



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék

Varga Benedek

**DEAD RECKONING ELJÁRÁS FEJLESZTÉSE
TÖMEGFELÜGYELETI RENDSZEREKHEZ**

KONZULENS

Dr. Simon Vilmos

BUDAPEST, 2017

Tartalomjegyzék

| | |
|---|----|
| Összefoglaló..... | 3 |
| Abstract | 5 |
| 1. Tömegfelügyelet - bevezető..... | 6 |
| 2. Tömegfelügyeleti rendszerek csoportosítása technológia alapján | 8 |
| 2.1 Bluetooth | 8 |
| 2.2 CCTV – Zárt láncú kamerarendszer | 9 |
| 2.3 Közösségi érzékelés | 10 |
| 3. A dead reckoning eljárás..... | 12 |
| 4. Mozgásérzékelő szenzorok | 15 |
| 4.1 Gyorsulásmérő | 15 |
| 4.2 Giroszkóp | 17 |
| 4.3 Magnetométer | 17 |
| 4.4 Szenzoradatok elérése iOS eszközökön | 19 |
| 5. Dead reckoning algoritmus tömegfelügyeleti rendszerekhez | 21 |
| 6. Dead reckoning mobilalkalmazás..... | 28 |
| 7. Teszt és validáció..... | 35 |
| 8. Konklúzió..... | 39 |
| Irodalomjegyzék..... | 41 |
| Táblázatjegyzék..... | 43 |
| Ábrajegyzék | 44 |

Összefoglaló

Évről évre egyre több fesztiválon, tömegrendezvényen vehetünk részt, amelyek sajnos a szórakozáson túl veszélyeket is hordoznak magukban. Megjelent az igény olyan tömegfelügyeleti rendszerekre, amelyek képesek monitorozni a tömegdinamikai változásokat, ezáltal előrejelezni a tömegkatasztrófákat, hogy a szervezők időben fel tudjanak készülni ezek megelőzésére. A kiépítendő infrastruktúra minimalizálása érdekében célszerű a látogatók mobiltelefonjait monitorozó eszközökként is felhasználni, amelyek során a dead reckoning eljárás segítségével energiatakarékos módon gyűjthetjük be a látogatók által bejárt útvonalakat. Ezeket feldolgozva a szervereken, képet kaphatunk a rendezvény tömegdinamikai jellemzőiről, előre láthatjuk a tömegkatasztrófák kialakulásának veszélyét, így meg tudjuk előzni azokat. A dead reckoning eljárás, szemben a folyamatos GPS kapcsolaton alapuló útvonalrögzítő megoldásokkal, hatékony energiafelhasználású, nem meríti nagy mértékben az okostelefon akkumulátorát, ami egy fesztiválon kritikus lehet. Viszont megfelelő pontosságú ahhoz, hogy a szervezők a tömegdinamikai leírókat naprakészen követhessék.

A rendszer egyik kritikus pontja az adatok tárolása és azok feldolgozása mellett az, hogy minél pontosabb információt kapjunk arról, hogy a résztvevők hol tartózkodnak a rendezvény területén belül és mindezt minél kevesebb infrastruktúrális beruházással kelljen megvalósítani. Az okostelefonokon futó dead reckoning algoritmus kifejlesztése és implementálása komoly kihívás, amelynek megoldására vállalkoztam a TDK dolgozatomban.

A TDK munkámban a tömegfelügyeleti rendszerek rövid bemutatása után ismertetem a dead reckoning eljárást, annak előnyeit és hátrányait a GPS-es útvonalrögzítéssel szemben, valamint hogy milyen szenzorokból érkező adatokat használok fel az útvonal kiszámításához. Részletesen kitérek az általam fejlesztett dead reckoning algoritmus egyes moduljaira: lépésmeghatározás, haladási irány meghatározása, pozíciók kiszámítása. Az algoritmust a fejlesztés támogatása érdekében implementáltam iOS platformra. Az alkalmazás funkcióinak és működési elvének ismertetése után bemutatom a fejlesztés lépéseit és nehézségeit, illetve az egyes modulok iOS specifikus implementálását.

Végezetül validációs tesztekkel támasztom alá a megoldást. Ehhez kézben és zsebben tartott okostelefonnal rögzített útvonalakat hasonlítok össze, amiket lemértem GPS-el és a

dead reckoning eljárással is, és összevetem a valós trajektóriával. A validáció után röviden kitérek a technológia továbbfejlesztésének további lehetőségeire is.

Abstract

We can see a rising popularity of festivals and events from year to year. Beside the fun they come with danger. That is why organizers are in need for an effective mass surveillance system that helps monitoring mass dynamics changes, to forecast and prevent mass disasters. To minimize the infrastructure costs, we can use the smartphones of participants to gather information about their location with the energy efficient dead reckoning method. By processing the data on the servers, we can gain information about the mass dynamics' descriptive parameters, which enables us to foresee and prevent disasters. Unlike GPS, dead reckoning is much more energy efficient for the smartphone users, does not drain the user's battery quickly, which is a scarce resource at a mass gathering. However it is accurate enough to provide us the mass dynamics most important state variables on time.

Developing and implementing the dead reckoning algorithm on a smartphone is a great challenge which I chose as the goal of this TDK work. In my work mass surveillance systems are presented, furthermore the development of my dead reckoning algorithm, with the pros and cons compared with the GPS based track recording, together with the sensors which I use to gather data to calculate the trace.

I have implemented my algorithm on an iOS platform, presenting the functions and features of the application in my work and I also showing examples of the iOS specific developing process. I conclude my work with a validation of my algorithm with real measurements tests. I have compared traces recorded with handheld and in-pocket smartphones with GPS based traces and the real trajectory.

1. Tömegfelügyelet - bevezető

Évről-évre egyre több több tömegrendezvényen vehetünk részt, amik óriási látogatottságnak örvendenek a színes programok miatt. Ugyanakkor a szórakozáson kívül komoly veszélyforrások is ezek az események, hiszen a nagy tömegben a legkisebb pánik is könnyen katasztrófához vezethet. Egy veszélyes helyzet bárhol és bármikor kialakulhat, ami a szervezőket gyakorlatilag felkészületlenül érheti, és megfékezni idővel már nagyon nehéz, épp ezért rendkívül gyorsan fel kell tudni ismerni és kezelni kell ezeket a helyzeteket. Emiatt az alábbi kérdések megválaszolása nagyban segítheti a felügyelő szervezetet a gyors és pontos reagálásban:

- A rendezvény egyes területein hozzávetőleg mennyi ember tartózkodik?
- Hogyan tudnánk detektálni azt, ha a tömeg egy adott területen belül kritikus méretűre nő?
- Előre tudjuk-e jelezni a tömegkatasztrófák kialakulását a tömeg mozgását figyelve?
- Értesítéseket küldve a vendégeknek megfékezhető-e a pánik kialakulása?

Sajnos minden évben előfordulnak tömegkatasztrófák, amik talán megelőzhetőek lennének, ha több információ állna rendelkezésükre. Tragikus példáért nem is kell messzire menni: Németországban a 2010-es Love Parade-on 21 ember halt meg és kb. 500-an sérültek meg, amikor egy alagútnál a szűk keresztmetszet miatt összezsúfolódtott a tömeg, és az emberek összenyomták egymást [1]. Ha a szervezők látják, hogy a tömeg sűrűsége egy kritikus pont felé mozog, időben kihelyezett kordonokkal és tereléssel talán megelőzhető lehetett volna a katasztrófa. De a mekkai zarándoklat, a *hajj* idején is történtek már borzalmas tömegkatasztrófák, 1990-ben 1492-en, 2004-ben 251-en, 2006-ban 432-en, 2015-ben 2236-an vesztették életüket [2]. Ezek a halálesetek mind abból adódtak, hogy az emberek a nagy tömegben összezsúfolódtak és összetaposták egymást.

A tendencia azt mutatja, hogy a tömegrendezvények egyre népszerűbbek az emberek körében és évről-évre egyre több látogatót vonzanak. A sok látogató miatt egyre nagyobb szükség van egy olyan támogató rendszerre, amely segítségével jobban megfigyelhetik, megérthetik a szervezők a tömeg mozgását, és ezen információk alapján gyorsan és pontosan avatkozhatnak be a katasztrófa elkerülése érdekében, amivel akár életet is menthetnek. A rendszer működésének kritikus pontja, hogy a felhasználók automatikusan tájékoztassák a

rendszer arról, hogy éppen merre járnak. Természetesen ez nem az egyes felhasználók nyomon követését jelenti, kizárólag anonim módon küldenék be ezeket az adatokat.

Komoly segítséget jelent, ha a tömeg mozgásáról begyűjtött adatokat vissza tudjuk játszani, és a beérkezett információkat újra és újra ki tudjuk értékelni. Ebből fontos következtetéseket vonhatunk le arra vonatkozóan, hogy mik voltak a legnépszerűbb helyek, mely kereszteződésekön haladt át a legtöbb ember, milyen irányban mozogtak az egyes szakaszokon az emberek, volt-e esetleg olyan pont, ahol az emberek mozgása hirtelen lelassult. Az utólagos kiértékeléssel meghatározhatók biztonsági szempontból kritikus pontok, zónák, amiket eleve nagyobb készültséggel figyelhet a biztonsági szolgálat, de akár az útvonalak utólagos módosításával biztonságosabbá is tehetőek ezek a területek.

A tömegrendezvények felügyeletére többféle technológiai megoldás is létezik már. Ezeknek vannak erősségeik és hátrányaik is. A következő fejezetben a technológia bemutatása mellett példákkal is illusztrálom a már meglévő rendszereket.

2. Tömegfelügyeleti rendszerek csoportosítása technológia alapján

Többféle módszer is létezik arra, hogy megtudjuk, ki merre tartózkodik a rendezvény területén belül. Ezeknek megvannak az előnyeik és a hátrányaik is, általában mindig egy speciális igényt vagy helyzetet figyelembe véve történik a rendszerek kiépítése, így nincs általánosan jó megoldás. A saját rendszerünk megtervezéséhez fontos, hogy ismerjük ezen technológiák előnyeit és hátrányait, így a számunkra leginkább megfelelő megoldást tudjuk választani. Ezért ebben a fejezetben sorra veszem a már létező technológiákat, amik segítségével adatokat gyűjthetünk egy tömegfelügyeleti rendszer számára.

2.1 Bluetooth

Az egyik ilyen technológia a Bluetooth, ami egy rövid hatótávolságú adatcseréhez használt, nyílt, vezeték nélküli szabvány. Előnye, hogy gyakorlatilag az összes okostelefonban megtalálható, valamint automatikusan létesíthetünk vele kapcsolatot eszközök között, így elég, ha csak be van kapcsolva a felhasználó készülékén. Elhelyezhetünk Bluetooth kapukat, beacon-öket, amik, ha érzékelnek egy-egy telefont, a jelerősségből és a háromszögelés módszerével viszonylag pontosan meg tudják határozni az egyes emberek helyzetét. Viszont ahhoz, hogy egy több hektáros területet - ahol adott esetben akár több tízezer is lehetnek – lefedjünk ezzel a technológiával nagymértékű infrastrukturális beruházásra van szükség, valamint a nagy terhelés miatt a pozíciók pontossága is kérdéses lehet. Ezenkívül a felhasználók mobiltelefonjában aktívnak kell lennie a Bluetooth-nak. Ez a tény nagyban befolyásolja a rendszer pontosságát. Ezt át lehet hidalni azzal, ha a belépőbe, karszalagba beépítünk egy alacsony fogyasztású Bluetooth chipet. Mivel a karszalag állandóan rajta van a résztvevőkön, ezért állandó információval látja el a szervezőket a felhasználó mozgásáról. Sajnos azonban ez meg nagyobb mértékben megnöveli az infrastrukturális költségeket.

Egy ilyen rendszert és egy hozzá fejlesztett mobilalkalmazást mutat be a [3], ahol a rendszer segítségével a szervezők információt kaphatnak a résztvevők helyzetéről, valamint üzeneteket tudnak küldeni nekik vészhelyzet esetén.

Fő céljaik a következők voltak:

- Tömegdinamikai jellemzők mérése
- Adatok felhasználásával hasznos információ előállítása a szervezők részére
- Dedikált kommunikációs csatorna biztosítása a szervezők és a résztvevők között.

A koncepció egy jól működő rendszert írt le, azonban a rendszer hibája az volt, hogy a Bluetooth kapukat kizárólag a be- és kijáratoknál helyezték el, aminek eredményeképp a fesztivál belsejében uralkodó tömegdinamikai viszonyokról nem kaptak semmiféle információt, csak az éppen a fesztivál területén tartózkodó látogatók létszámát ismerték.

2.2 CCTV – Zárt láncú kamerarendszer

A CCTV, vagyis a Closed Circuit Television rendszer egy belső elérésű kamerahálózat. A kezelők több kamera képét tudják nyomon követni egyszerre, valamint kívülről nem elérhető a kamerák adása. Talán ez a legelterjedtebb módszer manapság a tömegfelügyeleti technológiák közül. Nem csak a tömegrendezvények megfigyelésére alkalmazzák őket, gyakorlatilag bárhol jó szolgálatot tesznek, ahol valamilyen megfigyelésre van szükség. Használják kamerás rendszereket az autóforgalom monitorozására, aminek segítségével a balesetek azonnal észrevehetőek, és így időben értesíteni tudják a mentőket, rendőröket, esetleg figyelmeztetni lehet a baleset közelében autózókat, amivel csökkenthető a kialakuló dugó mérete. De a bűnmegelőzésben, bűnüldözésben, közúti szabálytalanságok észlelésében is nagy szerepük van. A megfigyelt területen a kamerák segítségével könnyen észrevehető egy-egy körözött személy, lefülelhetők a tolvajok és a gyorsajtók is.

Az Egyesült Királyság területén 2011 márciusáig 1,85 millió kamerát telepítettek [4] ilyen célból, mely közül rengeteg magánkézben van, és privát rendszerekben teljesít állandó szolgálatot, de ezek felvételéhez a hatóságok büntény esetén hozzáférhetnek, és bizonyítékként használhatják fel az anyagot.

A fent említett felhasználási területeken a kamerás rendszerek jól használhatóak, de a tömegrendezvényeken a tömeg mozgásának megfigyelésére sajnos csak részben alkalmasak. Jelentős hiányosságuk az, hogy nem tudnak egy átfogó, aggregált képet adni a tömegről. Hiába helyezünk el sok kamerát a rendezvény területén, azokat csak önmagukban lehetséges értelmezni, így a tömeg méretére és a tömegdinamikai jellemzők változására is csak durva becsléseket tudunk adni. Használhatunk ugyan drónokat a statikus földhöz közeli kamerák helyett vagy mellett, de ezeket nem tudjuk állandóan üzemeltetni. Viszont vészhelyzet esetén

elő képet tudnak biztosítani a tömeg egészségének mozgásáról, ami nagyban segítheti a szervezőket, valamint fény és hanghatások a tömeg irányítását is elősegíthetik.

2.3 Közösségi érzékelés

Talán a legelőreemutatóbb tömegfelügyeleti technológia a közösségi érzékelésen alapuló megoldás vagyis a crowdsensing. Ennek lényege, hogy az adatok begyűjtését és feldolgozásának egy részét kiszervezzük a felhasználók okostelefonjaiba, így nagy tömeg esetén sem lesz a rendszerünk leterhelt. Ezentúl a pozíciók meghatározását sem egy kiépítendő infrastruktúrára bizzuk, mint a Bluetooth-os vagy kameraképes megoldás esetén, hanem ezt is a felhasználók okostelefonjai végzik.

Erről az elképzelésről először 2013-ban jelent meg egy cikk [5], a rendszer két részből állt: egy okostelefonos alkalmazásból, ami másodpercenként rögzítette a résztvevők GPS koordinátáit, majd azokat aggregálva beküldte a másik komponensnek, egy szervernek, ami feldolgozta és kiértékelte a kapott információt. A feldolgozás után a rendszer hőtérképet állított elő a különböző tömegdinamikai jellemzők szemléltetése érdekében. A következő jellemzők alakulását figyelték meg a szervezők: sűrűség, sebesség, turbulencia, tömegnyomás [6]. A színek segítségével a szervezők könnyen felismerhették a kritikus területeket, így gyorsan tudtak reagálni a kialakulóban lévő vészhelyzetre.

A fenti rendszer kifejlesztői rámutattak a crowdsensing lényegére: a pozíciók meghatározását rá kell bízni a felhasználók eszközeire, ezáltal csökkenthetők az infrastrukturális költségek. Így a központi egységnek már csak aggregálni kell az adatokat, és azokat elérhetővé tenni azonnali, illetve későbbi kiértékelésre.

Ahogy a fenti példából is látszik, a felhasználók pozíciójának meghatározása történhet klasszikusan, GPS segítségével. Ez a módszer biztosítja az állandó és pontos információt az egyének pozíciójáról, azonban a folyamatos kapcsolatot igényel a műholdakkal, ami jelentős áramfelvétellel jár. Ez az akkumulátorok gyors lemerüléséhez vezet, ami kritikus erőforrás a fesztiválozók életében, mivel az akkumulátorok újratöltése nehézkes és olykor költséges is. Hiába ad ez a módszer megfelelő információt a résztvevők helyzetéről, sajnos az említett hátrány miatt nem lehet megfelelően kihasználni. Emiatt mindenképp egy olyan megoldást kell találni, ami amellet, hogy megfelelően pontos, kis energiaigényű is.

Egy másik eljárás szolgálhat válaszul a GPS hátrányaira. Ez a módszer a dead reckoning, ami kezdetben csupán egy GPS koordinátát igényel, a további pozíciót pedig az okostelefon

gyorsulásmérőjének és iránytűjének adatait felhasználva számítja ki. A módszer elég pontos ahhoz, hogy meg tudja mondani, hogy a felhasználó merre tartózkodik a fesztivál területén, és elég kevés energiát fogyaszt ahhoz, hogy folyamatosan futtasson a háttérben, hiszen nem állandóan van bekapcsolva GPS - csupán a mérések elején egy rövid ideig – és a telefonba szerelt mozgásérzékelő szenzorok rendkívül kis energiaigényűek. Ezek alkalmassá teszik egy crowdsensing rendszer információkkal való ellátására, és emiatt esett erre a módszerre a választásom. Sajnos azonban a szakirodalomban a dead reckoning eljárások rosszul dokumentáltak, valamint konkrét implementáció sem található meg. A legtöbb esetben csupán elméleti szinten valósították meg, nem próbálták ki élesben.

Munkám további részében azt fogom bemutatni, hogy fejlesztettem ki egy működő dead reckoning algoritmust, valamint ezt hogyan implementáltam egy iOS alkalmazásban, amivel aztán valós méréseket lehet végezni.

3. A dead reckoning eljárás

A második fejezetben felsorolt pozíciókövetési technológiák közül végül a dead reckoning módszerre esett a választásom. A döntés oka az volt, hogy alacsony energia használat mellett is megbízhatóan meghatározható, hogy a felhasználó a fesztivál területét felosztó cellák¹ közül éppen melyikben tartózkodik [7]. Nézzük meg részletesebben is a dead reckoning eljárást, eredetét, előnyeit és hátrányait.

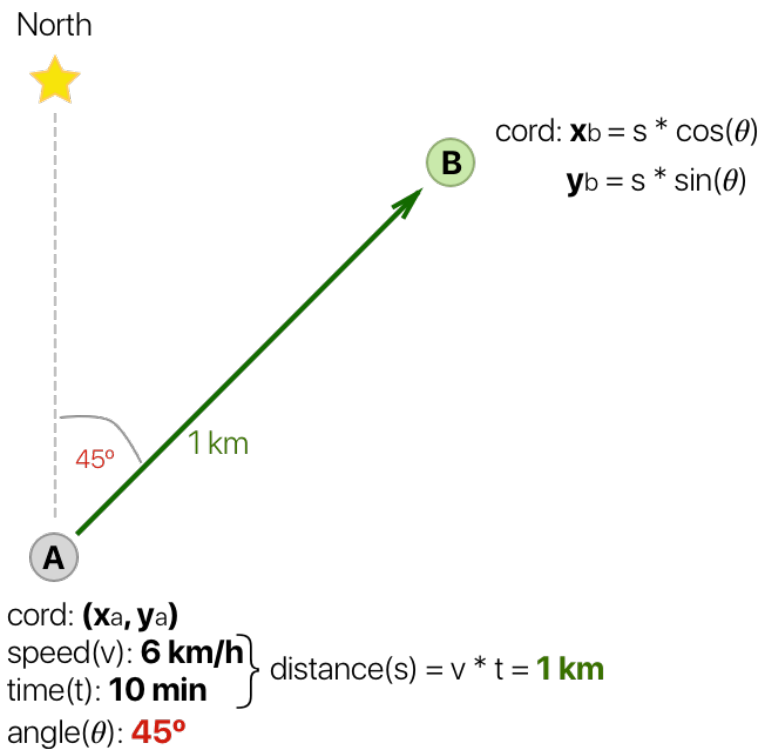
A módszer gyakorlatilag egyidős a tengeri navigációval. A hajósok egy fix pontból kiindulva, a térkép segítségével határozták meg, hogy az eltelt idő, a haladási irány és sebesség alapján hol tartózkodhatnak. Használták a jelenlegi helyzet kiszámítására és útvonaltervezésre is.

A repülés megjelenésével a pilóták is alkalmazták a módszert. 1927-ben például Charles Lindberg sikeres repülést hajtott végre az USA-ból Párizsba egy egymotoros géppel [8]. Útja során a navigációt a dead reckoning eljárással oldotta meg [9].

Láthatjuk, hogy a GPS megjelenéséig gyakorlatilag a bevett és biztos módszer a navigációra a dead reckoning volt. De miután megjelent a műhold alapú helyzetmeghatározás feleslegessé vált a módszer, mert a legtöbb esetben nem kritikus szempont a GPS eszköz áramfelvétele, viszont használatával rendkívül pontosan és biztosan meghatározhatjuk a lokációt.

A módszer nagyon egyszerű. Ismerjük a kiindulási pont koordinátáját, valamint azt, hogy a következő pont ilyen messze, és milyen szögben helyezkedik el az előző ponthoz képest. Így egy egyszerű trigonometrikus számítással megkapjuk a következő pont koordinátáját is. Ha nem visszamenőleg szeretnénk kiszámolni az útvonalat, hanem előre tervezni, akkor a haladási sebességet és az eltelt időt kell vennünk, amikből szintén könnyen megkaphatjuk a megtett utat, innen pedig már könnyen megkapjuk a végkoordinátát.

¹ A fesztivál területét a szervezők négyzetekre bontják, amiket celláknak nevezünk. Célszerű a celláknak követniük az úthálózatot, kiemelt helyeket. A felosztás utólag akár módosítható, finomítható is az első napok adatai alapján.



Ábra 3.1 – Következő pozíció kiszámítása az irány, a sebesség és az idő ismeretében.

A dead reckoning, úgy, mint bármelyik másik eljárás érzékeny a hibás bemenetre, ha bármelyik változó a formulában (kezdőkoordináta, sebesség, idő, orientáció) hibás, a kiszámolt pozíció már nem lesz pontos. Mivel a következő koordináta mindig relatív az előzőhöz képest, így minden hiba kumulatív a mérések során és a végeredmény jelentősen eltérhet a tényleges pozíciótól. Ennek a problémának a kiküszöbölésére, vagy legalábbis lecsökkentésére jó megoldás lehet az, hogy bizonyos időközönként, a megtett távolság után újramérjük a jelenlegi pozíciókat GPS-el. Így kapunk egy új biztos pontot, ahonnan folytatva a számolást pontosabb eredményhez jutunk.

Jelenlegi rendszerünkben az újramérést két esetben érdemes elvégezni. Egyrészt bizonyos eltelt idő után, másrészt, ha az alkalmazás úgy érzékeli, hogy a látogató cellát váltott. Mivel cellaváltáskor történik az adatok felküldése a szervereknek, ezért ezzel egy időben érdemes az új GPS koordinátát is mérni, így pontosabban tudjuk folytatni az útvonalrögzítést. Mivel ezek az események időben elég távol vannak egymástól, ezért a GPS időszakos ki- és bekapcsolása sokkal kisebb energiafelvételt jelent, mint ha folyamatosan vennénk a GPS jelet.

Mindezen hibák ellenére a dead reckoning eljárás alkalmazása a fesztiválozók nyomon követésére megfelelő módszer. A rendszer sajátosságaiból adódóan nincs szükségünk a felhasználók méterre pontos pozíciójára, csupán azt kell tudnunk, hogy adott időpillanatban

melyik cellában hányan vannak, illetve, hogy merre mozognak. Nem az egyes személyek egyedi mozgására vagyunk kíváncsiak, hanem a tömeg egészének dinamikájára, ami meghatározható kevésbé pontos adatokból is, emellett a legkritikusabb kérdésben, az akkumulátor használatban pedig jóval felülmúlja a GPS-es megoldást, így ezt érdemes alkalmazni.

Mivel a számításokhoz szükséges adatokat az okostelefonokban található mozgásérzékelő szenzorok szolgáltatják, ezért feltétlen tisztában kell lennünk azok működésével, elérésével, valamint azzal, hogy pontosan milyen mozgást tudunk jellemezni segítségükkel. Így a következő fejezetben ismertetem a népszerű mozgásérzékelő szenzorok technológiai hátterét, valamint azt, hogy ezek közül melyikre van szükségem az algoritmus működéséhez.

4. Mozgásérzékelő szenzorok

Manapság már szinte minden okostelefonban megtalálhatóak az alacsony fogyasztású mozgásérzékelő szenzorok. Három fő típusuk van: gyorsulásmérő, giroszkóp és magnetométer. Ezeken kívül egyes eszközökben barométer is megtalálható, ami a telefon tengersizint feletti magasságáról ad információt.

Ezek a szenzorok egyrészt az operációs rendszer működéséhez kellenek, mivel segítségükkel a felhasználók vezérelhetik azt, például a kijelző elforgatása, ha elfektetjük a telefont, kamera orientációjának állítása a telefon helyzetéhez képest. Használatuk népszerű a játékfejlesztők körében is: elsőként a Nintendo használt mozgásérzékelő szenzorokat a konzoljaiban, de mára már a legtöbb okostelefonos játék az irányítást kiegészíti a telefon döntögetésével, valamint már az újabb konzolok (PlayStation, XBox) controllerjeiben is megtalálható ez a technológia. A játékok irányításán kívül állandó információt is biztosíthatnak a mozgási aktivitásunkról is: figyelik a megtett lépések számát, a megtett kilométereket, megmászott emeleteket. Ahhoz, hogy ezek az információk állandóan a rendelkezésünkre álljanak, egy alacsony fogyasztású chipen helyezik el ezeket a szenzorokat, valamint általában egy szintén nagyon alacsony fogyasztású processzormag vagy co-processzor felel az üzemeltetésükért. Ezzel a megoldással érhető el az, hogy gyakorlatilag egyfolytában üzemeljenek, még akkor is, amikor a készülék alvó módban van, anélkül, hogy a felhasználó észrevenné azt az akkumulátora merüléséből.

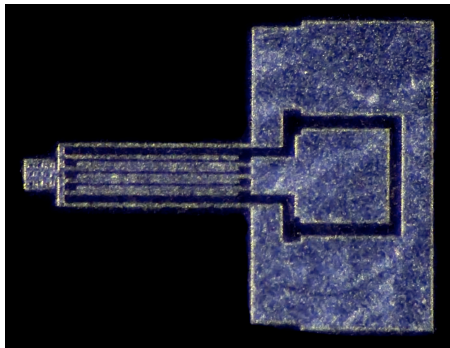
Az Apple például már 2013 óta, az iPhone 5S-ben először [10], alkalmazza ezt a megoldást „motion coprocessor” néven. Ez a processzor felel a mozgásérzékelő szenzorokból érkező adatok begyűjtéséért, feldolgozásáért és tárolásáért rendkívül energiahatékony módon. A technológia évről-évre fejlődik, kiegészült azóta barométerrel, valamint a beszédfelismerő rendszer (Siri) felébresztése érdekében a mikrofont is figyeli.

4.1 Gyorsulásmérő

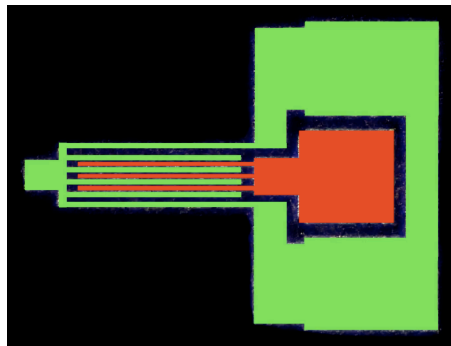
A gyorsulásmérőt (accelerometer) legegyszerűbben egy rugó végén tartott mozgó tömegként képzelhetjük el. Ahogy a tömeg a gravitációs gyorsulás hatására mozgatja a rugót, úgy adja vissza az értékeket a gyorsulásmérő. Ha a gyorsulásmérő nullát mutat, akkor szabadesésben van a rendszer, nyugalmi állapotban pedig a gravitáció értékével húzza a tömeg a rugót.

Természetesen az okostelefonokban nem rugóval működik a gyorsulásmérő. A MEMS (Micro Electro-Mechanical Systems) segítségével valósítják meg őket, amik olyanok, mint az integrált áramkörök, de működésükben mechanikaiak [11]. Hasonlóan készítik őket, mint az áramköröket, de valójában miniatűr mechanikus egységekből állnak, amik kapcsolódnak az áramkörökhöz és információt tudnak szolgáltatni számunkra.

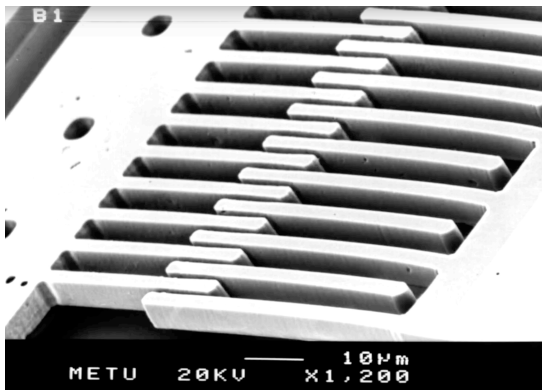
A gyorsulásmérőket valójában két fésűszerűen összekapcsolt kondenzátor alkotja. Viszont ezek a kondenzátorok képesek elmozdulni egy súly mozgásának hatására. A mozgás érzékelhető a fésűs szerkezetben, mivel a fésű ujjainak egymáshoz való távolsága változik, ami hatására megváltozik az ellenállás értéke. Ezt az értéket könnyen át tudjuk alakítani digitális értéké, így tudjuk értelmezni a gyorsulásbeli változásokat.



Ábra 4.1 – Egyszerű gyorsulásmérő [12]



Ábra 4.2 – A gyorsulásmérő kondenzátorai [12]



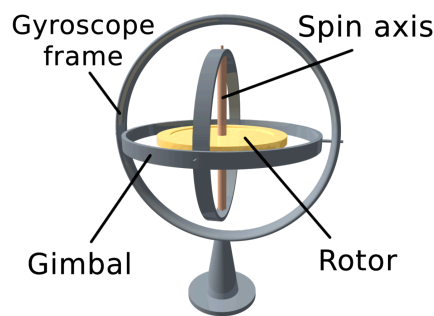
Ábra 4.3 – A modern gyorsulásmérőknek a fésűs szerkezete sok ujjból áll [12]

A gyorsulásmérőt használjuk arra, hogy detektáljuk, hogy a telefon állítva vagy fektetve van-e, de a pillanatnyi gyorsulási értéket lekérve és felhasználva a telefon mozgására is tehetünk következtetéseket. A munkámban a gyorsulásmérőből kinyert értékeket használom fel arra, hogy detektáljam a felhasználó lépéseit.

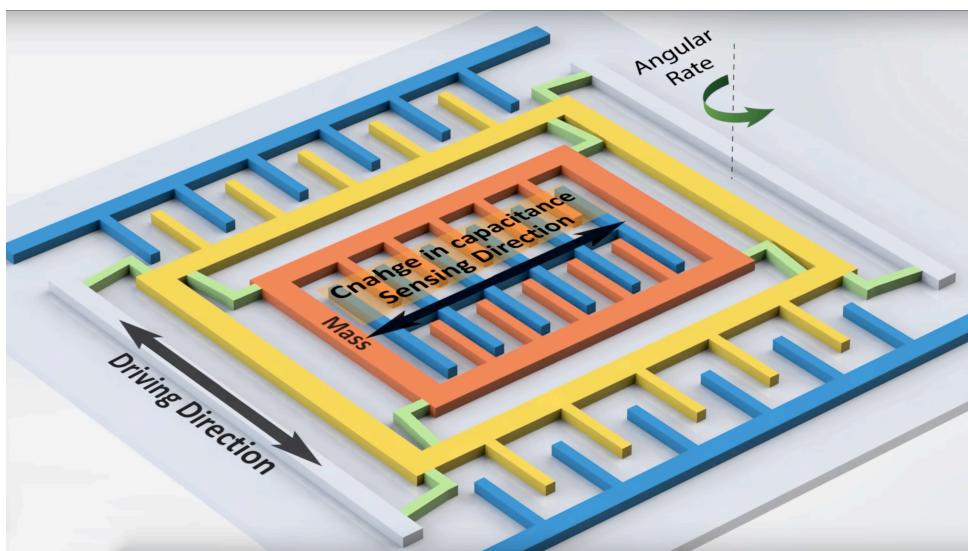
4.2 Giroszkóp

A giroszkóp a Coriolis-erőt kihasználva mérni tudja a szögsebességét a készüléknek. Ha elmozdul a készülék, akkor a giroszkóp tengelyére merőlegesen elmozdul a tömeg, amíg megfigyelve információt kaphatunk a készülék mozgásáról.

A MEMS giroszkópok közepén egy állandóan mozgó, oszcilláló tömeg található, aminek merőleges elmozdulása változást eredményez az ellenállásban. Az ellenállás mértékét figyelve kapunk eredményt. A giroszkópból származó adatokat nem használom a dead reckoning algoritmusban, mivel a szögsebesség értékeiből nem tudok semmi releváns adatra következtetni.



Ábra 4.4 – Klasszikus giroszkóp [13]

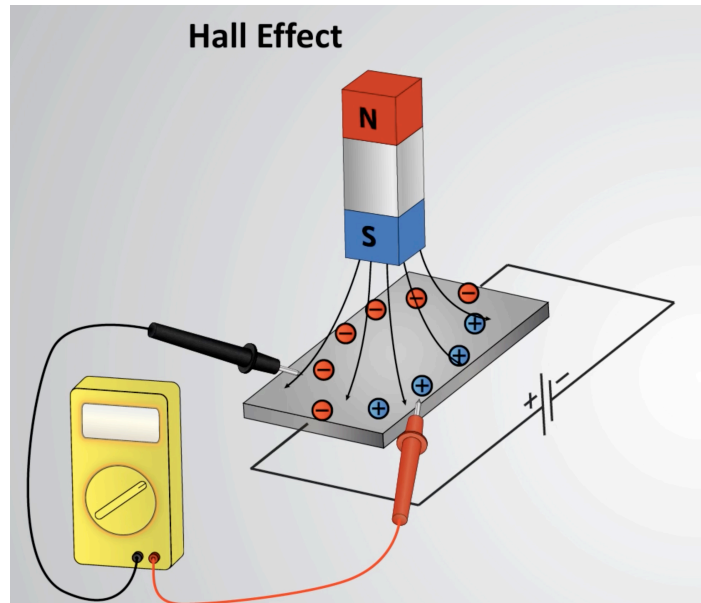


Ábra 4.5 – MEMS giroszkóp [14]

4.3 Magnetométer

A magnetométer iránytűként szolgál az okostelefonokban. A Föld mágneses mezejét méri a Hall-effektus segítségével. A készülékekben egy áramvezető lapkán keresztül közlünk

elektromos áramot. Az elektronok normális esetben egyenesen mennek át a lapka egyik oldalától a másikig. Ha azonban mágneses erő éri a rendszert, az elektronok eltérnek az egyenes iránytól. A protonok az egyik, míg az elektronok a másik oldalán fognak mozogni a lapkának. Ezáltal feszültséget mérhetünk a lapka két oldala között, aminek a nagyságából következtethetünk a készülék orientációjára.



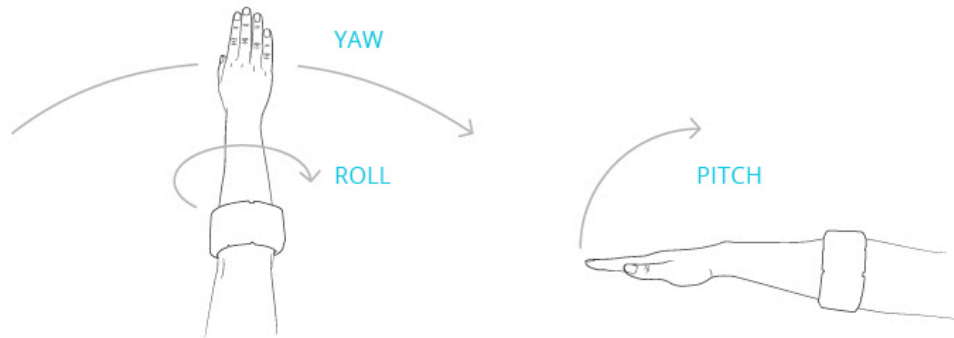
Kép 4.6 – Magnetométer működése a Hall-effektust kihasználva [14]

A magnetométer tengelyeinek értékeit önmagukban nem használom fel, viszont az iOS beépített iránytűje használja. Az iránytű aggregálja a három tengely értékeit és ezekből ad pontos értéket a telefon Északtól való helyzetéről. Érdekes az iránytűt használni a magnetométerrel szemben, mivel kevesebb zajt tartalmaz a belőle származó idősor. Én is az iránytű értékeit használom fel a dead reckoning algoritmushoz.

A felsorolt szenzorok mindegyike megtalálható a legtöbb okostelefonban, és természetesen mindhárom tengely mentén szolgáltatnak adatot. A három szenzorból származó adatokat önmagukban is értelmezhetjük, de kombinálhatjuk is őket. Ezt az eljárást sensor-fusion-nek nevezzük és virtuális szenzorokat hozunk velük létre. Így sokkal pontosabb és zajmentesebb adatokat kaphatunk a telefon mozgásállapot változásairól, valamint új információkhoz is juthatunk.

A három szenzor együttes értelmezésével létrehozhatunk egy úgynevezett Tilt-szenzort, amiből a telefon orientációját kapjuk meg a pitch, roll és yaw tengelyek mentén [15]. Kezdetben az volt az ötletem, hogy ebből határozom meg a telefon haladási irányát. Viszont a tilt-szenzorból származó idősorok elemzése során szembesültem vele, hogy

rendkívül zajos a belőle származó információ, nagyon érzékeny a legkisebb orientációs változásokra is, valamint egységes referenciapont sem biztosítható a felhasználók számára, így nem tudom meghatározni egymáshoz képest a helyzetüket. Ezért döntöttem végül az iránytű használata mellett.



Ábra 4.7 – A pitch, roll és yaw tengelyek értelmezése [16]

4.4 Szenzoradatok elérése iOS eszközökön

Munkám során megvizsgáltam, hogy az iOS API-t használva milyen adatokhoz juthatok hozzá. Az API dokumentációból [17] kiderült, hogy többféle szenzorból és virtuális szenzorból érkező adatot érhetek el: gyorsulásmérő, giroszkóp, magnetométer, gravitációs szenzor, magasságérzékelő, orientációs szenzor és iránytű, és a legtöbb adat három tengely mentén értelmezett, és az egyes tengelyeket külön külön érhetem el.

Számomra két szenzorból érkező információ a lényeges: a gyorsulásmérő segítségével detektálom a felhasználó lépéseinek időpillanatát és hosszát, az iránytű segítségével pedig a haladási irányát.

Az iOS API segítségével többféle módon is meghatározhatjuk a telefon orientációját. Egyrészt használhatjuk a Tilt-szenzorból érkező adatokat a pitch, roll és yaw tengelyeken értelmezve. A yaw tengely – ez felel meg az y-tengely menti forgásnak - adataiból meghatározhatnánk a telefon haladási irányát, viszont egyrészt a kapott információ sétálás közben rendkívül zajos, másrészt az adatok relatívak a telefonok saját tengelyéhez. Ha ezt a tengelyt a készülékek egyedi módon kalibrálják maguknak nem tudjuk összevetni több

készülék adatait, így az aggregáció sem lehetséges. Ezért ezt a módszert elvettem és az iránytűt használtam.

A telefon helyzetét a Föld mágneses mezejéhez viszonyítva többféle módon is elérhetjük. Lekérhetjük a magnetométerből x, y és z tengelyenként, vagy használhatjuk az iránytű [18] virtuális szenzort. Az iránytű a magnetométer mindhárom tengelyét figyelembe véve határozza meg, hogy a készülék Északhoz képest milyen irányba néz. Működik minden helyzetben, mivel mindig a telefon tetejéhez viszonyítva kapjuk meg az értékeket. Ez az adat több szempontból is jobb, mint a tilt-szenzorból érkező: egyrészt kevésbé érzékeny a zajokra, mivel a három tengelyt együttesen használva a zajok kiszűrhetőek, valamint egy biztos ponthoz, Északhoz képest kapjuk meg a telefon orientációját. Így a kalibrációval sem kell foglalkoznunk, tudjuk egyből aggregálni a több telefonból érkező adatokat. Valamint az Apple az iránytű dokumentációjában is azt javasolja a fejlesztőknek, hogy ezt az adatot használják, ha azt szeretnék megtudni, hogy milyen irányba néz a készülék.

5. Dead reckoning algoritmus tömegfelügyeleti rendszerekhez

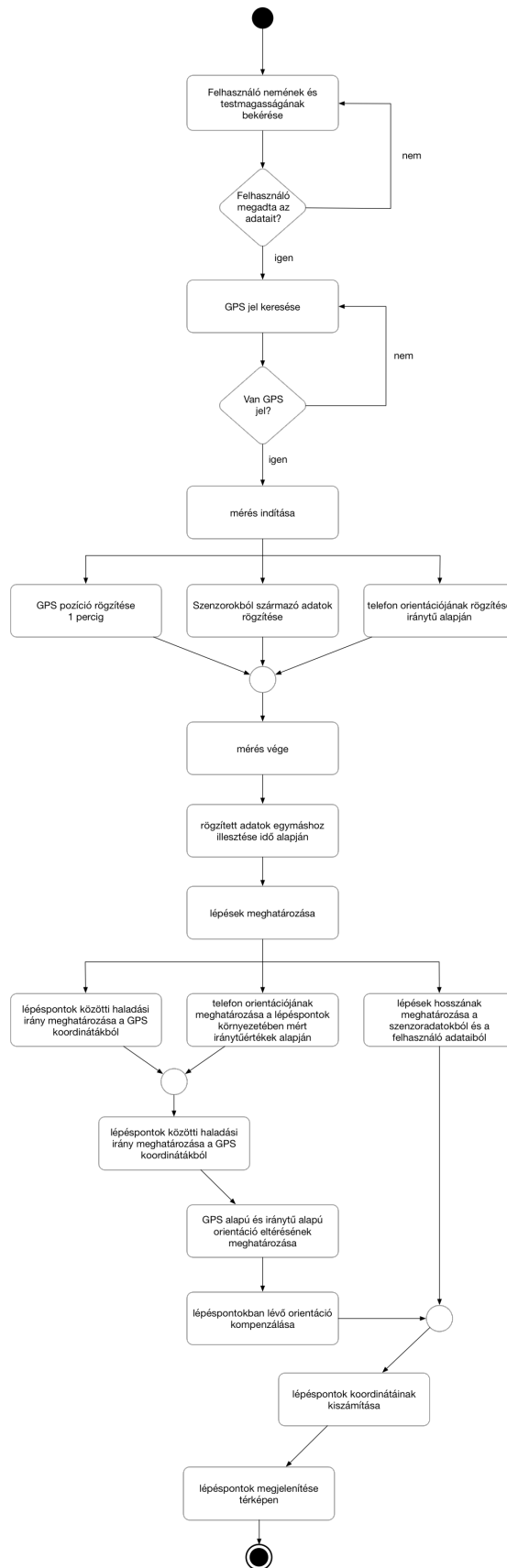
A tömegfelügyeleti rendszerekhez használt technológiák ismertetésénél már láthattuk, hogy minden szempontot figyelembe véve a dead reckoning eljárás a legoptimálisabb egy energiafogyasztás szempontjából kritikus alkalmazási területre. Mivel a BME Hálózati Rendszerek és Szolgáltatások Rendszerek Tanszéken fejlesztett tömegfelügyeleti rendszer [7] cellákra, vagyis négyszögletű területekre osztja a tömegrendezvény területét, ezért elég azt meghatározni, hogy melyik cellában hány vendég tartózkodik. Ennek biztosítására a dead reckoning eljárás elég pontos. Fontosabb szempont az energiatakarékosság. Mivel az okostelefonokban található mozgásérzékelő szenzorok rendkívül alacsony fogyasztásúak, ezért azok használata mindenképp előnyösebb a GPS-szel szemben. Ezért választottam a felhasználók nyomon követésére a dead reckoning eljárást. A továbbiakban az algoritmust és a fejlesztés lépéseit mutatom be.

Ahogy korábban már láttuk, a klasszikus dead reckoning eljárás nem historikus adatokkal dolgozik, hanem az aktuális haladási irányból és sebességből, valamint egy ismert pont koordinátáiból próbálja meghatározni az érkezési pontot. Viszont számomra ezek az adatok nem mindig állnak rendelkezésre. A haladási sebességet nem lehetséges GPS nélkül meghatározni, ezért kicsit át kellett alakítanom az algoritmust, hogy a lehető legkevesebb GPS használat mellett is működjön.

A működéséhez több dologra is szükségünk van: egy kezdő GPS koordinátára, fix pontokra az útvonal mentén, valamint a fix pontoknak egymáshoz képesti távolságára és pontokban a felhasználó haladási irányára. Ha ezek megvannak, akkor a pontoknak ki tudjuk számolni a tényleges koordinátáit, így közelíteni tudjuk a felhasználók útvonalát.

A dead reckoning algoritmushoz a fix pontoknak a felhasználó lépéseit választottam. Azért döntöttem emellett, mert ezek olyan fix pontok, amiket a mozgásérzékelő szenzorok segítségével meg tudok határozni, és mindig pontosan leírják a felhasználó aktuális helyét, valamint kiszámítható belőlük a lépések hossza, az egyes pontoknak egymáshoz való távolsága, valamint a haladási irányt is meg tudom határozni az iránytű segítségével. Így az útvonal felépíthető a lépések pontjaiból, ha kiszámoljuk azok GPS koordinátáit.

Az algoritmus működését szemléltető folyamatábra [5.1]:



Ábra 5.1 – Az algoritmus folyamatábrája

A mérés indításához szükségünk van a felhasználó nemére és testmagasságára. Ezek segítségével fogjuk tudni kiszámolni a lépései hosszát [19]. Ezen kívül aktív GPS kapcsolatnak is létre kell jönnie, hogy rögzíteni tudjuk a kezdő GPS koordinátákat. Ezen adatok nélkül a mérés meg sem kezdhető. Amint a rendelkezésünkre állnak ezek az adatok megkezdhetjük a mérést. Itt rögtön három szálon folytathatjuk a rögzítést. Egyrészt szükségünk van a felhasználó GPS pozíciójára a mérés indításától kezdve egy percen keresztül. Ezután a GPS modult ki is kapcsoljuk.

Ezzel párhuzamosan elmentjük az okostelefon gyorsulásmérő és gravitációs szenzorjaiból érkező adatokat mindhárom tengely mentén. A nyers adatból egy kalkulált értéket is elmentünk. Alapvetően a gyorsulásmérő függőleges (Y) tengelyének értékére van szükségünk, de ez nem mindig pontosan függőleges. Ezért a többi tengely és a gravitációs szenzorból származó adatok segítségével úgy módosítjuk az értékét, mintha pontosan függőlegesen állna a készülék. Ehhez az alábbi képletet használtam fel:

$$\varphi = \text{atan}(\text{abs}(\text{grav}_z/\text{grav}_y))$$

$$\text{acc}_{\text{vertical}} = \text{acc}_z * \sin\varphi + \text{acc}_y * \cos\varphi$$

A továbbiakban nem a gyorsulásmérő Y tengelyének nyers értékével fogok dolgozni, hanem ezzel a kiszámolt értékkel, nevezzük vertikális gyorsulásnak.

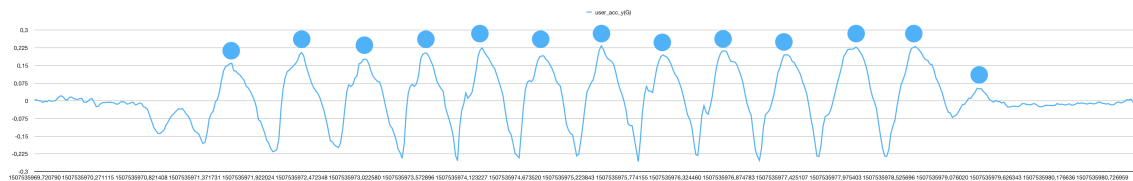
Harmadrészt pedig elmentem a telefon iránytűjének értékét is, amiből a haladási irányra lehet következtetni. Az iránytű értékeinek rögzítése egészen addig tart, amíg a mozgásérzékelő szenzoroké, a GPS-szel ellentétben ezt a modult nem kapcsoljuk le, mivel csak a magnetométert használja, ezért áramfelvétele elhanyagolhatóan alacsony.

Miután a mérés befejeződött következik az adatok kiértékelése. A crowdsensing nagy előnye, hogy ezeket nem egy központi szerveren kell elvégezni, hanem lokálisan, minden felhasználó a saját telefonján végzi el, így rengeteg számítási kapacitás spórolható meg. Sajnos a három rögzítendő adatsor közül egyedül a mozgásérzékelő szenzorokból lehet az adatokat lekérni előre meghatározott időközönként. Én 50 Hz-es mintavételezési időt választottam erre, ami már elég jó felbontást eredményez.

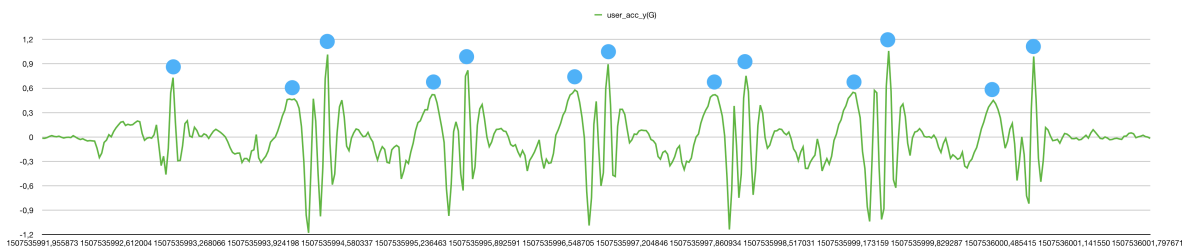
A GPS-ből és az iránytűből érkező adatok mintavételezési frekvenciáját sajnos nem lehet megválasztani, valamint nem is egyenlő időközönként érkezik adat, ezért a feldolgozás előtt idő szerint össze kell rendelni a három adatsort. A mozgásérzékelő szenzorok idősorát vettem alapul, mert előre meghatározott frekvenciájú és periodikus. Ehhez rendeltem hozzá a

GPS-ből és az iránytűből érkező adatokat az időbélyegek alapján. Ha nem volt azonos értékű időbélyeg, akkor mindig a legközelebbit választottam, ha pedig bizonyos ideig nem érkezett adat a GPS modulból vagy az iránytűből, akkor az előző értéket párosítottam a mozgásérzékelő szenzorok idősorának elemeihez. Így az összerendelés végére lett három, egyenlő felbontású és időben összerendelt idősorom, amin el tudom végezni a további számításokat.

Ezek után a következő feladat meghatározni a lépések pontos idejét. Ezt a vertikális gyorsulás idősorának vizsgálatával tudom megtenni. Az iOS API ugyan lehetőséget biztosít arra, hogy az Apple saját lépésszámlálójából lekérjünk adatokat, viszont ez sajnos csak arról ad információt, hogy két időpont között hány darab lépés történt. Azok intenzitását, valamint pontos idejét sajnos nem tudjuk. Ezért mindenképp egy saját megoldást kellett kifejleszteni, ami a gyorsulásmérőt használja.



Ábra 5.2 – 13 lépés kézben tartott telefonnal, saját mérésből készített ábra



Ábra 5.3 – 13 lépés zsebben lévő telefonnal, saját mérésből készített ábra

A grafikonok a vertikális gyorsulás idősorait ábrázolják kézben [Ábra 5.2] és zsebben [Ábra 5.3] tartott telefon esetén. A kékekkel jelölt csúcspontok jelentik az egyes lépéseket, ezeket kell algoritmikus módon felismerni. Látható, hogy a kézben és a zsebben elhelyezett telefon idősorai eltérő mintázatúak. Kézben csak a lépések hullámmozgása befolyásolja a telefon mozgását, minden lépés ugyan olyan mintájú. Ez abból adódik, hogy kezünkkel csillapítjuk a lépéseinknek a telefon mozgására gyakorolt hatását. Míg zsebben elhelyezve a telefont, a lépések mintázatát a csak a lábaink mozgása befolyásolja. Más alakú lesz a minta, ha éppen azzal a lábunkkal lépünk, aminek a zsebében a telefon van és más, ha a másikkal. Ezekből a különbségekből adódóan más threshold és minimális időköz beállításokat kell használni a

telefon két pozíciója között. Tehát nem elég egyszerűen felismerni a lépéseket, először azt kell meghatározni, hogy kézben vagy zsebben történt-e a mintavételezés.

A lépésfelismerés pontosítása érdekében több követelményt is felállítottam az egyes lépésekre vonatkozóan: egyrészt a csúcsoknak egy bizonyos érték (threshold) fölött kell lenniük, hogy lépésnek számítsanak, így kiszűrhető a zaj a rendszerből, valamint egy már megtalált lépéspont után bizonyos időnek el kell telnie, hogy egy következő pozitív csúcsra azt mondhassuk, hogy lépéspont. Ez a sétálás természetéből adódik. Jellemzően egy lépés nem rövidebb 200 ms-nál (5 Hz) [19]. Így két lépéspont között legalább ennyi időnek el kell telnie. A grafikonokat [Ábra 5.2], [Ábra 5.3] megnézve látható ennek az állításnak a fontossága, ugyanis látszik, hogy a zsebben tartott telefon esetében minden második lépés környezetében több tüske is található, ami átlépi a threshold értékét, így időbeli megkötés nélkül azokat is lépésként ismerné fel az algoritmus.

A lépésfelismerő algoritmus visszaad nekünk egy listát, ami a lépéspontokat tartalmazza. Egy lépéspontnak ismerjük a pontos idejét és a vertikális gyorsulását. Előbbire azért van szükség, mert ez alapján fogjuk megkeresni a vonatkozó adatokat a másik két idősorban, az utóbbinak pedig a lépés hosszának kiszámításában lesz szerepe.

Következő lépésben a dead reckoning algoritmus a megtalált lépések idejét alapul véve megkeresi a hozzájuk tartozó GPS koordinátákat, illetve az iránytű értékeit. GPS koordináta természetesen nem tartozik minden lépésponthoz, csak az első 1 percben történetekhez. Az iránytű értékeit pedig úgy párosítjuk az egyes lépéspontokhoz, hogy a lépés pillanata előtt és után rögzített pár mintát átlagoljuk, így az esetleges zajt csökkenthetjük.

A fejlesztés közbeni folyamatos tesztelés során vettem észre az alábbi problémát: kezdetben csak a telefon iránytűjéből származó értékekkel számítottam ki a lépések koordinátáit. Ez azt eredményezte, hogy ugyan a rögzített útvonal formája megegyezett a lesétált útvonallal, viszont valamilyen szögben el volt csúszva. Ezt az az egyszerű dolog okozta, hogy a telefon a felhasználó kezében vagy a zsebében nem pontosan a haladási irányba áll, így az iránytűje mindig egy adott szöggel eltolt értéket ad vissza. Ha meg tudjuk határozni, hogy mi a haladási irány szögének és a telefon orientációs szögének a különbsége, akkor a rögzített orientációs idősort ezzel a különbséggel kompenzálva már fel tudjuk használni arra, hogy kiszámítsuk vele a tényleges koordinátákat. Ezért van arra szükség, hogy a GPS az első egy percben be legyen kapcsolva és ezáltal tudjunk az első pár lépéshez pontos koordinátát kötni, ugyanis két szomszédos lépéspontot összekötő szakasz alapján meghatározható a felhasználó haladási iránya. Ezt összevetve az időben megegyező iránytű

értékkel megkapjuk a két haladási szög különbségét, amivel az összes többi iránytűből származó értéket kalibrálni tudjuk. Ez a különbség annál pontosabb, minél több GPS-ből származó és iránytűből származó értéket tudunk összevetni, és ezek átlagát felhasználni.

A lépéspontok orientációja mellett szükségünk van a lépések hosszára is, hogy tudjuk az egyes pontok egymáshoz való távolságát. A legegyszerűbb módszer az lenne, ha egy fix értéket használnánk minden felhasználónál, minden lépésénél. Ez egy nagyon egyszerű megoldás, viszont nem pontos. Rövid méréseknél nem lenne nagy probléma, de minél több lépést tesz meg a felhasználó egy mérés során, annál nagyobb lesz az eltérés a valós távolsághoz képest.

A szakirodalomban találhatunk rá példákat [19], [20], hogy lehetőségünk van meghatározni a lépések hosszát a felhasználó nemének, testmagasságának és lépései gyakoriságának felhasználásával. Így külön hosszúsággal számolhatunk minden lépés esetén, ami pontosabb eredményt ad hosszú séták esetén is. Erre az alábbi képletet használhatjuk, ahol a k egy konstans érték, ami a felhasználó nemétől függ, nők esetén ez az érték 0.2975 , férfiak esetén pedig 0.3139 . A h a felhasználó magassága méterben, a lépés hosszát szintén méterben kapjuk meg. Az f_s pedig a lépés frekvenciáját, vagyis időbeli hosszát jelenti. Természetesen az első lépésnél ezt nem tudjuk kiszámolni, mivel nincs mihez viszonyítani, ezért ezt utoljára állítjuk be, jó közelítés, ha az idősor többi lépéshosszának átlagával közelítjük.

$$L_{step} = k * h * \sqrt{f_s}$$

Ebben a pontban már ismerjük az egyes lépések hosszát, valamint irányát, így ki tudjuk számítani a koordinátájukat is. Mivel a mérések nem ölelnek fel nagy távolságokat, ezért eltekintünk a földfelszín görbületétől, valamint a tengerszint feletti magasságtól és a domborzattól is. A számítást egyszerű síkban tesszük meg.

Egy ismert koordinátából, irányból és távolságból a következő egyenletek megoldásával kiszámolhatjuk a következő pont koordinátáit:

$$\varphi_2 = \text{asin}(\sin\varphi_1 * \cos\delta + \cos\varphi_1 * \sin\delta * \cos\theta)$$

$$\lambda_2 = \lambda_1 + \text{atan2}(\sin\theta * \sin\delta * \cos\varphi_1, \cos\delta - \sin\varphi_1 * \sin\varphi_2)$$

Ahol φ szélesség, λ a hosszúság, θ az Északtól való eltérés foka óramutató járása szerint, és δ a szögtávolság, ami d/R vagyis $\text{lépéshossz}/\text{Föld sugara}$ [21].

Ezután már csak egyszerűen elhelyezzük a kapott koordinátákat a térképen és sorrendben összekötjük őket egy szakasszal és meg is kaptuk a séta útvonalát.

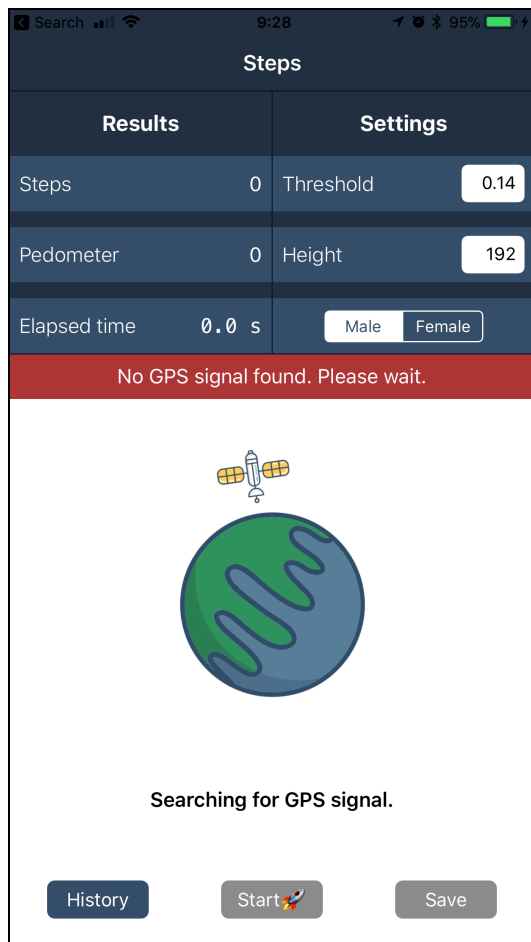
Ez a módszer önmagában alkalmas arra, hogy viszonylag pontosan rögzítse a felhasználó útvonalát. Hátránya, mint minden dead reckoning eljárásnak az, hogy minél hosszabb a mérés a távolabbi pontok pozíciója annál pontatlanabb, mert a hibák összeadódnak. Továbbá, ha a telefon pozíciója megváltozik, újra kell kalibrálni, mivel az előző szögeltérés már nem lesz érvényes. Ezért tényleges használatban több esemény bekövetkeztekor is szükséges újraindítani a mérést. Értelemszerűen, ha a telefon pozíciója megváltozik: kézből átkerül zsebbe, vagy fordítva. Mivel ilyenkor megváltozik a telefon iránytű szögének és a tényleges haladási iránynak a különbsége, ezért szükséges az újrakalibrálás, de ha már be kell kapcsolni a GPS-t, akkor a kezdőpontot is frissíthetjük, így a dead reckoning algoritmus friss értékekkel újraindítható, ami pontosabb értékeket eredményez. Mivel a telefonon futó szoftver tartalmazza a tömegrendezvény térképét és cellarendszerét (ezt belépéskor tölti le a felhasználó kötelezően a központból), ezért tisztában van azzal, hogy a felhasználó épp melyik cellában tartózkodik [7]. Ha úgy érzékeli a szoftver, hogy cellaváltás történt, minden esetben beküldi az útvonalinformációkat a központba (ehhez a cellás vagy/és WLAN hálózatot használja, ami a tömegrendezvényen rendelkezésre áll) és új mérést indít. Ezzel is pontosítható a helyzet megadása, valamint eliminálható a felhalmozott hiba. Ezen túl időhöz is köthetjük a mérések újraindítását. A cellák méretétől függően az időkorlát változhat, de 10-15 percenként mindenképp érdemes megtenni.

6. Dead reckoning mobilalkalmazás

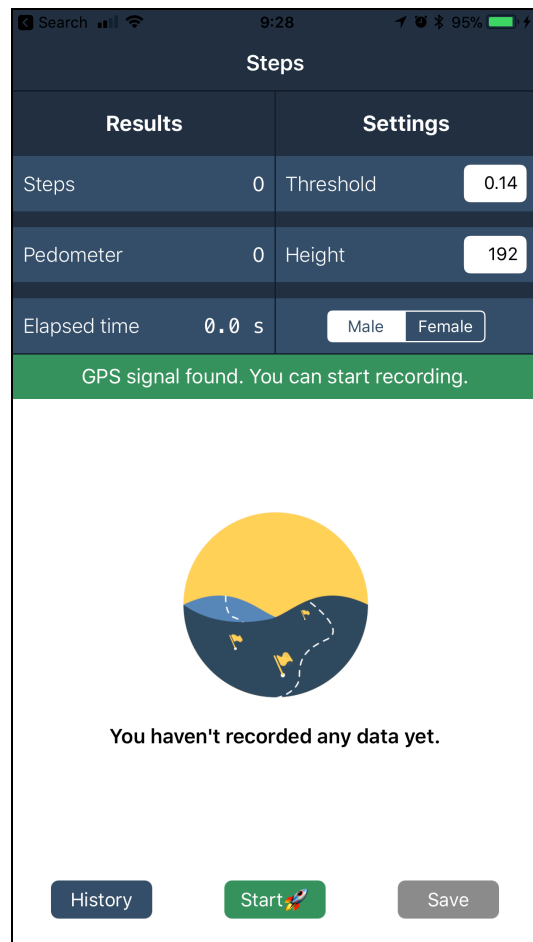
Az előző fejezetben ismertetett algoritmust mindenképp ki szerettem volna próbálni valós körülmények között is. Ez egyrészt segít a validációban, de ami fontosabb a továbbfejlesztést és a finomítást is elősegíti. Vannak már előre megírt alkalmazások, amik elérhetőek iPhone-ra, amik segítségével ki lehet menteni a szenzorokból érkező információkat. Ezekhez fel lehet dolgozni később, de egyszerűbbnek és valóságosabbnak találtam azt, hogy egyből egy iOS-en futó alkalmazást készítek. Ezzel egyrészt lehetőségem nyílik iPhone specifikusan megírni az alkalmazást, és felfedezni az esetleges korlátokat, másrészt ki tudom próbálni a valódi felhasználást: a telefonnal rögzítem az adatsorokat, és azokat szintén a telefonon dolgozom fel. Ezért döntöttem végül úgy, hogy elkészítek egy iOS alkalmazást, amivel ki tudom próbálni az elképzelésemet és esetleg tovább is fejleszteni, pontosítani azt.

Az alkalmazás legfontosabb funkciója a dead reckoning algoritmus megvalósítása, a szabad paraméterek beállítási lehetősége mellett. Ezenkívül elmenthetjük vele a méréseinket, amiket visszanezhetünk, illetve az elmentett adatsorra újra le tudjuk futtatni az algoritmust új paraméterekkel.

Az applikáció megnyitása után a mérést vezérlő képernyőre kerülünk. Itt egyrészt beállíthatjuk a mérés paramétereit: threshold, testmagasság, nem, valamint elindíthatjuk a mérést és elmenthetünk egy lezárt mérést. Ezen kívül át tudunk navigálni a korábbi méréseket tartalmazó oldalra is. Fontos, hogy amíg nincs megfelelően pontos GPS jel, addig az alkalmazás nem engedi a mérést elindítani. Erre megfelelően figyelmezteti is a felhasználót. Amint elérhető megfelelő jel, a mérés elkezdhető, amire szintén jön figyelmeztetés.



Ábra 6.1 - GPS jel keresése.



Ábra 6.2 – GPS jel elérhető.

Miután elindítottuk a mérést, az alkalmazás jelzi, hogy folyamatban van egy rögzítés. Közben a kezelőfelület inaktív, csupán leállítani tudjuk a mérést, megtekinteni az előző méréseket vagy változtatni a paramétereken nem lehetséges. A leállítás után automatikusan lefut a dead reckoning algoritmus, ami kiértékeli a rögzített idősorokat, és visszaad egy listát a megtett lépésekről. Minden lépésnek megkapjuk a sorszámát, hosszát, a mérés kezdete óta számított idejét, vertikális gyorsulását, valamint a haladási irányt. Ennek a listának segítségével leellenőrizhetjük, hogy a valósághoz képest milyen eredményt adott a dead reckoning algoritmus. Ezen kívül láthatjuk a mérés időtartamát, valamint, azt is, hogy az Apple által implementált lépésszámláló (Pedometer) mennyi lépést rögzített a mérésünk ideje alatt, összehasonlítási alapként. A tesztek során azt tapasztaltam, hogy dead reckoning eljárás pontosabb eredményt ad, mint az Apple-féle megoldás. A Pedometer jellemzően hosszú séták esetén működik jól, rövid, 100 alatti lépésszám esetén sokat téved, míg a dead reckoning algoritmus lépésszámlálóját a mérés hossza nem befolyásolja.

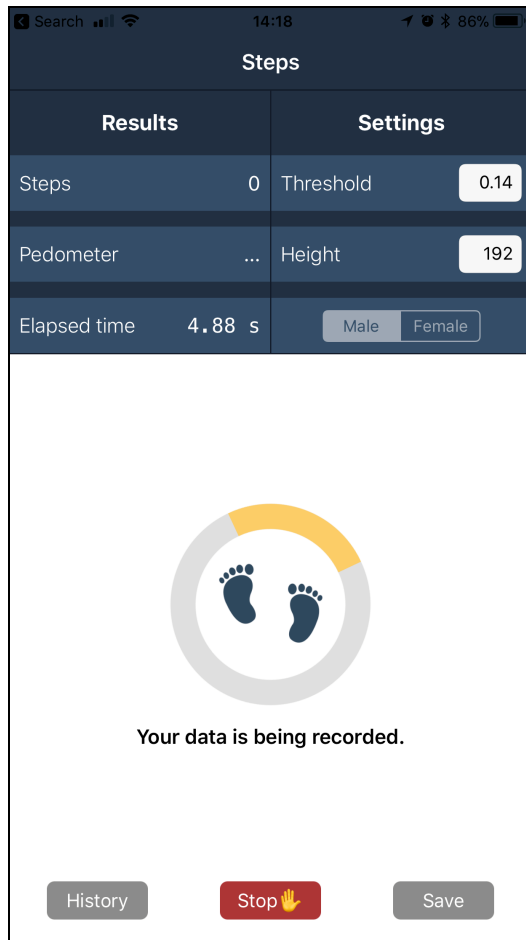
Az elvégzett tesztek alapján az alábbi értékeket kaptam dead reckoning alkalmazásom lépésszámlálója és az Apple lépésszámlálója alapján:

| Dead reckoning lépésszámláló | iOS lépésszámláló | Ténylegesen megtett lépés |
|---|--------------------------|----------------------------------|
| 31 | 22 | 30 |
| 32 | 27 | 30 |
| 190 | 182 | 190 |
| 184 | 176 | 185 |

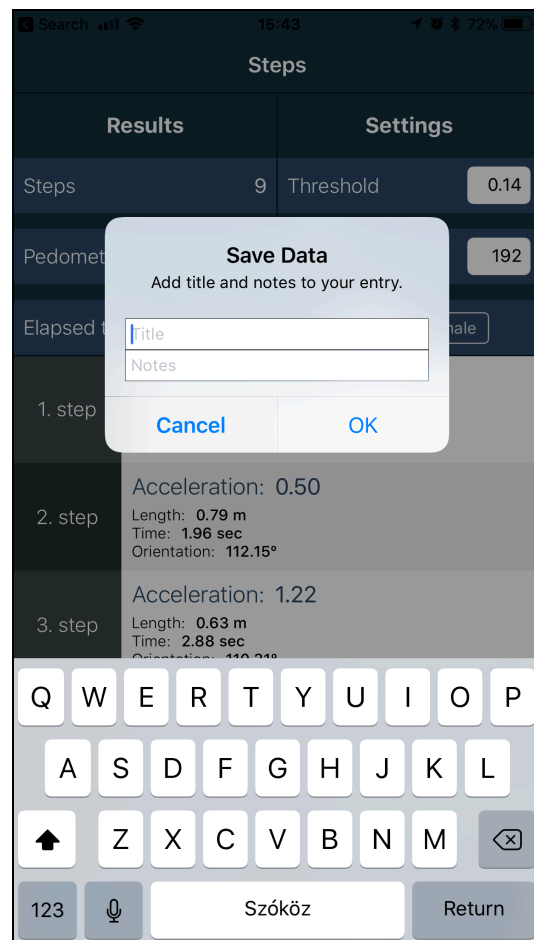
Táblázat 6.1 - Lépésszámlálók összehasonlítása

Látszik, hogy minél hosszabb időszakban kérjük le az iOS lépésszámlálójából a megtett lépések darabszámát, annál pontosabb eredményt ad. Ez valószínűleg abból következik, hogy a felismert lépések idejét átlagolja, valamilyen módszerrel közelíti az eltelt időt. Így hosszú távon jó közelítéssel képes meghatározni a lépések számát, viszont rövidtávon pontatlan. Valamint az iOS Pedométeréből az egyes lépések pontos idejét sem tudjuk kinyerni, ezért sem tudjuk használni a dead reckoning formuláihoz.

A rögzített adatokat lehetőségünk van elmenteni. Ilyenkor bekerül egy perzisztensen elmentett listába, minden információval együtt. Így később lehetőségünk van visszatölteni, újra kiértékelni és térképen megjeleníteni az eredményt. A mentés során elnevezhetjük mérésünket, valamint rövid jegyzetet is fűzhetünk hozzá.



Ábra 6.3 - Folyamatban lévő mérés.



Ábra 6.4 – A mentés képernyője.

A „History” gombbal tudunk átnavigálni az elmentett mérések listáját tartalmazó oldalra. A listában láthatjuk a mérések általunk megadott neveit, illetve a mentés dátumát. Utóbbit az alkalmazás generálja, megváltoztatni nem lehetséges, csak azonosítás céljából van jelen, mivel szerepelhet két azonos nevű mérés is a listában.

Az egyes listaelemekre bökkve megnyithatjuk azokat vagy ujjunk balra húzásával további opciókat hívhatunk elő: átnevezhetjük, elküldhetjük, illetve törölhetjük a bejegyzést. Ha a „Share”, vagyis a megosztás gombot választjuk, akkor az alkalmazás generál egy .csv fájlt az adatokból, amit egy tetszőleges email címre továbbíthatunk. Így lehetőségünk van az adatokat kimenteni az appból és azokat később számítógépen feldolgozni egy táblázatkezelő programmal.

Az átnevezés (Rename) és a törlés (Delete) műveletek nem visszavonhatók. Természetesen a lista perzisztens, tehát az alkalmazás bezárása és RAM-ból eltávolítása után is megmaradnak a mentett elemek.



Ábra 6.5 - Elmentett mérések listája.



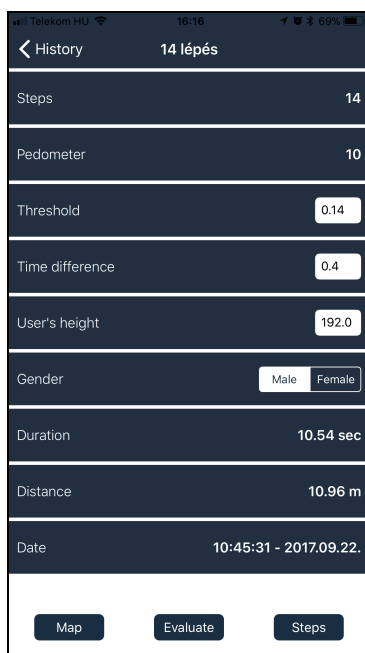
Ábra 6.6 – Opciókat tartalmazó contextusmenü.

Egy elem megnyitása után egy új képernyőre kerülünk. Ilyenkor az alkalmazás visszatölti a mérés összes adatát, amit újra ki tudunk értékelni. Ezzel a funkcióval az algoritmus ellenőrzését és a paraméterek pontosítását szeretném elősegíteni. Az újra kiértékeléssel több különböző paraméterrel is lefuttathatjuk az algoritmust az adatsoron, így más és más eredményt kapva, majd a legjobbat választva. Ennek segítségével állíthatjuk be pl. a megfelelő threshold értékét is, de az egyes lépések között minimális időkülönbséget is finomhangolhatjuk.

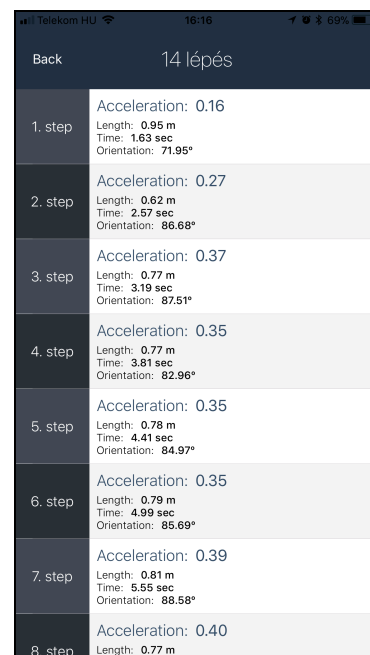
A mérés aktuális eredményét listaszerűen láthatjuk. Az alábbi adatokról kapunk információt:

- Lépések darabszáma (Steps)
- Lépések száma az Apple pedométeréből (Pedometer)
- Threshold
- Két lépés közötti minimális időkülönbség (Time difference)
- Felhasználó magassága (User's height)
- Felhasználó neme (Gender)
- Mérés időtartama (Duration)
- Megtett út hossza (Distance)
- Mentés időpontja (Date)

A lista alatt található még három gomb: Map, Evaluate és Steps. A „Map” gombbal a térképen tekinthetjük meg a rögzített lépések pozícióját, az „Evaluate” gombbal újra lefuttathatjuk a dead reckoning algoritmust a beállított paraméterekkel, a „Steps” gombbal pedig az eredményül kapott lépések listáját tekinthetjük meg hasonlóan, mint az alkalmazás első oldalán, a mérés befejezése után.

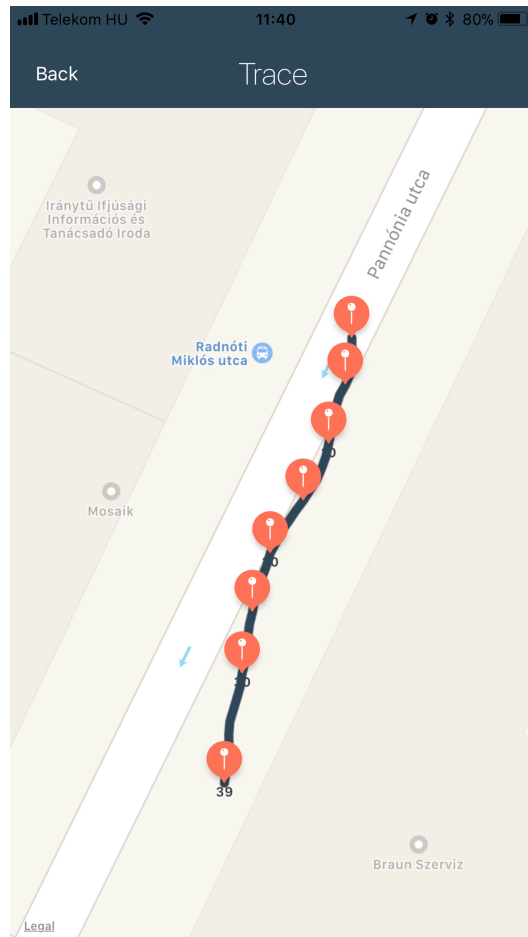


Ábra 6.7 – Kiértékelő képernyő.



Ábra 6.8 – Lépések listája.

A „Map” gombra kattintva térképen is megnézhetjük a megtett útvonalat. Itt a lépések tényleges pozícióit láthatjuk, az elsőt és utolsót, valamint minden 5. lépés ki van emelve az irány azonosítása érdekében. Ennek segítségével vizuálisan is leellenőrizhetjük, hogy helyes útvonalat számolt-e ki a dead reckoning algoritmus. A térkép az Apple által fejlesztett Maps, amit az iOS API-val natívan elérhetünk és szabadon felhasználhatunk az alkalmazásainkon belül. Természetesen a térkép interaktív, nagyíthatjuk, kicsinyíthetjük, forgathatjuk a nézetet.



Ábra 6.9 - Térképes nézet

A mérések újraszámolását nem tudjuk elmenteni és visszatölteni később. Arra sincs lehetőség, hogy kívülről betöltsünk adatokat az alkalmazásba és azokat dolgozzuk fel.

7. Teszt és validáció

Az elkészített alkalmazás az algoritmus tesztelésében is nagy szerepet játszik, az egyik fő motivációja az applikáció elkészítésének is ez volt. Segítségével valós használatot szimulálva tudjuk validálni a módszer helyességét, egyszerűen, azonnali módon. Használata megkímél attól, hogy az adatokat átjuttassuk a számítógépre, majd azokat ott dolgozzuk fel.

Két tesztet vettem figyelembe, ami a leggyakoribb esetek lehetnek a valós felhasználás során:

- a felhasználó a kezében tartott telefontal sétál
- a felhasználó az első zsebében tartott telefontal sétál

Mindkét tesztessel több útvonalat is bejártam, majd ezeket hasonlítottam össze a valóságban bejárt útvonallal és érkezési ponttal. Vizsgáltam továbbá a kiszámolt út hosszát is a valóságban megtetthez képest.

| | Pozíció | Lépésszám | Valós lépésszám | Eltérés | Megtett út | Valós úthossz | Eltérés |
|----|---------|-----------|-----------------|-------------|------------|---------------|-----------------|
| 1. | kéz | 91 | 90 | +1 (1,1%) | 73,57 m | 76,38 m | -2,81 m (3,4%) |
| 2. | zseb | 100 | 90 | +10 (11,1%) | 79,82 m | 76,38 m | +3,44 m (4,5 %) |
| 3. | kéz | 30 | 30 | 0 (0%) | 24,42 m | 23,77 m | -0,65 m (2,7 %) |
| 4. | zseb | 32 | 30 | +2 (6,6 %) | 23,31 m | 23,77 m | +0,46 m (1,9 %) |
| 5. | kéz | 190 | 190 | 0 (0 %) | 152,58 m | 154,49 m | -1,91 m (1,3 %) |
| 6. | zseb | 184 | 185 | +1 (0,6 %) | 145,75 m | 154,49 m | -8,74 m (5,7 %) |

Táblázat 7.1 - Elvégzett tesztek eredményei

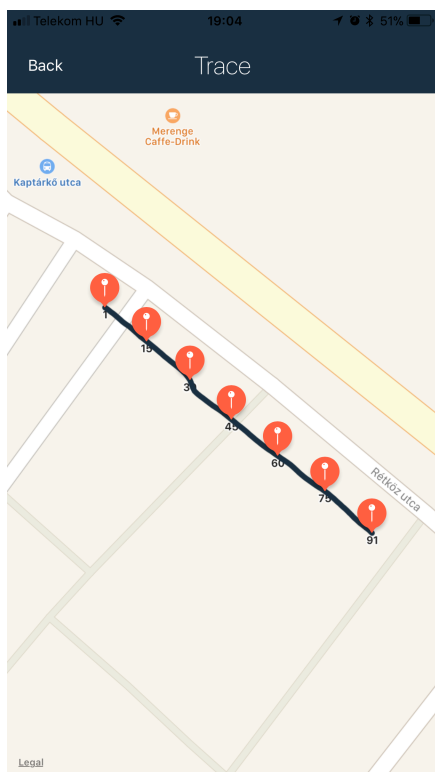
| | | |
|---------------------------|---------|---------------|
| Átlagos lépésszám eltérés | kézben | 0,4 % |
| | zsebben | 6,1% |
| Átlagos úthossz eltérés | kézben | 2,47 % |
| | zsebben | 4,03% |

Táblázat 7.2 - Tesztek átlagos értékei

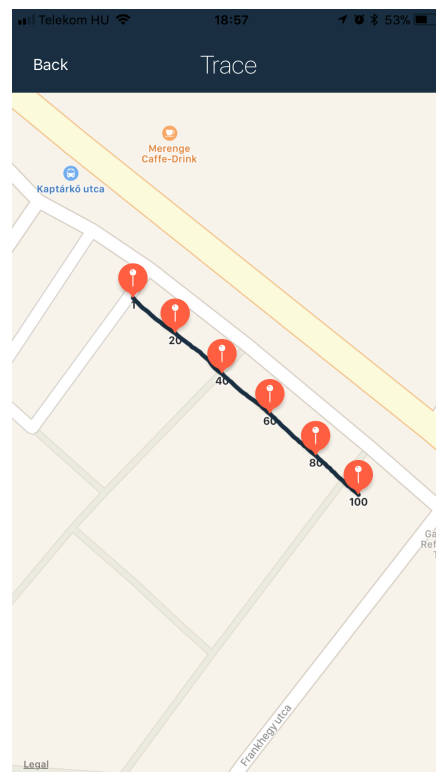
Az elvégzett tesztek eredményiből [Táblázat 7.2] látszik, hogy kézben pontosabb eredményeket kapunk, mint zsebben. Ez abból adódik, hogy a kézben tartott telefon mozgásához sokkal kevesebb zaj adódik hozzá a séta során, tisztábban felismerhetők a lépések. Látható, hogy a lépések felismerésében alig téved az algoritmus és a megtett útvonal pontossága is majdnem tökéletes. Ha zsebben tartottuk a telefont a mérés során a hibaarány nagyobb, viszont még így is elég jónak mondható. Észrevehető az is, hogy a zsebben tartott telefon esetén a lépésszám eltérés nagyobb, mint az útvonal eltérése. Tehát hiába nem ismerjük fel nagyon pontosan a lépéseket, attól még a közelített útvonal iránya és hossza megfelelő lesz.

A térképek [Kép 7.1 – Kép 7.6] alapján pedig látszik, hogy a tényleges útvonaltól kis mértékben tér el a kiszámolt útvonal. Attól függően, hogy milyen hosszú időközönként indul újra a mérés jó értékeket biztosíthatunk a felügyeleti rendszer számára.

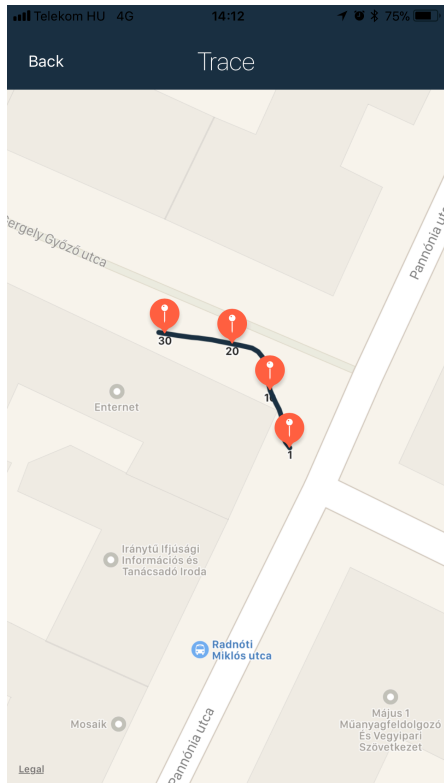
Elvégzett mérések:



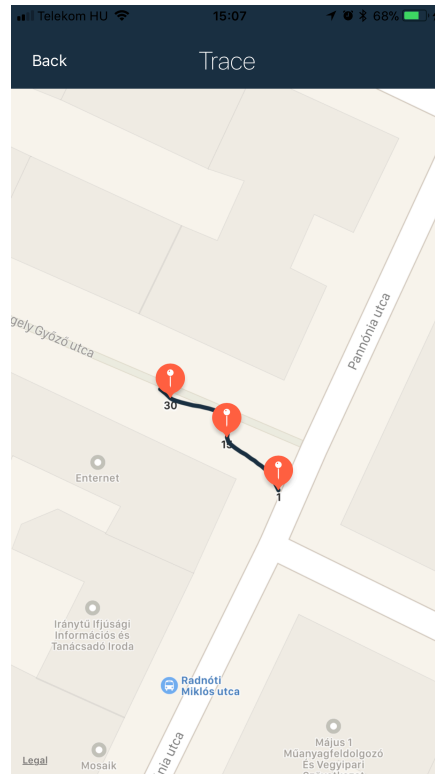
Ábra 7.1 - 1. mérés képernyőfotója (park)



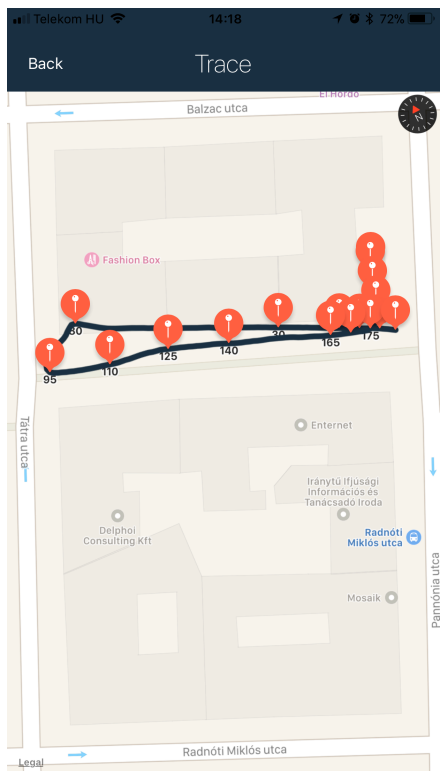
Ábra 7.2 - 2. mérés képernyőfotója (park)



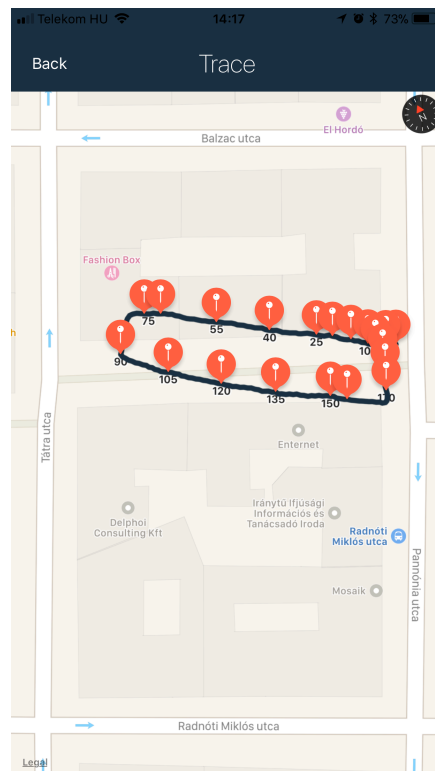
Ábra 7.3 - 3. mérés képernyőfotója (utca)



Ábra 7.4 - 4. mérés képernyőfotója (utca)



Ábra 7.5 - 5. mérés képernyőfotója (utca)



Ábra 7.6 - 6. mérés képernyőfotója (utca)

Az elvégzett mérések validálását [Táblázat 7.1] a Google Maps segítségével végeztem el. Az útvonal elejének és végének koordinátáit rögzítettem, majd a távolságmérés funkció segítségével meghatároztam a tényleges távolságot. A valóságban megtett lépésszámot pedig mérés közbeni számolással határoztam meg. Az útvonalakat mindig úgy választottam ki, hogy azok kövessék a járdát vagy az útestet, így később vizuálisan is leellenőrizhető annak alakja.

Fontos megjegyezni, hogy a módszer pontossága rendkívüli mértékben függ a GPS kapcsolat pontosságától. Ez határozza meg ugyanis a kiindulási pozíciót, valamint a kalibrálást is. Tehát ha nem megfelelő a kapcsolat a GPS műholdakkal a mérés elején, hiába jó a dead reckoning eljárás a pontatlan változók miatt a valóságtól nagyban eltérő is lehet az eredmény. Ezért garantálni a pontosságot csak kültéren és nyílt területen lehetséges. Szerencsére a tömegrendezvényeket általában kültéren, olyan helyeken rendezik, ahol nincsenek magas épületek és szűk utcák, így a GPS kapcsolat megfelelő pontosságú lehet. A teszteket igyekeztem parkban és utcán is elvégezni, hogy többféle környezetben is leellenőrizsem az alkalmazás pontosságát.

8. Konklúzió

Manapság a tömegrendezvények és a fesztiválok rengeteg embert vonzanak, ugyanakkor rengeteg veszélyt is hordoznak magukban. A nagy tömegben egy kisebb pánik is hamar tömegkatasztrófává nőheti ki magát, ezért nagyon fontos, hogy a felügyelő szervek időben értesüljenek a veszélyes helyzetek kialakulásának lehetőségéről, így meg tudják akadályozni azt. Azonban nem lehet minden esetre felkészülni, évről évre történnek halálos tömegkatasztrófák, ezért egy jól működő tömegfelügyeleti rendszer kiépítése igen sürgető, aminek segítségével a szervezők pontosabb információkat kaphatnak a tömegdinamikai állapotokról.

A technológiai fejlődésnek hála sok lehetőségünk van ezen rendszerek megvalósítására, többféle technológiai megoldás közül is válogathatunk. Fontos azonban, hogy mindig az adott feladatnak megfelelő megoldást részesítsünk előnyben, figyelembe véve a rendezvény jellegét és az infrastrukturális költségeket is. A résztvevők nagy része már rendelkezik olyan eszközzel, amely segítségével információt biztosíthat a szervezők számára az aktuális pozíciójáról, amiből a tömeg egészének mozgására lehet következtetni. Ez egy költséghatékony megoldás, mivel a számítási kapacitás eloszlik a felhasználók okostelefonjai között, így mind a szerverpark (nem kell plusz gépeket fenntartani arra, hogy futtassák a dead reckoning eljárást is az adatokon), mind az infrastruktúra kiépítésének költségei alacsonyan tarthatók. Az egyik legfontosabb feladat egy ilyen rendszer kiépítése során az, hogy a felhasználók pozícióját minél pontosabban és minél energiatakarékosabb módon határozzuk meg.

Mivel a helymeghatározásra több módszer is létezik, első lépésben ezek közül kiválasztottam a feladatra a legmegfelelőbbet, ami a dead reckoning eljárás, dolgozatomban bemutatva a több módszerrel szembeni előnyeit. Ezután tanulmányoztam ezt a módszert és fejlesztettem egy olyan algoritmust, amit fel lehet használni egy közösségi érzékelésen alapuló tömegfelügyeleti rendszer információval való ellátására. Azért volt szükség saját megoldás fejlesztésére, mivel a szakirodalomban nem található jól dokumentált megoldás, ráadásul ezek a megoldások túl általánosak is. Nekünk arra volt szükségünk, hogy az algoritmus minél jobban megfeleljen a tömegfelügyeleti rendszer igényeinek, működjön lépésfelismerés alapján, valamint az okostelefonok szenzorjaiból származó értékeket használja fel. Annak érdekében, hogy az elméleti algoritmust élesben is ki lehessen próbálni

és tesztek végeztetése a működéséről készítettem egy iOS alkalmazást, ami segítséget nyújt ebben. Segítségével egyszerűen rögzíthetünk útvonalakat, ezeket elmenthetjük, majd a paraméterek finomhangolása után újra futtathatjuk a méréseken a dead reckoning algoritmust. Valamint a probléma iOS specifikus részeire is rávilágított az alkalmazás fejlesztése, például a mozgásérzékelő szenzorok működése és adatainak elérésének módjára, GPS és iránytű periodikusságának hiányára, az adatok összerendelésére, a lépések térképen való megjelenítésére stb.

Azonban az elkészített applikáció önmagában még nem egy kész kereskedelmi termék, csupán egy teszteszköz. Magát az algoritmust ugyan át lehet ültetni egy kiadásra szánt verzióba, de az egyes use-case-ek, az adatok beküldése a szerverre, a mérés indításának és leállításának triggerei valamint a telefon pozíciójának (kéz vagy zseb) meghatározása még megoldandó feladat, amelyre a dolgozatomban nem tértem ki.

Végül az alkalmazás tesztelésével igazoltam, hogy képes a rendszer elég pontos út és helymeghatározásra, amennyiben a megfelelő bemenetek rendelkezésre állnak. Az elvégzett tesztek eredményei [Táblázat 7.1] [Táblázat 7.2] alapján látszik, hogy az algoritmus megfelelően működik és jól használható a felhasználók nyomon követésére. A tömegfelügyeleti rendszert ellátva az alkalmazásból származó adatokkal, útvonalakkal a szervezők megfelelő képet kaphatnak a vendégek aktuális pozíciójáról, aminek segítségével könnyebben és pontosabban tudnak reagálni a kialakulóban lévő vészhelyzetekre.

Irodalomjegyzék

- [1] <http://www.billboard.com/articles/business/7768135/love-parade-festival-tragedy-german-court-orders-trial> (2017. 10. 25.)
- [2] <https://www.theguardian.com/world/2015/sep/24/timeline-of-tragedies-in-mecca-during-hajj> (2017. 10. 25.)
- [3] Nelson, Donna. Proximity Information Resources for Special Events. No. SHRP 2 Reliability IDEA Project L-15B. 2012.
- [4] <http://www.securitynewsdesk.com/how-many-cctv-cameras-in-the-uk/> (2017. 10. 10.)
- [5] Wirz, Martin, et al. "Using mobile technology and a participatory sensing approach for crowd monitoring and management during large-scale mass gatherings." Co-evolution of Intelligent Socio-technical Systems. Springer Berlin Heidelberg, 2013. 61-77.
- [6] Helbing, Dirk, Anders Johansson, and Habib Zein Al-Abideen. "Dynamics of crowd disasters: An empirical study." Physical review E75.4 (2007): 046109.
- [7] Nagy, Attila Mátyás, and Vilmos Simon. "Integrated mass surveillance system for large scale events." Smart Cities Conference (ISC2), 2016 IEEE International. IEEE, 2016.
- [8] <https://www.space.com/16677-charles-lindbergh.html> (2017. 10. 27.)
- [9] Schroer, R. "Navigation and landing [A century of powered flight 1903-2003]." IEEE Aerospace and Electronic Systems Magazine18.7 (2003): 27-36.
- [10] https://support.apple.com/kb/SP685?locale=en_US (2017. 10. 27.)
- [11] Langfelder, Giacomo, et al. "MEMS motion sensors based on the variations of the fringe capacitances." IEEE Sensors Journal 11.4 (2011): 1069-1077.
- [12] <https://www.youtube.com/watch?v=i2U49usFo10> (2017. 10. 27.)
- [13] <http://basicgyro.blogspot.hu/2013/11/basics-of-gyroscope.html> (2017. 10. 27.)
- [14] <https://www.youtube.com/watch?v=eqZgxR6eRjo> (2017. 10. 27.)

- [15] Yosi, S., N. J. Agung, and S. Unang. "Tilt and heading measurement using sensor fusion from inertial measurement unit." Control, Electronics, Renewable Energy and Communications (ICCEREC), 2015 International Conference on. IEEE, 2015.
- [16] <http://developerblog.myo.com/gui-without-g-going-beyond-screen-myotm-armsband/> (2017. 10. 27.)
- [17] <https://developer.apple.com/documentation/coremotion> (2017. 10. 27.)
- [18] https://developer.apple.com/documentation/corelocation/getting_heading_and_course_information (2017. 10. 27.)
- [19] Tian, Qinglin, et al. "An enhanced pedestrian dead reckoning approach for pedestrian tracking using smartphones." Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on. IEEE, 2015.
- [20] Bertram, John EA, and Andy Ruina. "Multiple walking speed–frequency relations are predicted by constrained optimization." Journal of theoretical Biology 209.4 (2001): 445-453.
- [21] <http://www.movable-type.co.uk/scripts/latlong.html> (2017. 10. 27.)

Táblázatjegyzék

[Táblázat 6.1] – Lépésszámlálók összehasonlítása

[Táblázat 7.1] – Elvégzett tesztek eredményei

[Táblázat 7.2] – Tesztek átlagos értékei

Ábrajegyzék

[Ábra 3.2] – Következő pozíció kiszámítása az irány, a sebesség és az idő ismeretében

[Ábra 4.1] – Egyszerű gyorsulásmérő

[Ábra 4.2] – A gyorsulásmérő kondenzátorai

[Ábra 4.3] – Modern gyorsulásmérő

[Ábra 4.4] – Klasszikus giroszkóp

[Ábra 4.5] – MEMS giroszkóp

[Ábra 4.6] – Magnetométer működése a Hall-effektust kihasználva

[Ábra 4.7] – A pitch, roll és yaw tengelyek értelmezése

[Ábra 5.1] – Az algoritmus folyamatábrája

[Ábra 5.2] – 13 lépés kézben tartott telefonnal, saját mérésből készített ábra

[Ábra 5.3] – 13 lépés zsebben tartott telefonnal, saját mérésből készített ábra

[Ábra 6.1] – GPS jel keresése

[Ábra 6.2] - GPS jel elérhető

[Ábra 6.3] – Folyamatban lévő mérés

[Ábra 6.4] – A mentés képernyője

[Ábra 6.5] – Elmentett mérések listája

[Ábra 6.6] – Opciókat tartalmazó kontextusmenü

[Ábra 6.7] – Kiértékelő képernyő

[Ábra 6.8] – Lépések listája

[Ábra 6.9] – Térképes nézet

[Ábra 7.1] – 1. mérés képernyőfotója (park)

[Ábra 7.2] – 2. mérés képernyőfotója (park)

[Ábra 7.3] – 3. mérés képernyőfotója (utca)

[Ábra 7.4] – 4. mérés képernyőfotója (utca)

[Ábra 7.5] – 5. mérés képernyőfotója (utca)

[Ábra 7.6] – 6. mérés képernyőfotója (utca)