválasztás** A javasolt megközelítésünk egyik kritikus lépése a megfigyelt és fix objektumok kulcspontjainak meghatározása. Egy célravezető megoldás [16] minden objektumra egyetlen kulcspontot, az objektum tömegközéppontját használja (4(a) és (b) ábra). Azonban, ahogy korábban említettük, az érzékelt LiDAR pontfelhő számos részlegesen detektált objektumot tartalmaz, melyek alakja jelentősen különbözhet ugyanazon objektum az MLS pontfelhőben teljeskörűen szkennelt formájától, ami által a kiszámolt középpontok is jelentősen eltérhetnek.

Ennek megoldására többféle kulcspontválasztási stratégiát dolgoztunk ki, és a tisztán középpont alapú regisztráció mellett teszteltük az algoritmus teljesítményét 4, 8 és 16 kulcspont használatával is. Ezeknek az illesztéseit a 4(c), (d) és (e) ábra tartalmazza. Ahogy az ábrákon látszik, a 4- és 8-kulcspontú stratégiák során a megfigyelt és fix objektumok 3D határolókereteinek sarokpontjait használtuk fel. A 16-kulcspontú eset

során a 3D határolókereteket  $2 \times 2 \times 4$  darab egyenlő kockára bontottuk, és minden egyes területre kiválasztottuk a sarokpontok tömegközéppontját.



4. ábra. Kulcsponatok választása a regisztrációhoz.

Célunk több kulcsponthasználat esetén az volt, hogy a helyes párosítások részlegesen megfigyelt objektumok esetén is megtalálhatóak legyenek, amennyiben határolókeret bizonyos sarkai viszonylag jól detektáltak. A kulcsponthasználatának emelése másrészt a számítási igényt is növeli, illetve a lehetséges pont-pont megfeleltetések halmazának növekedése a transzformáció hibás optimumához is vezethet.

**Kompatibilitási feltételek a megfigyelt és fix objektumok között** Ahogy korábban is említettük, a pontfelhők közti optimális transzformációt két kulcsponthalmaz között szeretnénk meghatározni. Ezt egy objektum szintű eljárással valósítjuk meg, így az algoritmusunk már objektum szintű tudás mellett alkalmas hibás kulcsponthasználat kiszűrésére. Ezt felhasználva csak olyan pontpárokat párosítunk, melyek az adott szituáció kompatibilis objektumait alkotják.

A referencia HD térkép szegmentálásával (3.1. fejezet) már meg tudtuk különböztetni a *magas oszlop* és *utcai bútort* osztályokat, így az elérhetővé vált, hogy minden MLS kulcsponthasználat a fenti két osztályból származzon. Ezzel szemben a részletes objektum osztályozás nem működött a LiDAR pontfelhőn, így ezekhez a következő kompatibilitási feltételeket írjuk elő:

- a *magas oszlop* MLS osztály azokkal a LiDAR szegmensekkel kompatibilis, melyek oszlop formájú határolókerettel rendelkeznek, azaz a magasságuk legalább kétszerese a szélességüknek és a mélységüknek.

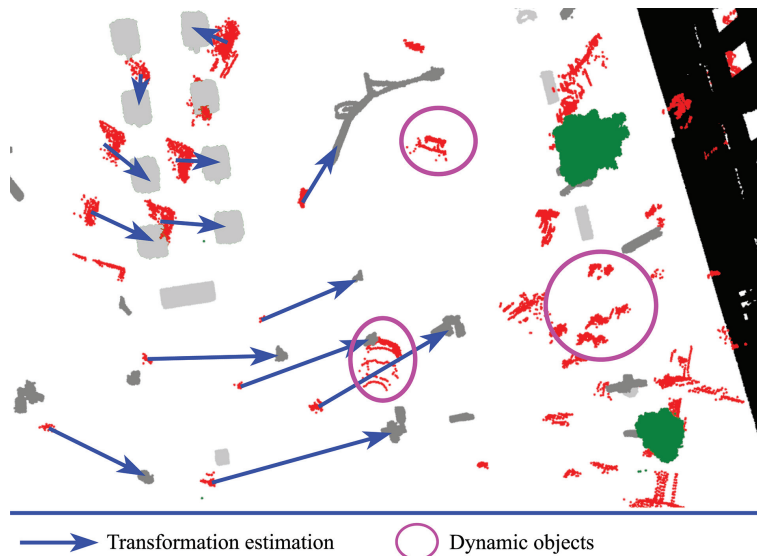


- az *utcai bútorzat* MLS osztály és a kompatibilis LiDAR objektumokhoz tartozó határolókeretek térfogatának aránya 0.75 és 1.25 között van.

A feltételeket alkalmazva megnöveljük annak a bizonyosságát, hogy az adott transzformáció csak a hasonló struktúrájú objektumpárok között zajlik le, és a sok felesleges számolás kiszűrésével az algoritmus sebessége is feljavul. Ugyan a LiDAR pontfelhő tartalmazhat dinamikus objektumokat is – gyalogos, jármű –, melyek a fenti kritériumokat teljesíthetik bizonyos MLS objektumokkal, de ezek az objektumok kilógó értékpárokat fognak mutatni a transzformáció során, így a hatásuk magasabb szinten kiküszöbölhető.

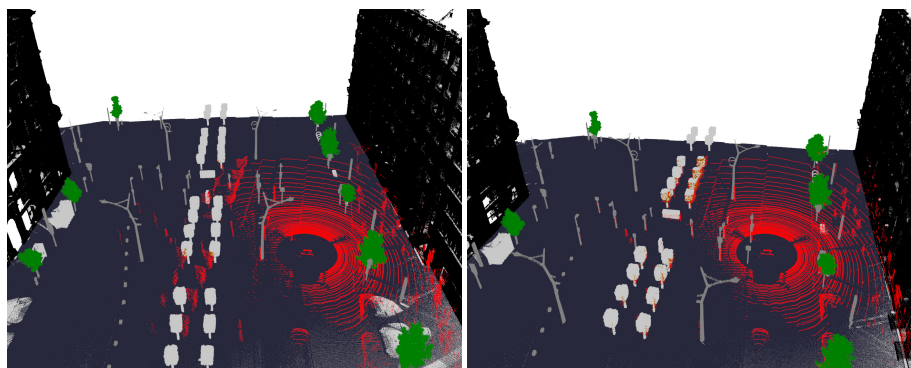
**Optimális transzformáció meghatározása** Legyen a LiDAR szenzorral *megfigyelt* és az MLS térképen *fix* objektumok halmaza  $\mathcal{O}_o$  és  $\mathcal{O}_l$ .

A legjobb transzformációt két objektumkulcspont-halmaz között egy többségi szavazási eljárással (5. ábra) keressük a Hough transzformáció 3D kiterjesztésének sémáját [17] felhasználva. Ehhez a transzformációs teret diszkrétizáljuk minden paraméter számára, és a translációs komponenshez  $0.2m$ , míg a rotációs komponenshez  $0.25^\circ$ -os lépésközt használunk a minimálisan és maximálisan megengedett érték között.



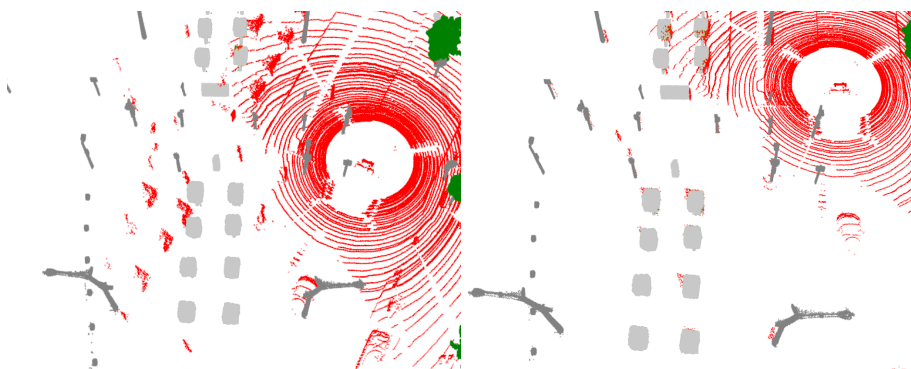
**5. ábra.** A minúcia alapú ujjenyomatillesztésen [17] alapuló javasolt objektumpárosító algoritmus kimenetének illusztrációja. A piros színű pontok a LiDAR pontfelhő megfigyelt objektumait alkotják.

Ezt követően allokálunk egy négy dimenziós tömböt, amivel az adott transzformációt leíró minden lehetséges négyesre  $(dx, dy, dz, \alpha)$  érkező szavazatokat összegezzük.



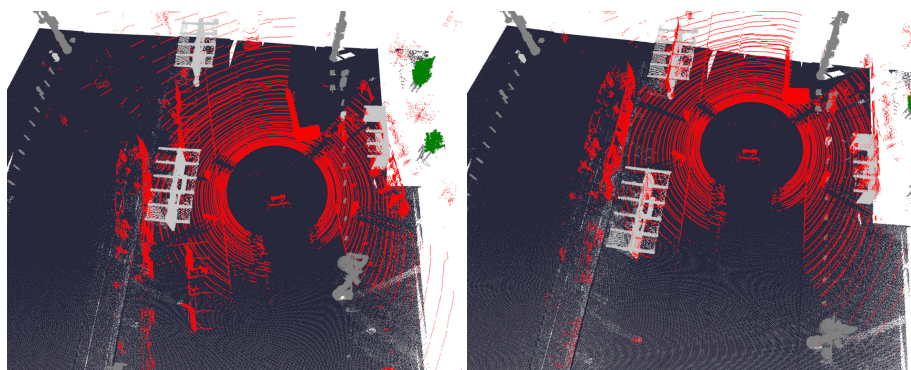
(a) Kezdeti GPS-alapú illesztés

(b) A regisztráció eredménye



(c) Kezdeti GPS-alapú illesztés

(d) A regisztráció eredménye



(e) Kezdeti GPS-alapú illesztés

(f) A regisztráció eredménye

**6. ábra.** A javasolt regisztrációs algoritmus eredménye 8-kulcspon t ű stratégia esetén. Az RMB LiDAR pontfelh ők et pirossal, míg az MLS adatot a szegmentációs osztálytól függ ők egy  b színnel jelöltük. Az első két sor ugyanazon szituációnak felel meg azzal a különbséggel, hogy a második sorban a talaj  s  p  let osztályokat eltávolítottuk.

A tömb minden tagját nulla kezdeti értékkel inicializáljuk. A szavazási folyamat során az  $\mathcal{O}_o \times \mathcal{O}_l$  Descartes-szorzat minden lehetséges  $\mathcal{O}_o, \mathcal{O}_l$  kompatibilis objektumpárját megvizsgáljuk, majd megpróbálunk párosítani minden  $o_o \in \mathcal{O}_o$  kulcspontot a megfelelő  $o_l \in \mathcal{O}_l$  kulcsponttal úgy, hogy az  $o_o, o_l$  kulcspont pár esetén adunk egy szavazatot minden lehetséges  $T_{dx,dy,dz,\alpha}$  transzformációnak, ami az  $o_o$  pontot  $o_l$  pontba képezi le. Ekkor egyesével végigiterálunk minden egyes  $\alpha^* \in [-45^\circ, +45^\circ]$  értéken, és minden  $\alpha^*$  esetén elforgatjuk az  $o_o$  pontot az aktuális  $\alpha^*$  szöggel, és kiszámoljuk a megfelelő translációs vektort  $[dx^*, dy^*, dz^*]^T$  a következőképpen:

$$\begin{bmatrix} dx^* \\ dy^* \\ dz^* \end{bmatrix} = o_l - \begin{bmatrix} \cos \alpha^* & \sin \alpha^* & 0 \\ -\sin \alpha^* & \cos \alpha^* & 0 \\ 0 & 0 & 1 \end{bmatrix} o_o$$

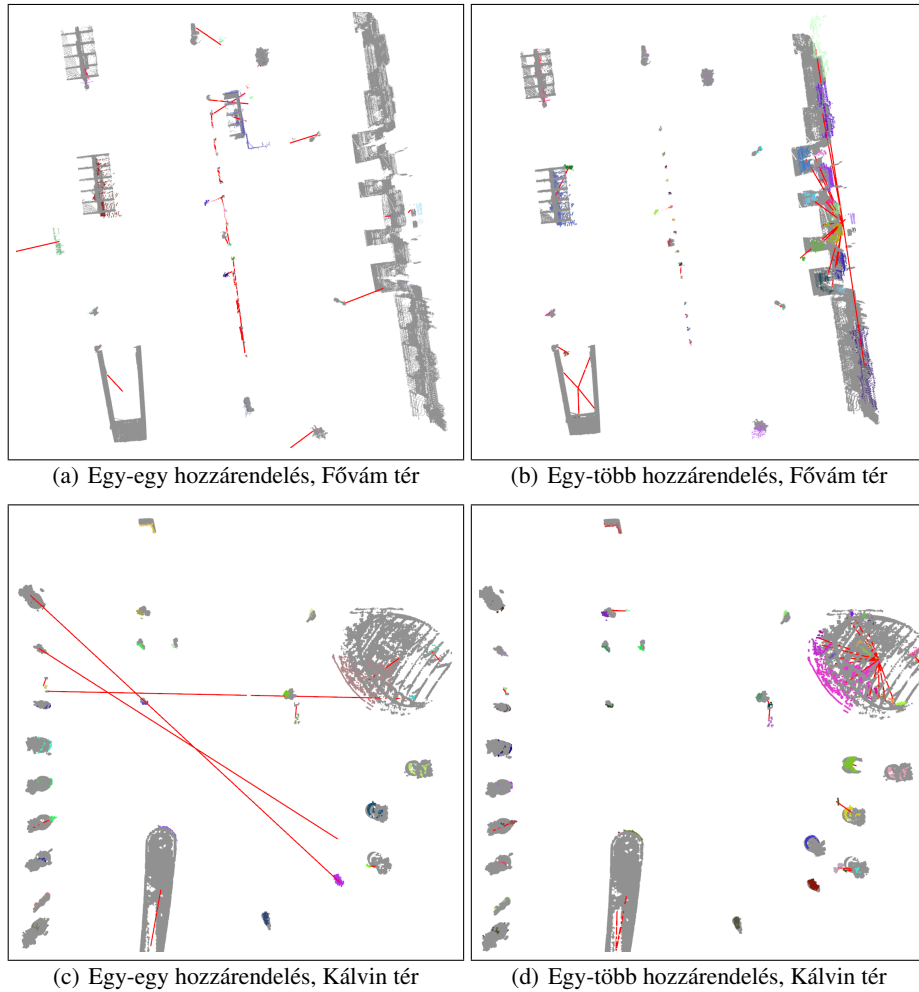
Ezt követően megnöveljük a szavazatok számát a  $T_{dx^*,dy^*,dz^*,\alpha^*}$  transzformációra. Az iteratív eljárás végén megkeressük a 4-D tömb maximumát, melynek paraméterei ( $\alpha, dx, dy$  és  $dz$ ) meghatározzák az optimális transzformációt a két objektumhalmaz között.

#### 4. Széteső objektumok egyesítése és változásdetekció

Az optimális transzformáció meghatározásával és végrehajtásával a ritka LiDAR pontfelhőt a nagy felbontású HD térképhez illesztjük. Mivel a javasolt eljárás az eddigi megoldásokhoz képest egy nagyságrenddel megnöveli a pontosságot, az eredmény közvetve már magasabb szintű autonóm funkciók számára is használható. Ehhez egy köztes lépés a környezet dinamikus, mozgó objektumainak – például *autó* és *gyalogos* – valós idejű kiszűrése, amit a precíz regisztráció, illetve a LiDAR pontfelhőn végzett robusztus objektumdetekció és a kivágott MLS pontfelhő szemantikus szegmentálásának felhasználásával valósíthatjuk meg.

A regisztrációs eljáráshoz hasonlóan a mobil lézershakkenner által készített HD térkép statikus objektumait ( $\mathcal{O}_r$ ) referenciaként használjuk fel, és egy optimális párosító algoritmus segítségével megkeressük a LiDAR pontfelhő azon klasztereit, melyek a legjobban illeszkednek a HD térkép egy-egy háttérobjektumához. A sikeres párok a jelenet statikus objektumhalmazát alkotják, melyek komplementere lesz maga a dinamikus, változó környezet. Fontos megemlíteni azonban, hogy a LiDAR pontfelhőn végzett robusztus objektumdetekció eredményeként a térbeli objektumok ( $\mathcal{O}_o$ ) – járművek, oszlopok és fák – nem minden esetben jelennek meg önálló klaszterként, a hosszú épületrészek pedig kisebb falszegmensekre esnek szét (3.2. fejezet), így kizárólag kölcsönösen egyértelmű megfeleletések keresése nem célravezető stratégia.

Egy általános megoldás erre [21], ami a magyar módszer továbbfejlesztésével több-több megfeleltetésekre képes. A megközelítés az alap algoritmushoz képest az egyik halmazt egy  $L$  minimális, míg a másik halmazt egy  $L_a$  maximális kapacitásértékkel látja el. Feltételezve, hogy a nagy felbontású (HD) térkép objektumainak *offline* szegmentálása pontos, minden Velodyne klaszterhez az egyetlen, neki megfelelő statikus MLS objektumot keressük. Ezáltal a párosítás az  $\mathcal{O}_o$  halmaz elemeire  $L = 1$  érték választásával egy-több hozzárendeléssé módosul. Az ugyanahhoz a háttérobjektumhoz



**7. ábra.** A párosítások eredménye egyszeres, illetve többszörös hozzárendelés esetén. Színek: *szürke*: MLS háttérobjektumok, *piros vonal*: párosított objektumok közép-pontjai közti szakaszok, *egyéb színek*: különböző detektált Velodyne klaszterek.

tartozó LiDAR szegmensek összekapcsolásával továbbá a pontfelhő széteső objektum-részeit is egyesíteni tudjuk.

A HD térkép statikus objektumainak ( $O_r$ ) kapacitása a helyszín függvényében változtatható, városi környezetben célravezető magasabb értéket beállítani. Végtelen értéket választva ( $L_a = \infty$ ) az algoritmus az objektumok függetlenségének következtében egy egyszerű minimumkereséssé módosul, viszont ez korlátozás hiányában több nagyvonalú párosítást is eredményezhet, így csak olyan esetben érdemes használni, ahol semmilyen korlátozás nem alkalmazható.

Az egyes párosítások illeszkedési jóságához a következő költségfüggvényt definiáljuk  $\forall o_r \in \mathcal{O}_r$  és  $\forall o_o \in \mathcal{O}_o$  esetén:

$$Cost(o_r, o_o) = 1000 + W_1 d(o_r, o_o) + W_2 dh(o_r, o_o) - W_3 A(o_r, o_o)$$

ahol  $d$  a két objektum középpontjának távolságát,  $dh$  a magasságaik különbségét,  $A$  a felülnézeti határolókeretek átlapolódását, míg  $W_1$ ,  $W_2$  és  $W_3$  a tényezők költségének súlyait jelölik. Fontos megjegyezni azonban, hogy egy adott jelenet alatt előforduló potenciális objektumok sok esetben nem rendelkeznek párral. Ez dinamikus LiDAR klaszterek esetén nem okoz gondot, viszont a pár nélküli háttérobjektumok ronthatják az algoritmus eredményét. Egy-egy jelenet irreleváns háttérobjektumainak szűréséhez ezért egy magas,  $D$  (disallowed) értékkel növeljük a költséget minden esetben,

- ha a Velodyne objektum legalább másfélszer magasabb a háttérobjektumnál,
- ha egy nagy háttérobjektum ( $A_r \geq 4m^2$ ) csak minimálisan, vagy egyáltalán nem lapolódik át a Velodyne objektummal,
- ha egy kis háttérobjektum ( $A_r < 4m^2$ ) a Velodyne objektum középpontjától legalább 2 méter távolságra van, vagy
- ha a Velodyne és egy kis háttérobjektum ( $A_r < 4m^2$ ) magasságkülönbsége nagyobb, mint 1 méter.

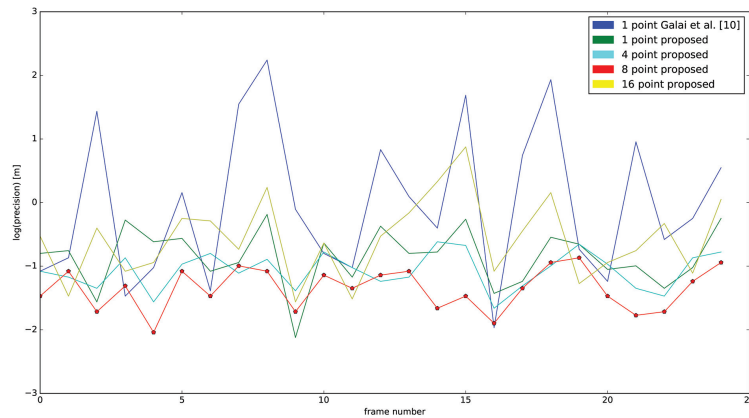
A szűrés után eltávolítjuk az MLS objektumok közül azokat, melyek minden Velodyne objektummal minimum  $D$  költséggel rendelkeznek, és virtuális objektumokkal helyettesítjük őket, melyek költsége a maximális és minimális költség átlaga, azaz a rossz, és a potenciális párok költségei közé esik. Ezt követően lefuttatjuk a párosítást (7. ábra), majd az objektumok komplementer halmazát véve megkapjuk a környezeti változást.

## 5. Kiértékelés

A javasolt pontfelhőregisztrációs algoritmust és a változásdetekciót többféle zsúfolt, nagyvárosi útszakaszon értékeltük ki. Néhány kvalitatív eredmény a 6. ábrán látható. A kiértékelés során a javasolt modell különböző kulcspontválasztási stratégiáit vizsgáltuk, és összehasonlítottuk a megközelítést a jelenleg ismert legjobb módszerrel [16] is. A kiértékelés mérőszámaként a optimális illesztést követően fennmaradó átlagos kulcsponttávolságot használtuk fel, ez a mennyiség ugyanis a szubjektív, vizuális ellenőrzés-szel szemben releváns numerikus összehasonlításnak bizonyult.

A 8. ábra az összehasonlítás eredményét demonstrálja 25 különböző RMB LiDAR pontfelhőn. A transzformáció pontosságát logaritmikus skálán jelenítettük meg. Az ábra adataiból láthatjuk, hogy az optimális stratégia a 8-kulcspontú megközelítés (4. (d) ábra), ami mindössze egy 0.15 - 0.5 méteres átlagos hibát eredményezett a tesztesetekre. Azt is megfigyelhetjük továbbá, hogy 1, illetve 4 kulcspontot használva a hiba némileg magasabb, mint a 8-pontú esetben, míg ezzel ellentétben, 16-kulcspontú stratégia esetén már túllilleszkedés jelenik meg, ami néhány esetben nagy hibákat eredményez.

A javasolt eljárást korábbi eljárásunkkal [16] összehasonlítva az új technika előnyeit bármilyen kulcspontválasztás esetén tapasztalhatjuk. Egyrészt, [16] során az optimális transzformációhoz csupán felülnézeti kép alapján vetített 2D objektumközéppontokat használtunk, ami csak egy hozzávetőleges illesztésre megfelelő a pontfelhők között.



**8. ábra.** A javasolt eljárás különböző kulcspontválasztási stratégiáinak kiértékelése és összehasonlítása a korábbi [16] megközelítésünkkel.

Másrésztől, [16]-ban nem használtunk semmilyen, a HD térképből kinyert objektum-specifikus tudást, ami a jelenlegi modellünkben nagyban hozzájárult a hibás párok kiszűréséhez. Ezenkívül, mivel az eljárásunk nem használ számításigényes, pont szintű NDT finomhangoló lépéseket, 10 fps sebességgel fut egy asztali gépen, ami jelentős különbség [16] 0.5 fps sebességéhez képest.

Javasolt módszerünk további előnye, hogy az illesztés pontosságát, illetve az algoritmus egyes lépéseit már autonóm viselkedésekre is közvetlenül fel tudtuk használni. A precíz regisztrációt követően először egy jól definiált költségfüggvénnyel, illetve [21] módszerével egy-több hozzárendelést végeztünk a LiDAR, illetve az MLS pontfelhő objektumai között. Ennek eredményét a kölcsönösen egyértelmű megfeleltetésekkel összehasonlítva (7. ábra) azt tapasztaltuk, hogy az egy-egy hozzárendelések adott városi helyszínen vagy néhány nagyon kedvezőtlen (7(c) ábra), vagy sok kis mértékben hibás párosítást okoznak (elcsúszott párosítás, 7(a) ábra), melyeket egy többszörös hozzárendelés viszont nagy hatékonysággal megold (7(b) és (d) ábrák).

A megfelelő párosítási eredmények alapján egyesíthetjük a LiDAR pontfelhő ugyanahhoz az MLS háttérobjektumhoz tartozó klasztereit, és az új objektumok felhasználásával még tovább hangolhatjuk a regisztrációs eljárást, valamint a párok halmazának komplementerét véve a jelenet pár nélküli Velodyne objektumait is kinyerhetjük a környezetből. Utóbbi halmazt egyrészt a dinamikus, változó objektumok – például *autó*, *gyalogos* –, másrészt olyan statikus objektumok alkotják, melyek az offline HD térkép felvétele óta kerültek a helyszínre (például új *oszlop*, *tábla*). A regisztráció és a párosítás kimenetelét így a jövőben *online* térképkészítési célokra is felhasználhatjuk.

## 6. Konklúzió

Cikkünkben egy objektum alapú precíz helymeghatározási algoritmust javasoltunk RMB LiDAR szenzorral felszerelt önjáró járművek (SDV) számára. Feltételezve, hogy a kör-

nyezetről elérhető egy nagy felbontású (HD) pontfelhőtérkép, például mobil lézerszkenner segítségével, a probléma egy jelentősen különböző sűrűségkarakterisztikájú pontfelhőregisztrációs feladattá módosul. Erre a HD térkép szemantikus információitól függetlenül különböző kulcspontválasztási stratégiát javasoltunk és hasonlítottunk össze. Végeredményben azt tapasztaltuk, hogy a 8-kulcspontú stratégia egy olyan hatékony megoldást ad a problémára, amely a többi kulcspontválasztási stratégiát, illetve a jelenleg ismert legjobb megoldást is felülmúlja, és pontossága már magas szintű autonóm funkciók számára is használható.

## Köszönetnyilvánítás

A szerzők köszönetet mondanak a Budapest Közút Zrt.-nek a Riegl VMX450 MLS tesztadatok biztosításáért. A projektet a Nemzeti Kutatási, Fejlesztési és Innovációs Alap (NKFIH KH-125681) és az Emberi Erőforrás Fejlesztési Operatív Program (EFOP-3.6.2-16-2017-00013) támogatta. Nagy Balázs részéről a közreműködés részben az Emberi Erőforrások Minisztériuma ÚNKP-18-3 kódszámú Új Nemzeti Kiválóság Programjának támogatásával történt.

## Hivatkozások

1. Nagy, B., Benedek, C.: Real-time point cloud alignment for vehicle localization in a high resolution 3D map. In: Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving at ECCV'18, Munich, Germany (2018)
2. Bayat, B., Crasta, N., Crespi, A., Pascoal, A.M., Ijspeert, A.: Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. *Current Opinion in Biotechnology* **45** (2017) 76 – 84
3. Kang, M., Hur, S., Jeong, W., Park, Y.: Map building based on sensor fusion for autonomous vehicle. In: International Conference on Information Technology: New Generations, Las Vegas, NV, USA (2014) 490–495
4. Seif, H.G., Hu, X.: Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry. *Engineering* **2** (2016) 159 – 162
5. Matthaei, R., Bagschik, G., Maurer, M.: Map-relative localization in lane-level maps for ADAS and autonomous driving. In: IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA. (2014) 49–55
6. Börcs, 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
7. 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: IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA (2012) 3033–3040
8. Yu, Y., Li, J., Guan, H., Wang, C.: Automated detection of three-dimensional cars in mobile laser scanning point clouds using DBM-Hough-Forests. *IEEE Trans. Geosci. and Rem. Sens.* **54** (2016) 4130–4142
9. Magnusson, M.: The Three-Dimensional Normal-Distributions Transform – an Efficient Representation for Registration, Surface Analysis, and Loop Detection. PhD thesis, Örebro University (2009)

10. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision* **13** (1994) 119–152
11. 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
12. Men, H., Gebre, B., Pochiraju, K.: Color point cloud registration with 4D ICP algorithm. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China (2011) 1511–1516
13. 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
14. Rusu, R.B., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D registration. In: *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan (2009) 3212–3217
15. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. *International Journal of Computer Vision* **89** (2010) 348–361
16. 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, IEEE (2016) 3374–3379
17. Ratha, N.K., Karu, K., Chen, S., Jain, A.K.: A real-time matching system for large fingerprint databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18** (1996) 799–813
18. Qi, C., Yi, L., Su, H., Guibas, L.: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: *Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA (2017) 5105–5114
19. Nagy, B., Benedek, C.: 3D CNN based phantom object removing from mobile laser scanning data. In: *International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA (2017) 4429–4435
20. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China (2011) 1–4
21. Zhu, H., Liu, D., Zhang, S., Zhu, Y., Teng, L., Teng, S.: Solving the Many to Many assignment problem by improving the Kuhn–Munkres algorithm with backtracking. *Theoretical Computer Science* **618** (2016) 30 – 41