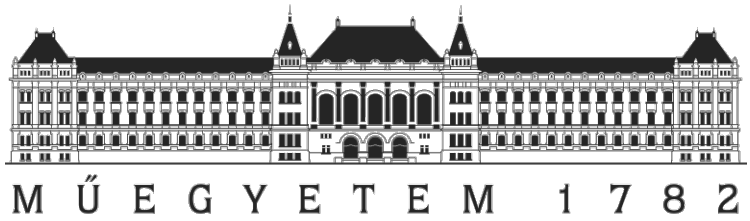


Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Távközlési és Médiainformatikai Tanszék

Mobil előfizetők városi közlekedésének elemzése

Készítette:
Bognár Péter

2017. október 26.

Konzulens: Dr. Toka László

Tartalomjegyzék

Kivonat

Napjainkban szinte minden ember rendelkezik okostelefonnal, melyek elengedhetetlenek a munkában, a kapcsolattartásban, és általában információhoz való gyors hozzájutásban. A távközlési cégek igazi kincsesbányája a felhasználók mobil hálózatos aktivitásáról befolyó hatalmas adattömeg, melynek megannyi felhasználása lehet. Ilyen alkalmazási terület a felhasználók pozíciós adatainak összevetése a tömegközlekedési adatokkal, amely rengeteg lehetőséget rejthet magában, mint például a különböző járatok optimalizálása, járatsűrűség meghatározása, útvonal terv készítése, stb. A pozíciós adatok feldolgozásakor gyakran kell számolni zajjal, illetve nem releváns, hibás mérésekkel, amelyek kiszűrése elengedhetetlen a pontos elemzések szempontjából.

Munkám során különböző pozícióbecslő eljárásokat vizsgáltam, melyek jellegét számításba véve egy, a problémámhoz legjobban idomuló megoldást választottam. A megfelelő eszköznek a Kálmán-szűrő bizonyult, ami mozgó rendszerek állapotáról ad optimális becslést sorozatos mérésekkel, figyelembe véve az állapotméréseket és zavaró tényezőket. A választott szűrő több változatát is implementáltam és kipróbáltam, melyek értékelését követően kiválasztottam a feladat szempontjából ideálisat, melynek segítségével a további felhasználás során el tudtam rejteni a mérésekkor fellépő hibákat.

Abstract

Nowadays every person owns a smartphone, which is necessary at working, in being connected with others and to get information immediately. The enormous amount of data about the activity of users of mobile networks means a treasure to telecommunications companies. This kind of big data can be used for a lot of solutions. For instance, one can compare data about the positions of users with the data of public transportation. This raising hides a lot of opportunities in itself, like optimising routes of transportation, ascertain the density of routes, creating new optimal traffic paths and so on. During the process of position data, noise can occur which causes outliers in the outcome. To have correct results, it is essential to filter these inaccuracies.

During my work, I examined different kind of smoothing and filtering algorithms, which properties were taken into account, when I chose the most appropriate one for my problem. There are numerous tools to solve the inaccuracy problem, like regression analysis, kernel-based smoother. Kalman-filter seemed to be the ideal solution that gives us an optimal estimation of the state of moving systems according to measured values and possible measure of errors. After evaluating some types of this method, I have chosen the one which fitted best the ideal trajectory, and interpreted the results on a real case example.

1. Bevezetés

A technológia fejlődése egyre inkább arra sarkalja az embereket, amire az idők kezdete óta is törekszenek, hogy az őket körülvevő világot adatosíthassák, számokkal leírhatóvá, mérhetővé tegyék. A számok, adatok lehetővé teszik, hogy modellezzük és megoldjuk problémáinkat, segítségükkel felfedezhessük környezetünket. Az alábbiakban ezeket a folyamatokat elősegítő technológiákat mutatom be.

1.1. A "dolgok internete" - Internet of Things

Manapság kimondottan felkapott kifejezésnek számít az IoT az informatikában. Az IoT az Internet of Things rövidítése és lényegében az az egyszerű gondolat húzódik mögötte, hogy ha valamit érdemes csatlakoztatni az internetre, akkor tegyük is meg. Rengeteg innováció jött létre, mely ezen a filozófián alapszik. Az IoT megvalósítások felhasználásuk jellegét tekintve kifejezetten széles skálán mozognak mivel gyakorlatilag bármilyen eszközt „felokosíthatunk" oly módon, hogy ellátjuk olyan szenzorokkal, hardware elemekkel, melyekkel képes az általa gyűjtött információt hálózaton keresztül továbbítani egy másik hálózati csomópont számára. Az „okos" eszközök lehetnek háztartási vagy egészségügyi berendezések, irodai eszközök, gépjárművek, de akár testünkön viselhető technológiák is, mint például egy okos óra, de a mobiltelefonunkat is ide sorolhatjuk.

Az IoT-nak vagy más néven a dolgok internetének megannyi gyakorlati alkalmazása ismert, melyek közül kiemelkednek az okos otthonok, okos útak, okos gyárak, okos városok [2]. Ezek olyan rendszerek melyek komponensei olyan fizikai tárgyak, eszközök, melyek beágyazott elektronika, szoftverek és érzékelők (szenzorok) segítségével kommunikálnak egymással és a működtetővel. Feltehetjük a kérdést, hogy miért csak az elmúlt években indult fejlődésnek az IoT. A válasz pedig az, hogy a megvalósításhoz szükséges technológiák, csak most értek egy olyan szintre, hogy lehetővé teszik a hatékony implementációt. Ehhez szükség volt a számítógépes hálózatok fejlődésére, arra, hogy a beépíthető szenzorok kisebbek, pontosabbak és legfőképp olcsóbbak legyenek, illetve a szenzorokból beérkező óriási mennyiségű adathalmazt fel lehessen dolgozni és eltárolni. Az adatok eltárolása minden eddiginél olcsóbb lett, mivel a háttértárolók előállítási költsége egyre csökken, így manapság egyre nő az igény, az adatok gyűjtésére.

Mint említettem, az okos városok az egyik legkiemelkedőbb és legjobban fejlődő IoT alkalmazás [8]. Az okos város egy olyan urbánus környezet, ahol különböző elektronikai adatgyűjtőket és szenzorokat használnak, hogy a környezetet és az erőforrásokat hatékonyan üzemeltethessék. Az adatok be-

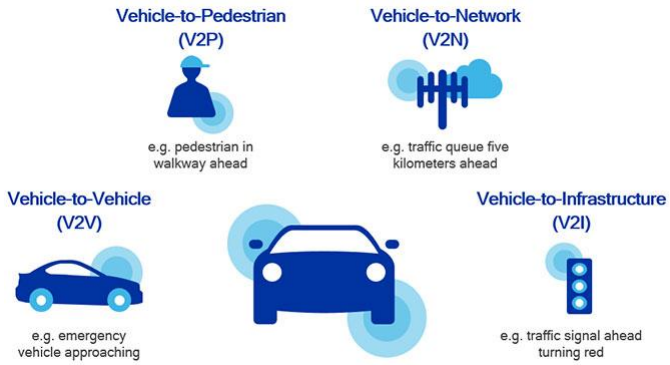
gyűjtése egyformán történhet személyektől, eszközöktől és az infrastruktúra különböző elemeitől is. Ezekkel az információkkal hatékonyan menedzselhetjük a közlekedést, úthálózatokat, vízhálózatokat, rendfenntartó egységeket, de akár erőműveket, iskolákat és kórházakat is.

Az okosvárosok létrehozását és fejlesztését sok tényező motiválja. Elsősorban nyilván gazdasági drive-okról beszélhetünk. A technológia bevonása a városi környezetbe rengeteget takaríthat meg mind az önkormányzat, mind pedig a városlakók számára. Elég abba belegondolnunk, hogy ha folyamatosan bejövő információnk van az egyes infrastruktúra elemek állapotáról, működéséről, akkor megspórolhatjuk azok időszakos vizsgálatát, felügyeletét és nagy valószínűséggel még azelőtt javítani, cserélni tudjuk őket, hogy azok meghibásodnának. Ez nem csak gazdaságos, de hatékony is. A városi közlekedés monitorozásával megkönnyíthetjük a közlekedők mozgását, dinamikusan igazíthatjuk az utak rendelkezésre állását a forgalomhoz, az esetleges balesetekre sokkal gyorsabban reagálhatunk. A segítség előbb érkezhet, a hatósági intézkedések felgyorsulhatnak, valamint a forgalom elterelése is optimálisan történhet. Láthatjuk, hogy a gazdasági vonatkozásokon túl, az emberi tényező biztonsága és kényelme is ösztönző faktor az innovációban, fejlesztésekben.

1.2. Okos gépjárművek - V2X technológiák

Az okosvárosokkal szorosan összefonódó technológia a V2X, melyet szintén érdemes pár mondatban bemutatni. A V2X jelentése Vehicle to Everything, mely a gépjárművek és egyéb eszközök kommunikációját és együttműködését jelenti [2]. Egy ilyen rendszerben a járművek képesek egymással, és az őket környező infrastruktúrával információt megosztani, cserélni valós időben, gyors és megbízható hálózaton keresztül. Ez a terület a sok újítás ellenére még nem ért el egy olyan kiforrott állapotot, mely szabványszerűen használható. Rengeteg szabványosító szervezet vesz részt a fejlesztésekben, hogy olyan irányelvek jöhessenek létre, melyek a kommunikáció technikai részleteit, szabályozásokat foglalnak magukba, annak érdekében, hogy a technológia biztonságosan, szabályszerűen és hatékonyan működhessen.

Mint az eddig említett okosalkalmazások, ez is rengeteg előnnyel járhat számunkra. Gondoljunk bele, ha az úton közlekedő járművek informálni képesek a környező járműveket, hogy baleset történt, hirtelen fékezésre kell számítani, egy tömegbaleset kerülhető el azáltal, hogy a sofőrök időben tudnak reagálni a történésekre, melyeket szerencsétlen esetben az előttük lévő járművektől nem látnak. Másik példa lehet, hogy valósidejű kommunikációval akár konvojok irányítása is megvalósítható, anélkül, hogy az összes sofőr közreműködésére szükség volna. Elég ha az első jármű vezetője irányít, a töb-



1. ábra. A V2X típusai, forrás: `https://www.qualcomm.com/news/onq/2016/06/07/path-5g-paving-road-tomorrows-autonomous-vehicles`

bi jármű csak leköveti azt, ha fékez, azzal arányosan az azt követő járművek is fékeznek, ha az kanyarodik a többi is így tesz, és így tovább.

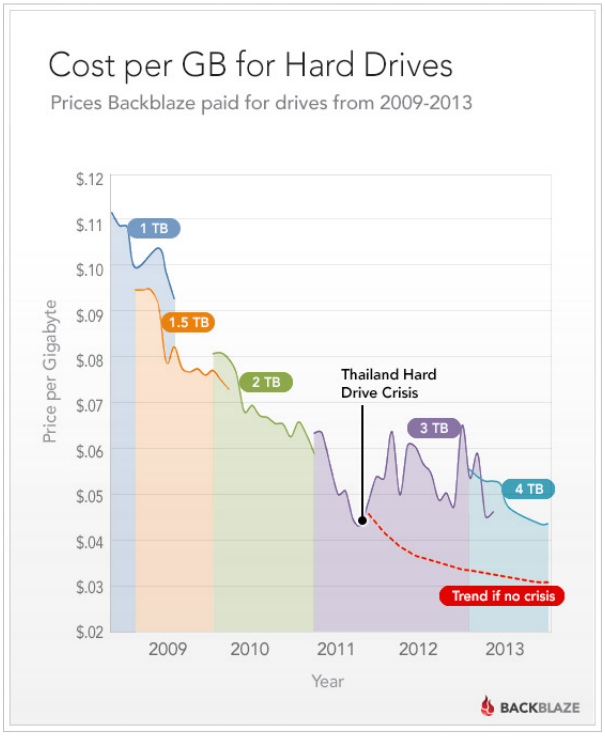
Rengeteg előnnyel jár a V2X technológia, optimalizálja a közlekedést, kényelmesebbé teszi azt és megelőzi a baleseteket. Természetesen a közlekedők épségén és kényelmén túl itt is megjelennek a gazdasági vezérlők. Egy amerikai szövetségi tanulmány kimutatásai alapján, csak az Egyesült Államokban évente mintegy 871 milliárd amerikai dollárnak megfelelő kár keletkezik az utakon történt balesetekből. Ez óriási veszteség, mely megspórolható az emberéletekkel együtt, ha a technológiát alkalmazzuk.

Ahol a V2X szorosan kapcsolódik az okos városokhoz, az a közlekedő gépjárművek és az infrastruktúra közötti kommunikáció, más néven a „Vehicle to Infrastructure" 1. ábra. A közlekedők információhoz juthatnak a környezettel kapcsolatban, ezáltal igazodhatnak az aktuális közlekedési, infrastrukturális és időjárási állapotokhoz. A gépjárművek tájékozódhatnak az útviszonyokról, dugókról, esetleges karbantartási munkálatokról, balesetekről, ezek alapján az utazás felgyorsulhat, könnyebbé válhat. Az elektromos autók kommunikálhatnak a töltőállomás hálózattal, így azok a kialakuló igényekhez mérten választhatják erőforrásaik mértékét, azok rendelkezésre állását. A szabványosítás mellett sokszor az implementáció sem egyértelmű és egyszerű, mivel ad-hoc hálózatokról, rendszerekről van szó, melyek dinamikusan változnak és általában nem figyelhető meg bennük semmilyen jellegű állandóság.

1.3. Nagy adatok bányászata - Big Data technológiák

Az IoT technológia rengeteg adatot szolgáltat számunkra, melyeknek feldolgozása és eltárolása komoly mérnöki megoldásokat igényel. Itt jön képbe az

úgynevezett Big Data technológia, mely jelenleg szintén az informatika egyik zászlóshajójának nevezhető [11]. A Big Data jelent egy szemléletmódot is, mely röviden azt mondja, hogy az adatok hatalmas tömege, méretükkel arányosan rengeteg információt rejt magában, melyek megtalálása, kiaknázása jelentős haszonnal járhat, továbbá a Big Data az ezt lehetővé tevő technológiákra is vonatkozik. Az elmúlt tíz évben a háttértárolók ára rohamosan csökkent, és ez a tendencia továbbra is fennáll. Minden eddiginél olcsóbbá vált az adatok tárolása, melyet a 2. ábra reprezentál. A világ számos pontján egymás után húzzák fel az elképesztő méreteket öltő adatközpontokat. Ezek az óriási, komplex rendszerek új lehetőségeket nyitnak azoknak, akik az óriási adattengerek meghódításával akarnak az informatika piacán élre törni. Az egészen apró startupoktól a multinacionális óriásvállalatokig sok piaci szereplő ide sorolható.



2. ábra. A háttértárolók árainak alakulása, egy becsléssel, ha a thaiföldi katasztrófa nem következik be, forrás:`https://www.backblaze.com/hard-drive.html`

Egyre több cég ismeri fel, hogy az adatokban rengeteg információ lehet rejtve, melyek jelentős értékkel bírhatnak. Egyes országokban a könyvvezetésbe próbálják bevezetni és számszerűsíteni az adatok értékét, hiszen nem

egy cég létezik, akinek az elsődleges tőkéjét ez jelenti. Rengeteg olyan alkalmazás van, amelynek erőssége, egyedisége és lényege az azt kiszolgáló óriási adatmennyiségnek köszönhető. A leghíresebb ilyen cégek közé sorolható a Google és a Facebook. A Google villámgyors keresést biztosít a világhálón szörfölő felhasználónak, mindezt úgy, hogy az egész internetet „bejárják", indexelik, és nyilvántartják. A Google főbevétele azonban mégsem a keresésekből adódik, hanem a célzott hirdetésekből. Ez a bevételeik több mint 90%-át jelenti és a teljes internetes hirdetésekből befolyó bevételek mintegy 54%-a az ő zsebükben landol, ez 2016-ban 9.5 milliárd dollárt jelentett. A felhasználók profilozásával személyre szabott hirdetéseket kínálnak az interneten böngészőknek és az imént említett számok a hirdetések hatékonyságáról árulkodnak. Az algoritmusok melyek ebben kulcsfontosságú szerepet játszanak, az óriási, Google által birtokolt adatrengeteggel dolgoznak, abból tanulnak.

Ez a példa megfelelően érzékelteti az adatok jelentőségét a mai piacon. Egy másik szemléletes példa, a számokban alig alulmaradó, második helyezett Facebook, melynek jelenleg, 2017-ben körülbelül 2 milliárd felhasználója van. A felhasználók egy jelentős része napi rendszerességgel használja az alkalmazást és a használatkor keletkezett adatokat a Facebook be is gyűjti. Az elképesztően értékes adatnak egyik felhasználását a Facebooknál is a célzott hirdetések jelentik. 2016-ban a Facebook hirdetésekből származó bevétele 3.4 milliárd dollár körül mozgott, mely az internetes hirdetésekből befolyó pénzek mintegy 45%-át teszi ki. A növekedés állandó.

A Big Data világában nemcsak az adatok feldolgozása és eltárolása jelent kihívást, de az információ kinyerése is azokból. Az úgynevezett data engineer-ek, adatbázismérnökök, szoftverfejlesztők mellett új szakmák formálódnak, melyek az informatika és statisztika világából növik ki magukat. Ilyenek az adattudósok, adatelemzők vagy az egykor gúnyos hangvételűnek számító megnevezésükkel élve, az adatbányászok, akik az adatokban rejlő értékes információk feltárásában, kinyerésében működnek közre. Ezeket a munkaköröket nehezen lehet körülírni és folyamatosan formálódnak. Noha alapjaikat az informatika és matematika, azon belül pedig statisztika jelentik, rengeteg területről érkeznek erre a pályára. Biológusok, közgazdászok, vagy akár a területtől távolinak mondható bölcsész szakma képviselői is. Az eddig megszokott adatelemzési módszerek ezeknél a méreteknél általában csődöt mondanak, így új megoldások után kell nézni. A sokszor bizonytalannak tűnő „tapogatózás" az adatrengetegben meghökkentő eredményeket adhat. Képesek lehetünk egy vállalat stratégiáját optimalizálni a bevételek növekedését elérve, jövőbeli történéseket „megjósolni" bizonyos minták alapján.

A Big Data alkalmazásán alapszik a Murder Accountability Project az Egyesült Államokban, melynek alapítója Thomas Hargrove, klaszteranalízis

segítségével képes azonosítani a sorozatgyilkosokat. Egy másik érdekes eset, amikor a New York-ban túlmelegedett csatornafedelek több méter magasba repültek. A probléma megoldására a Con Edison nevű szolgáltató statisztikusok segítségét kérte, akik a rendelkezésre álló adatok alapján nagy pontossággal jósolták meg, mikor fog egy fedél felrepülni, így jelentős károkat megelőzve. Ezek a történetek is azt bizonyítják, hogy a Big Data alkalmazásai rendkívül széles spektrumon mozognak és rengeteg más területhez kapcsolódhatnak.

1.4. Okos város okos eszközökkel

Dolgozatom témájához leginkább az okos városok, hálózatok és tömegközlekedés témák állnak közel, mindezeket egy adatelemzési perspektívából közelítem. Az okos város minden olyan innovációt magába foglal, mely a jövő, modern, urbánus környezetét jellemzi. Ezen belül pedig kiemelkedő helyen szerepel a közlekedés optimalizálása. A forgalom menedzselése optimális módon, sok előnyt rejthet magában, azt hatékonyabbá, biztonságosabbá és gyorsabbá téve. Felmerülhet a kérdés, hogy vajon a városi utakon megjelenő forgalmat hogyan lehet mérni, milyen szenzorok jöhetnek szóba ebben az esetben. A szenzoroknak alapvetően két fajtáját különböztethetjük meg ebben az esetben. Egyik maga a felhasználó, mely saját beépített szenzorai segítségével információt szolgáltat a rendszer számára, a másik megoldás pedig ha a szenzorok magában a rendszerben vannak, beépítve az infrastruktúrába, a környezetükben lévő tárgyakat így megfigyelve. Az előbbinek megvan az az előnye, hogy nem igényel nagyobb beruházást az eszközök szempontjából, hiszen azokat a felhasználók szolgáltatják, azonban így kevésbé strukturált, ad-hoc jellegű rendszerről beszélünk. Ebben az esetben a szenzorok lehetnek gépjárművekbe szerelt egységek, de mobiltelefonok is.

Az ilyen jellegű szenzorból a legelterjedtebb a mobiltelefon készülék. Manapság mindenki rendelkezik telefonnal, méghozzá a legtöbben okostelefonnal, mely képes csatlakozni az internetre és rengeteg szenzort használ a működéséhez és alkalmazásai kiszolgálásához. A Samsung Galaxy S4 mobiltelefon még 2013-ban került forgalomba és ekkor ez a készülék az alábbi szenzorokkal volt ellátva: barométer mely a légnyomás változását érzékeli, magnetométer mely a föld mágneses terét érzékeli, gyorsulásmérő, giroszkóp, fényérzékelő, gesztus felismerő, mely felismeri a felhasználó kézmozgását, GPS, hő és páratartalom mérő. Azóta ezeknek a kis eszközöknek a száma nő, ugyanígy a pontosságuk, az áruk pedig egyre csökken.

Ezek azok a tulajdonságok ami miatt úgy gondolom, hogy ezek ideális mérőeszközei lehetnek a városi forgalomnak. Ha ezek az adatok rendelkezésre állnak, számtalan megoldásra lehet őket felhasználni, például tömegköz-

lekedési járatok és azok terheltségének meghatározásra, járat kihasználtság optimalizálására, dugók elkerülésére, infrastruktúra karbantartási munkálatok optimális ütemezésére és még lehetne sorolni. A mobilhálózati adatok összevetése a tömegközlekedéssel és forgalommal olyan megoldásokhoz vezethetnek, melyek felgyorsítják és megkönnyítik az emberek mindennapos utazásait, további jelentős megtakarításokkal.

2. A megcélzott probléma

A kivonatban említett elképzelés, miszerint a mobilfelhasználók hálózati forgalma alapján optimalizálni lehet a közlekedést, egyáltalán nem áll távol a valóságtól. Több eddigi kutatásban is építettek modelleket a felhasználói adatokra. Egy tanulmány [9] során például azt vizsgálták, hogy különböző megközelítések alapján milyen kategóriákba sorolhatók egy város bizonyos kerületei. Sikerült megállapítaniuk az adatok alapján, hogy mely kerületek számítanak lakóövezetnek, és melyek minősülnek irodanegyedeknek. Az ilyen jellegű információkra megannyi alkalmazás épülhet.

A mobilhálózati adatokhoz való hozzáférés nyilván rendkívül korlátozott, mivel ezekhez az adatokhoz egyedül a mobilhálózati szolgáltatónak van jogosultsága, ők gyűjtik és dolgozzák fel ezeket. Adatvédelmi szempontok miatt ezeket kizárólag a szolgáltató kezelheti, mivel személyes adatokat tartalmaznak és segítségükkel akár konkrét személyeket is lekövethetnek. Érthető módon így ezek felhasználása külső fél által jogilag nem megengedett.

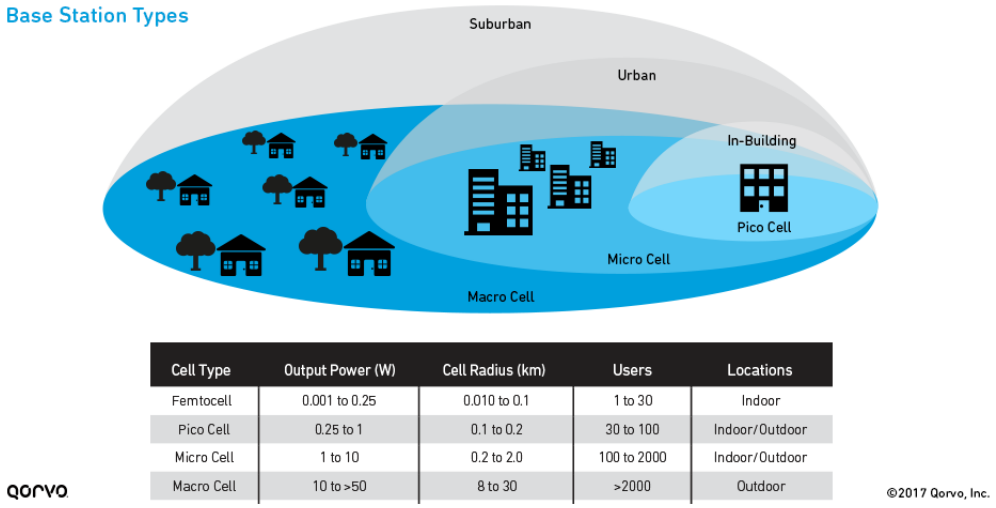
Nem csak a közlekedés szempontjából lehet hasznos kísérlet a fentebb vázolt elképzelés, de jelentős haszonnal járhat a szolgáltatóknak is, ha ezek az adatok olyan érdekességeket „mesélnek”, melyek segítségével javíthatják szolgáltatásaik minőségét és felismerhetik a felhasználók igényeit. Ez az elsődleges oka, hogy noha az információ- és kommunikációtechnológiai cégek nem kifejezetten motiváltak és jártasak adataik felhasználásában, mégis időnként létrejönnek különböző kooperációk kutatói szervezetekkel, melyekben megpróbálják ezeket az adatokat felhasználni kutatási és fejlesztési célokból.

2.1. Mobilhálózati adatok tulajdonságai

Ha az első nagy akadályt leküzdöttük és hozzáfértünk hálózati forgalmi adatokhoz, még közel sem állunk készen pontos becslésekre, mivel a hálózati adatok rendszertelenül érkeznek be, hiányosak lehetnek és rendkívül pontatlannak számítanak. Alapvetően kétfajta adattal kell számolnunk, egyik, hogy a felhasználó éppen egy hálózati cellában tartózkodik, a másik pedig, hogy a felhasználó éppen egyik cellából a másikba megy és ekkor cellaváltás

történik.

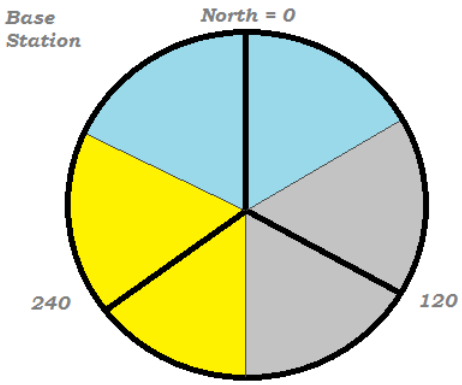
Érdemes tudni, hogy a mobilhálózatot sok komponens mellett a bázisállomások alkotják. Ezek olyan adó-vevő berendezések, melyek biztosítják a kapcsolatot a végfelhasználó és a hálózat között rádiós kapcsolat formájában. Méretüktől, hatótávolságuktól és teljesítményüktől függően sok típus létezik, az egészen nagy makrocelláktól a szobákban alkalmazható femtocellákig. Jellemzően makrocellákat a kevésbé lakott területeken alkalmaznak, melyek hatótávolsága körülbelül 1-10 km sugarú. Városi környezetben a mikrocellák jellemzőek, melyek körülbelül 300-1000 méteres hatósugarúak 3. ábra.



3. ábra. Mikro- és makrocellák alkalmazásai, forrás: `http://www.qorvo.com/design-hub/blog/small-cell-networks-and-the-evolution-of-5g`

A sűrűn lakott, beépített területeken hatványozottan jelentkezik a fading hatása, mely megköveteli, hogy a bázisállomások egymást akár fedve sűrűn legyenek telepítve. A hatékonyság és nagy rendelkezésre állás céljából egy bázisállomáshoz több cella is tartozik és ezek a cellák rétegződhetnek. Jellemzően egy bázisállomáshoz városi környezetben három darab cella tartozik, melyek különböző frekvenciatartományokon további rétegeket tartalmaznak, a terheltség mértékétől függően. A 4. ábrán látható, hogy a cellák 120°-onként helyezkednek el.

A hálózati adatok többnyire a következő lényeges információkat tartalmazzák: a cella azonosítója, melyhez a felhasználó csatlakozik, a cella típusa, a hálózat típusa, egy időbélyeg, a cella iránya fokban megadva. Ilyen



4. ábra. Egy bázisállomás cellái.

rekordok akkor keletkeznek, ha a felhasználó valamilyen hálózati aktivitást végez, ami lehet híváskezdeményezés, hívásfogadás, üzenetküldés és fogadás, valamilyen periodikus signaling vagy adatforgalom. Mivel nem álltak rendelkezésemre ilyen jellegű adatok, ezért ezek sajátosságaival a továbbiakban nem foglalkoztam, azt viszont érdemes megjegyezni, hogy ezek rendszertelenül beérkező és pontatlan adatok, így feldolgozás és optimalizálás nélkül a felhasználó helyzetét nem könnyű meghatározni.

2.2. Célok

A célom, hogy hálózati adatok alapján a felhasználók helyzetére, mozgására ideális becslést adjak minél kisebb hibával. Annak érdekében, hogy a felhasználók helyzetét megbecslő, ideális trajektóriát meghatározó algoritmusokat tesztelhessem, adatokra van szükség. Ezeket az adatokat mesterségesen állítottam elő. Egy mobilos applikáció segítségével rögzítettem utazásaimat, mely feljegyezte, hogy melyik időpillanatban hol tartózkodtam. Az applikáció által adott feljegyzések közötti idő teljesen változó, de átlag és medián számítását követően, körülbelül 15 másodpercenként érkezett állapotfrissítés. Az applikáció a helyzet meghatározásához GPS-t használt, amely jóval pontosabb eredményt ad a mobile hálózati cellás adatoknál, ezért az így kapott mérési adatok minőségén szándékosan rontani kellett, hogy a hálózati környezetet jellemző mérési eredményeket szimulálhassam. Az adatok pontatlanságát úgy kalkuláltam az eredménybe, hogy minden pozícióhoz egy random számot adtam. Ezen számok eloszlása normális, mely várható értéke nulla, szórása pedig 100 méter.

Persze felmerülhet a kérdés, hogy mi szükség erre az egészre, ha ma már léteznek olyan technológiák, mint például az úgynevezett „Mobile Subcell

Localization" melynek segítségével pár méteres pontossággal meg lehet határozni a felhasználó helyzetét. Ez a technológia figyelembe veszi a felhasználót kiszolgáló cellák jelerősségét és háromszögeléssel határozza meg a pontos pozíciót. Annak ellenére, hogy kiválóan működik, ilyen esetekben mégsem alkalmazható, mivel ennek a műveletnek rendkívül magas az overhead-je, így kifejezetten költséges. Esetleges vészhelyzetek esetén szokták használni, de tömegesen már kivitelezhetetlen volna.

3. Városi közlekedést becslő eljárás

A mérések pontatlanságának kiküszöbölésre számos módszer ismert, így munkám első felében az ilyen megoldások megismerése, működésük megértése volt a célom. Vizsgálatuk során szem előtt tartottam, hogy a módszernek, városi környezetben mozgó járművek trajektóriáját kell megbecsülnie, így olyan munkákat is kerestem, ahol hasonló jellegű problémákra alkalmaztak szűrő, simító eljárásokat. Az alábbiakban bemutatom, hogy milyen szűrő és becslő eszközöket találtam.

3.1. Kapcsolódó irodalom és eredmények

Ahol adatokkal dolgoznak, szinte mindenhol szükség van valamilyen szűrésre vagy becslésre. Mind a mérések, mind pedig a feldolgozás alatt keletkezhet nem kívánt zaj, amely torzítja a valós értékeket. A zajok kiszűrésére sok-sok módszer ismert, melyek segítségével, ha tökéletesen nem is rekonstruálhatjuk, de közelíthetjük a valóságot. Az ilyen jellegű metodikák egyaránt megjelennek a digitális jelfeldolgozás és a statisztika területén is. A jelfeldolgozás esetében egy említésre való példa a Savitzky-Golay szűrő [7], mely konvolúciós együtthatók segítségével közelíti a jel adott szakaszait, csúszóablakos módszerrel. Egy alapvető eljárás az is, ha az adathalmaz szomszédos, egymást követő pontjaiból alkotott részhalmazokon végig iterálva lokálisan próbálunk egyenest illeszteni legkisebb négyzetes hiba alapján [3]. Ha a függvényhez parametrikus modell nem ismert, akkor a kernel regresszió [1] lehet megoldás a keresett érték közelítésére.

A szűrők esetében kiemelkedő helyen áll a Kálmán-szűrő [6], mely egyszerűsége ellenére rendkívül erős és hatékony eszköz, ezért lett méltán népszerű és alkalmazzák rengeteg helyen. A klasszikus Kálmán-szűrő csak olyan esetekben működik ahol a jelentkező zaj normális eloszlású, azonban számtalan továbbfejlesztett változata jelent meg [12], ahol igyekeznek megszabadulni ettől a korláttól. A szűrő nagy előnye, hogy tanulást nem igényel, az adott érték becsléséhez elegendő az előzőleg becsült érték és a hiba kovarianciá-

ja, ebből adódik, hogy rekurzív módon működik, ezáltal streaming jellegű, folyamatosan bejövő adatok esetén is effektíven és gyorsan számol. Létezik olyan változata is, mely vagy nagyobb csúszóablakkal, vagy batch, tehát kötegelt feldolgozás esetén alkalmazható. Ez a megközelítés általában pontosabb becsléssel szolgál, mivel az adott pont egy környezete is „segít" a becslésben. Az alábbiakban röviden bemutatom az említett algoritmusokat és azok közül egyet választok a további munkámhoz, mely leginkább idomul a problémámhoz.

3.2. Vizsgált eljárások

3.2.1. Kálmán-szűrő

A Kálmán-szűrő [12] egy olyan algoritmus, mely mozgó, változó rendszerek állapotáról ad optimális becslést sorozatos mérésekkel, figyelembe véve az állapotméréseket és a zavaró tényezőket (zajok, bizonytalanságok, pontatlanságok). Más szóval a Kálmán-szűrő a zajos bemenő adatok rekurzív mérésével egy optimális becslést ad a mérés tárgyának állapotáról.

A Kálmán-szűrő kizárólag Gauss zaj esetén működik, tehát abban az esetben, ha a rendszerben megjelenő zaj normális eloszlást követ. További korlátot jelenthet, hogy a rekurzív állapotbecslés során, a vizsgált tárgy mozgását egy előre definiált transzformációs mátrix írja le, így egyik állapotból az azt követőbe csak affin transzformációval juthat. Az algoritmus népszerűségét a számos alkalmazásán túl az is jelzi, hogy megjelenése óta sok specifikus változata született, mint például a Kálmán-Bucy-módszer vagy a kiterjesztett Kálmán-szűrő [10]. A Stratonovich–Kalman–Bucy szűrő nemlineáris rendszerek stabilizálására alkalmas. Ezek között olyan megoldások is szerepelnek, melyek olyan problémára alkalmazhatóak, ahol az említett korlátozások nem teszik lehetővé az eredeti szűrő alkalmazását.

A Kálmán-szűrőnek számos felhasználási területe van, általánosan használják navigációra, irányításvezérlésnél, különösen repülőgépeknél, űrhajóknál, robotrepülőgépeknél. A Kálmán-szűrőt széles körben alkalmazzák jelfeldolgozó rendszerekben és az ökonometria területén, de lényegében minden olyan esetben használható, ahol az algoritmus által definiált modell jellemzi a rendszerünket és célunk a rendszer egy rejtett változójának optimális becslése más, ismert változókkal és esetleges vezérlőjelekkel. Optimális esetben a mérési adatok több forrásból származnak. Egy gépjármű pozíciójának becslése esetén ezek a mérési források lehetnek a GPS adatok, IMU(Inerciális mérőegység) adatok, amelyek a jármű gyorsulását és szögsebességét adják, de akár a kilométeróra is. A Kálmán-szűrő ezen információk vegyítésével találja meg az optimális becslést, figyelembe véve a források pontosságát.

A Kálmán-szűrőt [6] Kálmán Rudolf Emil (1930–2016) magyar származású amerikai villamosmérnökről nevezték el, aki 1960–1961-ben fejlesztette ki az algoritmust.

A Kálmán-szűrő működését az alábbi matematikai modell írja le, mely a (1) és (2) képletekből áll:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{1}$$

$$z_k = Hx_k + v_k \tag{2}$$

Az A mátrix a tranzíciós mátrix, mely az előző becsült jelet transzformálja, oly módon, hogy az egy reális becslést ad, figyelembe véve a rendszer fizikai tulajdonságait. Ehhez adódhat még esetleges u_k vezérlőjel és a feldolgozási zajnak az értéke, w_{k-1}. Ezek összege adja a becslést. A második egyenlet azt mondja, hogy z_k mérés az x_k rejtett változó és v_k zaj összegéből adódik. Az esetek többségében a H mátrix egységmátrix, mivel legtöbbször igaz, hogy a mérésünk a valós értékből és mérési zajból adódik.

A szűrő egy rekurzív algoritmus, ahol két lépés ismétlődik. Az első becslési lépésben az állapotváltozók meghatározása a cél. Itt az x értéknek egy előbecslését és a hiba kovarianciáját számolja ki. A következő, korrekciós lépésben pedig kiszámolja K-t, az úgynevezett Kálmán-szorzót, mely meghatározza a mért és előbecsült értékek arányát $\hat{x}_k$ kiszámolásához, valamint újraszámolja a hiba kovarianciáját.

A két lépést egymást követően alkalmazzuk, minden egyes pontjára a rendszernek, ezért rekurzív a módszer, melyet az 5. ábra mutat.

Esetünkben az egyenletek némileg leegyszerűsödnek, ugyanis a H mátrixszal nem kell számolnunk, mivel az itt egy egységmátrix. Tudjuk, hogy a rejtett változót a mérés és mérési zaj adják, így szükségtelen ebben az esetben. A képletek egyszerűsítve (3,4,5,6,7):

$$K_k = \frac{P_k^-}{P_k^- + R} \tag{3}$$

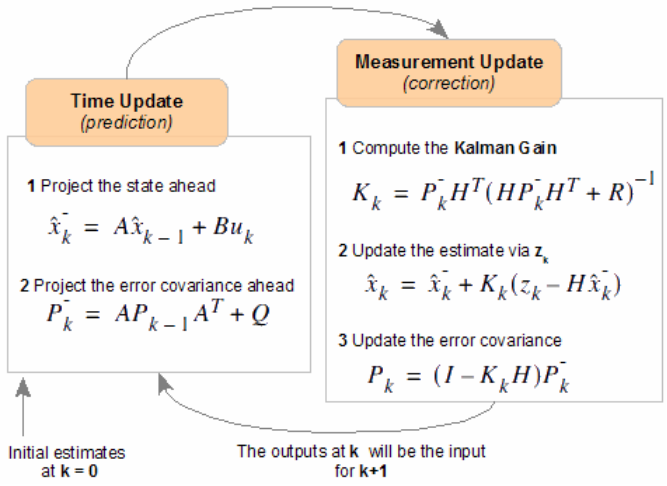
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) \tag{4}$$

$$P_k = (I - K_k)P_k^- \tag{5}$$

$$\hat{x}_k^- = A\hat{x}_{k-1} \tag{6}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{7}$$

Láthatjuk, hogy a becslés a súlyozott előbecslés és súlyozott mérés összege, ahol a súlyokat a Kálmán-szorzó határozza meg. Érdemes megfigyelni, hogy amennyiben a mérésünk kevésbé pontos a Kálmán-szűrő inkább az előző



5. ábra. A Kálmán-szűrő rekurzív algoritmusa, forrás:`http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies`

becslést veszi figyelembe, míg ha a becslés kovariancija kicsi, akkor a mérés kap nagyobb súlyt. Az alábbi levezetés bemutatja e tulajdonságát:

$$\lim_{R\to 0} K_k = \lim_{R\to 0} \frac{P_k^- C^T}{CP_k^- C^T + R} = C^{-1} \tag{8}$$

$$C^{-1} = 1$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k + C\hat{x}_k^-) = \hat{x}_k^- + C^{-1}(y_k + C\hat{x}_k^-) = y_k \tag{9}$$

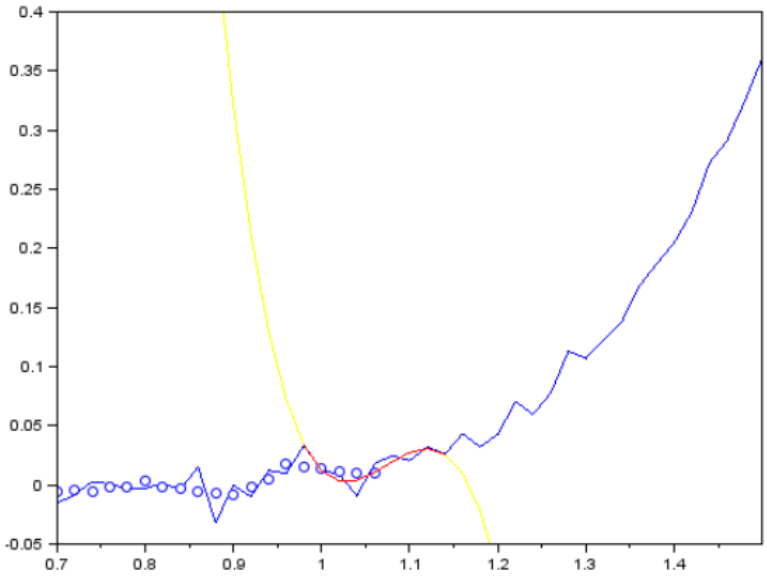
$$\lim_{P_k^-\to 0} K_k = \lim_{P_k^-\to 0} \frac{P_k^- C^T}{CP_k^- C^T + R} = \lim_{P_k^-\to 0} \frac{0}{0 + R} = 0 \tag{10}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) = \hat{x}_k^- + 0(y_k - C\hat{x}_k^-) = \hat{x}_k^- \tag{11}$$

3.2.2. Savitzky-Golay szűrő

A Savitzky-Golay szűrő [7] egy digitális szűrő, melynek alkalmazásával digitális jeleket simíthatunk úgy, hogy a szűrő növeli a jel-zaj viszonyt anélkül, hogy a jel minősége jelentősen romlana. Ezt konvolúció segítségével éri el úgy, hogy az adathalmaz egymást követő, szomszédos pontokból álló részhalmazaira egy alacsony-fokú polinomot illeszt lineáris „least square” függvény segítségével. Abban az esetben ha pontok egyenlő távolságra vannak egymástól, egy analitikus megoldást kapunk a lineáris least square egyenletekre, melynek eredménye a konvolúciós együtthatók halmaza. Ezekkel a konvolúciós együtthatókkal becsülhetjük a simított jelet, vagy annak valahányad

fokú deriváltját minden részhalmaz középső eleménél. A 6. ábrán látható, ahogy az algoritmus végighalad a jelen (kék) és az adott környezetben (piros) mindig meghatároz egy alacsony-fokú polinomot (sárga), melynek eredménye egy becsült pont (kék kör).



6. ábra. A Golay-Savitzky szűrő működése, forrás:`https://en.wikipedia.org/wiki/Kernel_(statistics)`

Az Y_j predikátum meghatározását a (12) képlet mutatja. Konvolúciós együtthatókkal való számolásra példa a (13) egyenlet. A Savitzky-Golay filter egy rendkívül hatékony ám - a konvolúciós együtthatókat tekintve - viszonylag komplex számításokat igénylő algoritmus. Az együtthatók előre meghatározásával azonban az algoritmus futása gyors. Az együtthatókat először Abraham Savitzky és Marcel J. E. Golay gyűjtötték táblázatba.

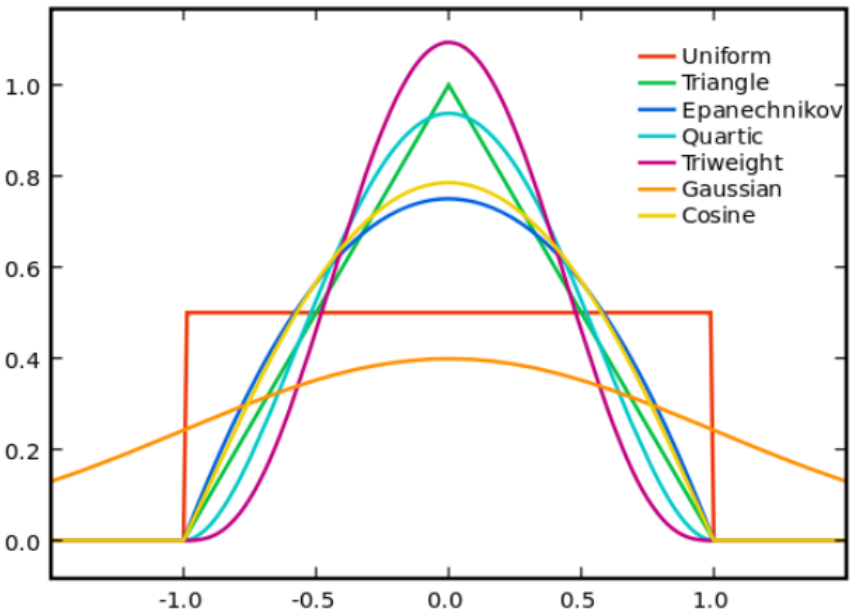
$$Y_j = \sum_{i=-\frac{m-1}{2}}^{\frac{m-1}{2}} C_i y_{j+i} \tag{12}$$

$$\text{, ahol } \frac{m-1}{2} \leq j \leq n - \frac{m-1}{2}$$

$$Y_j = \frac{1}{35}(-3y_{j-2} + 12y_{j-1} + 17j_j + 12y_{j+1} - 3y_{j+2}) \tag{13}$$

3.2.3. Kernel-alapú simító

A kernel-alapú simító egy statisztikai technika, arra, hogy egy valós függvényt megbecsüljünk annak zajos méréseivel, amikor nincs ismert parametrikus modell a függvényhez. A becsült függvény egy simított görbe, melynek simasága egyetlen paraméterrel hangolható, mely a sávszélesség. A sávszélesség megválasztása nem egyértelmű, de számos módszer ismert ideális értékének meghatározására. A kernel függvényre igaznak kell lennie, hogy a végtelenen vett integráljának egynek kell lennie, mivel egy eloszlás függvényt kell megadni. Rengeteg ilyen kernel ismert, melyek közül a legelterjedtebbek az Epanechnikov (parabolikus), Tricube és Gauss.



7. ábra. Kernel típusok, forrás: `https://en.wikipedia.org/wiki/Kernel_regression`

A $\hat{Y}$ predikátumot a (14) képlettel lehet meghatározni. A kernel általános alakját a (15) képlet mutatja, annak egy típusáét, a Gauss kernelét pedig a (16) képlet. Alapvetően a sávszélesség megválasztása fontosabb, mint maga a kernel típus. A kernel sávszélessége a csúszóablak szélességét jelenti, mely végighalad az adatpontokon és figyelembe veszi minden pont esetében az azt környező további pontokat is. Azt, hogy az adott függő változóhoz milyen érték tartozik, befolyásolják a csúszóablakban lévő pontok, melyek súlyát a kernel függvény határozza meg. A cél, hogy a két változó között egy nem lineáris kapcsolatot találjunk.

$$\hat{Y}(X_0) = \frac{\sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i) Y(X_i)}{\sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i)} \tag{14}$$

$$K_{h_\lambda}(X_0, X) = D\left(\frac{\|X - X_0\|}{h_\lambda(X_0)}\right) \tag{15}$$

$$K(x^*, x_i) = \exp\left(-\frac{(x^* - x_i)^2}{2b^2}\right) \tag{16}$$

3.3. Választott eljárás

A Kernel regresszió esetében meg kell jegyezni, hogy ezt a szűrőt általában két dimenzió esetén alkalmazzák, amikor a függő változó valamilyen mért becsülni kívánt érték, míg a független változó például az idő. Ebben az esetben valaminek az időbeni változását tudjuk optimálisan megbecsülni a rendelkezésre álló zajos mérésekből. Természetesen olyan problémákon is lehet ezeket alkalmazni ahol több dimenzióról van szó, ehhez azonban adatredukciós eljárásokra van szükségünk. Ezek, ha nem is nagy mértékben, de ronthatnak az eredményen. Ilyen esetben alkalmazhatjuk az úgynevezett főkomponens-analízist [4], mely egy többváltozós statisztikai eljárás. Lényege, hogy egy nagy adathalmaz dimenzióit lecsökkentse, melynek változói kölcsönös kapcsolatban állnak egymással, úgy, hogy közben a jelen lévő varianciát a lehető legjobban megtartja. A főkomponens-analízis érzékeny az eredeti változók relatív skálázására.

A kapcsolódó irodalomban, kutatói munkákban [5], fórumokon is egyhangúan a Kálmán-szűrő bizonyult a legnépszerűbb módszernek a pozíciós adatok finomításához, outlierek szűréséhez. Mivel a problémám idomulni látszott a Kálmán-szűrő által definiált modellhez, ezért a választásom erre az eszközre esett. Sokrétű alkalmazása, viszonylag gyors implementálhatósága sokat ígért számomra.

3.4. Finomító eljárás

A bemutatott szűrők általánosságban jól alkalmazhatóak, olyan speciális esetekben mint a megcélzott probléma viszont esetleges plusz információkat nem vesznek figyelembe. Jelen esetben azt, hogy egy úthálózatra kell illeszteni a becsült pontokat. Ez persze nem az algoritmusok hibája, de ha ilyen jellegű információk rendelkezésünkre állnak, finomíthatunk az általuk adott eredményen. Rövid keresgélés után az interneten, találtam egy olyan oldalt, ahonnan tetszőleges városok úthálózata tölthető le json formátumban. Ezek között szerepelt Budapest úthálózata is. Mivel az általam mért adatok mind

a fővároson belül lettek rögzítve, ezért ebben az esetben ez elegendő, nyilván ha ezt a finomítást bárhol szeretnénk alkalmazni, akkor globális adatokra lenne szükség.

Miután feldolgoztam a json állományt, megjelenítettem azt térképen és meglepve vettem észre, hogy az adatok viszonylag sűrűn és pontosan fedik le az utakat. Ezt az információt sokféleképpen lehetne vegyíteni az algoritmusokkal, például a Kálmán-szűrőbe is be lehetne építeni, úgy, hogy a pontokat próbálja a legközelebbi utcára vetíteni, azonban az idő szűkössége és az egyszerűség kedvéért arra az elgondolásra jutottam, hogy elég ha a kapott végeredményből azokat a pontokat tartom meg, melyek utcára, vagy annak egy bizonyos környezetébe esnek.

Az egyes utcák szegmensekből állnak, melyeket pontok írnak le, így egy utca egymáshoz csatlakozó szakaszokból áll. Ezeket a szakaszokat úgy alakítottam téglalapokká, hogy a két pont egyenesére merőlegesen állítottam egy-egy h hosszúságú szakaszt. Ezzel nem csak azt modelleztem, hogy az utcák rendelkeznek bizonyos szélességgel, de így egy bizonyos mértékű hibát is megengedtünk.

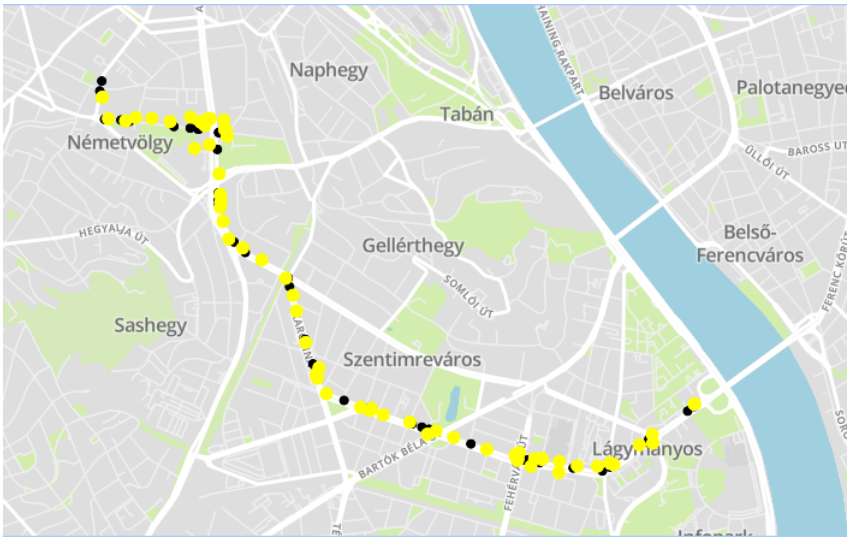
4. Értékelés

Az algoritmusok teszteléséhez és értékeléséhez elsőként referencia adatokra volt szükség. Ezek után a bemeneti adatok megfelelő formátumra való alakítása következett. A helyes működéshez bizonyos paramétereket előre kellett definiálni, melyek meghatározását a következőkben bemutatom. A sikeres lefutást követően az eredmények értékelése volt hátra, melynek szempontjait szintén ismertetem.

4.1. A referencia adathalmaz kiválasztása

A Kálmán-szűrő viszonylag könnyen implementálható algoritmus, megértése viszont alapos vizsgálatot igényel és akik nem jártasak a témában, azoknak komoly fejtörést okozhat rájönni, hogyan is működik. Az interneten rengeteg anyag található melyek képekkel illusztrálva mutatják be az algoritmus működését konkrét eseteken keresztül. A megértés megkönnyítése végett a klasszikus Kálmán-szűrőt implementáltam és egyszerű példákon keresztül próbáltam rájönni, mi alapján becsüli a rejtett változót. A továbbiakban azonban egy jóval több lehetőséget biztosító könyvtárat hívtam segítségül, melyben nem csak a klasszikus szűrő volt megvalósítva, de annak további variánsai is, illetve egy EM-algoritmus (Expectation Maximization algorithm), mely iteratív módon segít megtalálni a számunkra ismeretlen paramétereket.

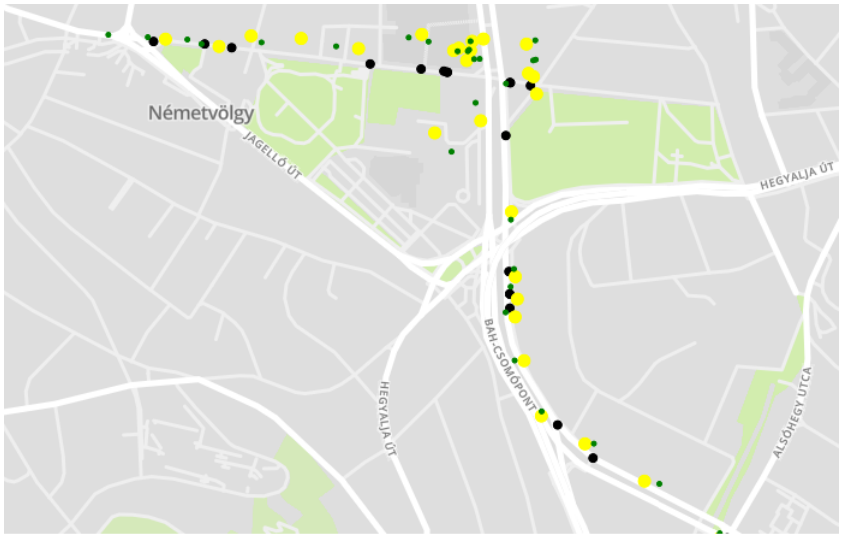
Mivel pontos referencia adat nem állt rendelkezésre számomra, ezért egyetlen megoldásnak az bizonyult, ha a mért adatokhoz hasonlítom a becsült útvonalakat. Ehhez pedig elsőként meg kellett vizsgálnom, hogy ezek a GPS adatok mennyire tükrözik a valóságot, azaz mennyire simulnak az útra. Ennek érdekében a térképen bejelöltem egy utazásomhoz olyan pontokat, melyeket biztosan érintettem. Ezt az utat a 212-es busszal tettem meg és a biztos pontokat feketével jelöltem a következő ábrákon. A mérésekből adódó pontokat sárga színnel jelöltem.



8. ábra. A mérések pontossága a referenciához képest.

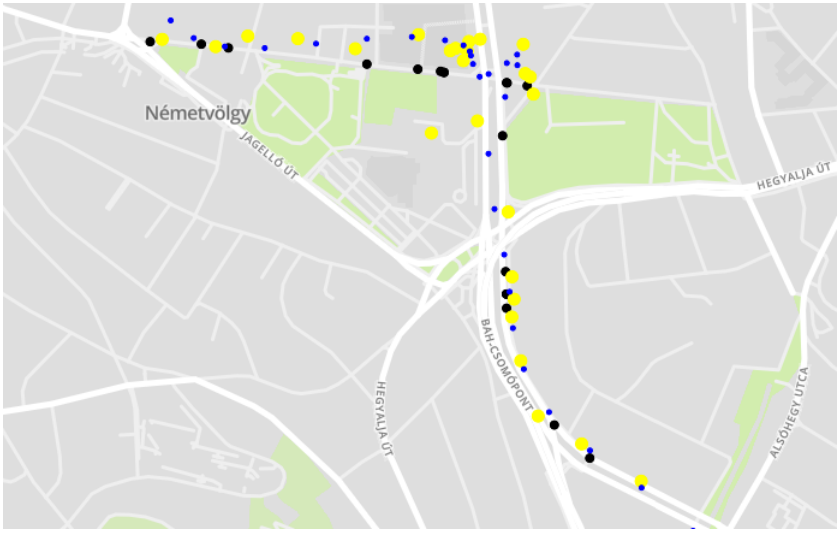
A 8. ábrán látható módon a mérési adatok kifejezetten pontosnak mondhatóak, szinte végig követik a busz útvonalát. A következőkben arra voltam kíváncsi, hogy lehet-e ezeket a méréseket még jobban közelíteni, elvégre erre valók a vizsgált algoritmusok. Ha sikerül olyan eredményt elérnem, ami jobban simul az úthálózatra mint a nyers mérési adat, akkor ezt használhatom referencia adatként.

A 9. ábrán a hagyományos Kálmán-szűrő eredménye látható zöld pontok formájában. Abból a jellegéből adódóan, hogy rekurzív módon csak az előző pont és a jelen mérés alapján becsül, látszik, hogy az egyes kiugró pontokat is némileg követi és ezeket nem becsüli túl pontosan. Itt jegyezném meg, hogy a Kálmán-szűrő ebben a kontextusban inkább mondható becslő algoritmusnak, mint szűrőnek mivel a pontokat nem szűri, azaz a kiugró értékeket nem dobja el, hanem egy kedvezőbb becslést ad rájuk, tehát ugyanannyi pontot ad ki a kimenetén, amennyi a bementén beérkezett, ellentétben egy hagyományos értelembe vett szűrővel.



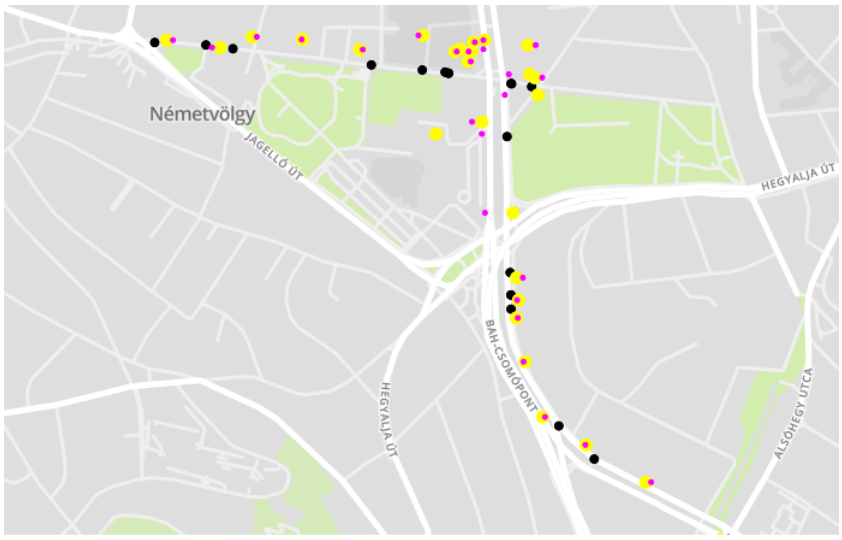
9. ábra. A Kálmán-szűrő által adott eredmény zölddel.

Az 10. ábrán már a Kálmán simító eredménye látszik, ami kimondottan meggyőző. Látszik, hogy ebben az esetben a kiugró értékek nincsenek akkora hatással a becslésekre, mivel ez az algoritmus a jövőbeli értékeket is figyelembe veszi. Láthatjuk, ahol nagyobb szünet van két pont között, ott szépen interpolál és egy kifejezetten sima, az útra jól illeszkedő görbét ad. A Kálmán simító által becsült pontokat kékkel jelöltem.



10. ábra. A Kálmán-simító által adott eredmény kékkel.

Kíváncsiságból kipróbáltam a Savitzky-Golay szűrőt is a problémára és meglepően jó eredményt adott. Noha a Kálmán simítónál láthatóan nem tudott jobb becslést adni, de a sima szűrőhöz közeli eredménnyel szolgált. A Savitzky-Golay szűrő által becsült pontokat rózsaszínnel jelöltem a 11. ábrán:



11. ábra. A Savitzky-Golay szűrő által adott eredmény rózsaszínnel.

Csúszóablak méretének 5-öt adtam meg. Fontos megjegyezni, hogy páratlan számú méretet kell meghatározni, mivel mindig egy pontot és az attól a listában mindkét irányba adott távolságra lévő pontok halmazát veszi számításba az algoritmus. A közelítő polinomok fokszámaként 3-at határoztam meg. A paraméterek meghatározása nem magától értetődő, ezeket próbálgatással, következtetésekkel és a vizuális visszaigazolások alapján állítottam be. Ahogy azt az ábrák mutatják, a legjobb eredménnyel szolgáló módszer a Kálmán-simító, mely némi hiba mellett viszonylag egyértelműen rásimítja a mért pontokat az úthálózatra. A továbbiakban referenciaként a simító által adott eredményeket használtam.

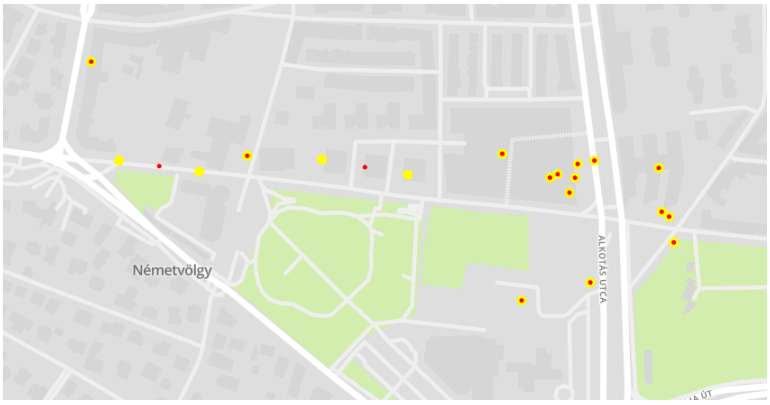
4.2. Útvonal becslő eljárás Kálmán-szűrő alapokon

Esetünkben a Kálmán-szűrőnek egy négyelemű vektort kell bementként megadni, melynek első két tagja a szélességi és hosszúsági fokok, a második kettő pedig a két koordinátához tartozó sebesség. Mivel a hálózati adatok alapján csak koordináták állnak rendelkezésünkre, ezért a sebességet nekünk kell meghatároznunk. Az esetek többségében külön forrásból érkeznek az olyan

információk mint sebesség, gyorsulás vagy megtett kilométerek száma. A több forrásból származó adatok fúziója egy sokkal pontosabb becslést adhat, mintha a sebességet származtatjuk a pozíciós adatokból. A hálózati adatoknál sajnos nincs semmi további segédinformáció, így a pontok közti távolságból és a köztük eltelt időből számoltam átlagsebességet. A következő előbecsült pozíciót tehát úgy kapjuk, ha az előző pozíció koordinátáihoz hozzáadjuk a hozzájuk tartozó származtatott sebesség és az eltelt idő szorzatát, ami a mérési adatok sajátosságaiból adódóan 15 másodperc.

A Kálmán-szűrő paramétereit már az inicializáláskor meg kell adni, melyek az algoritmus lefutása alatt dinamikusan nem változhatnak. Ez azt jelenti, hogy a tranzíciós mátrixban előre meg kelll határozni a két pont közt eltelt időt, aminek így egységesnek kell lennie, tehát a bemeneti méréseknek egyenlő távolságra kell lenniük időben. Ahogy arról beszámoltam, sajnos a mérési adatok beérkezése teljesen kiszámíthatatlan, nincs bennük fix periodicitás. Ezt valamilyen módon ki kell küszöbölni.

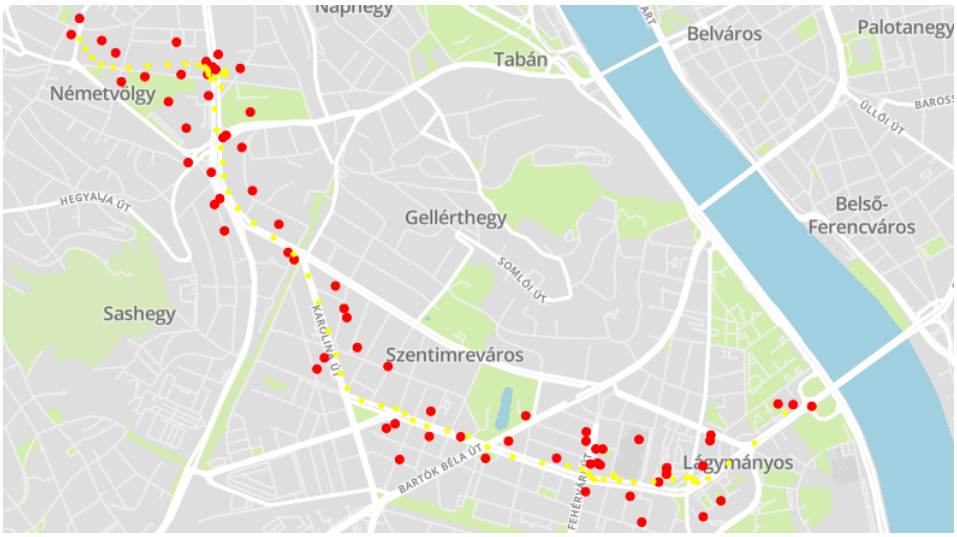
Legkézenfekvőbb megoldásnak az mutatkozott, ha az utazás időintervallumát felosztom 15 másodperces egységekre és ezekre az időpontokra mint, centroidokra tekinthetünk. A mérési pontokat pedig valamelyik centroidhoz rendeljük aszerint, hogy melyikhez van legközelebb az időbélyege alapján. Mivel a pontok lehetnek akár éppen a centroidon vagy a sugarának határán, ezért ezeket súlyozni kell, hogy megfelelő arányban vegyenek részt a centroid koordinátáinak meghatározásában. A mérési pontok súlyai fordítottan arányosak a centroidoktól vett távolságukkal. Az így kapott átlagolt GPS mérési adatokat a 12. ábrán piros színnel jelöltem.



12. ábra. Az átlagolás eredménye.

A 12. ábrán a sárga színű pontok a mérési adatok, amelyekből az átlagolt pontokat számoltam. Az útvonal nem torzul és lehet látni, hogy a legtöbb helyen egy 15 másodperces blokkba csak egy mérés tartozik, így ugyanazt a

pontot kapjuk mint a mérési pont. A 12. ábrán egy olyan szakaszt mutatok be ahol az átlagolás is megfigyelhető, hisz két olyan pont esetén amelyek egy blokkban vannak, a piros pont közéjük kerül. Az egy pont blokkonként egyébként reálisnak mondható, mivel ez körülbelül 15 m/s sebességet jelent, ami körülbelül 50 km/h. A mérési adatokat ezek után még szándékosan el kellett rontani ahhoz, hogy mobil hálózati cellás pozíció adatokat emuláljak. Ezt úgy oldottam meg, hogy minden pozícióhoz egy véletlenszerű számot adtam: ezen számok eloszlása normális, várható értéke nulla, szórása pedig 100 méter. A referencia adatok és a rontott minőségű adatok együttesen a 13. ábrán láthatóak.



13. ábra. A mobil hálózat cellás pozíció adatok emulálása (piros pontok) GPS pozíció adatokból (zöld pontok).

Miután rendelkezésemre álltak az emulált mérési adatok, a szűrő paramétereinek meghatározása következett. Elsőként a tranzíciós mátrixot írtam fel. Mivel a bemenet 4 elemű vektor, ezért ennek 4x4-es mátrixnak kell lennie. A cél, hogy az előbecsült koordináta az előző koordináta értéke plusz a hozzájuk tartozó sebesség és az eltelt idő szorzata. Így kapjuk meg a következő állapotot, ha csak a fizikai tulajdonságokra hagyatkozunk. Az A tranzíciós mátrix tehát így néz ki:

$$A = \begin{bmatrix} 1 & 0 & 15 & 0 \\ 0 & 1 & 0 & 15 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Mint említettem a H mátrix egységmátrix mivel a mi esetünkben a rejtett változó a mért adat és valamekkora gauss-i zaj összege, így nincs szükség további transzformációra. A H mátrix:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A mérési kovarancia mátrixot is előre kell definiálni, ez határozza meg, hogy az adataink mennyire pontatlanok. A GPS adatok esetében ez körülbelül 10-20 méter. Ha ezt az értéket szélességi és hosszúsági fokokban akarjuk kifejezni, akkor durván becsülve az 1 méterre 10^{-5}-t kell számolnunk, így a GPS kovarancia mátrixja az alábbi lesz:

$$R = \begin{bmatrix} 10^{-4} & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{bmatrix}$$

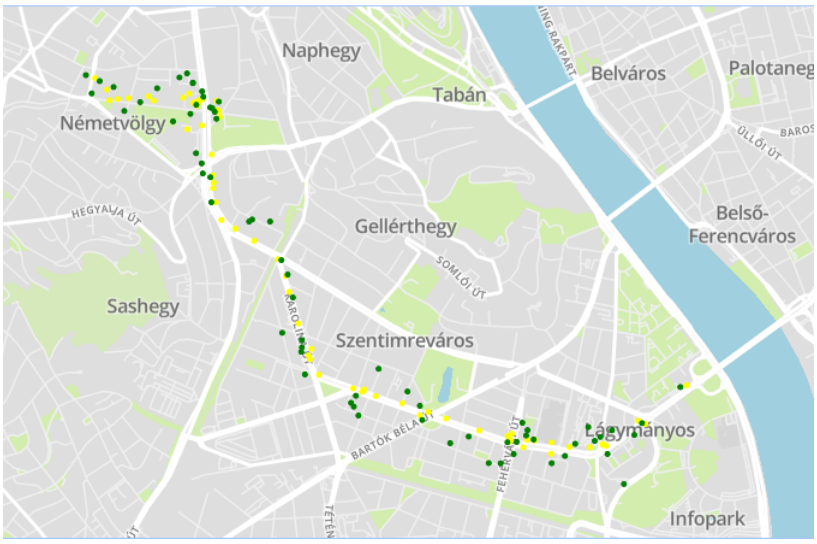
Mivel az első pont esetében nyilván nincs előző pont mely alapján meg lehetne határozni az előbecslést illetve a hiba kovarianciáját, ezért ezeket is meg kell adnunk. A kezdő állapot egyértelműen az első pont koordinátái és zérus sebességek. A kezdő hiba kovarianciának pedig az alábbi mátrixot adtam meg (lényeges, hogy ne csupa nulla mátrixot adjunk meg, mert az azt jelenti, hogy nincsen zaj a környezetben, így továbbra is a kezdeti állapotot kapnánk vissza):

$$P_0 = \begin{bmatrix} 10^{-1} & 0 & 0 & 0 \\ 0 & 10^{-1} & 0 & 0 \\ 0 & 0 & 10^{-1} & 0 \\ 0 & 0 & 0 & 10^{-1} \end{bmatrix}$$

Miután a paramétereket megfelelően meghatároztam, már csak a tranzíciós kovarancia mátrix volt hátra, ennek megadása már korán sem jelentett egyszerű feladatot. Ekkor hívtam segítségül a beépített EM-algoritmust, mely iteratív módon segít megtalálni a legjobb közelítést. Az iterációk számának növelésével egyre pontosabb eredményt érhetünk el, ezzel együtt sajnos a futási idő is növekszik. 300-szori iterációval az alábbi tranzíciós kovarancia mátrixot kaptam:

$$Q = \begin{bmatrix} 3.09^{-5} & -1.48^{-5} & 1.53^{-6} & -2.097^{(}-6) \\ 1.49^{-5} & 3.09^{-5} & 2.1^{-6} & 1.53^{(}-6) \\ 2.01^{-6} & -5.11^{-7} & 2.5^{-7} & -2.32^{(}-7) \\ 5.15^{-7} & 2.01^{-6} & 2.33^{-7} & 2.5^{(}-7) \end{bmatrix}$$

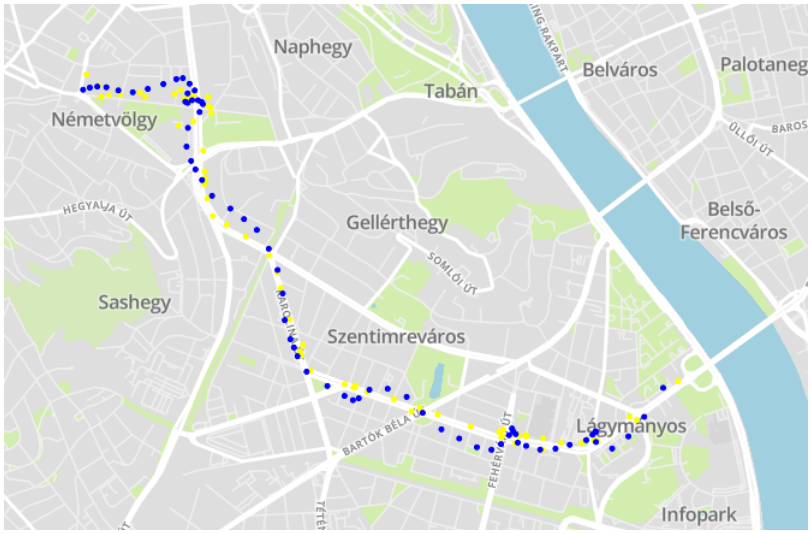
A paraméterek meghatározása után lefuttattam az algoritmusokat és térképen megjelenítettem az eredményeket. Az ábrákon a szintén Kálmán-filterrel becsült referencia adatokat sárga színnel jelöltem a Kálmán-szűrő becslését pedig zölddel.



14. ábra. A Kálmán-szűrő eredménye emulált mobil hálózati cellás pozíció adatokon.

A 14. ábrán látszik, hogy ebben az esetben a szűrő már kevésbé ad pontos becslést. A mozgás iránya nagyjából meghatározható, ám az utca szerinti pontosság már elveszett, így ez a megoldás nem alkalmas a további felhasználások esetében.

Ugyanerre az adatszettre a simítót is lefuttattam, melynek eredményét kék színnel jelöltem a 15. ábrán. Ez a módszer már sokkal jobb közelítést ad



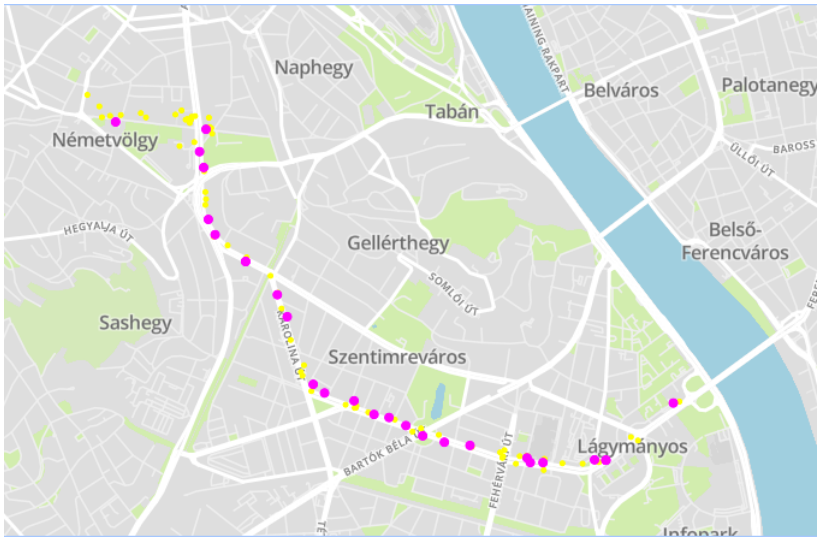
15. ábra. A Kálmán-simító becslése hálózati környezetben.

mint a klasszikus algoritmus. Ha konkrét utcák nem is minden esetben meghatározhatóak, de az utazás pályája viszonylag pontosan kirajzolódik. Ettől a megoldástól nem is lehet elvárni, hogy az utcák összetett struktúrájához igazodjon, ehhez jóval sűrűbb és pontosabb adatra lenne szükség.

Végül a már említett eljárást alkalmaztam a Kálmán-simító által adott eredményen: az eredményként kapott becslést a legközelebbi utcára vetítettem. A 16. ábrán látszik, hogy a kapott becslés minden eddiginél pontosabb. Kizárólagosan csak utcák vonalára „becsüli" a pontokat. Sajnos a nem megfelelő pontok eldobásával a megmaradó pontok kicsit ritkábbak, de így is egyértelműen meghatározható az útvonal. A térképes adatokkal további lehetőségek nyílnak a Kálmán-filter felhasználását illetően.

4.3. Értékelési szempontok

Az implementált Kálmán-szűrőket valami alapján össze kell hasonlítani, hogy ne csak vizuálisan győződhessünk meg arról, hogy melyik eredményez pontosabb becslést, de számszerű adatok is alátámasszák. Sokféle módszer van arra, hogy két becslő algoritmus teljesítményét összehasonlítsuk, erre egy bevált módszer az átlagos négyzetes hiba (Mean Square Error, MSE) számítása, ahogy a (17) képlet mutatja.



16. ábra. Az utcákra vetített becslések által adott útvonal.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{17}$$

Minden pontpárra vesszük a két pont közötti távolság négyzetét, ahol egyik pont a becsült, másik pedig a referencia pont, ezeket összeadjuk, majd elosztjuk a pontok számával. Ez a módszer sokszor reálisabb eredményt ad mint a sima átlag számítás, mivel az átlagszámítást esetében sokszor nem befolyásolják döntően az eredményt a kiugró értékek. Egy egyszerű példával szemléltetve: Ha $d1$ távolság 5 egység, $d2$ távolság pedig szintén 5, akkor az átlaguk is 5 és az átlagos négyzetes hiba ekkor 25. Amennyiben $d1$ távolság 1 egység, $d2$ pedig 9, abban az esetben az átlag ismét 5 lesz, pedig a távolságértékek nagyban eltérnek az első esetben meghatározottaktól. Az átlagos négyzetes hiba azonban ebben az esetben 41 lesz, mely már hűen tükrözi a két eset közti nyilvánvaló különbséget.

Ahhoz, hogy az átlagos négyzetes hiba számítását alkalmazni tudjam, arra volt szükség, hogy a referencia pontok időbélyegével azonos időben a becsült értékek között is legyen egy koordinátapár, így az adott időpontra tudtam számolni egy távolságot, a becsült és valós helyzet alapján. Mivel a Kálmán-szűrő egy adott frekvenciával érkező, azonos távolságra lévő pontokat várt bemenetként, ezért a mért adatokat 15 másodpercenként átlagolni kellett. A beérkező mérési adatok között eltelt idő teljesen változó volt, de egy átlag és medián számolása után a 15 másodperces időintervallum tűnt ideálisnak. Ebben az esetben a két adathalmaz időbélyegei már nem egyeznek,

így nem elég végig iterálni a két halmazon és a két i. elem közti távolságot kiszámítani.

Mivel a referencia adatokhoz kell hasonlítani a becsült adatokat, ezért azt a módszert választottam, hogy a referencia pontokon végigiterálva, mindegyik időbélyegéhez rendelek egy becsült értéket. Ehhez először az adott referencia ponthoz, a becsült pontok között megkerestem azt a két pontot melyek időbélyege alapján definiált időintervallumba esik. Ezután a két pontra egyenest illesztettem. Az első becsült pont időbélyege és a referencia pont időbélyege közti különbség és az első és második becsült pont közti különbség alapján megkaphatjuk az első becsült pont és a referencia pont közti távolságot, mivel a két becsült pont közti távolság ismert. Ekkor felírható annak a körnek az egyenlete, melynek középpontja az első becsült pont, sugara pedig az előbb számolt távolság. Az egyenes és a kör egyenletéből pedig már könnyedén meghatározható annak a pontnak a koordinátája a becsült adatokon, melyhez a referencia pont időbélyege tartozik.

A fenti módszerrel így pontosan ki lehetett számolni a referencia és becslés közti különbséget. Így az algoritmusok teljesítménye összehasonlítható. Ennek az eljárásnak az eredménye látszólag a legtöbb helyen pontosabb mint a referencia útvonal (mely csak közeli becslése a helyes útvonalnak), így eredményének számszerűsítése nem lett volna releváns. A kapott eredményeket az 1. táblázatba foglaltam.

1. táblázat. Algoritmusok becsült hibája:

Kálmán-szűrő hibája	68.53 m
Kálmán-simító hibája	39.44 m

5. Befejezés

5.1. Konklúzió

A megvizsgált algoritmusok közül egyértelműen a Kálmán-simító tűnt a legjobb megoldásnak a városlakók mozgásának becslésének problémájára. A zajos adatok ellenére viszonylag pontosan becsülte meg a valós trajektóriát. Nem hiába alkalmazzák számos esetben. A térképes adatok segítségével pedig szinte pontosan adta vissza a vizsgált utazásokat. Persze ez a modellezés nem feltétlenül tükrözi a valóságot, hiszen lehetséges, hogy a hálózati adatok még annál is pontatlanabbak, megbízhatatlanabbak mint az emulált zajos adathalmaz. A térképes adatok hatékony bevonása és további finomítások javíthatnak ezeken az eredményeken.

5.2. Kitekintés

A Kálmán-szűrő alkalmazása térképes adatokkal ígéretes eredményeket adott. Érdemes lehet követni mások példáját, és oly módon módosítani a klasszikus szűrőt, hogy a plusz információkat figyelembe vegye. Egy megoldás lehet az, hogy az általa adott becslést megpróbálja a legközelebbi utcára vetíteni. Erre egy heurisztika, hogy megtaláljuk azt a kört (melynek középpontja a becsült pont) és azt az utat leíró szakaszt, melyek esetén a kör és a szakasz metszik egymást és a kör sugara minimális. Ekkor csak merőlegest kell állítani a kapott szakaszra a becsült pontból és megtalálhatjuk a végső becslés eredményét. Ez egy kvadratikus programozást igénylő feladat. További megoldások lehetnek, hogy az előbecslést vagy a mérést próbáljuk útra illeszteni. Úgy gondolom, a továbbikban érdemes lehet kipróbálni ezen ötletek valamelyikét.

Hivatkozások

[1] N. S. Altman. An introduction to kernel and Nearest-Neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[2] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[3] Norman Richard Draper and Harry Smith. *Applied regression analysis*. A Wiley-Interscience publication. Wiley, New York, NY [u.a.], 3. ed. edition, 1998.

[4] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[5] Jungwook Jun, Randall Guensler, and Jennifer Ogle. Smoothing methods to minimize impact of global positioning system random error on travel distance, speed, and acceleration profile estimates. *Transportation Research Record: Journal of the Transportation Research Board*, 1972:141–150, 2006.

[6] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[7] Per-Olof Persson and Gilbert Strang. *Smoothing by Savitzky-Golay and Legendre Filters*, pages 301–315. Springer New York, New York, NY, 2003.

[8] Riccardo Petrolo, Valeria Loscrì, and Nathalie Mitton. Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Transactions on Emerging Telecommunications Technologies*, 28(1):n/a–n/a, 2017.

[9] Jonathan Reades, Francesco Calabrese, and Carlo Ratti. Eigenplaces: Analysing cities using the space–time structure of the mobile phone network. *Environment and Planning B: Planning and Design*, 36(5):824–836, 2009.

[10] Maria Isabel Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. 44(1):n/a–n/a, 2004.

[11] Saint John Walker. Big data: A revolution that will transform how we live, work, and think. *International Journal of Advertising*, 33(1):181–183, 2014.

[12] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.