

Valós idejű járműdetekció LIDAR pontfelhősorozatokon

Börcs Attila^{1,2}, Nagy Balázs¹ és Benedek Csaba¹

¹ Elosztott Események Elemzése Kutatólaboratórium, Magyar Tudományos Akadémia, Számítástechnikai és Automatizálási Kutatóintézet

² Irányítástechnika és Informatika Tanszék, Budapesti Műszaki és Gazdaságtudományi Egyetem

{vezetéknév.keresztnév}@sztaki.mta.hu

Absztrakt. Járművek automatikus felismerése zsúfolt városi környezetben kihívásokkal teli feladatnak tekinthető a robot látás és érzékelés problémáitól kezdve egészen az önjáró járművek látórendszereinek a fejlesztéséig. Cikkünkben egy modell alapú megoldást ajánlunk, ami járművek felismerésére képes háromdimenziós pontfelhősorozatokon. Munkánk során egy autóra szerelhető Velodyne HDL-64 S2 típusú LIDAR lézerszkenner által szolgáltatott pontfelhősorozatokon dolgoztunk. Az általunk kifejlesztett keretrendszer fogadja a berendezésből érkező nyers pontfelhőfolyamot, és a következő három fő feldolgozási lépést hajtja végre rajta: 1) Objektum detekció: összetartozó háromdimenziós pontok halmazainak a kinyerése, amik az utcai környezetben elhelyezkedő egyes objektumokhoz tartoznak. 2) 3D-s leírók előállítás: járművek felismerésére használható jellemzők valós idejű kinyerése. 3) Járműfelismerés: az előzőleg előállított jellemzők alapján az objektum pontfelhők bináris osztályozása. Az összesen 2690 járművet tartalmazó adathalmazon kvantitatívan és kvalitatívan igazoltuk, hogy az általunk kifejlesztett 3D-s alakleírók jelentős sebesség növekedést és pontosabb felismerést biztosítanak a szakirodalmi főkomponens analízis alapú jármű detekciós algoritmusokkal szemben.

1. Bevezetés

¹Napjainkban az automatikus objektumfelismerési feladatok központi szerepet töltenek be a robot látás és érzékelés kutatásában. A vizuális felismerési módszereknek számtalan alkalmazási területe ismert, az önjáró autók látórendszerétől elkezdve, a vezetés segítő eszközökön át, egészen az automatikus ütközésselhárító rendszerekig [3, 4]. A jövőbeli járművekbe építhető számítógépes látórendszerek számtalan módon nyújtanak segítséget társadalmi szinten. Segítségükkel megelőzhetőek és csökkenthetőek lesznek a közúti balesetek, hiszen ezek a rendszerek képesek lesznek folyamatosan megfigyelni a jármű környezetét ezzel is segítséget és nagyobb komfortot biztosítva a járművezetőknek. A kültéri lézeres mérőberendezések - mint például a LIDAR lézerszkenner - különösen fontos eszközeivé váltak az automatikus megfigyelési feladatokhoz kapcsolódó adatgyűjtésnek, hiszen képesek valós idejű és nagy kiterjedésű háromdimenziós mérési adatot szolgáltatni a környezetről. Ezen lézeres mérőberendezések fő előnyei, hogy pontos

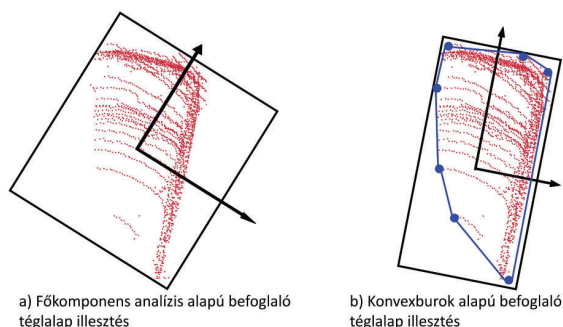
¹ A cikkben közölt eredmények eredetileg angol nyelven, az *ECCV 2014* [1] és *ACCV 2014* [2] konferenciák kiadványaiban jelentek meg.

háromdimenziós geometriai információt biztosítanak a helyszínről. A lézeres LIDAR rendszerek ezenfelül több hasznos tulajdonsággal is rendelkeznek a hagyományos optikai kamerákhoz képest: 1) nem érzékenyek a változó kültéri fényviszonyokra 2) nagyobb látószögben és távolságról képesek adatot gyűjteni 3) a technológia sajátossága miatt éjszaka is megbízhatóan használhatók. Cikkünkben különböző városi területeken (*szűk mellékutca, főút, útkereszteződés*) rögzített nagy kiterjedésű pontfelhősorozatokban elhelyezkedő járművek felismerésével foglalkozunk. Az adatrögzítéshez egy Velodyne HDL-64 S2 típusú földi lézerszkennert használtunk. A pontfelhő folyamatokban történő valós idejű objektum felismerés kihívásokkal teli feladat több okból is. Elsősorban a berendezésből érkező adat zajjal terhelt, és számtalan olyan régió található a mért helyszínről, ahol a pontfelhő hiányos. Másodsorban, zsúfolt városi környezetben gyakran előfordul, hogy az egyes járművek, gyalogosok és egyéb utcai objektumok takarásba kerülnek. A takarásban lévő objektumok kinyert alakzatai gyakran hiányosak, vagy több darabra esnek a szét a pontfelhőben. Végül számolnunk kell a Velodyne LIDAR lézerszkennert tipikus mérési karakterisztikájának kihívásaival, mint például az erősen csökkenő pontfelhő sűrűség a szenzortól távol eső régiókban [5], ami miatt bizonyos fajta objektumok (például járművek) többféle alakban és geometriai sajátosságokkal jelenhetnek meg a mérésben, ezzel megnehezítve a felismerési eljárást. További nehézségek lépnek fel, ha olyan felismerési algoritmust akarunk megvalósítani ami valós időben működik, ugyanis ilyen esetben erősen számítás igényes feladatot kell hogy végrehajtsunk, egy nagyon szűk időtartományon belül.

1.1. Szakirodalmi áttekintés

A szakirodalomban számos módszer található, amelyek lézerszkennerral mért háromdimenziós adatokon kínálnak megoldást különböző felismerési feladatokra. A hatékony alakleírók (*jellemzők*) kinyerése esszenciális részét képezik a publikált eljárásoknak, tipikusan a következő két stratégiát felhasználva: Az *első* stratégia szerint a felismerendő objektumok méretének a becslése 3D-s befoglaló téglatestek segítségével történik. A [6] munkában a szerzők eljárást dolgoztak ki objektumok osztályozásához és követéséhez. Az algoritmus alapötlete egy oktális fa alapú rács struktúra, aminek a segítségével egy adott 3D-s pont szomszédsága származtatható. A módszer a kinyert lokális térrészekben elhelyezkedő szomszédos pontokra próbál egy 3-D befoglaló téglatestet illeszteni, majd ezen téglatest oldal arányait - mint jellemzőt - használja fel az objektumok osztályozásához, úgy mint *gyalogos, kerékpáros, jármű*. Esetünkben a megfigyelt környezetről készített mérés komplex városi szcenáriókat tartalmaz, változatos objektumtípusokkal, mint növényzet, oszlopok, közlekedési táblák és takarásban lévő objektumok. Az itt bemutatott jellemzők alkalmasak lehetnek egyszerűbb városi színtereken történő objektum felismerésre, viszont kevésbé robusztusak zsúfolt, komplex szcenáriók esetében, ahol a jelenlévő objektumok változatos alakú sajátosságokkal bírnak. Más módszerek főkomponens analízis (*Principal Component Analysis - PCA*) alapú technikákkal származtatnak 3D-s befoglaló téglatesteket a felismerendő objektumok köré. A [7] és [8] dolgozatok szerzői statisztikai leírókat számolnak a feladat végrehajtása érdekében: jellemzőket generálnak kovariancia analízis segítségével, amelyek képesek egy lokális 3D-s térrészben mérni a pontok eloszlását. A felismerendő objektum jelöltek fő orientációját (kiterjedését) a kovariancia analízis során számolt kovari-

ancia mátrix sajátértékeiből és a hozzá tartozó sajátvektorokból származtatják. Az objektum osztályozás három jellemző alapján történik, az objektum pontok *minden irányú szóródása - scatterness*, *egyirányú szóródása - linearness*, illetve egy *síkszerűsége - surfaceness*. A szerzők által generált jellemzők sajátosságait a sajátértékek lineáris kombinációjaként számítják. A *második* alkalmazott stratégia különböző alaki tulajdonságokat reprezentáló jellemzők előállítására fekteti a hangsúlyt. A [9], [10], [11] munkákban a szerzők különböző objektum klasszifikációs eljárásokat javasolnak alaki és kontextuális sajátosságokat kihasználó 3D-s leírók (jellemzők) alapján. A [9] dolgozatban a szerzők egy rendszert dolgoztak ki objektum felismeréshez, ahol először egy gráfvgás alapú eljárással különítenek el előtér és háttér régiókat a háromdimenziós adathalmazból, majd az előtéren történő klaszterezés után jellemzőket építenek fel a kinyert pontfelhőszegmenseken, melyeket egy felügyelt gépi tanulási módszerrel osztályoznak. Más szakirodalmi módszerek az alaki karakterisztikákat pontosabban leíró jellemzőket (*spin images*, *harmonic descriptors*) használnak objektumok robusztus detekciójához, sok esetben ezen jellemzők kinyerése elég számítás igényes feladat, így esetünkben nem alkalmazható a feladatunk valós idejű igényét szem előtt tartva [10].



1. ábra: A főkomponens analízis alapú befoglaló téglalap illesztés limitációjának szemléltetése, illetve a javasolt konvex burkoló alapú befoglaló téglalap illesztő előnye egy felülnézeti pontfelhő részleten

2. Tudományos hozzájárulás

Ebben a munkában egy valós időben működő modell alapú rendszert mutatunk be járművek felismeréséhez. A rendszer egy földi LIDAR lézerszkenner által mért pontfelhősorozatot fogad bementként, ami különböző városi scenáriókon került rögzítésre. A modell megalkotása érdekében, három újszerű jellemző kinyerési technikát fejlesztettünk ki. A három leíró kombinációja alkotja a járműmodellt, ami segítségével végrehajtjuk a felismerést. Cikkünkben egy újszerű konvex burok alapú 2D-s téglalap illesztési technikát javasoltunk, amit a jármű kandidánsokra illesztünk annak érdekében, hogy pontosan és gyorsan tudjuk becsülni az adott járművek pozícióját, orientációját és térbeli

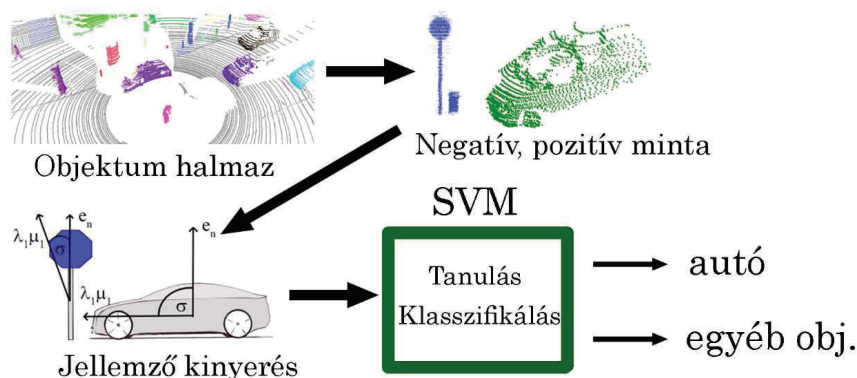
kiterjedését. A járművek felületén tipikus görbületek figyelhetők meg. Ezen görbületek felismerésére gömb leíró alapú jellemzőt javasoltunk, amit a 4. fejezetben részletezünk. Végül a jármű modell felépítése során felhasználtuk azt a tényt, hogy oldalnézetből a járműveknek jól azonosítható és egyedi kontúrja van. Ezen kontúrok kinyerésével egy robusztusabb jármű modell hozható létre, ami nagyban javítja a felismerés pontosságát. A jelen munkában kifejlesztett algoritmus és modell a következő két tudományos hozzájárulást adja a szakirodalmi technikákhoz képest:

◇ *Gyors 2D-s befoglaló téglalap illesztés erősen hiányos és zajos objektumokra:*

A feladat kapcsán célunk a pontfelhősorozatban lévő objektumok köré történő befoglaló téglalapok illesztése, ugyanakkor a Velodyne földi LIDAR szkennerek által szolgáltatott pontfelhőkben számtalan - a szenzor adatból származó - hátráltató tényezővel kell megbirkóznunk. A mért 3D-s pontfelhőnek változó a pontsűrűsége és a takarások miatt az objektumok gyakran hiányosak és zajosak, így egy adott objektum típus változó méretben és megjelenésben érzékelhető a pontfelhőben. A szakirodalmi főkomponens analízis alapú technikák ilyen adaton nem mindig nyújtanak megbízható teljesítményt [6, 7]. Tipikusan a földi lézerek esetében a mért 3D-s adat csak a szkennelés irányából látható teljesen, a környezetben lévő objektumok egyes részei - amelyek takarásban vannak a szenzor pozícióhoz viszonyítva - legtöbbször részlegesen hiányosak és változó pontsűrűséggel bírnak. Ahogy az 1. ábra is szemlélteti az említett hátráltató tényezők miatt a főkomponens analízis alapú technikák nem becsülik elég robusztusan az objektum kandidánsok fő orientációját, tekintve hogy kovariancia mátrix sajátvektorjait használják fel a feladat végrehajtására ami változó sűrűségű adatban pontatlan méret és orientáció becslést eredményez. A főkomponens analízis alapú eljárásokkal ellentétben, ebben a munkában más megközelítést alkalmazunk. Kiszámoljuk minden egyes objektum jelölt felülnézeti konvex burkolóját, majd közvetlenül a burkolóból származtatjuk a befoglaló téglalapokat az objektumok köré. Ez a stratégia kevésbé érzékeny az inhomogén pontsűrűségből származó részlegesen hiányos objektumokra, mivel ahelyett hogy egy lokális térrészben számolnánk térbeli ponteloszlást, közvetlenül az objektumok alakjait próbáljuk felhasználni a burkoló segítségével, annak érdekében hogy minél pontosabb befoglaló téglalapot tudjunk illeszteni rájuk.

◇ *Objektumok alak jellemzőinek gyors vizsgálata valós idejű feldolgozáshoz:* Számos szakirodalmi eljárás [9–11] használja az úgynevezett *spin images* alakleíró annak érdekében, hogy növelje a felismerési algoritmusok hatékonyságát. Az alakleíró hátránya, hogy az objektumok alakját egy felület modellel becsüli, ami 3D-s adat esetén egy erősen számítás igényes feladatnak tekinthető, így valós idejű végrehajtás esetén nem alkalmazható. A mi megoldásunkban két újszerű alakleíró fejlesztettünk ki annak érdekében, hogy valós időben mégis robusztusan tudjuk az egyes jármű kandidánsok alakjait becsülni 3D-ben. A kinyert alakjellelmezőket egy manuálisan annotált tanító adatbázis elemeivel hasonlítjuk össze, így osztályozva a jellemzőkhöz tartozó objektumok halmazát *jármű*, illetve *egyéb városi objektumok* kategóriákba.

A saját modell alapú jármű-felismerő rendszer algoritmusának a bemutatása a következők szerint strukturálható (2. ábra). A 3. fejezetben röviden bemutatjuk a pontfelhő



2. ábra: A kifejlesztett modell alapú jármű-felismerő keretrendszer feldolgozási lépései

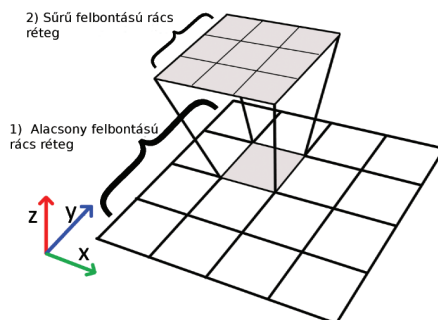
szegmentációját és az objektumok szeparációját végrehajtó előfeldolgozó lépést. Annak érdekében, hogy a pontfelhőt *előtérre*, illetve *háttérre* tudjuk szeparálni egy szegmentációs technikát ajánlottunk. Az *előteret* tartalmazó pontfelhő régiók tipikusan álló és mozgó járműveket, gyalogosokat, táblákat és egyéb utcai objektumokat tartalmazhatnak, míg a *háttérhez* tartozó pontfelhő régiók az úttestet, házak oldalfalait foglalják magukban. Az *előtér* címkével rendelkező pontfelhő szegmenseket bemenetként fogadva, egy hatékony objektum detekciós (*connected component analysis*) eljárást fejlesztettünk ki, aminek segítségével meghatározhatóak az egyes objektumokhoz szemantikailag tartozó 3-D pontok egy halmaza. A 4. fejezetben bemutatásra kerül az általunk ajánlott jármű modell, ami alakleírók egy halmazaként áll elő, és segítségével hatékonyan felismerhetők a városi környezetben elhelyezkedő járművek. A 5. fejezetben kifejtésre kerül egy SVM (*Support Vector Machine*) alapú tanító eljárás, ami az előzőleg kinyert alak jellemzők bináris osztályozására szolgál. Végül a kísérletekről és teszt eredményekről a 6. fejezetben számolunk be.

3. Pontfelhő szegmentáció és objektum szeparáció

Ebben a fejezetben bemutatásra kerül a modell alapú jármű-felismerő rendszerünk előfeldolgozó lépése, ami felkészíti a mért adatot a jármű detekcióra. Egy kétdimenziós hierarchikus rács alapú módszert [1] dolgoztunk ki annak érdekében, hogy hatékony előtér-szegmentációt tudjunk végrehajtani zsúfolt városi környezetről készített pontfelhősorozatokban, ahol sokszor a jelenlévő objektumok szorosan egymás mellett helyezkednek el. A motivációt a hierarchikus rács struktúra kifejlesztéshez a következő tapasztalatok adták: 1) A hagyományos egyrétegű 2D-s rács struktúrák [12] hatékonyan használhatók a pontfelhő szegmentációjára, bár az objektum szeparációs feladatoknál nem mindig szolgáltatnak pontos eredményt az objektum határoló felülete közelében, továbbá nem működnek megbízhatóan közel elhelyezkedő objektumok esetén. Ha nagyméretű cellákat használunk az egyrétegű rács struktúrán, a rács alacsony felbontása

miatt a közel elhelyezkedő objektumok szeparálása nehézkes. Kisméretű cellák használata esetén, viszont fennáll az a veszély, hogy kevés mérés esik egy cellában ami nem elegendő robusztus statisztikai jellemzők számolására. 2) A szakirodalomban használt fastruktúrák (oktális fa, kd-fa) [13] szintén közkedvelt eszközei a szegmentációs és detekciós feladatok megvalósításának. Ezekkel a struktúrákkal hatékonyan lehet pont szomszédságot származtatni, viszont a fa többszöri felépítése és inicializálása egy nagy számítási igényű feladat, így pontfelhő folyamokban nem alkalmazható hatékonyan.

◇ Hierarchikus rács struktúra bemutatása: A rács struktúra kialakításához egy kétdimenziós S rácsot feszítünk a $P_{z=0}$ síkra W_S rács cella mérettel, ahol $s \in S$ jelöl egy önálló cellát a rács struktúrában. A talajszík azonosításához a szenzor pozícióját használtuk referencia koordinátaként. A pontfelhő minden $p \in \mathcal{P}$ pontját hozzárendeljük egy s_p cellához, ha az tartalmazza a p pont projekcióját a $P_{z=0}$ talajszíkra. Jelölje $\mathcal{P}_s = \{p \in \mathcal{P} : s = s_p\}$ azt a ponthalmazt ami az s cellába került levetítésre. Továbbá a cellákban eltároljuk a pontok magasság koordinátáit, és egyéb magasság jellemzőket, úgymint a maximális $z_{\max}(s)$, minimális $z_{\min}(s)$ és átlagos $\hat{z}(s)$ magasságérték. Ezeket a jellemzőket később a szegmentációs és objektum szeparációs lépésben használjuk majd fel.



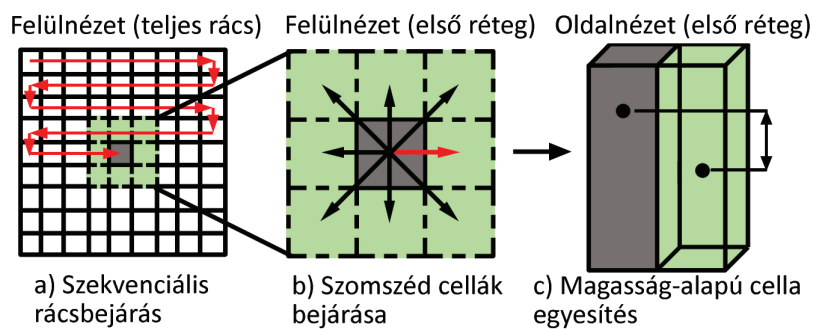
3. ábra: A hierarchikus grid struktúra szemléltetése - alul az alacsony felbontású rács réteg: A 3D-s tér 2D-s nagyméretű rács cellákra való felosztása, felül a sűrű felbontású rács réteg: minden nagyméretű rács cella felosztása kisebb cellákra.

Robusztus objektum szeparáció kivitelezéséhez egy sűrűbb rács struktúrára is szükségünk van, ezért az előbb bemutatott rács s celláit tovább daraboljuk kisebb cellákra $s'_d | d \in \{1, 2, \dots, \xi^2\}$, $W_{s'_d} = W_s / \xi$ cellamérettel, ahol ξ jelöli a felbontás sűrűségét (munkánk során a $\xi = 3$ értéket használtuk).

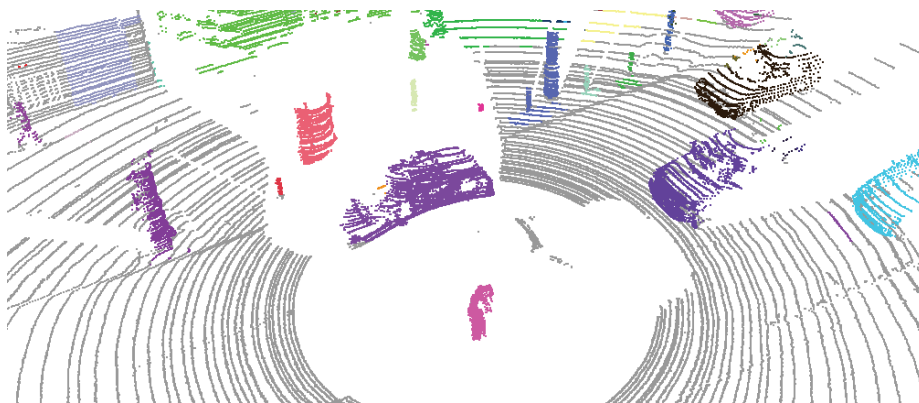
◇ Előtér szegmentáció és objektum detekció:

Az előtér szegmentáció a fent bemutatott hierarchikus rács struktúra ritka felbontású rétegén történik. A célunk egy *előtér* maszk létrehozása ami utcai objektumokról, gyalogosokról, járművekről, házak oldal falairól tartalmaz pontfelhő régiókat, és emellett a végrehajtandó feladat egy *háttér* maszk létrehozása amivel tipikusan utakat és talajpontokat tartalmazó pontfelhő régiókat igyekszünk kiszűrni a mért adatból. A pontfelhőben lévő talajpontok eltávolítására a [12] munkához hasonlóan egy lokálisan adaptív eljárást használtunk, ami képes eltávolítani a talajpontokat tartalmazó pontfelhő régiókat, még akkor is, ha a felület nem teljesen sík. A feladat végrehajtásához a bemutatott rács struktúra ritka felbontású celláiban tárolt pontokból származtatott magasság jellemzőket használjuk fel. Első lépésként megkeressük és eltávolítjuk azokban a cellákban lévő pontokat, amelyek darabszáma nem halad meg egy előre definiált küszöbértéket (tipikusan 4-8 pontot). Ezek a cellák gyakran zajos és ritkás - a szenzortól távol eső régiókban - találhatóak, és sok esetben megnehezítik és hátráltatják a felismerési feladatot. A ritka pontfelhő régiók eltávolítása után, a rács struktúra fennmaradó cellái *háttér* osztály-címkeket kapnak, ha a minimális és maximális magasság érték egy cellán belül nem halad meg egy előre definiált küszöbértéket (mi 25cm-t használtunk munkánk során), továbbá az adott cella 3×3 szomszédságában található cellákból származtatott átlagos magasságérték nem lép túl egy globális küszöbértéket. Az előző algoritmikus lépések után, a pontfelhő még címkézetlen régiói nagy valószínűséggel potenciális *előtér* régiókat tartalmaznak különböző utcai objektumokat, gyalogosokat és járműveket magukban foglalva, ezért a rács struktúra még osztály címke nélküli celláit *előtérnek* osztályozzuk. Az előtér szeparáció után, a kifejlesztett keretrendszer objektum detekciós modulja csak az előtér címkével ellátott pontokon dolgozik tovább. A cél különböző városi objektumok detektálása az előtérmaszk pontfelhőjében. A detekciós lépés eredményeként minden városi objektum, ami a szintéren szerepel egyedi osztálycímkével lesz ellátva. A feladat megvalósításához a hierarchikus rács struktúra ritka és sűrű felbontású rétegét használtuk fel: Egyrészt a ritka felbontású rács réteg alkalmas szorosan összetartozó 3D-s pontok detekciójára, ilyen módon becsülhető a lehetséges objektum kandidánsok mérete és pozíciója is. Másrészt a hierarchikus rács struktúra sűrű felbontású rétege alkalmas arra, hogy sokkal pontosabban számoljunk különböző jellemzőket a pontfelhőben, így lehetőség nyílik az alacsony felbontású rács rétegből érkező detekciós eredmény finomítására.

A kifejlesztett objektum detekciós algoritmus három fő lépésből épül fel: *Először*, bejárjuk az alacsony felbontású rács réteg minden egyes celláját, és megvizsgáljuk minden s cella 3×3 szomszédságát (4a) - 4b) ábra). A szomszédos cellák bejárása által lehetőségünk van egy cella lokális környezetéből származó jellemzőket számolni: (i) $Z_{max}(s)$ maximális magasság érték az alacsony felbontású cellákon belül, és (ii) pontsűrűség (pontok darabszáma) a sűrű felbontású réteghez tartozó cellákból számolva. *Másodszor*, az algoritmus célja, hogy összetartozó pontok egy halmazát határozza meg a pontfelhőből szeparált előtér maszkon, úgy hogy az alacsony felbontású rács rétegen elhelyezkedő nagy cellákat összevonja (azonos osztály címkét rendel hozzá), abban az esetben ha a mért pontfelhőben lévő 3D-s pontok valóban közel helyezkednek el egymáshoz képest, és nagy eséllyel egy objektumhoz tartoznak. A $\psi(s, s_r) = |Z_{max}(s) - Z_{max}(s_r)|$ kritérium segítségével azonos osztálycímké rendelhető azon cellákhoz az alacsony felbontású rács rétegen, ahol az s cella maximális magassága és a szomszédos

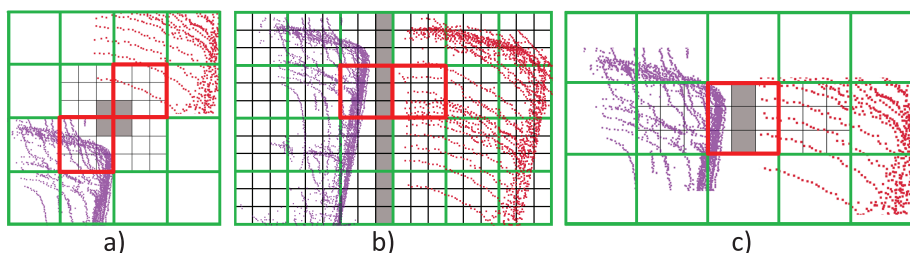


4. ábra: Az objektum detekciós algoritmus egyes lépéseinek bemutatása



5. ábra: Az objektum detekciós lépés eredménye. A különböző városi objektumok különböző színnel jelennek meg a szintéren.

s_r cellák maximális magasságai között mért különbség nem halad meg egy előre definiált magasság küszöböt (4c) ábra). Harmadszor, végrehajtottunk egy finomítási lépést a detekciós eredményen a sűrű felbontású cellákat felhasználva. A magasság alapú kritérium gyakran nem működik megbízhatóan egymáshoz közel elhelyezkedő objektumok esetén, ugyanis az alacsony felbontású rács rétegen lévő cellák mérete túlságosan nagy ahhoz, hogy robusztusan kezelje ezeket az eseteket. A hibás detekciók kiküszöbölése érdekében megmérjük a cella kitöltöttségét a sűrű felbontású rács réteghez tartozó s'_d cellákban. Ahogy a 6. ábra is szemlélteti, azok a közel elhelyezkedő objektumok amelyek hibásan azonos osztálycímkét kaptak az alacsony felbontású cella rétegen, hatékonyan szétválaszthatók a sűrűbb rács rétegben elhelyezkedő cellákban történő pontsűrűség vizsgálatával. A következőkben bemutatunk néhány tipikus városi szituációt amikor az alacsony felbontású rács réteg hibásan egy objektumnak detektál két közel elhelyezkedő objektumot a pontfelhőben, viszont a sűrűbb rács rétegben ezek a hibás esetek kezelhetők. Ahogy a 6a) és 6b) ábrákon is látható, jelölje piros négyzet a két szomszédos cellát az alacsony felbontású rács rétegen. Mindkét esetben a nagy cellák egyaránt tartalmaznak pontokat az egyik, illetve a másik objektumból is, azonban a sűrű rács réteg kisebb celláin található olyan régiók (szürkével jelölve az 6a) - 6c) ábrákon) amelyek mentén elválasztható a két objektum. Ezekben a régiókban a pontsűrűség hirtelen változását vizsgáljuk az objektum szeparálás érdekében.



6. ábra: Közeli objektumok szétválasztása a sűrű rács rétegen. [színek: zöld vonalak = alacsony felbontású rács réteg, fekete vonalak = sűrű felbontású rács réteg, szürke cellák: az objektum szeparáció során vizsgált régiók]

4. A jármű modell és az alakleírók kinyerése

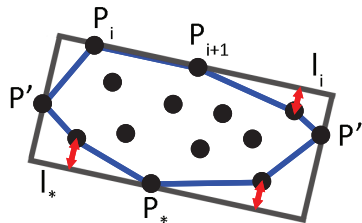
A 3. fejezetben bemutatott algoritmus kimenete különböző pontfelhő szegmensek egy listája, amelyek az egyes objektum várományosokat reprezentálják a szintéren. Ebben a fejezetben bemutatjuk, hogy hogyan ismerjük fel a járműveket az előzőleg előállított pontfelhő szegmensekből. A feladat végrehajtására különböző alakjellemzőket fejlesztettünk ki, amiket a jármű modell alapú felismerő keretrendszerünkben integráltunk. A jármű modell három alakleíró kombinációját használja fel, szem előtt tartva a végrehajtási időt, illetve a felismerés pontosságát.

Először, egy felülnézeti 2D-s befoglaló téglalapot illesztünk a lehetséges jármű jelöltekre annak érdekében, hogy megfelelően azonosítsuk a méretét, kiterjedését, orientációját és pozícióját. A szakirodalmi megoldásokkal [6–8] ellentétben, a saját eljárásunkban nem számolunk lokális ponteloszlást az egyes pontfelhő szegmensekből, a 2. fejezetben már kifejtett indokok miatt. Ahelyett, meghatározzuk az egyes jármű jelöltek 2D-s konvex burkát, így az eljárást nem befolyásolja a pontfelhő szegmensek erősen változó pontsűrűsége, továbbá robusztusabb felismerési teljesítményt érhetünk el részlegesen hiányzó és takarásban lévő jármű jelöltek esetén is. A szegmensek pontos lokalizációja és méretbecslése után, az egyes árulkodó alakjellemzők kinyerésére koncentráltunk. Megfigyeltük, hogy a járműveknek meghatározó alak karakterisztikája, például a szélvédő környékén lévő tipikus görbületek, vagy az autó autó kontúrja oldalnézetből. Ezeket a megfigyeléseket szem előtt tartva, fejlesztettük ki a dolgozatban szereplő alakleírókat.

◊ *2D-s befoglaló téglalap illesztés konvex burkoló segítségével:* A konvex burkolót számító algoritmus bemenetéhez a 3. fejezetben bemutatott hierarchikus rács struktúra sűrű rács felbontásában elhelyezkedő cellákat fogjuk felhasználni. Ebben a lépésben a cellákban tárolt pontok szélesség és hosszúság (X,Y) koordinátáit használjuk fel, a magasság (Z) koordinátákat figyelmen kívül hagyjuk. Először megvizsgáljuk, hogy a sűrű rács réteg cellái közül melyekben helyezkednek el pontok (nevezzük inentől foglaltak), illetve melyek üresek. A következő lépésben végig járjuk az összes foglalt cella 3×3 szomszédságát, és töröljük azokat a cellákat ahol az összes szomszéd egyaránt foglalt. Ezzel az eljárással hozzávetőlegesen becsülni tudjuk a pontfelhő szegmens kontúrára illeszkedő cellákat. Ezután az úgynevezett *monotone chain* algoritmust [14] felhasználva létrehozuk a konvex burkolót a szegmens kontúrján elhelyezkedő cellák pontjaiból. Ahogy a 7. ábrán látható, a következő lépésben megkísérelünk egy *optimális* 2D-s befoglaló téglalapot illeszteni az pontfelhő szegmens konvex burkolójára.

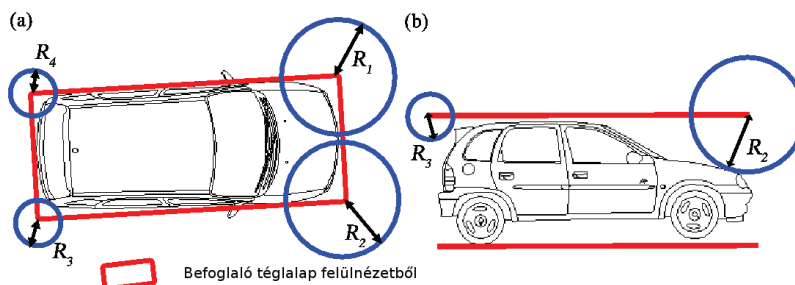
- Bejárjuk a konvex burok egymás utáni p_i és p_{i+1} pontpárjait ($i = 1, 2, \dots, i_{\max}$):
 1. Tekintsük az l_i szakaszt, úgy mint a p_i és p_{i+1} pontpár között értelmezett befoglaló téglalap egyik lehetséges oldala
 2. Keressük meg a konvex burok p_* pontját, aminek a távolsága maximális az l_i szakaszhoz képest, és generáljunk az l_i szakasszal párhuzamos szakaszt amely metszi a p_* pontot. Ezek után az l_* szakaszt a befoglaló téglalap második oldalaként értelmezzük.
 3. Vetítsük le a konvex burok összes pontját az l_i szakaszra, és keressük meg p' and p'' pontokat, amik a legközelebb vannak az l_i szakasz végpontjaihoz. A p' and p'' pontokat metsző és az l_i szakasszal párhuzamos szakaszokat illesztve, előállítható a téglalap jelölt maradék két hiányzó oldala.
- Minimalizáljuk az átlagos távolságot a konvex burkoló pontjai és az illesztett téglalap pontjai között, annak érdekében, hogy az optimális befoglaló téglalapot illesszük a pontfelhő szegmens köré a fent generált téglalap jelöltek halmazából.

◊ *Jellemző görbületek becslése gömbök segítségével:* Az algoritmus célja, hogy meghatározó alakjellemzőket becsüljön a jármű jelöltek felületén, különös tekintettel a szélvédő közelében megfigyelhető tipikus görbületekre. A feladat megvalósításához



7. ábra: A 2D-s konvex burok illesztő algoritmus bemutatása. A befoglaló téglalapot szürke színnel jelöltük.

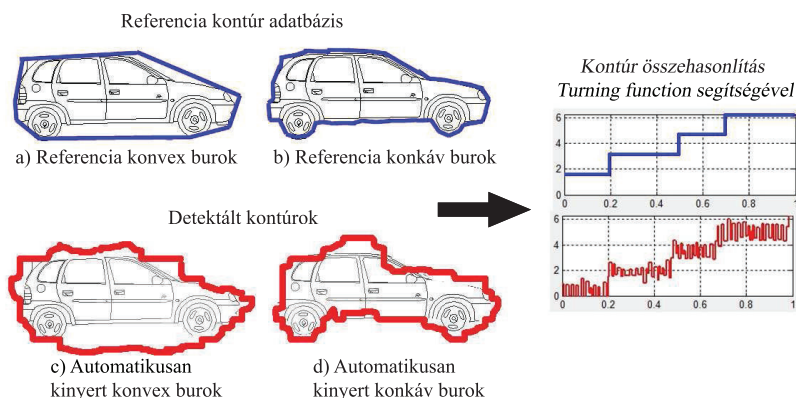
négy darab gömböt illesztünk az előzőleg már meghatározott befoglaló téglalap sarkaihoz. Ahogy a 8. ábra is szemlélteti először függőlegesen eltoljuk a jármű jelölt köré illesztett befoglaló téglalapot a pontfelhő szegmens maximális magasságáig. Ezután a négy gömb középpontját beállítjuk a befoglaló téglalap sarkaiban, majd elkezdjük a sugaraikat növelni mindaddig míg valamelyik gömbnek a felülete el nem éri a pontfelhő szegmens egy pontját. A mi feltételezésünk az, hogy a szélvédő környékén lévő erős görbület miatt, a gömbök sugaraik között jelentős eltérést tapasztalhatunk. Továbbá a 8. ábrán megfigyelhető, hogy a szélvédőnél elhelyezkedő gömb pár sugara (R_1 és R_2) nagyobbak mint a jármű jelölt hátuljánál elhelyezkedő gömbpár sugara (R_3 és R_4). A gömbpáronkénti sugár arányokat használjuk fel a később felépítendő jellemző vektorban.



8. ábra: Főbb alakjellelmezők becslése gömb alapú alakleíró segítségével.

◇ *Oldalnézeti kontúr becslése konvex és konkáv burkolók segítségével:* Ebben a lépésben levetítjük a jármű jelölt pontjait egy oldalnézeti síkra. Ezután konvex és konkáv burkolók illesztésével meghatározzuk az oldalnézeti sziluettjét az adott jármű jelöltnek. Az eljárásból származó jellemző a konvex és konkáv burkoló pontjai 20 cm-es felbontással mintavételezve. Az eltárolt kontúrponokat egy referencia adatbázissal hasonlítjuk össze, ahol előzőleg több jármű prototípus kontúrjait tároltuk el. Az összehasonlítást egy úgynevezett *turning function* segítségével vé-

gezzük, ami a mért és referencia kontúrok pontjai között lévő szögeltérésekből egy függvényt származtat, majd a két függvény eltérését méri p-norma segítségével. A módszer nullához közeli értéket ad vissza ha a két kontúr egyezik, illetve egyet ha a vizsgált kontúrok teljesen különböznek [15]. Ezt a skalárt a SVM tanulás során beépítjük a jellemző vektorban, illetve a kiértékelésnél is felhasználjuk.



9. ábra: A mért oldalkontúr és referencia kontúr összehasonlítása *turning function* segítségével

5. Osztályozás - jármű felismerés

A jármű-felismerő keretrendszer utolsó moduljának feladata, hogy a 4. fejezetben bemutatott pontfelhő szegmensekből kinyert alakjellemzőkön osztályozást hajtson végre *jármű*, illetve *háttér* osztálycímkeket rendelve az egyes jármű jelöltekhez. A jellemző vektor a következő komponensekből épül fel: 1) A konvex burkolóból származtatott befoglaló téglalap hosszúsága és szélessége. 2) A tipikus felületi görbületet becsülő jellemzőből származtatott gömbök sugarai és a gömbpáronkénti sugár arányok 3) Az oldalnézeti konvex és konkáv burkoló és a referencia kontúr adatbázis közötti eltérés (nulla és egy közötti szám). Következésképpen a felépített jellemző vektor nyolc dimenziós lesz. Az osztályozáshoz egy felügyelt tanító eljárást alkalmaztunk, ahol először egy manuálisan annotált tanító adatbázist hoztunk létre. Ebből a célból egy annotáló szoftvert fejlesztettünk ki, aminek a segítségével kézzel címkézhetünk különböző városi objektumokat a pontfelhőben. A Budapest belvárosában készített pontfelhőkből több mint 1600 pozitív és 4000 negatív tanító mintát hoztunk létre, továbbá a németországi KITTI Vision Benchmark Suite [16] pontfelhőiből további 12715 pozitív és 3396 negatív mintával bővítettük az adatbázist. Egy SVM *Support Vector Machine* alapú megoldás [17] segítségével valósítottuk meg a jellemző vektor bináris osztályozását *jármű*, illetve *háttér* objektum osztályokra vonatkoztatva.

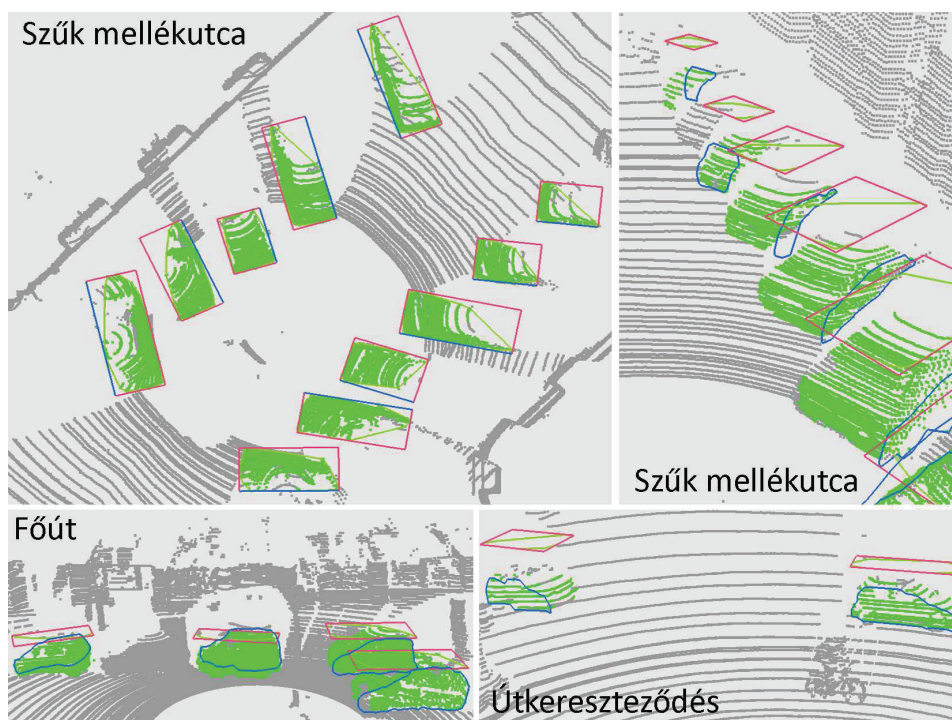
6. Kiértékelés

Adathalmaz	JSZ	főkomponens analízis alapú megoldás [7]		saját modell alapú megoldás	
		F-rate(%)	átlagos feldolgozási sebesség (fps)	F-rate(%)	átlagos feldolgozási sebesség (fps)
Budapest #1	567	73	15	89	24
Budapest #2	1141	71	12	90	21
Budapest #3	368	57	13	80	22
KITTI adatbázis [16]	614	62	14	78	25
Teljes	2690	68	13.5	86	23

1. táblázat: A szakirodalmi főkomponens analízis alapú eljárás [7] és a saját modell alapú eljárás felismerési eredményeinek kvantitatív kiértékelése és összehasonlítása. [JSZ = járművek száma]

A kifejlesztett módszerünket négy különböző LIDAR pontfelhősorozaton értékeltük ki, amelyek különböző városi területeken kerültek rögzítésre, úgymint *főút*, *szűk mellékutca*, *útkereszteződés*. Három adathalmaz Budapest belvárosában készített pontfelhősorozatokot tartalmaz, a negyedik adathalmaz a Németországban készített KITTI Vision Benchmark Suite [16] adatbázisból lett kiválasztva. Az összes tesztadat egy Velodyne HDL-64 S2 típusú autóra szerelhető lézerszkennerrel került rögzítésre, 10 Hz-es forgási sebességgel. Az általunk kifejlesztett modell alapú eljárás teljesítményét összehasonlítottuk egy szakirodalmi eljárással, ami egy egyszerű rács struktúrát használ előtér szeparációhoz, és főkomponens analízis alapú (PCA) jellemzőket használ az objektumok osztályozásához [7]. A bemutatott modell alapú eljárás kvalitatív eredményeit a 10. ábra szemlélteti. A kvantitatív kiértékelés során saját módszerünket és a kiválasztott PCA alapú szakirodalmi eljárást 2690 darab járművön teszteltük, egy manuálisan annotált *Ground Truth (GT)* adatbázis alapján. A módszerünk hatékonyságának teljesen automatizált numerikus kiértékeléséhez, egy megfeleltetést kell alkalmaznunk a felismert járművek és a *Ground Truth (GT)* adatbázisban szereplő járművek között. A felismert járművek és *GT* járművek közötti optimális párosításhoz az úgynevezett magyar módszert alkalmaztuk [18]. Azután megszámoltuk a hiányzó járműveket (*Missing Vehicle (MV)*), és a hibásan felismert járműveket (*Falsely detected Vehicles (FV)*). Ezek az értékek a valós járművek *Number of real Vehicles (NV)* darabszámával kerültek összehasonlításra, továbbá a felismerési algoritmus F-mértékét (pontosság és fedés harmonikus közepe) szintén meghatároztuk. A felismerés hatékonysága mellett, összehasonlítottuk a saját algoritmusunk és a szakirodalmi módszer futási sebességét is *frame/szekundumban (fps)* kifejezve. A numerikus kiértékelést a 1. táblázat részletezi. A cikkben kifejlesztett járműfelismerő keretrendszer egyes lépéseinek a futási sebessége a következőképpen alakulnak: 1) a hierarchikus rács struktúra felépítése - 13ms 2) a pontfelhő szegmentálása és az egyes objektumok szeparációja - 10ms 3) Az alakjellelzők kinyerése és a jármű modell felépítése - 18ms 5) az SVM alapú tanítóeljárással történő objektum osztályozás - 2ms. A teszteredmények igazolják, hogy az F-mérték szerint a saját modell alapú felismerő eljárás felülmúlva a szakirodalmi PCA alapú technikát az

összes tesztadatra vonatkoztatva. Továbbá a saját modell alapú jármű-felismerő gyorsabb futási eredményt produkál pontfelhőfolyamokon, és sokkal megbízhatóbb felismerési teljesítményt biztosít zsúfolt városi környezetekről készített pontfelhőkben (#2 és #3 tesztadat), ahol a szcenárió különböző típusú objektumokat tartalmaz (jármű, közlekedési lámpa, gyalogos, kerékpáros és egyéb utcai objektumok), amelyek takarásban vannak. Ilyen adathalmazban az objektumok gyakran közel helyezkednek el egymáshoz, így az egyes járművek pontfelhőszegmensei részlegesen hiányoznak, vagy több részre esnek szét. Az általunk ajánlott módszer csak azokban az esetekben hibázik, ahol a jármű jelölt erősen takarásban van, és a járműről készített pontfelhő szegmens nagy része hiányzik. A futási sebesség tekintetében átlagosan 13.5 fps-t mértünk a főkomponens analízis alapú szakirodalmi technika [7] esetén, és 23 fps-t mértünk a saját modell alapú járműfelismerő eljárásunk esetén.



10. ábra: A kifejlesztett járműfelismerő keretrendszer eredményei különböző városi pontfelhőkben. Pirossal a járművek köré illesztett felülnézeti téglalap látható, kék színnel a felismert járművek oldalnézeti kontúrja figyelhető meg.

7. Összefoglalás

Ebben a munkában egy modell alapú járműfelismerő eljárást fejlesztettünk ki, amely három új alakjellemzőt használ járművek felismeréséhez. Az eljárás bemenete egy Velodyne LIDAR lézerszkenner által készített pontfelhősorozat. A cikkben bemutatott jellemzők megfelelő teljesítményt nyújtanak zsúfolt, kihívásokkal teli városi pontfelhősorozatokban, ahol a szintéren szereplő objektumok takarásban vannak és hiányosak. A kifejlesztett modell előnyeit egy szakirodalmi módszerrel összehasonlítva igazoltuk, továbbá teljesítményét kvantitatívan értékeltük ki egy kézzel annotált *Ground Truth* adatbázis segítségével.

8. Köszönetnyilvánítás

Ezt a munkát részben az Európai Űrügynökség és az (OTKA #101598) ”Távérzékelte adatok átfogó elemzése” projekt finanszírozta.

Irodalom

1. 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, Lecture Notes in Computer Science, Zürich, Switzerland (2014)
2. Börcs, A., Nagy, B., Baticz, M., Benedek, C.: A model-based approach for fast vehicle detection in continuously streamed urban lidar point clouds. In: Workshop on Scene Understanding for Autonomous Systems at ACCV, Lecture Notes in Computer Science, Singapore (2014)
3. McNaughton, M., Urmsen, C., Dolan, J.M., Lee, J.W.: Motion planning for autonomous driving with a conformal spatiotemporal lattice. In: ICRA. (2011) 4889–4895
4. Levinson, J., Montemerlo, M., Thrun, S.: Map-based precision vehicle localization in urban environments. In: Proceedings of Robotics: Science and Systems, Atlanta, GA, USA (2007)
5. Behley, J., Steinhage, V., Cremers, A.B.: Performance of histogram descriptors for the classification of 3d laser range data in urban environments. In: ICRA, (IEEE) 4391–4398
6. Azim, A., Aycard, O.: Detection, classification and tracking of moving objects in a 3D environment. In: IEEE Intelligent Vehicles Symposium (IV), Alcalá de Henares, Spain (2012) 802–807
7. Himmelsbach, M., Müller, A., Luettel, T., Wuensche, H.J.: LIDAR-based 3D Object Perception. In: Proceedings of 1st International Workshop on Cognition for Technical Systems, Munich (2008)
8. Lalonde, J.F., Vandapel, N., Huber, D., Hebert, M.: Natural terrain classification using three-dimensional lidar data for ground robot mobility. *Journal of Field Robotics* **23** (2006) 839 – 861
9. Golovinskiy, A., Kim, V.G., Funkhouser, T.: Shape-based recognition of 3D point clouds in urban environments, Kyoto, Japan (2009)
10. Douillard, B., Underwood, J., Vlaskine, V., Quadros, A., Singh, S.: A pipeline for the segmentation and classification of 3d point clouds. In: In ISER. (2010)
11. Wang, D.Z., Posner, I., Newman, P.: What could move? finding cars, pedestrians and bicyclists in 3d laser data. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), Minnesota, USA (2012)

12. Józsa, O., Bőrös, 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
13. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: International Conference on Robotics and Automation, Shanghai, China (2011)
14. Andrew, A.: Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters* **9** (1979) 216 – 219
15. Kovács, L., Kovács, A., Utasi, A., Szirányi, T.: Flying target detection and recognition by feature fusion. *SPIE Optical Engineering* **51** (2012)
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). (2012)
17. King, D.E.: Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* **10** (2009) 1755–1758
18. Kuhn, H.: The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly* **2** (1955) 83–97