LIDAR mérések időbeli felskálázása mono kamera alapján ⋆

Rózsa Zoltán[1,2], Szirányi Tamás[1,2]

[1] Gépi Érzékelés Kutatólaboratórium, Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI), Eötvös Loránd Kutatási Hálózat (ELKH), H-1111 Budapest, Kende u. 13-17., Hungary

[2] Anyagmozgatási és Logisztikai Rendszerek Tanszék (ALRT), Közlekedésmérnöki és Járműmérnöki Kar (KJK), Budapesti Műszaki és Gazdaságtudományi Egyetem (BME), H-1111 Budapest, Műegyetem rkp. 3., Hungary

{zoltan.rozsa,tamas.sziranyi}@sztaki.hu

Absztrakt. Az autonóm vezetésben használt 3D LIDAR szenzorok többsége lényegesen alacsonyabb frame rátával rendelkezik, mint az azonos járműre felszerelt modern kamerák. Ez a tanulmány megoldást javasol a LIDAR-ok időbeli frekvenciájának virtuális növelésére, mono kamera alapján. Ezáltal lehetővé téve a környezetben gyorsan mozgó, dinamikus objektumok sűrűbb nyomonkövetését. A rendszer először a dinamikus objektumjelölteket észleli és követi a kamera képeken. Ezután az objektumoknak megfelelő LIDAR pontokat azonosítja. Majd virtuális kamerapózokat számol ki, ezen pontok kamera képre történő visszavetítése, és követése alapján. Végül, a virtuális kamerapózokból az objektum mozgása (az objektumot képkockák között transzformáló mátrix) kiszámítható (a valós kamerapózok ismeretében) ahhoz az időpillanathoz, amelyhez nem állt rendelkezésre LIDAR mérés. A statikus tárgyak helyzetét is megbecsülhetjük ebben az időpillanatban ismerve az ego-mozgást. A javasolt módszert az Argoverse adatkészleten teszteltük, és teljesítményben felülmúlta a korábbi, hasonló célú módszereket.

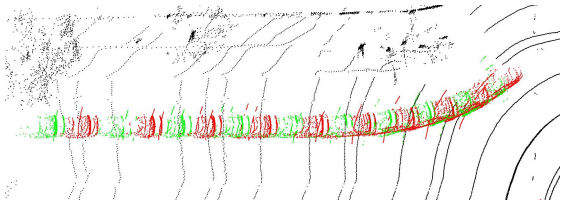
1. Bevezetés

A 3D LIDAR-ok alapvető elemeit képezik a legtöbb autonóm szállítási szenzorrendszernek mivel pontos mélységinformációt nyújtanak. Ezeknek a mintavételezési sebessége tipikusan 5-20 Hz tartományban mozog. Ha változtatható a szenzor szögsebessége (pl. Velodyne [1]) a vízszintes felbontás a forgási sebesség növelésével csökken. Sok esetben nem engedhető meg, hogy az időbeli felbontás növelése érdekében, csökkentsük a térbeli felbontást, ugyanis szeretnénk minél több részletet megismerni a környezetről. Például, a távoli objektumok 3D-ben való észleléséhez elengedhetetlen a nagy vízszintes felbontás. Ha megbecsüljük a LIDAR mérések közötti távolságokat (pontfelhőt), az nagyban javíthatja az egész rendszer 3D-s észlelési és nyomonkövetési képességeit.

⋆ A cikk angol nyelvű változata az ICIAP 2022 konferencia kiadványában (Lecture Notes in Computer Science) jelent meg ”Temporal Up-Sampling of LIDAR Measurements Based on a Mono Camera” címen (https://doi.org/10.1007/978-3-031-06430-2_5)

[1] https://velodynelidar.com

A gyakoribb mintavételezés helyett, a virtuális sűrítés segíthet a dinamikus objektumok növelt frekvenciájú megfigyelésében. Így lehetőségünk nyílik észlelni a gyors irányváltozásokat és időben reagálni azokra, elkerülve a veszélyes szituációkat. Emellett, megbecsülhető egy dinamikus objektum sebessége vagy a jövőbeli pozíciója is. Az autópályákon közlekedő járművek 130 km/h = 36 m/s sebességgel haladhatnak (vagy még gyorsabban), ami azt jelenti, hogy 5 Hz-es LIDAR mérési sebesség mellett egy jármű körülbelül 7,2 métert tud mozogni két képkocka között (és első észlelés esetében, nem tudjuk megbecsülni a sebességét, vagy megjósolni a mozgását). Emiatt szeretnénk minél előbb megtudni a tárgy következő pozícióját, annak érdekében, hogy ráépülő becslések is végrehajthatóak legyenek, így szükséges a LIDAR frame ráta növelése.



1. ábra:: Időben felskálázott objektum trajektória. A piros pontfelhők valós méréseket mutatnak, a zöldek a javasolt módszerrel generált virtuálisakat. Minden pont az első LIDAR koordinátarendszerében került illusztrálásra. Az első LIDAR frame környezeti pontjai fekete színnel lettek jelölve. A jobb láthatóság kedvéért nincs minden képkocka illusztrálva.

A LIDAR mellett az autonóm környezetérzékelő rendszerek másik létfontosságú eleme a kamera. A legtöbb esetben, az alkalmazott kamerák mérési sebessége legalább 30 FPS (pl. [2] [22]), de ennek akár többszöröse is lehet. Így sokkal gyakrabban nyerhető ki belőlük információ, mint a LIDAR-okból. A nagy frame ráta ellenére azonban a LIDAR-ok is szükségesek ahhoz, hogy pontos 3D információhoz jussunk, ezért javasolunk egy szenzorfúzión alapuló megoldást.

Ebben a cikkben, egy olyan módszert javasolunk valós mérések közötti virtuális LIDAR ponfelhők generálására (dinamikus objektumokra fókuszálva), amely a kamerákból sokkal gyakrabban kinyerhető információra épül. Egy eredményül kapott jármű trajektória az 1. ábrán látható, míg egy teljes pontfelhő a 4. ábrán.

Kontribúció: A cikk fő hozzájárulása egy új módszertan javaslata a LIDAR pontfelhők időbeli, kamera alapú felskálázására. A módszer előnyei a jelenleg elérhető pontfelhő predikciós módszerekkel (2.2. fejezet) szemben:

- A virtuális pontfelhő generálásához csak egy korábbi LIDAR mérés szükséges (a korábbi módszerek általában 5-öt használnak).
- Mivel a javasolt módszertan geometrián alapul, nincs szükség tanulásra.
- Nem függ a LIDAR felbontástól vagy a pontfelhő jellemzőitől.
- A metódus nagyobb tartományt tud feldolgozni valós időben, mint mások.
- A teljes folyamat valós idejű futtatása már magában foglalja az objektumdetekciót, a követést és az adatfúziót. Ezek végrehajtandó műveletek a legtöbb magasabb

szintű feldolgozási algoritmust megelőzően, például változádetekció, SLAM, online kalibráció stb. [1] [3].

Állításaink alátámasztására egy nagy nyilvános adatbázison (5. fejezet) készítettünk kiértékelést a módszer teljesítményéről.

2. Irodalmi áttekintés

A fejezet első részében a LIDAR felsklázással kapcsolatos szakirodalmak kerülnek bemutatásra, a második részben pedig az alternatív módszerek kerülnek kiértékelésre.

2.1. LIDAR mérési pontok felskálázása és interpoláció

A jelenleg elérhető LIDAR szenzorok felbontása mind időbeli, mind térbeli frekvencia tekintetében jóval elmarad a modern kamerákétól.

Míg egy kamera frame-nél általános, hogy több millió pixelből áll, a mai LIDAR-ok egy frame-hez tartozó mérési pontjainak száma a százezres tartományba esik. A LIDAR mérések térbeli felskálázása relatív hosszú múltra tekint vissza, és számos módszer áll rendelkezésre. A KITTI adatszett [21] készítői például folyamatos versenyt tartanak az ún. ”depth completion” feladatban, ami a probléma általánosított megnevezése. Elérhetőek hagyományos módszerek, mint a bilaterális szűrés [14] vagy a szemantikai alapú felskálázás [19], de manapság a mélytanulás alapú módszerek ([26] [7]) teljesítenek a legjobban ebben a feladatban.

A LIDAR és kamera mérések időbeli gyakoriságának aránya szintén alacsony. Ennek ellenére a szakirodalomban nem áll rendelkezésre módszer az időbeli felskálázásra.

A problémához legközelebb álló kutatások azok, amelyek a LIDAR felhők időbeli interpolációt célozzák. Ezeket azonban nem lehet összevetni a mi javaslatunkkal, mivel szükségük van egy jövőbeni (az online feldolgozás során nem elérhető) LIDAR framere a köztes frame generálásához. Ezen módszerek előfeldolgozási lépésként használhatók a miénkhez, ha a LIDAR és kamera adatok között szinkronizálási problémái áll fenn. A [11] és [10] módszerek célja a kamerák és LIDAR-ok szinkronizálási problémájának megoldása. A gyakorlatban (ha a szinkronizálás nem megoldott) a LIDAR képkocka időbélyegéhez legközelebbi kamera képkockát veszik figyelembe (lásd bővebben a 3.1. részben).

2.2. Pontfelhő predikció

A szakirodalomban megtalálhatóak módszerek, amelyek elméletileg képesek lennének helyettesíteni az ebben a cikkben javasolt virtuális LIDAR pontfelhő generálást. Ezeket a módszereket pontfelhő predikcióra tervezték. Az olyan módszerek, mint a [4] [24] [23], jellemzően mélytanuláson alapulnak, és körülbelül 5 korábbi képkockát használnak a soronkövetkező mérés predikciójára. Ezen módszerek használatának azonban számos hátránya van:

– Több korábbi képkocka szükséges. (Ha egy objektum csak az utolsó képkockában jelenik meg, nagy valószínűséggel statikusnak feltételezik vagy nagy hibával jelzik előre annak mozgását.)

– Csak korábbi információkat használnak fel, így nem tudnak megbirkózni a nagy gyorsulásokkal vagy irányváltásokkal. Így elméletileg sem érhetiek el az olyan módszerek pontosságát, mint a miénk, amely aktuális információt használ fel a virtuális mérések generálásához.
– Ezeket a módszereket tanítani kell, különböző jellemzőkkel vagy felbontással rendelkező LIDAR pontfelhőkre pedig újratanítani.
– A predikció alapú módszerek (kvázi) valós időben is csak kis távolságra működnek. A mielőbbi távol objektum detekció azonban elengedhetetlen. [17].

3. ELőZMÉNYEK

Mielőtt rátérnénk a javasolt módszer részleteire, ismertetünk néhány szükséges előzetes ismeretet.

3.1. LIDAR-kamera szinkronizálás

Ahogy korábban említettük, a gyakorlatban, a legtöbb esetben a LIDAR képkockához az időbélyegéhez legközelebbi időbélyeggel rendelkező kamera képkockát rendeljük hozzá. Ennek a feltételezésnek a helyessége több tényezőtől is függ. Egyrészt a forgási sebességtől, a LIDAR képkockán belüli adatpontok is eltérő időbélyeggel rendelkeznek a valóságban. Ily módon az ego-mozgás torzítást vihet a mérésünkbe, mivel az első mért pont lényegesen eltérő nézőpontból mérhető, mint az utolsó. Mindazonáltal, a megfelelő szenzor (nagyfrekvenciás lokalizációra alkalmas) használatával, és olyan módszerekkel, mint [6], egy képkocka összes adatpontja azonos koordinátarendszerbe transzformálható. Más dinamikus objektumok is okozhatnak problémát, de ha ugyanazon jármű pontjairól beszélünk (mivel ezek közel vannak egymáshoz) az adatpontjaik mérései között elhanyagolható lesz az időkülönbség. A fennmaradó hibát a kamera és a LIDAR mérési frekvenciájának eltérése okozza. 0,03 s mintavételi idővel (kb. 30 FPS) rendelkező kamerát feltételezve azt kapjuk, hogy a LIDAR képkocka időbélyege csak kb. 0,015 s-mal térhet el a legközelebbi kamera képkocka bélyegétől (nagyobb FPS esetén még kevesebbel). Ezek azok az okok, ami miatt a gyakorlatban is elfogadott, hogy az időben legközelebbi kamera képkockát rendeljük az adott LIDAR frame-hez. Mivel ez a pontosság áll alapigazságként rendelkezésre, ezért becslésünkkel mi is ezt a pontosságot kívánjuk elérni.

3.2. Időpillanat jelölések

A dolgozatban az időpillanatok két jelölését különböztetjük meg. A T_{t-1} a referencia időbélyeget, a T_t pedig az aktuális időbélyeget jelzi (a $t \in \mathbb{N}^+$ index a t-edik mérést jelöli). Különböző szenzorok esetében összefoglaljuk ezek értelmezését:

T_{t-1} időpillanat:

– *Kamera:* Kamera esetén a T_{t-1} az utolsó elérhető teljes LIDAR képkockának megfelelő kamera frame időbélyegére hivatkozunk.

– *LIDAR:* A LIDAR esetében a T_{t-1} az érzékelő utolsó elérhető teljes körülfordulás (képkocka) időbélyegét jelenti. A feltételezésünkben ez megegyezik a kamera T_{t-1} értékével. (További információ a kamera és a LIDAR T_{t-1} figyelmen kívül hagyott időbeli eltéréseiről a 3.1. részben található.)

T_t időpillanat:

– *Kamera:* A legegyszerűbb esetben a T_t a T_{t-1} után következőleg elérhető kamerakép időbélyege. Ha nem az elérhető legmagasabb frekvenciára skálázunk, akkor a köztük eltelt idő (lásd az 1. egyenletet további magyarázatért) eltérő lehet a kamera mintavételezési gyakoriságától.
– *LIDAR:* Pontosan ugyanaz a virtuális időbélyeg, mint a T_t kamera esetén. Ebben az időpillanatban alapesetben nem áll rendelkezésre teljes körülforduláshoz tartozó LIDAR mérés. Ezért generálunk egy virtuális mérést, $P_{t,v}$-t erre az időpontra.

A két időpillanat közötti időkülönbség:

$$\Delta t = T_t - T_{t-1} = \frac{n}{f_C} \tag{1}$$

Ez azt jelenti, hogy az eltelt idő $n \in \mathbb{N}^+$ és a kamera mintavételi frekvenciájának f_C hányadosa. n választható igényünk és számítási teljesítményünk alapján. A kamera és a LIDAR mintavételi frekvencia aránya korlátozza a maximumot: $n < \frac{f_C}{f_L}$. $n = 1$ a lehető legmagasabb mintavételezést eredményezi, a legnagyobb számításigénnyel. $n = \frac{f_C}{f_L}$ nem eredményez felskálázást; ezt az értéket használjuk a kvantitatív értékelési részben, mivel ebben az esetben a becsléshez tartozik valós (ground truth) mérés.

A következőkben bemutatjuk módszerünket virtuális pontfelhő generálására a T_t időpillanatra. Ha megismételjük ezeket a lépéseket tetszőleges t és $t-1$ esetén, akkor felül mintavételezett LIDAR mérést kapunk eredményül.

4. A JAVASOLT MÓDSZER

A következőkben részletesen ismertetjük a javasolt módszert. A rendszer folyamatábrája a 2. ábrán látható. A virtuális LIDAR frame generálás (időbeli felskálázás) öt lépése:

1. Dinamikus objektumjelöltek detekciója (T_{t-1} időpillanat)
2. A detektált objektumoknak megfelelő LIDAR pontok meghatározása, és a kamera képre való visszavetítése (T_{t-1} időpillanat)
3. Az objektumok és pontjaik követése a kamera képén (T_t időpillanatig)
4. Virtuális kamera mozgás számítása a követett pontok alapján objektumonként
5. LIDAR pontok transzformációja a becsült transzformációs mátrixok segítségével

4.1. Dinamikus objektum detekció RGB képeken

Kísérleteink során a Yolo_v2 [16] hálózatot használtuk jármű detekcióra. (Ez a lépés bármilyen más detektorral helyettesíthető.) Példák a detekcióra a 2. ábrán láthatók. A tesztek során csak járművekre fókuszáltunk (de bármely kategóriával kibővíthető), az

alábbi okok miatt: A két leggyakoribb közlekedési résztvevő és dinamikus objektum az utakon a járművek és a gyalogosok. A gyalogosok, akik körülbelül $5km/h$ $1,4m/s$ sebességgel sétálnak, a legalacsonyabb LIDAR mérési sebesség mellett ($5Hz$) körülbelül $0,28m$-t mozoghatnak (a mintavételezés $10Hz$-ra növelése ennek a felét jelentené - $0,14m$). Ez a $10cm$-es tartomány (nagyobb LIDAR mérési frekvencia esetén még kisebb) hasonló a LIDAR-ok pontosságához, így a gyalogos pontok jó közelítéssel statikusnak tekinthetők ebben a kis időtartamban. Ezzel ellentétben az autók ez idő alatt irányt változtathatnak és jelentős mértékben mozoghatnak (lásd a leírt példát az 1.. fejezetben).

A járművekre többször is dinamikus objektumjelöltekként hivatkozunk. Ennek az az oka, hogy minden egyes járműnél kiszámítjuk a transzformációt, amely az előző LIDAR koordináta rendszeréből képezi le az aktuális (virtuális) koordináta rendszerre. Parkoló járművek esetében ez az ego-mozgásból adódó transzformációra lesz. Azonban az előzetes ismeretektől függetlenül érdemes az összes autót dinamikus objektumjelöltként kezelni, mivel azt nem tudjuk előre megmondani, hogy egy parkoló jármű nem indul-e el, illetve az is előfordulhat, hogy nem rendelkezünk korábbi információval a jármű statikus vagy dinamikus állapotáról.

4.2. LIDAR objektum pontok visszavetítése a kamera képre

Első lépésben a belső kamera kalibrálációt [25] és a LIDAR-kamerakalibrációt [27] szükséges elvégezni.

Ebben az alfejezetben a T_{t-1} időbélyeggel rögzített mérésekkel foglalkozunk. Van egy pontfelhőnk a P_{t-1} LIDAR érzékelőtől és egy I_{t-1} RGB kép, amelyet egy kamera vett fel. Feltételezzük, hogy T_{t-1} és T_t között eltelt idő azonos a kamera mintavételezési idejével, vagyis a lehető legmagasabb felskálázást célozzuk (1. egyenlet).

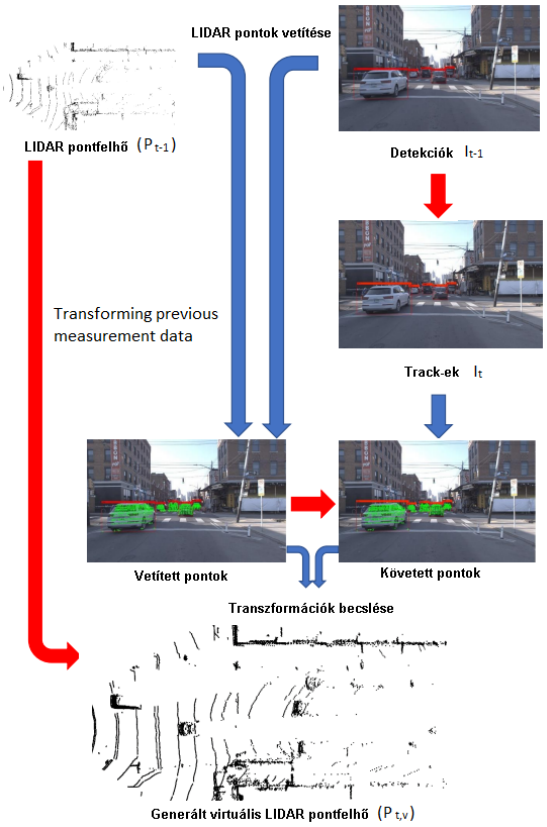
Ismerve a $\mathbf{K}$ belső mátrixot és a LIDAR és a kamera koordinátarendszerek közötti $\mathbf{T}_{L,C}$ transzformációs mátrixot, 3D LIDAR adatpontokat vetíthetünk a 2D képre. Ezek közül kiválasztásra kerülnek azok a LIDAR adatpontok, amelyek a kamera előtt vannak és kép határain belülre vetülnek, és csak ezekkel foglalkozunk a továbbiakban. Második lépésben minden detektált objektumhoz kiválasztjuk a hozzátartozó LIDAR pontokat, ezek vetületei az objektum 2D határoló téglalapjába esnek. Ezek a pontok több objektumhoz is tartozhatnak 3D-ben, így euklideszi klaszterezéssel [18] kerülnek kiválasztásra a végső objektum pontok, amik a legnagyobb klaszterhez tartoztak. Az egyes objektumok LIDAR-ból visszavetített 2D pontjainak illusztrációja a 2. ábrán látható az utolsó előtti sor bal oldalán.

4.3. Objektumok és pontok követése RGB képeken

Ez az alfejezet két részre oszlik. Az első rész az objektumszintű követést ismerteti; a második rész a pontszintű követéssel foglalkozik. Innestől fogva olyan adatfeldolgozásról beszélünk, amelyet a T_t időpillanatban szükséges végrehajtani. Természetesen rendszerünk minden frame-n ugyanazt a detektort használja.

Az aktuális képkockán detektált és a megelező képkockáig trackelt objektumok összerendelésére (határoló téglalapjaik alapján) a magyar [12] algoritmust használjuk.

Az előző képkockából (ahol a LIDAR és a kamera mérése is elérhető volt) származó (a jelenlegiben is azonosított) objektumokhoz az utolsó LIDAR mérésből hozzárendelt

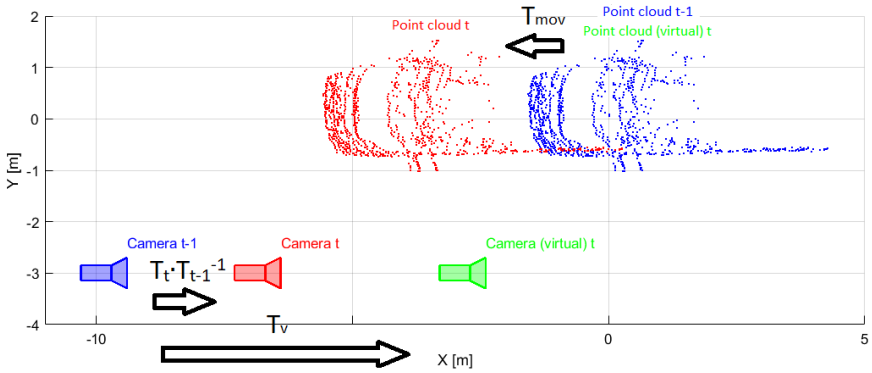


2. ábra:: A javasolt módszer áttekintése. Képek és pontfelhők illusztrálják a folyamat különböző állapotait. A nyilak a folyamatokat jelzik, a pirosak a különböző mérési idővel rendelkező adatokon keresztüli folyamatokat jelölik.

pontok tartoznak. Ezeket a pontokat a Kanade-Lucas-Tomasi (KLT) algoritmus [8] követi az aktuális frame-ig.

4.4. 3D transzformációk becslése

Ennek a lépésnek az előfeltétele, hogy ismerjük a T_{t-1} és T_t statikus hátterek közötti koordináta-transzformációt. Ez kiszámítható a két méréshez tartozó lokalizációs szenzor adatokból (kísérleteinkben az Argoverse [2] adatkészlet által biztosított GPS-alapú pózokat használtuk), vagy alternatívaként meghatározható kizárólag a képekből pl., olyan módszerekkel, mint az ORB-SLAM [13]. $\mathbf{T}_{t-1} = [\mathbf{R}_{t-1}|\vec{t_{t-1}}]$ és $\mathbf{T}_t = [\mathbf{R}_t|\vec{t_t}]$ homogén transzformációs mátrixok, amik T_{t-1} és T_t kamera koordináta rendszerekből a globális koordináta rendszerekbe transzformálnak, az $\mathbf{R}$-k és $\vec{t}$-k rendre forgatási mátrixokat és transzlációs vektorokat jelölnek. (Ha két kamera pozícióról beszélünk, akkor az esetek többségében az első kamera koordinátarendszerét szokás globális ko-



3. ábra:: A dinamikus objektum pontfelhők (és transzformációk) számításának szemléltetése (abban az időpillanatban amikor csak kamera mérés érhető el), egyszerű esetben: előrefelé haladó ego mozgással és szemben mozgó járművel. Az autók kamera pozíciói és pontjai a globális koordináta-rendszerben kerültek ábrázolásra, kék színnel a T_{t-1} időpontban, pirossal pedig a T_t időpontban. $\mathbf{T}_{t-1}$ és $\mathbf{T}_t$ (a valós kamerák koordinátarendszere és a globális koordinátarendszer közötti transzformációk) ismertek. Az ábrázolt autó (i-edik objektum) $\mathbf{T}_{mov}$ mozgását a globális koordinátarendszerben leíró transzformációt keressük. Ennek érdekében feltételezzük, hogy az autó statikus, és egy transzformációs mátrixot becsülünk a T_{t-1} időpontban lévő valós kamera és egy virtuális kamera (zöld színű) között T_t, $\mathbf{T}_v = \mathbf{T}_{t,i} \cdot \mathbf{T}_{t-1,i}^{-1}$ a I_t képpontjai alapján (ami I_{t-1}-ből lett nyomon követve). Ezekből az adatokból a $\mathbf{T}_{mov}$ kiszámítható. Megfigyelhető, hogy a virtuális kamera lokális koordinátarendszerében az objektumpontok ugyanazzal a koordinátával rendelkeznek mint a valósban, mivel ezek a koordinátra rendszerek megegyeznek.

ordinátarendszerként definiálni, de mi a következőkben továbbra is ezt az általános terminológiát használjuk.)

Minden egyes követett objektum esetében megbecsülhetünk egy virtuális kamerapózt a 3D-2D (3D P_{t-1} és 2D I_t) pontpárok alapján (4.3. alfejezet) . Kísérleteink során legalább 50 pontpár egyezéssel rendelkező track-eket értékeltük ki (így kiszűrve a nagyon távoli objektumokat). A [5] módszert használtuk az MLESAC [20] robusztus becslővel a Perspektív-n-pont probléma megoldására. A transzformációk illusztrációja (és egy egyszerű eset az objektum és kamera mozgást tekintve) a 3. ábrán látható. A becsléssel $\mathbf{T}_{D,i}$ transzformációs mátrixokat kapunk a LIDAR (T_{t-1}) a virtuális kamerák (az T_t) között az i-edik dinamikus objektumra.

$$\mathbf{T}_{t,i} = \mathbf{T}_{L,C} \cdot \mathbf{T}_{D,i} \cdot \mathbf{T}_{L,C}^{-1} \tag{2}$$

Végezetül, ha meg szeretnénk határozni az i-edik 3D objektum valós mozgását ($\mathbf{T}_{mov,i}$) leíró transzformációs mátrixot a globális koordinátarendszerben, ezt a valós kamerakoordináta-rendszer és az $i$edik virtuális kamera koordinátarendszere közötti egyenlőséget leíró egyenlet átrendezésével kaphatjuk meg:

$$\mathbf{T}_t \cdot \mathbf{T}_{mov,i} \cdot \mathbf{T}_{t-1}^{-1} = \mathbf{T}_{t,i} \cdot \mathbf{T}_{t-1,i}^{-1} \tag{3}$$

4.5. (T_{t-1} időpillanathoz tartozó) LIDAR pontfelhő transzformációja T_t időpillanatra, a becsült transzformációs mátrixok használatával

Ebben a lépésben kétféle pontfelhőt különböztetünk meg. Az első típusú pontfelhő a statikus háttérnek felel meg. Ebbe a kategóriába tartozik az összes pont, amely vetülete kívül esett a dinamikus objektumok befoglaló téglalapjain. Ezek a következő mátrix segítségével transzformálhatok T_t virtuális időbélyegre:

$$\mathbf{T}_S = \mathbf{T}_{L,C}^{-1} \cdot \mathbf{T}_t \cdot \mathbf{T}_{t-1}^{-1} \cdot \mathbf{T}_{L,C} \tag{4}$$

A fennmaradó pontokhoz különböző transzformációs mátrixokat használunk aszerint, hogy melyik dinamikus objektumjelölteknek felelnek meg:

$$\mathbf{T}_{D,i} = \mathbf{T}_{L,C}^{-1} \cdot \mathbf{T}_t \cdot \mathbf{T}_{mov,i} \cdot \mathbf{T}_{t-1}^{-1} \cdot \mathbf{T}_{L,C} \tag{5}$$

$\mathbf{T}_S$ és $\mathbf{T}_{D,i}$-s használatával a megfelelő pontokhoz létrehozhatjuk a $P_{t,v}$ virtuális LIDAR pontfelhőnket T_t időpillanatban (amikor eredetileg nem volt LIDAR mérésünk). Meg tudjuk határozni ezeket a transzformációs mátrixokat és transzformálni tudjuk az előző LIDAR mérést az összes kamera mérés idejére, így időben felskálázhatjuk a LIDAR méréseinket.

5. TESZT EREDMÉNYEK

Módszerünket az Argoverse [2] adatkészletén teszteltük, amely egy nagy nyilvános adatbázis az autonóm vezetéshez. Összehasonlítottuk a javasolt módszerünk pontosságát state of the art pontfelhő prediktáló módszerekkel, mint [4] és a [23]. Chamfer távolságot (CD) és Earth Movers távolságot (EMD) használtunk a hiba mérésre, amelyeket a hivatkozott szakirodalmak is alkalmaznak. A CD méri a távolságot a prediktált pontfelhő és a ground truth pontfelhő legközelebbi szomszéd pontpárjai között mindkét irányban, míg az EMD leírja a megjósolt pontfelhő és a ground truth pontfelhő hasonlóságát a globális illesztési probléma költségének kiszámításával.

A Chamfer távolság definíciója két pontfelhő között:

$$CD = \frac{1}{N_t} \sum_{p_{t,v} \in P_{t,v}} \min_{p_t \in P_t} ||p_t - p_{t,v}||^2 + \frac{1}{N_t} \sum_{p_t \in P_t} \min_{p_{t,v} \in P_{t,v}} ||p_t - p_{t,v}||^2 \tag{6}$$

ahol $p_{t,v} \in \mathbb{R}^{N_t \times 3}$ és $p_t \in \mathbb{R}^{N_t \times 3}$ a prediktált (virtuális) $P_{t,v}$ és a ground truth P_t pontfelhők pontjai.

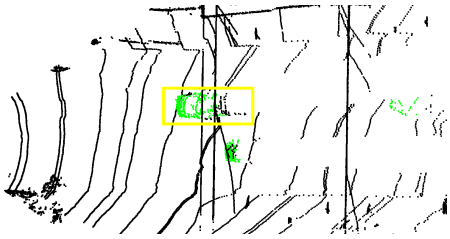
Az Earth Movers távolság két pontfelhő között:

$$EMD = \min_{\phi: P_t \to P_{t,v}} \frac{1}{N_t} \sum_{p_t \in P_t} ||p_t - \phi(p_t)||^2 \tag{7}$$

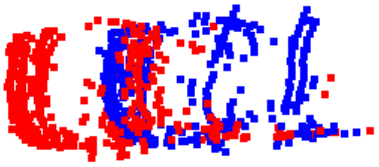
ahol a ϕ bijekció. Az EMD kiszámítja a pont-pont leképezést két pontfelhő P_t és $P_{t,v}$ között a legközelebbi globális szomszédok megtalálásával.

(a) T_{t-1} pontfelhő (a detektált dinamikus objektumok kék színűek)

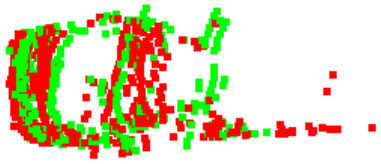
(b) T_t ground truth pontfelhő (a detektált dinamikus objektumok pirossal vannak színezve)



(c) A T_t becsült pontfelhője (a dinamikus objektumok zölddel vannak színezve)



(d) Sárgával kiemelt jármű T_{t-1} (kék) és T_t (piros)



(e) Sárgával kiemelt jármű ground truth (piros) és becsült (zöld) pozíciók T_t

4. ábra:: A javasolt módszer pontfelhő-becslésének illusztrációja. (a) mutatja az utolsó LIDAR frame-t (P_{t-1}), (b) a ground truth-t és (c) a pontfelhő becslést T_t időpillanatban. A (d) kiemeli a jármű pontokat a T_{t-1} és T_t időpillanatokban. Míg (e) ground truth pontfelhőt a becslésünkkel együtt jeleníti meg.

Az értékeléshez az Argoverse adatkészlet (LIDAR és első középső kamera) összes tesztsorozatát (4168 képkocka) használtuk. Esetünkben a hibametrikákat csak a dinamikus objektum jelöltekre számoltuk ki, mivel a becslés a pontfelhőnek erre a részére terjedt ki (a többi rész transzformációja előzetes tudással került meghatározásra). Az eredményeket az 1. táblázat foglalja össze.

Az 1. táblázatban szereplő mély tanulási módszerek 5 korábbi LIDAR képkockát használtak az egymást követő becsléshez, míg mi csak egy korábbi LIDAR frame-t. Az aktuális kamera információira támaszkodva azonban a pontosság egy másik szintjét értük el. Mivel virtuális pontfelhőnket valós mérésekből, transzformációval állítjuk elő, az EMD érték (pontfelhők hasonlósága) jóval kisebb, mint a versenytársaknál.

1. táblázat:: A javasolt algoritmus kvantitatív kiértékelése.

Methods	CD [m^2]	EMD [m^2]
Copy last input	0.5812	1.0667
PointNet++ · LSTM [15]	0.3826	1.0011
PointCNN · ConvLSTM [9]	0.3457	0.9659
[4]'s PointGRU	0.2994	0.9084
[4]'s PointLSTM	0.2966	0.8892
[4]'s PointRNN	0.2789	0.8964
[23]'s proposal	0.3010	0.9010
[23]'s alignment	0.5250	0.7710
Proposed	**0.1983**	**0.3007**

Számításigény: A módszert Matlab környezetben implementáltuk, konfiguráció: Intel Core i7-4790K @ 4,00 GHz processzor, 32 GB RAM, Nvidia GTX 1080 GPU, Windows 10 64 bit. A feldolgozási idő nagymértékben függ a képfelbontástól (de bizonyos rendszerelemeknél - pl. detekció - kisebb felbontás is használható). Az eredeti 1920x1200 felbontású Argoverse képek eredeti méretének 25 %-ára kicsinyítésével meg tudtunk tartani minden fontos detekciót az értékeléshez. A fenti módszerek csak alulmintavételezett és kivágott pontfelhőket használnak a valós idejű futás biztosítására (például a [23] körülbelül 5 FPS sebességről számolt be n NVIDIA RTX2080Ti GPU-val rendszerükhöz, mindössze 4096 adatponttal). A teljes rendszerünkre e nélkül 105 ms átlagos futási időt mértünk, vagyis a jelenlegi korlátozott, kutatási konfigurációval időben felskálázhatunk 5 Hz-es LIDAR méréseket.

[23] és az összes többi fenti módszer (ami [4]-ben került kiértékelésre) levágta a képkockákat *[10m × 10m]*, *[-5m, 5m]* tartományban (*x* és *y* irányban is). Mivel mi első kamerát használtunk, így az *x* irány esetén a *[0m, 10m]* tartományon belüli középpontú objektumokat értékeltük ki. Megjegyzés: a távolság növelésével nehezebb becslési feladatot kaptunk.

6. Konklúzió

Ez a dolgozat egy új megközelítést mutatott be a LIDAR mérések időben felskálázására monokamerás alapon. A módszer state of the art pontosságot és valós mérésekhez való hasonlóságot biztosít mindösszesen két kamera frame és egy megelőző LIDAR frame felhasználásával. A jövőben más adatkészletek tesztelését tervezzük, a módszer kiterjesztését a kitakart objektumok becslésére, és tovább tervezzük javítani a generált pontfelhő jellemzőinek valós mérésekhez való hasonlóságát.

Köszönetnyilvánítás

A Kulturális és Innovációs Minisztérium ÚNKP-22-4-II-BME-54 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával készült.

A munkát a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal (NKFIH) OTKA K 139485 pályázata támogatta. A publikációban szereplő kutatást a SZTAKI az Európai Unió támogatásával valósította meg, az Autonóm Rendszerek Nemzeti Laboratórium keretében. (RRF-2.3.1-21-2022-00002)

Irodalom

1. Benedek, C., Majdik, A., Nagy, B., Rozsa, Z., Sziranyi, T.: Positioning and perception in LIDAR point clouds. Digital Signal Processing p. 103193 (2021)
2. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., Hays, J.: Argoverse: 3D tracking and forecasting with rich maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
3. Debeunne, C., Vivet, D.: A review of visual-LiDAR fusion based simultaneous localization and mapping. Sensors **20**(7) (2020)
4. Fan, H., Yang, Y.: Pointrnn: Point recurrent neural network for moving point cloud processing. arXiv **1910.08287** (2019)
5. Gao, X.S., Hou, X.R., Tang, J., Cheng, H.F.: Complete solution classification for the perspective-three-point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(8), 930–943 (2003)
6. He, L., Jin, Z., Gao, Z.: De-skewing lidar scan for refinement of local mapping. Sensors **20**, 1846 (03 2020)
7. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Towards precise and efficient image guided depth completion (2021)
8. Kalal, Z., Mikolajczyk, K., Matas, J.: Forward-backward error: Automatic detection of tracking failures. In: 20th International Conference on Pattern Recognition. pp. 2756–2759 (2010)
9. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on X-transformed points. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
10. Liu, H., Liao, K., Lin, C., Zhao, Y., Guo, Y.: Pseudo-LiDAR point cloud interpolation based on 3D motion representation and spatial supervision. IEEE Transactions on Intelligent Transportation Systems pp. 1–11 (2021)
11. Liu, H., Liao, K., Zhao, Y., Liu, M.: Plin: A network for pseudo-LiDAR point cloud interpolation. Sensors **20**, 1573 (03 2020)
12. Miller, M.L., Stone, H.S., , Cox, I.J., Cox, I.J.: Optimizing murty's ranked assignment method. IEEE Transactions on Aerospace and Electronic Systems **33**, 851–862 (1997)
13. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
14. Premebida, C., Garrote, L., Asvadi, A., Ribeiro, A., Nunes, U.: High-resolution LIDAR-based depth mapping using bilateral filter. pp. 2469–2474 (11 2016)
15. Qi, C., Yi, L., Su, H., Guibas, L.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NIPS (2017)
16. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6517–6525 (2017)
17. Rozsa, Z., Sziranyi, T.: Object detection from a few LIDAR scanning planes. IEEE Transactions on Intelligent Vehicles **4**(4), 548–560 (2019)
18. Rusu, R.B.: Semantic 3D object maps for everyday manipulation in human living environments. KI - Künstliche Intelligenz **24**(4), 345–348 (2010)

19. Schneider, N., Schneider, L., Pinggera, P., Franke, U., Pollefeys, M., Stiller, C.: Semantically guided depth upsampling. In: Rosenhahn, B., Andres, B. (eds.) Pattern Recognition. pp. 37–48. Springer International Publishing, Cham (2016)
20. Torr, P.H.S., Zisserman, A.: MLESAC: A new robust estimator with application to estimating image geometry. Computer Vision and Image Understanding **78**, 138–156 (2000)
21. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: International Conference on 3D Vision (3DV) (2017)
22. Wang, P., Huang, X., Cheng, X., Zhou, D., Geng, Q., Yang, R.: The apolloscape open dataset for autonomous driving and its application. IEEE transactions on pattern analysis and machine intelligence (2019)
23. Wencan, C., Ko, J.H.: Segmentation of points in the future: Joint segmentation and prediction of a point cloud. IEEE Access **9**, 52977–52986 (2021)
24. Weng, X., Wang, J., Levine, S., Kitani, K., Rhinehart, N.: Inverting the Pose Forecasting Pipeline with SPF2: Sequential Pointcloud Forecasting for Sequential Pose Forecasting. CoRL (2020)
25. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(11), 1330–1334 (2000)
26. Zhao, S., Gong, M., Fu, H., Tao, D.: Adaptive context-aware multi-modal network for depth completion. IEEE Transactions on Image Processing **30**, 5264–5276 (2021)
27. Zhou, L., Li, Z., Kaess, M.: Automatic extrinsic calibration of a camera and a 3D LiDAR using line and plane correspondences. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5562–5569 (2018)