

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Denis Avguštin

**Analiza tresljajev s senzorji v
mobilnih napravah**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana 2013

To diplomsko delo je ponujeno pod licenco Creative Commons Priznanje avtorstva 3.0 Nedoločena ali (po želji) novejšo različico. To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela, kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, dajejo v najem, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani <http://creativecommons.si/> ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Android aplikacija je spisana v razvijalskem okolju Eclipse in prevedena s pomočjo zbirke razvojnih orodij Android SDK.

Grafi so narisani s pomočjo programa Gnuplot.



Št. naloge: 00393 / 2013
Datum: 3.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DENIS AVGUŠTIN**

Naslov: **ANALIZA TRESLJAJEV S SENZORJI V MOBILNIH NAPRAVAH**
ANALYSIS OF VIBRATION BY USING SENSORS IN MOBILE DEVICES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Senzorji v mobilnih napravah omogočajo zaznavanje različnih veličin in zato postajajo vse bolj zanimivi za razvoj mobilnih aplikacij. Posebej aktualna sta pospeškomer in žiroskop: prvi omogoča prepoznavanja gibanja, drugi pa naklona. Proučite možnosti uporabe pospeškometra in žiroskopa v operacijskem sistemu Android. Na podlagi tega analizirajte možnosti kombinirane uporabe teh dveh senzorjev pri zaznavanju tresljajev. Predstavite različne načine zaznave tresljajev in težave, do katerih lahko pride pri tovrstni zaznavi ter pristope, kako se jih znebiti. Analizo usmerite tudi v prepoznavanje trenutnega gibanja mobilne naprave in njenega uporabnika.

Mentor:

doc. dr. Rok Rupnik

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Denis Avguštin, z vpisno številko **63060013**, sem avtor diplomskega dela z naslovom:

Analiza tresljajev s senzorji v mobilnih napravah

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 16. oktobra 2013

Podpis avtorja:

Zahvaljujem se vsem, ki so kakorkoli pripomogli pri izvedbi diplomskega dela.

Diplomo posvečam vsem, ki jih zanima.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Pregled uporabljenih tehnologij in principov	3
2.1	Android	3
2.2	Android SDK	4
2.3	Eclipse	5
2.4	Gnuplot	6
2.5	Naprave, senzorji in principi zaznave	7
3	Matematična in fizikalna pomoč pri analizi	11
3.1	Newtonov prvi zakon, oziroma zakon o vztrajnosti	11
3.2	Absolutni pospešek	12
4	Zaznava gibanja naprave	13
4.1	Priprava	13
4.2	Zaznava tresljajev	14
4.3	Prepoznavna gibanja na osnovi zaznanih tresljajev	20
5	Sklepne ugotovitve	23
6	Slike	25
7	Literatura in viri	27

KAZALO

Slike

4.1	Graf absolutnega pospeška od mirovanja mobilnega telefona na mizi do dviga z mize.	16
4.2	Absolutni pospešek izmerjen z mobilno napravo med hojo z mobilnih telefonom v žepu.	16
4.3	3D graf poti na osnovi zaznanega pospeška v prostoru.	17
4.4	Absolutni pospešek na mobilno napravo med padcem	18
4.5	3D graf poti na osnovi pospeškov na napravo med padcem.	19
4.6	3D graf na osnovi pospeškov na napravo med padcem (obrnjen pogled).	20
6.1	Preprost graf narisani v orodju Gnuplot (vir: gnuplot.sourceforge.net)	25
6.2	Trodimenzionalni graf narisani v orodju Gnuplot (vir: gnuplot.sourceforge.net)	26
6.3	Koordinatni sistem senzorjev gibanja na Android napravah (vir: http://developer.android.com/guide/topics/sensors/sensors_overview.html)	26

Povzetek

Diplomsko delo se ukvarja z analizo tresljajev, zaznanih s pomočjo pospeškometra v mobilnih napravah in prepoznavo gibanja naprave in uporabnika na osnovi zaznanih tresljajev. Pospeškometri so naprave, oziroma senzorji za merjenje pospeška, ki so prisotne v širokem naboru naprav, tudi mobilnih napravah, kot so mobilni telefoni in tablični računalniki. Pospeškometer se v mobilnih napravah primarno uporablja pri nadzoru uporabniškega vmesnika, a ga je mogoče uporabiti tudi za kaj več. V nekaterih primerih pa sam pospeškometer ni dovolj, tako je v diplomskem delu predstavljeno tudi kombiniranje pospeškometra z žiroskopom, napravo za merjenje naklona, oziroma vrtenja.

Delo predstavlja analizo podatkov, zbranih s pomočjo aplikacije za operacijski sistem Android OS, ki odčitava stanje pospeškometra in žiroskopa. Iz zbranih podatkov pa nakaže slabosti posameznega senzorja in kako se teh slabosti znebimo s kombiniranjem obeh senzorjev.

V diplomskem delu so predstavljeni načini zaznave tresljajev na operacijskem sistemu Android OS in težave, do katerih lahko pride pri tovrstni zaznavi in kako se jih znebiti.

Zaključni rezultati diplomskega dela bi naj pomagali prepoznati trenutno gibanje mobilne naprave in njenega uporabnika.

Ključne besede:

pospeškometer, žiroskop, tresljaj, gibanje, mobilna naprava, Android

Abstract

This thesis presents an analysis of vibrations detected by accelerometers in mobile devices and identification of device and user movement based on detected vibrations. Accelerometers are devices or sensors for acceleration measurements, which are present in a wide range of devices, including mobile devices such as mobile phones and tablet computers. Accelerometer in mobile devices is primarily used for user interface control, but it can also be used for more than that. In some cases, the accelerometer alone is not enough, so the thesis presents a combination of accelerometer and gyroscope, a device which measures device inclination or rotation.

Thesis presents an analysis of data collected using an application for Android operating system, which collects readings from accelerometer and gyroscope. It shows weaknesses of each sensor, based on collected sensor readings, and how to avoid these weaknesses by combining both sensors.

This thesis presents methods of detection of vibrations on Android operating system and problems that may occur in this type of detection and how to get rid of them.

The final results of the thesis should help identifying current movement of mobile device and its user.

Key words:

accelerometer, gyroscope, vibration, movement, mobile device, Android

Poglavje 1

Uvod

Moderni "pametni" mobilni telefoni, tablični računalniki in druge mobilne naprave vsebujejo ogromno različnih senzorjev. Od senzorjev orientacije, GPS, kompasov, barometrov, termometrov in nenazadnje tudi pospeškometerov in žiroskopov[4].

Izraz pametni telefon je sredi 90. let 20. stoletja opisoval mobilno napravo, ki zmora več kot navaden mobilni telefon, ki ima večje računske zmogljivosti in omogoča nekaj, česar navadni mobilni telefon ne zmora[25]. Prvotni pametni telefoni so zato vključevali funkcionalnosti dlančnikov, kot so urejanje kontaktov, sestankov, seznam opravil, kasneje tudi bralnik elektronske pošte in spletni brskalnik. Z razvojem tehnologije in nižanjem cen posameznih komponent računalniške in telefonske opreme so pametni telefoni začeli dobivati vedno več dodatnih senzorjev, ki jih je bilo mogoče uporabiti tako, da so te naprave prikazovale še več informacij, kot kadarkoli prej, postajale so "pametnejše". A pametni telefon je samo toliko pameten, kot so pametne aplikacije, ki so na tej napravi nameščene in senzorje uporabljajo.

Redke aplikacije za mobilne naprave dejansko izkoristijo potencial teh senzorjev, tako da bi mobilne naprave naredile pametnejše. V večini primerov aplikacij se senzorje uporablja kot dodatek osnovni interakciji z napravo. Pospeškometer, oziroma merjenje naklona, ki ga je mogoče realizirati s pospeškometerom, se tako v splošnem uporablja za prilagajanje postavitve uporabniškega vmesnika glede na naklon naprave ali pa za premikanje v igrice ali izboljšanje uporabniške izkušnje v smislu grafičnih efektov (premikajoče ozadje, občutek tekočega ali odzivnega grafičnega vmesnika).

Primer pametne rabe senzorjev v pametnih telefonih je naprimer utišanje zvonjenja telefona obrnjenega tako, da je s sprednjo stranjo, oziroma z zaslonom, obrnjen proti tlo[24]. Ali pa samodejni sprejem klica, če je zaznan gib tak, da ustreza dvigu telefona proti ušesu[3], ali pa samodejna izbira predvajane glasbe, glede na tempo teka. Z namenom so omenjeni taki primeri, kjer se uporablja predvsem pospeškomer, ali senzor, ki temelji na rezultatih pospeškometra. Možnosti za uporabo tega sensorja je ogromno, le odločiti se je treba kako jih prepoznati in kako na osnovi prepoznanih gest reagirati, oziroma kako je potrebno obravnavati zaznane geste ali premike mobilne naprave.

Za zaznavo gest in premikov obstajajo naprave, ki se povezujejo s telefonom in služijo kot sledilci aktivnosti, Jawbone UP¹[16], Nike+ Fuelband²[17] in Fitbit Zip³[11], a to so posebne naprave, ki beležijo aktivnost in zbrane podatke lahko posredujejo na pametni telefon ali na internet. Uporabljajo pa podobne senzorje, kot jih najdemo v sodobnih pametnih telefonih. Pametni telefoni so sposobni sami zaznavati aktivnost uporabnika, če le pravilno interpretiramo rezultate odčitavanj senzorjev.

Cilj diplomske naloge je izkoristiti potencial mobilnih naprav in senzorjev v mobilnih napravah za prepoznavanje gibanja mobilne naprave in njenega uporabnika.

V poglavju 2 bomo predstavili orodja, ki smo jih pri diplomski nalogi uporabili in principe oziroma ideje za prepoznavo gest. V poglavju 6 bomo podrobneje predstavili matematične formule in fizikalne zakone, ki nam bodo koristili pri prepoznavi tresljajev in analizi le-teh. V poglavju 4 pa bomo opisali način zaznave, predstavili rezultate testiranja in jih interpretirali, oziroma pretvorili v uporabne rezultate.

¹<https://jawbone.com/up>

²http://www.nike.com/us/en_us/c/nikeplus-fuelband

³<http://www.fitbit.com/zip>

Poglavje 2

Pregled uporabljenih tehnologij in principov

2.1 Android

Android¹ je odprtokodni operacijski sistem za mobilne naprave in vgrajene sisteme, ki temelji na jedru Linux. Prvotna ideja razvijalcev Androida je bila razviti operacijski sistem za digitalne fotokamere, a so ugotovili, da je trg teh naprav prešibek in so se odločili razviti operacijski sistem za pametne telefone, ki bi konkuriral takratnim Symbianu, Windows phone in Blackberry. Razvit je bil predvsem za naprave z zaslonom na dotik. Za njegov razvoj od leta 2005 skrbi podjetje Google, v začetku kot sponzorji projekta in kasneje kot lastniki[8].

Podjetje Google je leta 2007 ustanovilo združenje Open handset alliance²[6], v katero so združeni mobilni operaterji, podjetja, ki izdelujejo programsko opremo, podjetja, ki izdelujejo strojno opremo in podjetja, ki izdelujejo čipe in vezja. V Open handset alliance so tako združena podjetja kot so: Samsung, Sony, HTC, LG, Motorola, Intel, NVIDIA, Qualcomm, Vodafone, T-mobile, China mobile, NTT DoCoMo in druga velika podjetja iz sveta mobilnih komunikacij. Skupno je trenutno v združenje vključenih 84 podjetij[7]. Cilj združenja je razvijati odprte standarde za mobilne naprave.

¹<http://www.android.com>

²<http://www.openhandsetalliance.com/>

Android operacijski sistem je izredno fleksibilen in zato primeren za različne naprave, od vgrajenih sistemov, do zapestnih ur, televizorjev in igralnih konzol, predvsem pa se uporablja za telefone in tablične računalnike. [1]

Programski jezik za razvoj Android aplikacij je prilagojena različica programskega jezika Java, tako da je mogoče aplikacije za operacijski sistem Android pisati v kateremkoli programskem orodju, ki podpira programski jezik Java. Prevedena koda pa se za razliko od Jave ne izvaja v JVM (*Java virtual machine*), ampak se izvaja v navidezni napravi Dalvik (*Dalvik virtual machine*). Android aplikacij tako ni mogoče uporabljati kot javanske programe, lahko pa jih izvajamo na emulatorju, ali na Android napravah.

Število prodanih naprav z operacijskim sistemom Android predstavlja približno $\frac{2}{3}$ vseh mobilnih naprav, razen prenosnih računalnikov[13].

2.2 Android SDK

Android SDK³ (*Android software development kit*) je zbirka orodij za razvoj aplikacij za operacijski sistem Android. Zbirka vključuje knjižnice programskih vmesnikov, vzorčne aplikacije in orodja za gradnjo, testiranje in razhroščevanje aplikacij[9].

Zbirka vključuje tudi navidezne naprave in emulator za simuliranje navideznih naprav. Navidezne naprave so koristne predvsem za testiranje aplikacij na različnih verzijah Android operacijskega sistema in na različnih velikostih prikazovalnika. Tako nam ni potrebno aplikacije preizkušati na širokem naboru fizičnih naprav, ampak osnovne funkcionalnosti, izgled in postavitve preizkusimo na navideznih napravah.

Razvoj aplikacij s pomočjo Android SDK je sicer mogoč v raznih urejevalnikih, vendar je trenutno uradno podprto programsko orodje Eclipse. Google sicer od junija 2013 pripravlja novo uradno integrirano programsko okolje Android Studio[12], ki pa je v času pisanja te diplomske naloge še v zgodnji fazi razvoja in ga zato ne bomo uporabili.

Večina orodij Android SDK je v okviru Android ADT (*Android development tools*) tesno integrirana v Eclipse. Med njimi so:

³<http://developer.android.com/sdk/>

- Traceview
Orodje za spremljanje izvajanja aplikacije. Integrirano v pogled *Window > Open perspective > Traceview*.
- android
Vključuje dostop do upravitelja SDK (*SDK Manager*) za nadgradnje zbirke orodij za razvoj aplikacij in upravitelja navideznih naprav *AVD Manager*, ki omogoča upravljanje z virtualnimi napravami. Ostale funkcionalnosti so kreiranje in posodobitve projektov, ki so prav tako vključene v Eclipse.
- Hierachy Viewer
Orodje za iskanje napak in neoptimalnosti v postavitvi uproabniškega vmesnika. Integrirano v pogled *Window > Open perspective > Hierarchy View*.
- Pixel Perfect
Za natančnejšo postavitev elementov grafičnega vmesnika. Integrirano v pogled *Window > Open perspective > Pixel Perfect*.
- DDMS (Dalvik debug monitor service)
Orodje za razhroščevanje in nadzor aplikacije. Omogoča vpogled v informacije o nitih, kopici, dnevnik dogodkov. Omogoča tudi komunikacijo z napravo tako, da lahko ročno ustvarimo klic, sms ali pošljemo poljubno GPS lokacijo in opazujemo odziv aplikacije na posredovano. Integrirano v pogled *Window > Open perspective > DDMS*.
- adb
Razhroščevalni most Android (*Android debug bridge*) omogoča polni dostop do naprave. Funkcionalnosti so tesno integrirane v Eclipse, za nekatere funkcionalnosti (shell - dostop do ukazne vrstice na mobilni napravi) pa je potrebno adb uporabljati v ukazni vrstici[28].

2.3 Eclipse

Eclipse⁴ je prilagodljivo razvijalsko orodje zgrajeno v programskem jeziku Java in namenjeno razvoju programov in aplikacij v raznih programskih jezikih. Vključki

⁴<http://eclipse.org/>

(*plugins*), s katerimi je mogoče razširiti razvijalsko okolje Eclipse, razširjajo funkcionalnosti od podpore dodatnih programskih jezikov, do grafičnih urejevalnikov, sistemov za nadzor verzij, in drugih funkcionalnosti.

Razvijalsko okolje Eclipse je leta 2001 nastalo pod okriljem podjetij IBM, Borland, Rational Software, Red Hat, SuSE in drugih. Prvotna verzija je temeljila na IBMovem zastarelem razvijalskem orodju VisualAge⁵, ki je obstajal že od leta 1980. Prva javno dostopna verzija razvijalskega okolja Eclipse je bila verzija 3.0, izdana leta 2004[10].

Zaradi prilagodljivosti in podpore raznih operacijskih sistemov je Eclipse primerno orodje za programiranje Android aplikacij v povezavi s programi in pripomočki zbirke Android SDK.

2.4 Gnuplot

Gnuplot⁶ je program za risanje raznovrstnih grafov. Program se uporablja v ukazni vrstici in ga je mogoče uporabljati na raznih operacijskih sistemih, od Microsoft Windows, Linux, Mac OS X in drugih.

Gnuplot lahko grafe izriše direktno v grafičnem vmesniku, ali izvozi v mnoge slikovne formate, med njimi so JPG⁷, PNG⁸, SVG⁹, EPS¹⁰ in pa tudi L^AT_EX.

S pomočjo Gnuplota lahko rišemo dvodimenzionalne ali trodimenzionalne grafe, po tabeliranih podatkih ali funkcijah. Nekaj primerov grafov narejenih v gnuplotu najdemo na strani 25.

Gnuplot za risanje grafov uporabljajo druga programska in računska orodja, med njimi tudi odprtokodna alternativa matematičnemu orodju MatLab, GNU Octave, ki ga uporabljamo za numerično matematiko[14].

⁵http://en.wikipedia.org/wiki/IBM_VisualAge

⁶<http://www.gnuplot.info/>

⁷Izgubna kompresija slik (*Joint photographic experts group*)

⁸Brezizgubna kompresija slik (*Portable network graphic*)

⁹Umerljiva vektorska grafika (*Scalable vector graphics*)

¹⁰Vdelan PostScript dokument (*Encapsulated PostScript*)

2.5 Naprave, senzorji in principi zaznave

Senzorji mobilnih naprav se v osnovi delijo na tri družine senzorjev [22]:

- Senzorji gibanja
V to družino senzorjev spadajo pospeškometer, žiroskop, senzor gravitacije, senzor vrtenja.
- Senzorji za zaznavo okolice
To so senzorji za zaznavo svetlosti, bližine, temperature, tlaka. V to družino senzorjev spadajo termometer, barometer, fotometer.
- Senzorji položaja
To so kompas, magnetometer, GPS sprejemnik.

V tem diplomskem delu nas bodo zanimali samo senzorji gibanja, natančneje pospeškometer in žiroskop. Vsi senzorji gibanja na Android napravah uporabljajo isti koordinatni sistem, kot je prikazan na sliki 6.3 na strani 26.

2.5.1 Pospeškometer

Pospeškometer (*accelerometer*) je naprava, ki meri magnitudo in smer pospeška. Pospeškometri v mobilnih napravah so bili v osnovi namenjeni zaznavi spremembe orientacije naprave[5]. Uporabniški vmesnik aplikacij se lahko prilagodi glede na orientacijo naprave, pokončno ali ležeče. Na osnovi smeri največjega pospeška na napravo je mogoče določiti kako je naprava obrnjena, če je naprava mirujoča ali enakomerno pospešuje¹¹, saj v tem primeru nanjo deluje samo gravitacija.

Tipična meritev mirujoče Android naprave zaznava gravitacijski pospešek ($9.81 \frac{m}{s^2}$) v smeri od tal navzgor, v ostalih dveh smereh tridimenzionalnega koordinatnega sistema pa vrednost $0 \frac{m}{s^2}$. Ta meritev je pravilna, saj se upošteva Newtonov prvi zakon, oziroma *zakon o vztrajnosti*, ki je pojasnjen v poglavju 3.1 na strani 11.

Druge mobilne naprave, naprimer Apple iPhone, lahko uporabljajo gravitacijske enote g_0 (standardni gravitacijski pospešek), ki označujejo gravitacijski pospešek

¹¹1. Newtonov zakon

objekta v breztežnostnem prostoru blizu zemeljskega površja. Pretvorba med enotami je $1g_0 = 9.80665 \frac{m}{s^2}$, kar pomeni, da v primeru mirovanja naprava zaznava pospešek $1g_0$ v smeri od tal navzgor.

S pospeškometerom je mogoče zaznavati tudi relativni (linearni) pospešek.

Linearni pospešek

Linearni pospešek je pospešek na napravo brez upoštevanja gravitacijskega pospeška.

Na kratko:

linearni pospešek = pospešek - gravitacijski pospešek

Mirujoča naprava zaznava linearni pospešek v vseh smereh blizu $0 \frac{m}{s^2}$,

2.5.2 Žiroskop

Žiroskop (*gyroscope*) je naprava za merjenje ali ohranjanje orientacije na osnovi vrtilne količine.

Žiroskop (merilna naprava) omogoča merjenje orientacije in vrtenja, v souporabi s pospeškometerom pa omogoča natančnejšo zaznavo gibanja v 3D prostoru, kot uporaba samega pospeškometra [15].

Natančnost pospeškometra je odvisna predvsem od frekvence zaznave. V primeru, da se naprava vrti, oziroma naglo spreminja orientacijo, so zaznani pospeški pogosto napačni. V primeru nižje frekvence zaznave, so lahko posamezna odčitavanja toliko napačna, da zaznava nima nobenega smisla, ker rezultati izgledajo naključni. Pri mobilnih napravah z operacijskim sistemom Android je privzet časovni zamik (`SENSOR_DELAY_NORMAL`) med dvema zaznavama odvisen od senzorja v napravi, a se tipično giblje okrog 5Hz (200 ms med dvema vzorcema, ali 5 vzorcev na sekundo). Za natančnejše odčitavanje je potrebno povečati frekvenco zaznav, na `SENSOR_DELAY_GAME` (20 ms med vzorcema) ali `SENSOR_DELAY_FASTEST` (0 ms med vzorcema, oziroma najboljša možna)[23].

Ko se naprava vrti, ali zaznava izredno velike pospeške, pa niti višja frekvenca zaznav ni dovolj za natančna odčitavanja. V tem primeru je potrebno zaznane tresljaje popraviti v pomočjo odčitavanj žiroskopa[2]. Podroben način računanja popravljenega pospeška s pomočjo pospeškometra in žiroskopa je opisan v poglavju ?? na strani ??.

Žiroskop se na podoben način za popravljanje zaznanega pospeška uporablja tudi v igralnih konzolah, oziroma upravljalnikih PlayStation®Move [26] in Wii Remote [29].

2.5.3 Uporabljene naprave

Pri testiranjih aplikacije za zaznavo tresjajev je bilo potrebno uporabiti dejanske mobilne naprave, saj navidezne naprave nimajo vseh potrebnih senzorjev, pa tudi tresljaje je bilo potrebno zaznavati med gibanjem naprave in uporabnika. Za testiranje sta bili uporabljeni dve različni Android napravi.

Samsung Galaxy 551 (Samsung GT-I5510)

Letnik	2010
Verzija OS	2.3.6
Hitrost procesorja	600Mhz
Pospeškometer	STmicro KR3DM [19]
Natančnost pospeškometra	$0.37 \frac{m}{s^2}$
Maksimalna zaznava pospeškometra	$20 \frac{m}{s^2}$
Najmanjši interval med dvema vzorcema pospeškometra	ob spremembi
Poraba energije med delovanjem pospeškometra	$0.005mA \frac{m}{s^2}$

Žiroskop v tej napravi ni prisoten.

Samsung Galaxy S3 mini (Samsung GT-I8190)

Letnik	2012	
Verzija OS	4.1.2	
Hitrost procesorja	800Mhz	
Pospeškometer	MPU-6050	
Natančnost pospeškometra	$0.15328126 \frac{m}{s^2}$	
Maksimalna zaznava pospeškometra	$39.24 \frac{m}{s^2}$	
Najmanjši interval med dvema vzorcema pospeškometra	10 ms	[20]
Poraba energije med delovanjem pospeškometra	0.2 mA	
Žiroskop	MPU-6050 Gyroscope	
Natančnost žiroskopa	$0.06103546 \frac{rad}{sec}$	
Maksimalna zaznava žiroskopa	$2000.01 \frac{rad}{sec}$	
Najmanjši interval med dvema vzorcema žiroskopa	10 ms	
Poraba energije med delovanjem žiroskopa	0.2 mA	

Poglavje 3

Matematična in fizikalna pomoč pri analizi

3.1 Newtonov prvi zakon, oziroma zakon o vztrajnosti

Izrek 3.1 Če je vsota vseh sil na telo enaka 0, je gibanje telesa premočrtno ali mirujoče.[18]

Dokaz. Iz česar sledi, da je pospešek \vec{a} pri mirujoči mobilni napravi nasprotno enak gravitacijskemu pospešku \vec{g}

$$\vec{F}_g = m\vec{g} \quad (3.1)$$

$$\vec{F}_a = m\vec{a} \quad (3.2)$$

$$\sum F = \vec{F}_a + \vec{F}_g = 0 \quad (3.3)$$

$$\sum F = m(\vec{a} + \vec{g}) = 0 \quad (3.4)$$

$$\vec{a} + \vec{g} = 0 \quad (3.5)$$

$$\vec{a} = -\vec{g} \quad (3.6)$$

□

3.2 Absolutni pospešek

Absolutni pospešek dobimo tako, da se vektorsko sešteje vrednost pospeška izmerjenega v smereh x , y in z tridimenzionalnega koordinatnega sistema.

$$|\text{absolutniPospesek}| = \sqrt{x\text{Pospesek}^2 + y\text{Pospesek}^2 + z\text{Pospesek}^2}. \quad (3.7)$$

S pomočjo absolutnega pospeška lahko zaznamo, da se naprava premika. V primeru mirovanja, kar pomeni, da na napravo deluje samo gravitacija, je absolutni pospešek $1g_0$ (približno $9.81 \frac{m}{s^2}$). Ko se naprava premika, pa je absolutni pospešek različen od $1g_0$.

Padajoča naprava, na katero ne deluje nobena druga sila, razen gravitacije, ima tako absolutni pospešek blizu 0, saj če objekt pada prosto, doseže pospešek blizu gravitacijskega pospeška, torej se vsota pospeška \vec{a} in gravitacijskega pospeška \vec{g} na sam objekt v tem trenutku zmanjša. Če naprava potuje v nasprotni smeri gravitacije, se delovanje gravitacije na ta objekt poveča in je absolutni pospešek večji od $1g_0$.

V tem diplomskem delu se absolutni pospešek uporablja zato, da odstranimo vpliv smeri oziroma rotacije naprave na rezultate, tam kjer tega ne potrebujemo.

Poglavje 4

Zaznava gibanja naprave

Za zaznavo gibanja naprave in uporabnika smo pripravili Android aplikacijo, ki beleži vzorce senzorjev. Konkretno obdelavo podatkov in prikaz v obliki grafov, smo naredili s pomočjo programa gnuplot, omenjenega v poglavju 2.4, zato je opisan samo del programske kode, ki je ključen za zaznavo in iz katerega je razvidno kako se rezultati pridobivajo.

4.1 Priprava

Za programiranje aplikacije smo pripravili razvijalsko okolje Eclipse opisano v poglavju 2.3 in zbirko orodij za izdelavo Android aplikacij Android SDK opisano v poglavju 2.2.

V programu Eclipse smo ustvarili nov projekt Android.

Za pisanje zaznanih podatkov v datoteko, je bilo potrebno v *AndroidManifest.xml* datoteki dodati dovoljenje za pisanje na zunanjo spominsko kartico, saj lahko količina podatkov pri hitri zaznavi naglo raste in notranji pomnilnik mobilnih naprav je bolj omejen, kot so spominske kartice.

`AndroidManifest.xml`¹ je datoteka, prisotna pri vsakem Android projektu. Datoteka vsebuje informacije o aplikaciji, imena aktivnosti in servisov v aplikaciji, posebne nastavitve teh aktivnosti in servisov, zahtevana dovoljenja, verzija gradnje in minimalna in ciljna verzija Androida[27].

¹<http://developer.android.com/guide/topics/manifest/manifest-intro.html>

V datoteko `AndroidManifest.xml` torej dodamo:

```
<uses-permission  
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Dostop do senzorjev je v Androidu omogočen preko *SensorManager* programskega vmesnika. *SensorManager* vklopimo v *onCreate* metodi glavnega razreda.

```
SensorManager sensorManager = (SensorManager) getSystemService(  
    SENSOR_SERVICE );
```

Odločiti se moramo tudi katere senzorje bomo uporabljali in s kakšno frekvenco osveževanja bi želeli dobivati nove meritve.

```
Sensor pospeskometer =  
    sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
Sensor ziroskop =  
    sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
```

```
int frekvenca = SensorManager.SENSOR_DELAY_FASTEST;
```

```
sensorManager.registerListener(this, pospeskometer, frekvenca);  
sensorManager.registerListener(this, ziroskop, frekvenca);
```

Zaznane tresljaje in spremembe naklona smo zapisovali v datoteko v metodi *onSensorChanged*, kjer je mogoče dobiti objekte tipa *SensorEvent*, ki hranijo informacije o zaznavi, imenu senzorja, natančnosti in času zaznave[21].

4.2 Zaznava tresljajev

Aplikacija, ki dobiva rezultate meritev senzorjev mora implementirati *SensorEventListener*:

```
public class ZaznavaTresljajev extends Activity implements  
    SensorEventListener {
```

Kar pa samodejno zahteva, da implementiramo metodi *onAccuracyChanged* in *onSensorChanged*.

```
public void onAccuracyChanged (Sensor sensor, int accuracy) {  
    }  
  
public void onSensorChanged(SensorEvent sensorEvent) {  
    // tukaj obdelamo rezultate  
}
```

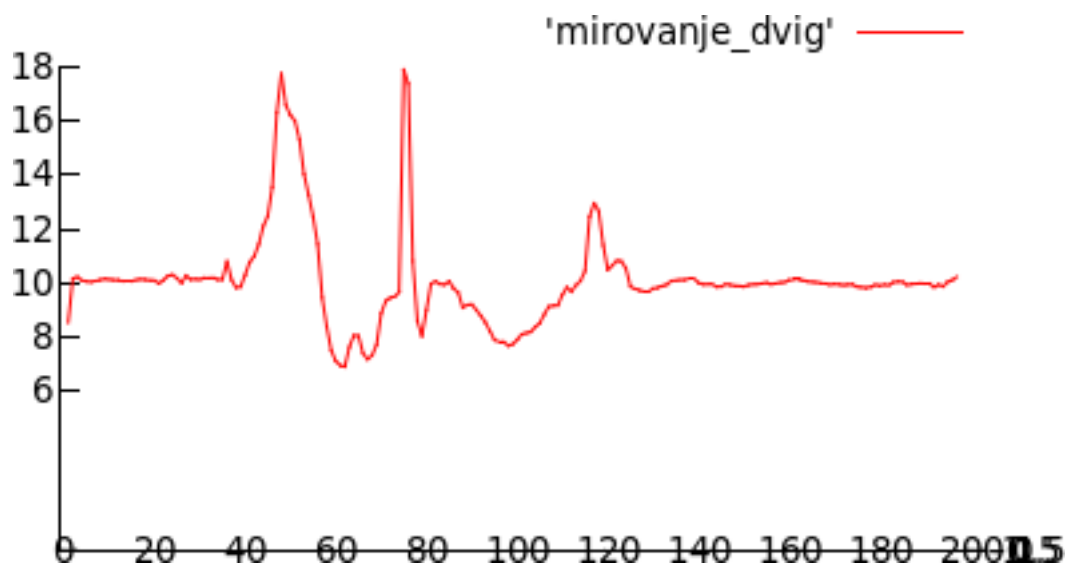
V metodi *onAccuracyChanged* ne rabimo narediti ničesar, metoda *onSensorChanged* pa bo obdelala vsako spremembo, ki jo bo senzor zaznal, zato je tam implementirano shranjevanje zaznanih podatkov v datoteko.

4.2.1 Mirovanje in dvig telefona

Scenarij obdela primer v katerem je mobilna naprava položena na mizo in dvignjena iz mize tako, da uporabnik pogleda v prikazovalnik.

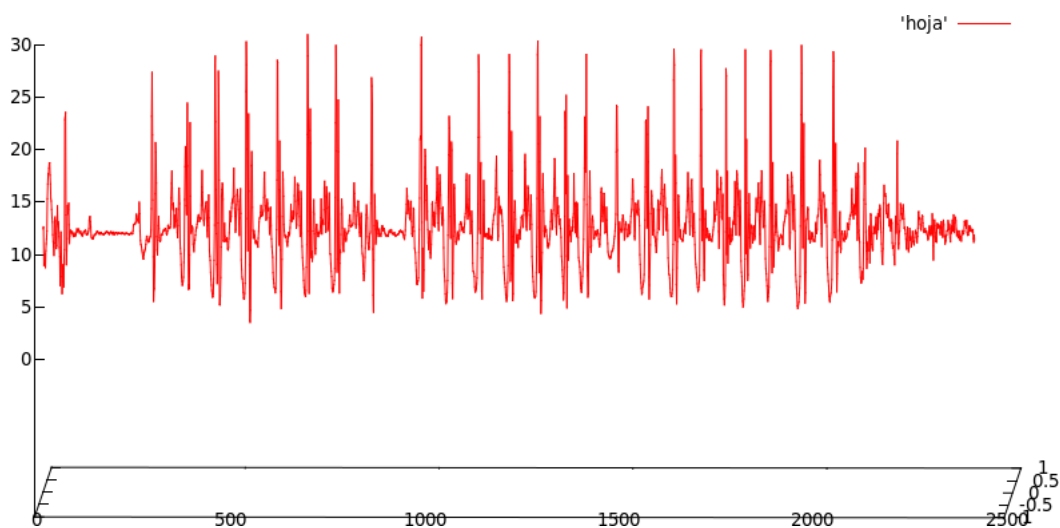
Za zaznavo premika mobilne naprave je dovolj že sama zaznava tresljajev neodvisno od smeri pospeška. Za ta namen smo uporabili absolutni pospešek, opisan v poglavju 3.2 na strani 12.

Na grafu 4.1 lahko vidimo absolutni pospešek v primeru dviga telefona z mize z uporabo Android senzorja za zaznavo pospeška, ki upošteva tudi vpliv gravitacije na napravo (`Sensor.TYPE_ACCELEROMETER`). V času mirovanja je absolutni pospešek delujoč na napravo okrog $10 \frac{m}{s^2}$, v primeru premikanja naprave, pa je različen od $10 \frac{m}{s^2}$. Podoben graf bi dobili tudi, če bi zaznavali linearni pospešek (Na operacijskem sistemu Android označen kot `Sensor.TYPE_LINEAR_ACCELERATION`), vendar bi se v tem primeru vrednosti pospeškov v mirovanju gibale okrog $0 \frac{m}{s^2}$, saj odčitavanja ne upoštevajo gravitacije.



Slika 4.1: Graf absolutnega pospeška od mirovanja mobilnega telefona na mizi do dviga z mize.

4.2.2 Hoja, telefon v žepu

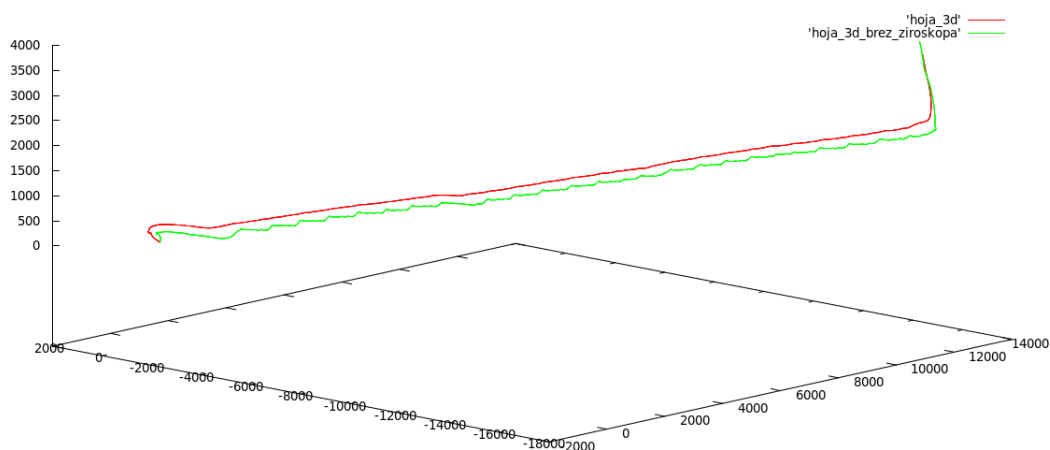


Slika 4.2: Absolutni pospešek izmerjen z mobilno napravo med hojo z mobilnih telefonom v žepu.

Scenarij obdela primer v katerem uporabnik mobilno napravo pospravi v hlačni žep in naredi nekaj korakov po prostoru.

Za hojo je značilno periodično tresenje, koraki se ponavljajo, sam korak pa se po zaznanih tresljajih vidi kot povečan absolutni pospešek na napravo.

Z grafa absolutnega pospeška med hojo je mogoče razpoznati frekvenco korakov, število korakov v opazovanem času, tudi morebitne ustavitve je mogoče opaziti. Opazovani vzorec traja 40 sekund, v tem času je jasno vidnih 25 korakov, saj je naprava med vsakim korakom zaznavala povečan absolutni pospešek.



Slika 4.3: 3D graf poti na osnovi zaznanega pospeška v prostoru.

S 3D grafa poti, ki smo ga dobili tako, da smo posamezne pospeške sešteli na sledeč način:

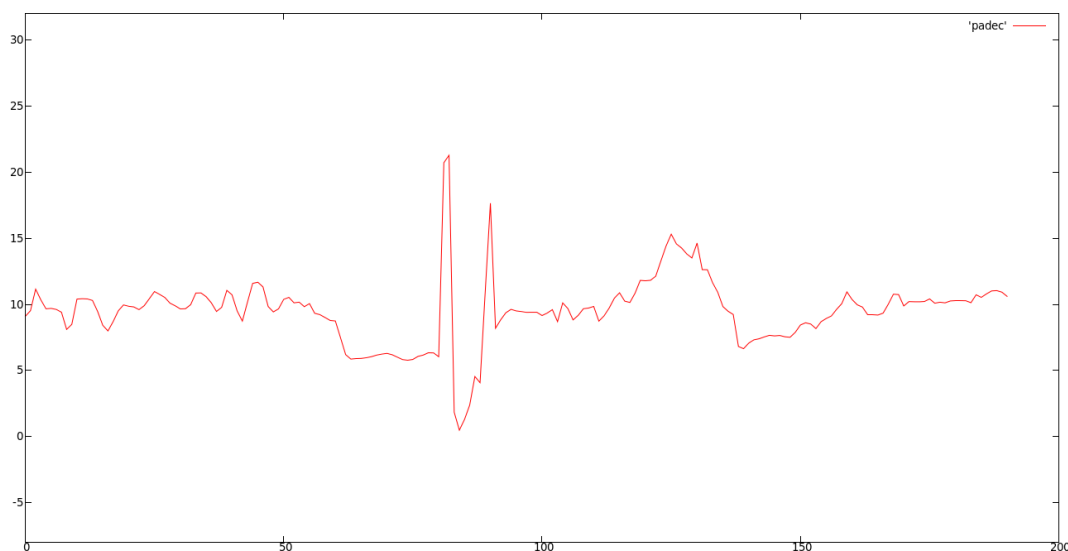
$x_n = 0.98 * ziroskop_n + 0.02 * x_n$, s čimer smo eliminirali vpliv posameznih manjših tresljajev na rezultat, zaporedne rezultate pa smo sešteli, da smo dobili pot, lahko vidimo naravo gibanja (rdeč graf). Prema pot, ki je vidna na grafu 4.3, je znak, da je gibanje bilo premo, kar je značilno za hojo, vožnjo ali ostala enakomerna gibanja, brez nenadnih pospeškov in sprememb smeri.

Zeleni graf prikazuje 3D graf poti, ki ne upošteva meritev ziroskopa. V grobem se grafa prekrivata, a se v primeru nenadnih vrtenj ali nenavadnih gibanj podobnost hitro zmanjša. Iz zelenega grafa, ki je bolj odvisen od manjših sprememb, je mogoče, podobno kot pri grafu absolutnega pospeška, razbrati posamezne korake.

4.2.3 Padec telefona

Scenarij obdela primer v katerem uporabnik spusti mobilno napravo z višine 1.5 metra, med padanjem pa se naprava nekontrolirano vrti.

Za padec telefona je značilno precej bolj nenavadno gibanje, kot pri hoji. Posamezni lokalni maksimumi si ne sledijo periodično, lahko se zgodi, da naprava med padanjem ne zaznava pretirano odklonskega absolutnega pospeška, na osnovi katerega bi lahko vedeli, da se karkoli dogaja z napravo, zato merjenje absolutnega pospeška med padanjem v tem primeru ni dobro, saj razen izrednega povečanja absolutnega pospeška med stikom s tlemi, oziroma površino s katero naprava trči, ne dobimo nobene informacije o gibanju. Na sliki 4.4 vidimo graf absolutnega pospeška na mobilno napravo med padcem.



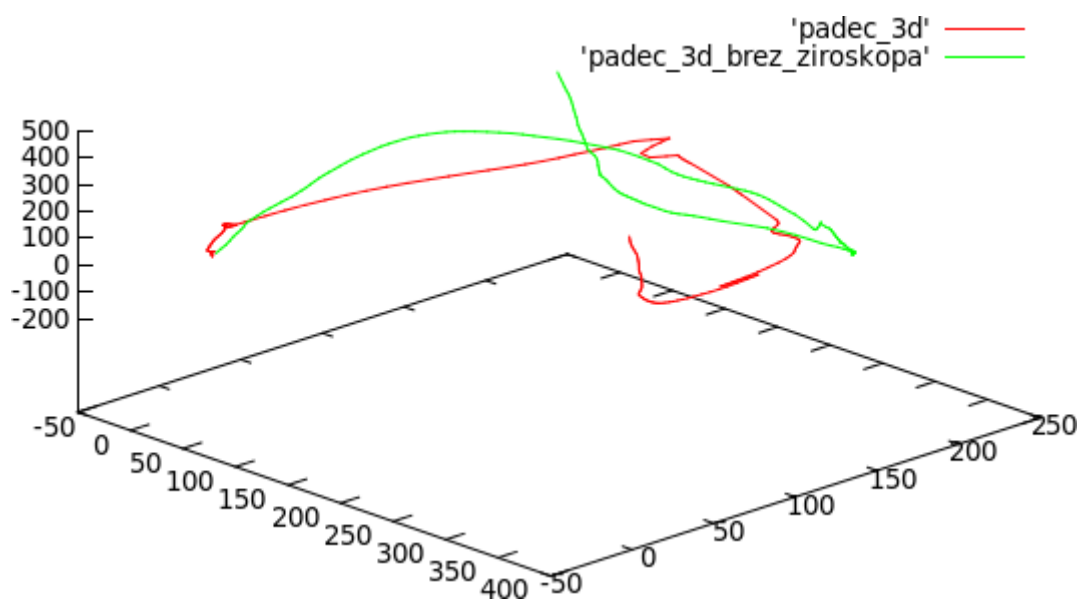
Slika 4.4: Absolutni pospešek na mobilno napravo med padcem

Na slikah 4.5 in 4.6 vidimo 3D graf pospeška na napravo med padcem, ki se precej razlikuje, glede na to a upoštevamo meritve žiroskopa (rdeč graf) ali jih ne upoštevamo (zelen graf). Grafa prikazujeta iste meritve, a sta različno obrnjena v prostoru.

Zeleni graf, brez upoštevanja meritev žiroskopa, iz katerega vidimo samo delovanje pospeškov na napravo, je bolj umirjen, iz česar bi lahko sklepali, da gre za graf, ki prikazuje relativno premo gibanje, naprimer zamah z roko, med katerim

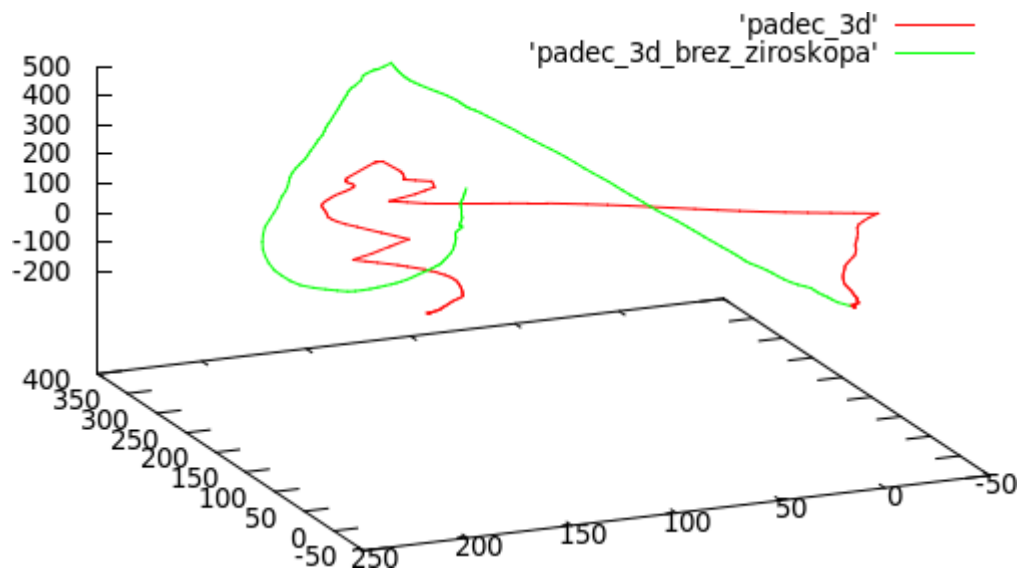
bi rahlo spremenili naklon mobilne naprave.

Analiza rdečega grafa, ki upošteva meritve žiroskopa, pa kaže na to, da gre za precej bolj kompleksno gibanje. Vidne so nagle spremembe orientacije, ki vplivajo na meritev popravljenega pospeška, ki upošteva tudi meritve z čiroskopa, kar nakazuje na to, da se je naprava med meritvijo vrtela.



Slika 4.5: 3D graf poti na osnovi pospeškov na napravo med padcem.

4.3. PREPOZNAVA GIBANJA NA OSNOVI ZAZNANIH TRESLJAJEV



Slika 4.6: 3D graf na osnovi pospeškov na napravo med padcem (obrnjen pogled).

4.3 Prepoznavna gibanja na osnovi zaznanih tresljajev

Gibanje naprave in zaznani tresljaji torej niso samoumevni in enostavni za uporabo. Nemogoče je sklepati o gibanju na osnovi samo enega senzorja oziroma zbirke senzorjev samo enega tipa, za več informacije je potrebno senzorje kombinirati in v različnih situacijah izbrati tako kombinacijo, ki nam o posameznem gibanju lahko pove čimveč.

Če se je mogoče omejiti na posamezno dejavnost, za katero vemo kaj pričakovati od meritev, se izplača pričakovane meritve preoblikovati v pravilo in v zaznanih tresljajih iščemo skladnost s temi pravili. Tako lahko v primeru sledilcev športne aktivnosti (ang. *sport tracker*) pričakujemo periodično gibanje uporabnika, naj gre za hojo, tek, mogoče kolesarjenje, dvigovanje uteži. Če je mogoče mobilno napravo imeti pozicionirano tako, da se giba hkrati z uporabnikom, kot je to v testnem primeru hoje z mobilno napravo v žepu, lahko vnaprej definirano, da nas zanima periodično gibanje z izrazitimi absolutnimi pospeški, neodvisno od orientacije naprave. Graf takega gibanja bo podoben grafu 3.2 na strani 12. Z

rezultatov meritev pa lahko ugotovimo frekvenco korakov, čas trajanja vadbe, lahko bi tudi ocenili porabo energije in jo uporabniku prikazali v prijazni obliki, na primer v kilokalorijah (Kcal) porabljenih med vadbo.

Če ne vemo kaj pričakovati od meritev, oziroma aplikacija nima enoznačno določenega gibanja, pa je smiselno uporabljati več kot absolutne pospeške. Kot smo prikazali v primeru padca naprave, je pri nenadzorovanih, oziroma neenakomernih gibanjih težko pričakovati kakšne bodo meritve, uporaba samo enega senzorja pa nam ne da pravih informacij o naravi gibanja. V tem primeru je potrebno senzorje kombinirati.

Za primer stalnega nadzora uporabnikovega gibanja bi tako lahko uporabili meritve pospeškometra in žiroskopa skupaj. Na osnovi meritev samega žiroskopa bi lahko sklepali o naklonu, oziroma vrtenju naprave. Če so zaporedne meritve žiroskopa v povprečju okrog 0 rad (ali stopinj, če naprava meri naklon v kotnih stopinjah), naprava miruje, oziroma ne spreminja orientacije. Če ima uporabnik mobilno napravo v žepu, bi na osnovi teh meritev lahko sklepali, da tudi uporabnik miruje, torej je lahko v postelji, na stolu, tudi v avtu ali na vlaku. Sama zaznana sprememba orientacije oziroma ohranitev orientacije v tem primeru ne povesta ničesar. V kombinaciji s pospeškom, pa bi lahko vedeli, če gre za mirovanje, torej ležanje, sedenje, ali gre za sedenje na avtobusu, saj bi se pri sedenju za mizo tudi na pospešku poznalo, da se ne spreminja, pri sedenju na avtobusu pa je za pričakovati, da se pospešek spreminja pri pospeševanju in zaviranju avtobusa.

4.3. PREPOZNAVA GIBANJA NA OSNOVI ZAZNANIH TRESLJAJEV

Poglavje 5

Sklepne ugotovitve

Diplomsko delo prikaže idejo za način prepoznave gibanja uporabnika mobilne naprave. Na osnovi primerov scenarijev analizira ustreznost senzorjev v predvidenih situacijah in predlaga rešitev pomanjkljivosti posameznih senzorjev s kombiniranjem le-teh.

Moderne mobilne in tudi druge naprave imajo ogromno senzorjev s katerimi je mogoče zaznavati parametre naprave in okolice. Med priljubljenimi senzorji je pospeškometer, ki ga je mogoče najti v širokem naboru naprav, od mobilnih naprav, do zabavne elektronike in tudi do bolj nepričakovanih mest, kot so avtomobili (merjenje pospeškov delujočih na vozilo za elektronski nadzor stabilnosti koles ESP), trdi diski (predvsem v prenosnih računalnikih, ki se pogosto premikajo, kjer služi za izključitev vrtenja trdega diska med zaznavo večjega pospeška na napravo za zmanjšano možnost okvar diska zaradi malomarnosti ali neprevidne rabe prenosnega računalnika) in celo stavbe (za zaznavo potresov).

V napravah, ki so pretežno statične, je mogoče večino zaznav ali odzivov na pospeške realizirati s samim pospeškometerom, ki lahko meri pospeške v treh dimenzijah, ali samo eni, odvisno od potreb naprave. V mobilnih napravah, kot so mobilni telefoni in tablični računalniki, ki hitro in pogosto spreminjajo orientacijo, pa to ni dovolj, zato je potrebno pospeškometer kombinirati z žiroskopom.

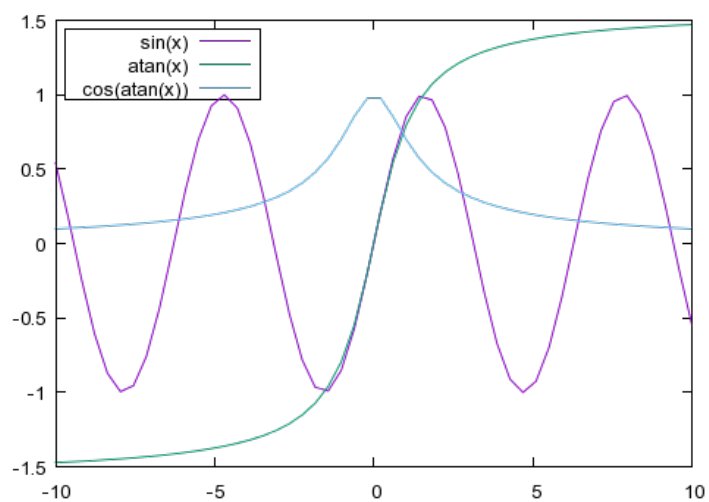
Mobilne aplikacije za pametne telefone je mogoče pripravljati tako, da se odzivajo res "pametno", če upoštevamo več dejavnikov, ki na napravo delujejo.

Če bi želeli znanje, ki ga diplomsko delo prinaša, uporabiti v dejanski aplikaciji, bi bilo potrebno analizirati dodatne scenarije gibanja, ki se pojavlja v problemu, ki

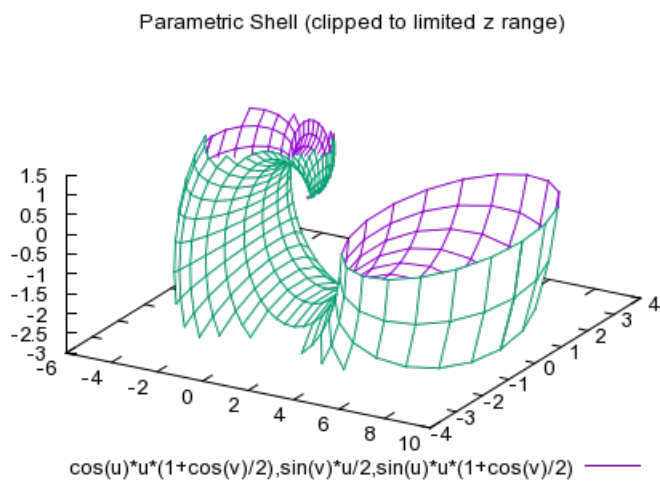
bi ga aplikacija reševala in iz analize sklepati na pravila gibanja, s pomočjo katerih bi lahko natančneje ocenili za kako gibanje gre. Na enak način, s kombiniranjem večih senzorjev bi bilo mogoče sledilce aktivnosti realizirati kot aplikacijo za mobilne naprave, brez potrebe po dodatnih sledilcih aktivnosti, ki trenutno obstajajo na tržišču kot samostojne naprave.

Poglavje 6

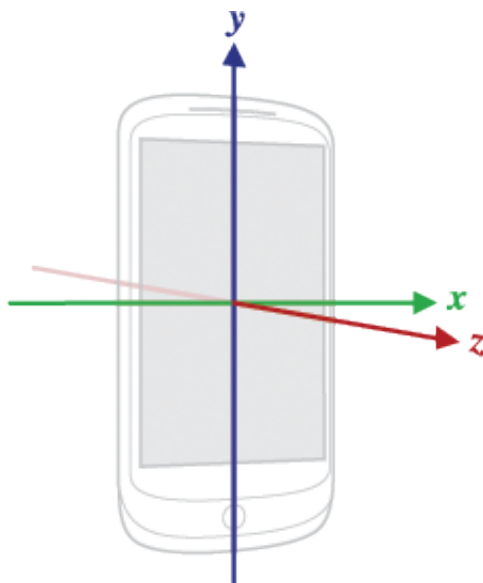
Slike



Slika 6.1: Preprost graf narisan v orodju Gnuplot (vir: gnuplot.sourceforge.net)



Slika 6.2: Trodimenzionalni graf narisan v orodju Gnuplot (vir: gnuplot.sourceforge.net)



Slika 6.3: Koordinatni sistem senzorjev gibanja na Android napravah (vir: http://developer.android.com/guide/topics/sensors/sensors_overview.html)

Poglavje 7

Literatura in viri

- [1] (2012) Android Everywhere: 10 Types of Devices That Android Is Making Better. Dostopno na:
<http://www.androidauthority.com/android-everywhere-10-types-of-devices-that-android-is-making-better-57012/>.

- [2] (2012) Better motion control using accelerometer/gyroscope sensor fusion, str 17. Dostopno na:
<http://www.slideshare.net/paller/better-motion-control-using-accelerometergyroscope-sensor-fusion>.

- [3] (2012) How to use Motion gestures on the Galaxy S3 — Android Central. Dostopno na:
<http://www.androidcentral.com/how-use-motion-gestures-galaxy-s3>.

- [4] (2012) Sensors and Cellphones. Dostopno na:
www.stanford.edu/class/cs75n/Sensors.pdf \begin{group}\let\relax\relax\endgroup [Pleaseinsert\PrerenderUnicode{ }intopreamble].

- [5] (2013) Accelerometer — Wikipedia, the free encyclopedia. Dostopno na:
<http://en.wikipedia.org/wiki/Accelerometer>.

- [6] (2013) Alliance FAQ — Open Headset Alliance. Dostopno na:
http://www.openhandsetalliance.com/oha_faq.html.

- [7] (2013) Alliance Members — Open Headset Alliance. Dostopno na:
http://www.openhandsetalliance.com/oha_members.html.
- [8] (2013) Android (operating system) - Wikipedia, the free encyclopedia.
Dostopno na:
[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [9] (2013) Android software development - Wikipedia, the free encyclopedia.
Dostopno na:
[http://en.wikipedia.org/wiki/Android_software_development#
Android_SDK](http://en.wikipedia.org/wiki/Android_software_development#Android_SDK).
- [10] (2013) Eclipse (software) - Wikipedia, the free encyclopedia. Dostopno na:
[http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software)).
- [11] (2013) Fitbit - Wikipedia, the free encyclopedia. Dostopno na:
http://en.wikipedia.org/wiki/Fitbit#Fitbit_Tracker.
- [12] (2013) Getting started with Android Studio — Android Developers. Dostopno na:
<http://developer.android.com/sdk/installing/studio.html>.
- [13] (2013) Global mobile statistics 2013 Part A: Mobile subscribers; handset market share; mobile operators — mobiThinking. Dostopno na:
<http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#smartphoneos>.
- [14] (2013) Gnuplot - Wikipedia, the free encyclopedia. Dostopno na:
<http://en.wikipedia.org/wiki/Gnuplot>.
- [15] (2013) Gyroscope - Wikipedia, the free encyclopedia. Dostopno na:
http://en.wikipedia.org/wiki/Gyroscope#Modern_uses.
- [16] (2013) HowStuffWorks "How Jawbone UP Works". Dostopno na:
[http://electronics.howstuffworks.com/gadgets/other-gadgets/
jawbone-up1.htm](http://electronics.howstuffworks.com/gadgets/other-gadgets/jawbone-up1.htm).
- [17] (2013) Inside the Nike+ FuelBand...Nike chip inside! — Chipworks Blog.
Dostopno na:

- <http://www.chipworks.com/en/technical-competitive-analysis/resources/blog/inside-the-nike-fuelband-nike-chip-inside/>.
- [18] (2013) Newton's laws of motion - Wikipedia, the free encyclopedia. Dostopno na:
http://en.wikipedia.org/wiki/Newtons_laws_of_motion#Newton.27s_first_law
- [19] (2013) Samsung Galaxy 551 - Full phone specifications. Dostopno na:
http://www.gsmarena.com/samsung_galaxy_551-3515.php.
- [20] (2013) Samsung I8190 Galaxy S III mini - Full Phone Specifications. Dostopno na:
http://www.gsmarena.com/samsung_i8190_galaxy_s_iii_mini-5033.php.
- [21] (2013) SensorEvent — Android Developers. Dostopno na:
<http://developer.android.com/reference/android/hardware/SensorEvent.html>.
- [22] (2013) Sensors Overview — Android Developers. Dostopno na:
http://developer.android.com/guide/topics/sensors/sensors_overview.html.
- [23] (2013) Sensors overview — Android Developers. Dostopno na:
http://developer.android.com/guide/topics/sensors/sensors_overview.html#sensors-monitor
- [24] (2013) Silence your Samsung Galaxy S3 phone quickly. Dostopno na:
<http://jimlynch.com/technologytips/mobile/silence-your-samsung-galaxy-s3-phone-quickly/>.
- [25] (2013) Smartphone - Wikipedia, the free encyclopedia. Dostopno na:
<http://en.wikipedia.org/wiki/Smartphone>.
- [26] (2013) Sony PlayStation Move tear down report — News — TechRadar. Dostopno na:
<http://www.techradar.com/news/gaming/sony-playstation-move-tear-down-report-717793>.

- [27] (2013) The AndroidManifest.xml file — Android Developers. Dostopno na:
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>.
- [28] (2013) Tools Help — Android Developers. Dostopno na:
<http://developer.android.com/tools/help/index.html>.
- [29] (2013) Wii Remote - Wikipedia, the free encyclopedia. Dostopno na:
http://en.wikipedia.org/wiki/Wii_Remote.