

Univerza
v Ljubljani
Fakulteta
za gradbeništvo
in geodezijo



**ODDELEK ZA
GEODEZIJO**

*Univerzitetni študij
geodezije
Smer: Prostorska
informatika*

Kandidat:

GAŠPER GENORIO

Diplomska naloga št.:

**Dopolnjevanje spletne baze GPS sledi
Supplementing Of GPS Track Web Database**

Graduation thesis No.:

Mentor:
doc.dr. Mitja Lakner

Predsednica komisije:
doc.dr. Mojca Kosmatin Fras

Somentor:
doc.dr. Dušan Petrovič

Član:

Datum zagovora: Ljubljana, _____ 2008

Ljubljana, januar 2008

BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK	004.6:528.28:528.9:659.2(043.2)
Avtor	Gašper Genorio
Mentor	doc. dr. Mitja Lakner
Somentor	doc. dr. Dušan Petrovič
Naslov	Dopolnjevanje spletne baze GPS sledi
Obseg in oprema	69 strani, 17 preglednic, 25 slik, 36 enačb, 7 prilog
Ključne besede	strežnik, HTML, PHP, SQL, spletno mesto, GPS sled, programiranje, DMR, višina

Izveček

V diplomski nalogi je predstavljeno dopolnjevanje spletnega mesta, na katerem so brezplačno objavljene GPS sledi planinskih in kolesarskih poti ter turnih smuk po Sloveniji. Predvsem sta dopolnjena podatkovna baza in uporabniški vmesnik spletnega mesta z namenom učinkovite uporabe baze, dostopnosti nabora opisnih podatkov in prijaznosti uporabniku. Opisane so osnovne značilnosti in uporaba brezplačnih spletnih tehnologij, na katerih spletno mesto temelji. GPS sledi so od zajema do objave na spletnem mestu podvržene različnim postopkom obdelave. Del diplomske naloge je namenjen izpeljavi in primerjavi štirih različnih matematičnih metod za določitev višine točke GPS sledi na podlagi DMR. Za najbolj učinkovito se je izkazala najbolj preprosta, linearna metoda.

BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION AND ABSTRACT

UDC	004.6:528.28:528.9:659.2(043.2)
Author	Gašper Genorio
Supervisor	Assist. Prof. Mitja Lakner
Co-supervisor	Assist. Prof. Dušan Petrovič
Title	Supplementing Of GPS Track Web Database
Notes	69 pages, 17 tables, 25 figures, 36 equations, 7 annexes
Key words	server, HTML, PHP, SQL, web site, GPS track, programming, DEM, elevation

Abstract

The graduation thesis introduces supplementing of a web site, that offers free published GPS tracks of mountaineering, cycling ways and tour skiing. Database and user interface of the web site are supplemented with intention of efficient use of stored data, accessibility of descriptive datasets and user friendliness. Thesis describes basic features and use of free web technologies, on which web site is based upon. From acquisition to publication, GPS tracks are subjected to different procedures of data processing. Therefore a part of thesis is assigned to derivation and comparison of four different mathematical methods for calculation of elevation of a GPS track point. For the most efficient method turned out to be the simplest, linear method.

STRAN ZA POPRAVKE, ERRATA

Stran z napako	Vrstica z napako	Namesto	Naj bo
----------------	------------------	---------	--------

IZJAVA O AVTORSTVU

Podpisani Gašper Genorio izjavljam, da sem avtor diplomske naloge z naslovom:

»DOPOLNJEVANJE SPLETNE BAZE GPS SLEDI«.

Izjavljam, da prenašam vse materialne avtorske pravice v zvezi z diplomsko nalogo na UL,
Fakulteto za Gradbeništvo in Geodezijo.

Ljubljana, 11.1.2008

Gašper Genorio

IZJAVE O PREGLEDU NALOGE

Nalogo so si ogledali:

BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK	004.6:528.28:528.9:659.2(043.2)
Avtor	Gašper Genorio
Mentor	doc. dr. Mitja Lakner
Somentor	doc. dr. Dušan Petrovič
Naslov	Dopolnjevanje spletne baze GPS sledi
Obseg in oprema	69 strani, 17 preglednic, 25 slik, 36 enačb, 7 prilog
Ključne besede	strežnik, HTML, PHP, SQL, spletno mesto, GPS sled, programiranje, DMR, višina

Izveček

V diplomski nalogi je predstavljeno dopolnjevanje spletnega mesta, na katerem so brezplačno objavljene GPS sledi planinskih in kolesarskih poti ter turnih smuk po Sloveniji. Predvsem sta dopolnjena podatkovna baza in uporabniški vmesnik spletnega mesta z namenom učinkovite uporabe baze, dostopnosti nabora opisnih podatkov in prijaznosti uporabniku. Opisane so osnovne značilnosti in uporaba brezplačnih spletnih tehnologij, na katerih spletno mesto temelji. GPS sledi so od zajema do objave na spletnem mestu podvržene različnim postopkom obdelave. Del diplomske naloge je namenjen izpeljavi in primerjavi štirih različnih matematičnih metod za določitev višine točke GPS sledi na podlagi DMR. Za najbolj učinkovito se je izkazala najbolj preprosta, linearna metoda.

BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION AND ABSTRACT

UDC	004.6:528.28:528.9:659.2(043.2)
Author	Gašper Genorio
Supervisor	Assist. Prof. Mitja Lakner
Co-supervisor	Assist. Prof. Dušan Petrovič
Title	Supplementing Of GPS Track Web Database
Notes	69 pages, 17 tables, 25 figures, 36 equations, 7 annexes
Key words	server, HTML, PHP, SQL, web site, GPS track, programming, DEM, elevation

Abstract

The graduation thesis introduces supplementing of a web site, that offers free published GPS tracks of mountaineering, cycling ways and tour skiing. Database and user interface of the web site are supplemented with intention of efficient use of stored data, accessibility of descriptive datasets and user friendliness. Thesis describes basic features and use of free web technologies, on which web site is based upon. From acquisition to publication, GPS tracks are subjected to different procedures of data processing. Therefore a part of thesis is assigned to derivation and comparison of four different mathematical methods for calculation of elevation of a GPS track point. For the most efficient method turned out to be the simplest, linear method.

ZAHVALA

Zahvaljujem se mentorju dr. Mitji Laknerju za velikodušno strokovno pomoč in vzpodbudo, somentorju dr. Dušanu Petroviču in vsem, ki so kakorkoli pripomogli k nastanku tega diplomskega dela. Zahvaljujem se tudi staršema, Jelki in Jožetu, ki sta mi ves čas študija stala ob strani in verjela vame.

KAZALO VSEBINE

1. UVOD	1
2. UPORABLJENE TEHNOLOGIJE	2
2.1 Strežnik Apache	2
2.2 HTML, XHTML	3
2.3 PHP	4
2.4 SQL	5
2.5 Strežniški programski svežnji, WAMP.....	6
2.6 Vtičniki, PHPlot.....	7
3. OBSTOJEČE STANJE SPLETNEGA MESTA »PLANINSKE IN KOLESARSKE POTI TER TURNI SMUKI Z GPS PODPORO«	7
3.1 Programska osnova	8
3.2 Struktura spletnega mesta	9
3.2.1 Informacijska arhitektura.....	9
3.2.2 Mape in datoteke	10
3.3 Vhodni podatki.....	10
3.3.1 Primeren GPS sprejemnik	11
3.4 Struktura SQL podatkovne baze	11
3.5 Obdelava podatkov	12
3.5.1 Datoteka <i>index.php</i>	13
3.5.2 Datoteka <i>gradnja_poti.php</i>	13
3.5.3 Datoteka <i>rezultati.php</i>	14
3.6 Rezultati	15
3.7 Uporaba.....	15

4.	PRILAGAJANJE VHODNIH DATOTEK	16
4.1	Prenos na osebni računalnik, pretvorba formata in zaslonska vektorizacija	17
4.2	Določanje višin točk	19
4.2.1	Digitalni model reliefa	19
4.2.2	Kriging	20
4.3	Prekrivanje datoteke PLT in baze DMR 12,5.....	21
4.3.1	Transformacija v državni koordinatni sistem.....	21
4.3.2	Iskanje območja z bisekcijo	23
4.4	Primerjava preprostih metod za določitev višine na podlagi DMR.....	24
4.4.1	Metoda uteži težiščne točke	24
4.4.2	Metoda regresijske ravnine	28
4.4.3	Metoda dveh ravnin	33
4.4.4	Linearna metoda.....	39
4.5	Primerjava rezultatov.....	42
4.6	Glajenje z drsečim povprečjem	44
5.	DOPOLNJEVANJE SPLETNEGA MESTA	44
5.1	SWOT analiza	44
5.2	Možne dopolnitve in njihova racionalizacija.....	46
6.	IZVEDENE DOPOLNITVE	47
6.1	Možnost izbiranja začetnih točk po atributih	48
6.1.1	Dopolnjevanje SQL tabele <i>tocke</i>	49
6.1.1.1	Regionalni položaj	49
6.1.1.2	Vrsta točke	50
6.1.1.3	Obstoj koče	50
6.1.2	Izdelava grafičnega vmesnika za izbiranje točk po atributih.....	51

6.1.2.1	Datoteka <i>index.php</i>	51
6.1.2.2	Datoteka <i>izbira.php</i>	54
6.2	Popravljanje skrajšanih imen točk v polna imena.....	57
6.2.1	Dopolnjevanje SQL tabele <i>tocke</i>	58
6.2.2	Dopolnjevanje uporabniškega vmesnika.....	59
6.3	Povezava na stran z opisom točke.....	59
6.3.1	Dopolnjevanje SQL tabele <i>tocke</i>	60
6.3.2	Dopolnjevanje uporabniškega vmesnika.....	60
6.4	Opremljanje poti z atributi	61
6.4.1	Dopolnjevanje SQL tabele <i>poti</i>	62
6.4.2	Dopolnjevanje uporabniškega vmesnika.....	63
7.	SKLEP	65
VIRI	67

KAZALO PREGLEDNIC

Preglednica 1: Obrazložitev simbolov, uporabljenih v enačbah od (1) do (7).....	24
Preglednica 2: Izračun višin točk z metodo uteži sredinske točke.....	28
Preglednica 3: Obrazložitev simbolov, na novo uporabljenih v enačbah od (8) do (16).....	29
Preglednica 4: Izračun višin točk z metodo regresijske ravnine.....	32
Preglednica 5: Koeficienti enačb ravnin, pripadajočih posameznima celicama.....	32
Preglednica 6: Razlage simbolov, uporabljenih v enačbah od (17) do (29).....	34
Preglednica 7: Območja za izbor ravnin z ustreznimi pari ravnin.....	37
Preglednica 8: Izračun višin točk z metodo dveh ravnin.....	38
Preglednica 9: Razlage simbolov, uporabljenih v enačbah (29) do (35).....	40
Preglednica 10: Izračun višin točk z linearno metodo.....	41
Preglednica 11: Primerjava rezultatov izračunov višin točk po posameznih metodah.....	42
Preglednica 12: SWOT analiza spletnega mesta »Planinske in kolesarske poti ter turni smuki z GPS podporo«.....	45
Preglednica 13: Kodiranje regij, kot je uporabljeno v SQL tabeli <i>tocke</i>	50
Preglednica 14: Razlaga uporabljenih HTML značk in ukazov v kodi obrazca za izbiranje...	52
Preglednica 15: Razlaga uporabljenih HTML značk in ukazov v kodi datoteke <i>izbira.php</i>	54
Preglednica 16: Razlaga uporabljenih PHP funkcij in ukazov v kodi datoteke <i>izbira.php</i>	55
Preglednica 17: Razredi zahtevnosti poti, ki so uporabljeni v SQL tabeli <i>poti</i>	63

KAZALO SLIK

Slika 1: Grafični vmesnik orodja PhpMyAdmin.	6
Slika 2: Prikaz naslovne strani v brskalniku.	8
Slika 3: Bločni diagram informacijske arhitekture spletnega mesta.	9
Slika 4: Struktura tabele <i>poti</i> (izsek iz PhpMyAdmin).	11
Slika 5: Struktura tabele <i>tocke</i> (izsek iz PhpMyAdmin).	12
Slika 6: Prikaz strani <i>gradnja_poti.php</i> v brskalniku.	13
Slika 7: Prikaz strani <i>rezultati.php</i> v brskalniku.	14
Slika 8: Potek obdelave GPS sledi od zajema do objave na spletnem mestu.	17
Slika 9: Primer zapisa GPX 1.0.	18
Slika 10: Bločni diagram poteka transformacije koordinat iz GPS v državni koordinatni sistem.	22
Slika 11: Primer zapisa DMR 12,5 v datoteki tipa XYZ.	23
Slika 12: Skica za izračun metode uteži sredinske točke.	25
Slika 13: Razporeditev testnih točk v celici DMR.	27
Slika 14: Skica za izračun višine točke na podlagi regresijske ravnine.	29
Slika 15: Skica za izračun višine točke z metodo dveh ravnin.	33
Slika 16: Tloris celice DMR s prikazanimi območji za izbor ustreznih ravnin in premicami.	37
Slika 17: Skica za izračun višine točke na podlagi presekov premic.	39
Slika 18: Grafični prikaz primerjave izračunanih višin točk po posameznih metodah.	43
Slika 19: Struktura dopolnjene tabele <i>tocke</i> (izsek iz PhpMyAdmin).	49
Slika 20: Grafični vmesnik za izbor začetne točke po atributih.	52
Slika 21: Obrazložitev naslova datoteke <i>izbira.php</i> , ki vsebuje spremenljivke z definiranimi vrednostmi.	54
Slika 22: Sestava SQL stavka za izbor točk po uporabnikovih kriterijih.	56
Slika 23: Prikaz strani <i>izbira.php</i> v brskalniku.	60
Slika 24: Struktura dopolnjene tabele <i>poti</i> (izsek iz PhpMyAdmin).	62
Slika 25: Prikaz strani <i>gradnja_poti.php</i> v brskalniku.	63

KAZALO PRILOG

PRILOGA A: Dokaz k linearni metodi v programu Mathematica

PRILOGA B: Programska koda datoteke *index.php*

PRILOGA C: Programska koda datoteke *izbira.php*

PRILOGA D: Programska koda datoteke *gradnja_poti.php*

PRILOGA E: Programska koda datoteke *rezultati.php*

PRILOGA F: Izsek SQL tabele *tocke*

PRILOGA G: Izsek SQL tabele *poti*

OKRAJŠAVE IN SIMBOLI

ASP	Active Server Pages
DMR	Digitalni model reliefa
DMV	Digitalni model višin
DTK	Državna topografska karta
GPL	General Public License
GPS	Global Positioning System
GPX	GPS Exchange Format
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IMG	Image File, Bitmap Graphic
JPG	Joint Photographic Experts Group
KML	Keyhole Markup Language
LAMP	Linux, Apache, MySQL, PHP
NCSA	National Center for Supercomputing Applications
PDF	Portable Document Format
PDOP	Positional Dilution of Precision
PHP	Hypertext Preprocessor
PLT	OziExplorer Plot File
REZI	Register zemljepisnih imen
SQL	Structured Query Language
SWOT	Strengths, Weaknesses, Opportunities, Threats
TIN	Triangulated Irregular Network
TTN	Temeljni topografski načrt
URL	Universal Resource Locator
USB	Universal Serial Bus
W3C	World Wide Web Consortium
WAMP	Windows, Apache, MySQL, PHP
WGS	World Geodetic System
WPT	Waypoint File
XAMPP	Apache, MySQL, PHP, Perl

XHTML Extensible Hypertext Markup Language

XML Extensible Markup Language

1. UVOD

Prostorski podatki so z razvojem informatike doživeli pravi razcvet. S pojavom svetovnega spleta se je obseg uporabnikov skokovito povečal na strani odjemalcev in ponudnikov. Množična uporaba prenosnih GPS sprejemnikov je privedla do tega, da so tudi laični uporabniki postali tvorci prostorskih podatkov, saj lahko s svojimi napravami zajemajo podatke o točkah in sledih v naravi. Vključevanje laikov v proizvodnjo in distribucijo prostorskih podatkov povzroča povečevanje obsega, posledično pa zmanjševanje kakovosti in celovitosti podatkov. Na medmrežju se je pojavila vrsta spletnih mest in aplikacij, preko katerih si uporabniki lahko izmenjujejo podatke, zajete s prenosnimi GPS napravami.

Spletno mesto »Planinske in kolesarske poti ter turni smuki z GPS podporo« je dober primer takšnega spletnega mesta, za razliko od mnogih pa so v njem objavljene GPS sledi kakovostno obdelane in v urejeni obliki ponujene uporabnikom. Z vdelanim izvornim in učinkovitim orodjem je mogoče združevanje posameznih sledi v mreže sledi. Od zajetja do objave na spletnem mestu gredo GPS sledi skozi vrsto postopkov, ki odpravijo napake zajema. Horizontalne koordinate točk sledi so na novo določene z zaslonko vektorizacijo, višine točk pa so določene na osnovi digitalnega modela reliefa (DMR) ločljivosti 12,5 metra. Del diplomske naloge je namenjen izpeljavi in primerjavi štirih različnih postopkov za določitev višine točke na podlagi DMR.

Obravnavano spletno mesto je povsem brezplačne narave. Brezplačna je tudi vsa tehnologija, na kateri spletno mesto temelji. Uporabljena programska jezika sta HTML in PHP, ki krmilita zbirko podatkov MySQL. Kot spletni strežnik služi program Apache. Tehnologija je zelo izpopolnjena in uporabna, kljub temu pa njena aplikacija terja kar nekaj znanja in časa. Predvsem slednji dejavnik predstavlja pogost vpliv na pomanjkljivo obdelane rešitve brezplačnih projektov in tudi obravnavano spletno mesto ni izjema. Možnih dopolnitev in izpopolnitev je veliko, nekatere izmed njih so izdelane in predstavljene v sklopu te diplomske naloge. Predvsem je dopolnjena baza podatkov in uporabniški vmesnik za učinkovito uporabo baze.

2. UPORABLJENE TEHNOLOGIJE

Programiranje za spletne strani je zelo razširjena panoga, za katero je razvitih veliko različnih tehnologij. Izbor tehnologije je odvisen od vrste, namena in obsega spletne strani, seveda pa ne gre zanemariti cene tovrstnega programja. Poleg plačljivih programskih paketov obstaja veliko brezplačnih, ki so večinoma zaščiteni pod splošno javno licenco (General Public Licence - GPL).

Spletno mesto, obravnavano v tej diplomski nalogi, je povsem brezplačne narave, zato bi bilo nesmiselno zanj uporabljati plačljivo tehnologijo. Avtorja spletnega mesta sta uporabila brezplačne tehnologije, ki so hkrati zelo uporabne. Uporabljeni programski jeziki s pripadajočo programsko podporo so HTML (Hypertext Markup Language), PHP (Hypertext Preprocessor) in SQL (Structured Query Language), vse skupaj pa povezuje in poganja strežniški program Apache. Naštete tehnologije so zelo razširjene med razvijalci, zato je na voljo tudi veliko podpore, ki je brezplačno dostopna na medmrežju (<http://www.wampserver.com/en>, 17.12.2007).

2.1 Strežnik Apache

Apache je zmogljiv strežniški program za izmenjavo in usmerjanje podatkov po protokolu HTTP (Hypertext Transfer Protocol). Zgrajen je bil leta 1994 na osnovi popravkov za svojega predhodnika NCSA HTTPd (National Center for Supercomputing Applications HTTP Daemon). Takrat je bil edina odprtokodna alternativa Netscapeovemu strežniku. Od tedaj je napredoval na vseh področjih in njegova priljubljenost že od leta 1996, ko je prvič prišel na vodilno mesto, še vedno narašča.

Spletni strežniški program Apache danes domuje v več kot polovici vseh spletnih strežnikov na svetu. Priljubljen je predvsem zaradi vzdržljivosti, prilagodljivosti in zanesljivosti. Uporaba Apache-ja je brezplačna (licenca GPL), kar v praksi pomeni, da na spletu obstaja veliko spletnih mest namenjenih podpori in izobraževanju in da so vse napake zelo hitro odpravljene. Skriptni programski jezik PHP je v strežniku podprt že od samega začetka (<http://www.e-uspeh.com/baza-znanja/entry/12/61>, 15.12.2007).

Apache deluje v veliki večini najbolj razširjenih operacijskih sistemov, kot so Windows, Linux, OS/2 in Unix, pa tudi v številnih bolj eksotičnih sistemih. V praksi je Apache skupek programov, modulov in nadzornih datotek, v katere lahko pišemo in tako (pre)nastavljamo strežnikove nastavitve, na primer, kaj naj strežnik naredi s posameznimi zahtevki za ogled določene spletne strani. Apache poskrbi, da je uporabnikov zahtevek izpolnjen in mu je poslana zelena spletna stran (Štancar, Klemen, 2002).

2.2 HTML, XHTML

HTML je označevalni skriptni jezik, v katerem so zapisane preproste spletne strani, pa tudi dinamičnih brez njega ne bi bilo možno izdelati. Kodo HTML, ki je lahko kombinirana z drugimi jeziki, brskalnik pretvori v berljive spletne strani. HTML je nekakšen temeljni kamen spleta in je dolgoleten standard za izdelavo spletnih strani. Za razvoj jezika skrbi konzorcij W3C (World Wide Web Consortium) in tudi določa pravila glede pisanja kode.

Ukazi, ki so v HTML imenovani tudi značke, so vedno zapisani med lomljenima oklepajema (< in >). Vse ostalo je izključno besedilo. Vseh ukazov je nekaj čez sto, vendar vsi brskalniki ne upoštevajo vseh, oziroma jih ne tolmačijo povsem enako. HTML je preprost označevalni skriptni jezik, zato mu manjkajo matematične in druge funkcije. Te pridobimo z vdelovanjem drugih programskih kod v HTML kodo (Štancar, Klemen, 2002).

XHTML (Extensible Hypertext Markup Language) je naslednji korak v razvoju jezikov, ki se uporabljajo za objavljanje v spletu in velja za nadomestilo jezika HTML. XHTML je bolj prilagodljiva različica hiperbesedila in je popolnoma združljiva z XML (Extensible Markup Language) standardom, kar omogoča boljšo integracijo različnih tehnologij. Dokumente v jeziku XHTML 1.0 je mogoče napisati tako, da so popolnoma združljivi z jezikom HTML od različice 4.0 naprej. V pripravi je tudi različica XHTML 2.0 (<http://www.s-sers.mb.edus.si/gradiva/w3/html/index.html>, 15.12.2007).

Oba jezika obstajata v treh različnih vrstah: Frameset, Transitional in Strict. Vrsta Strict je najboljše priporočena s strani konzorcija W3C za pisanje običajnih dokumentov. Vrsto

Transitional uporabimo, če želimo attribute, sloge in slogovne podlage medsebojno mešati, vrsta Frameset pa je namenjena uporabi okvirjev na spletni strani (<http://www.webstandards.org/learn/articles/askw3c/oct2003>, 15.12.2007).

XHTML je bolj univerzalen od svojega predhodnika, zaradi združljivosti s shemo XML je tudi lažji za učenje in za vzdrževanje. Za razliko od HTML pa ne dopušča nedoslednosti v skladnji.

2.3 PHP

PHP je strežniški skriptni jezik, ki ga lahko po želji pišemo med vrstice kode HTML. Prav tako lahko v skripto PHP vključimo HTML značke in besedilo. Večina skladnje (oblika programske kode) temelji na jezikih C, perl in java, dodanih pa je tudi veliko unikatnih funkcij. Jezik je nastal leta 1995, danes pa je v razvoju že šesta različica. PHP je splošno namemben, deloma predmeten programski jezik, katerega glavni namen je, da omogoča izdelavo dinamičnih spletnih strani.

Najbolj se PHP izkaže pri pisanju manj in srednje obsežnih spletnih strani. Je zelo prilagodljiv in združljiv s številnimi jeziki in aplikacijami v različnih operacijskih sistemih. Poleg spletnih strani lahko z njim delamo kakršnekoli aplikacije, tudi grafične vmesnike, igre in namenske programe. PHP je posebno močan v sodelovanju z raznimi zbirkami podatkov. Prav tako je prenosljiv med različnimi platformami operacijskih sistemov. Tako lahko npr. v Windows okolju razvijemo stran, ki vsebuje združena jezika PHP in SQL in jo objavimo v Linux ali Unix okolju. (Štancar, Klemen, 2002).

Besedna zveza strežniški skriptni jezik označuje dejstvo, da PHP deluje na strežniku. Ko uporabnik preko brskalnika pošlje strežniku zahtevo po neki informaciji, PHP pripravi podatke in jih vrne v obliki HTML, ki se izpiše v brskalniku. Za skriptne jezike je značilno, da je pomemben vrstni red pisanja stavkov. Strežnik po vrsti obdela vsako vrstico skripte posebej in rezultate pošlje nazaj uporabniku.

PHP je brezplačen (odprtokodni) jezik, zato je tudi množično uporabljen in predstavlja dobro konkurenco plačljivim primerljivim jezikom, kot je Microsoftov ASP (Active Server Pages) (http://www.w3schools.com/php/php_intro.asp, 15.12.2007).

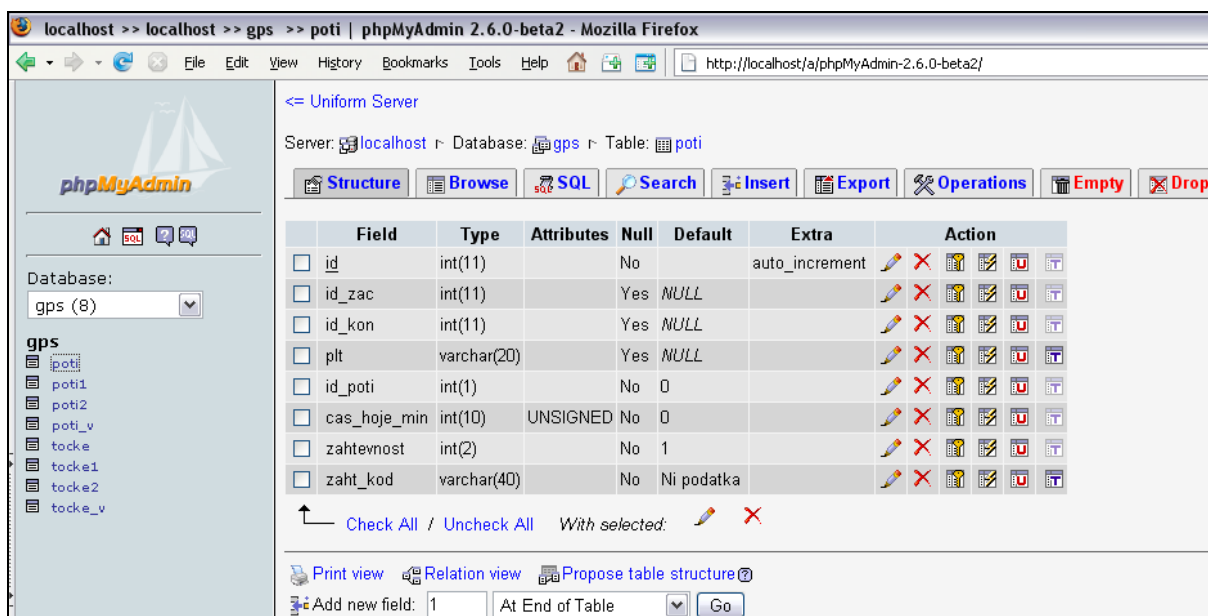
2.4 SQL

Strukturiran poizvedovalni jezik (SQL) je oblikovan za urejanje podatkov v relacijskih podatkovnih bazah. Z njim lahko ustvarimo in brišemo podatke, jih spreminjamo in kontroliramo dostop do njih. Prvo različico SQL so razvili pri IBM-u že v 70-ih letih prejšnjega stoletja. Danes poznamo mnogo različic jezika, ki jih razvijajo različne družbe. Dodani so različni postopki, podpora posebnim podatkovnim tipom, kontrolni stavki in drugo (<http://en.wikipedia.org/wiki/SQL>, 17.12.2007).

Najbolj razširjena različica jezika je MySQL, ki je klasična zbirka podatkov, združljiva z različnimi operacijskimi sistemi. Lahko deluje povsem samostojno, kot podatkovni strežnik, ali pa v kombinaciji z drugimi tehnologijami. MySQL je leta 1992 razvilo švedsko podjetje MySQL AB in ga ponudilo na trg pod pogoji uporabe javne licence GNU (<http://www.gnu.org>, 17.12.2007). Napisan je v jezikih C in C++.

Jezik je izdelan tako, da je skladnja ukaznih stavkov zelo podobna govorjeni angleščini. Uporabnik večinoma uporablja do deset osnovnih ukazov, zato učenje jezika ni ravno zapleteno. Potrebno pa je dobro razumevanje delovanja relacijske baze podatkov (Štancar, Klemen, 2002).

Delo z jezikom MySQL si najbolj olajšamo z uporabo orodja za administracijo podatkovne baze MySQL. Orodje PhpMyAdmin je napisano v jeziku PHP, upravljamo ga preko spletnega brskalnika (slika 1). Nabor funkcij vsebuje nujne ukaze, ki so urejeni v pregleden grafičen vmesnik. Poleg pregledovanja in manipuliranja s podatki omogoča tudi uvoz in izvoz podatkov v različnih formatih.



Slika 1: Grafični vmesnik orodja PhpMyAdmin.

2.5 Strežniški programski svežnji, WAMP

Vzpostavitev medmrežnega strežnika zahteva veliko znanja in izkušenj o delovanju strežniških sistemov. V zgornjih poglavjih opisane tehnologije so vse brezplačne in dostopne preko spleta. Vendar, če bi želeli sami vzpostaviti Apache strežnik s podporo tehnologijama PHP in SQL, bi se pošteno namučili ob nastavljanju mnogih parametrov. Kompleksna arhitektura takšnih sistemov je privedla do razvoja programskih svežnjev, v katerih je združenih in pravilno nastavljenih več tehnologij. S preprosto namestitvijo svežnja na računalnik vzpostavimo strežnik, ki že vsebuje nastavljene podpore različnim jezikom in podatkovnim bazam.

Za spletno mesto »Planinske in kolesarske poti ter turni smuki z GPS podporo« sta avtorja uporabila programski sveženj tipa WAMP. W v kratiki označuje operacijski sistem Windows, kateremu je strežnik namenjen, A pomeni strežnik Apache, M pomeni MySQL in P pomeni programski jezik PHP. Poleg naštetih svežnjev podpira še množico tehnologij, ki so najbolj pogosto uporabljene na spletnih strežnikih (<http://www.wampserver.com/en>, 17.12.2007).

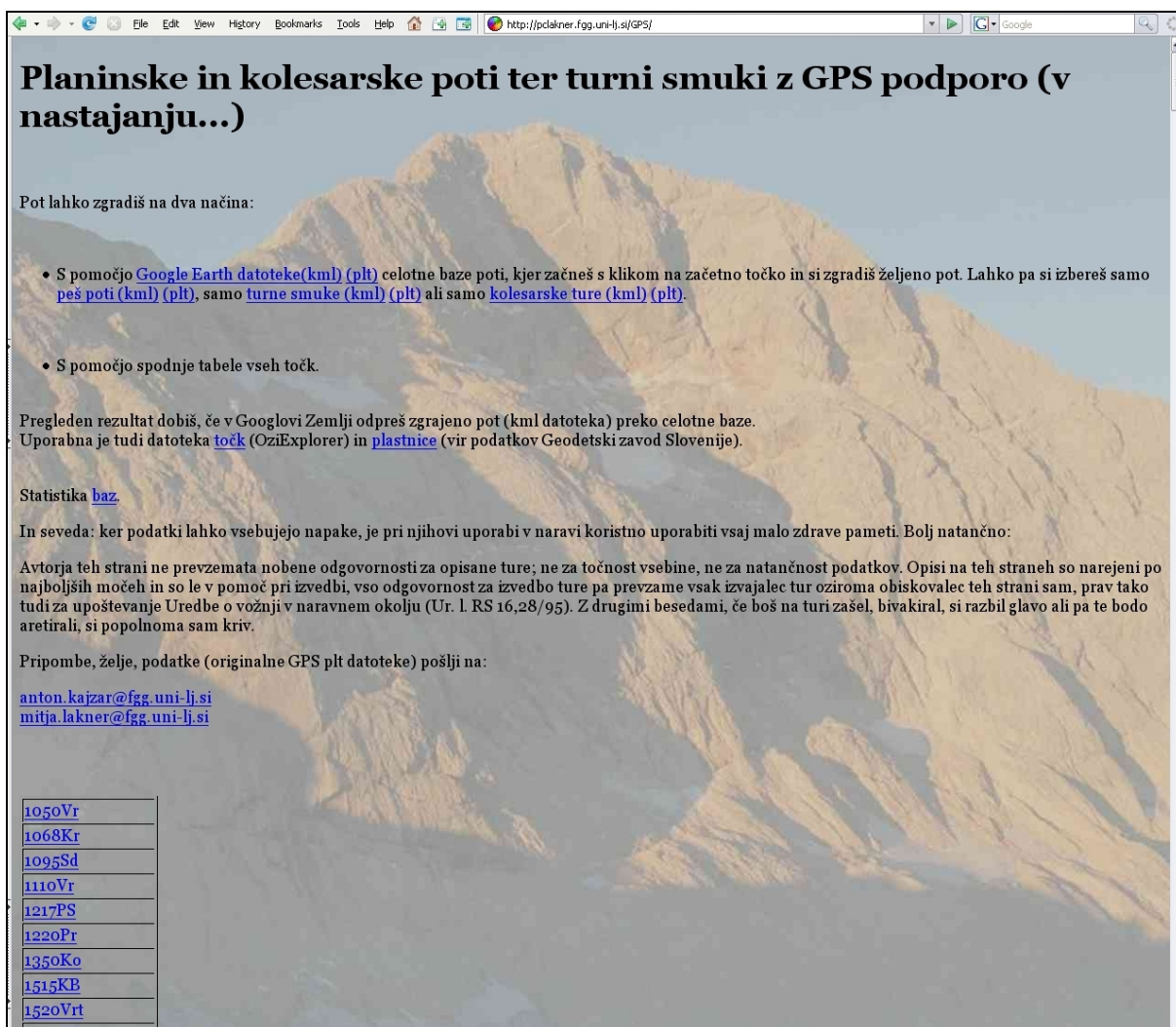
Podobna priljubljena svežnja sta še LAMP in XAMPP. Prvi je namenski za okolja Linux, drugi pa temelji na štirih različnih operacijskih sistemih, to so MS Windows, Linux, Sun Solaris in Mac OS X.

2.6 Vtičniki, PHPlot

Jezik PHP je zelo obsežen, z uporabo vtičnikov (»plug-in-ov«) pa se obseg in namembnost še povečata. Obravnavano spletno mesto uporablja vtičnik PHPlot, ki omogoča dinamično tvorjenje grafičnih prikazov v okolju PHP. Najbolj se orodje obnese pri risanju grafov, na spletnem mestu je uporabljeno za izris vertikalnega profila GPS sledi. PHPlot podpira tudi uvoz slik za ozadje, združevanje več grafov v eno sliko in shranjevanje slik na trdi disk z namenom uvoza v PDF dokumente (<http://freshmeat.net/projects/phplot>, 17.12.2007). Podoben vtičnik je GraPHPite.

3. OBSTOJEČE STANJE SPLETNEGA MESTA »PLANINSKE IN KOLESARSKE POTI TER TURNI SMUKI Z GPS PODPORO«

Spletno mesto GPS sledi »Planinske in kolesarske poti ter turni smuki z GPS podporo« je delo avtorjev doc. dr. Mitje Laknerja, univ. dipl. mat. in g. Antona Kajzarja, ki je tehniški sodelavec na Fakulteti za gradbeništvo in geodezijo Univerze v Ljubljani. Osnovna ideja spletnega mesta je na enem mestu zbrati GPS sledi, katerih večino sta avtorja sama zajela na izletih po Sloveniji in bližnji okolici. Večina GPS sledi je zajetih na pohodih po slovenskih gorah in hribovjih, nekatere so zajete na turnih smukah, ostale pa na kolesarskih izletih po hribovjih in ravninah.



Slika 2: Prikaz naslovne strani v brskalniku.

Namen objave teh sledi na spletu je, da lahko uporabniki spleta shranjene sledi ponovno uporabijo na sebi primeren način.

3.1 Programska osnova

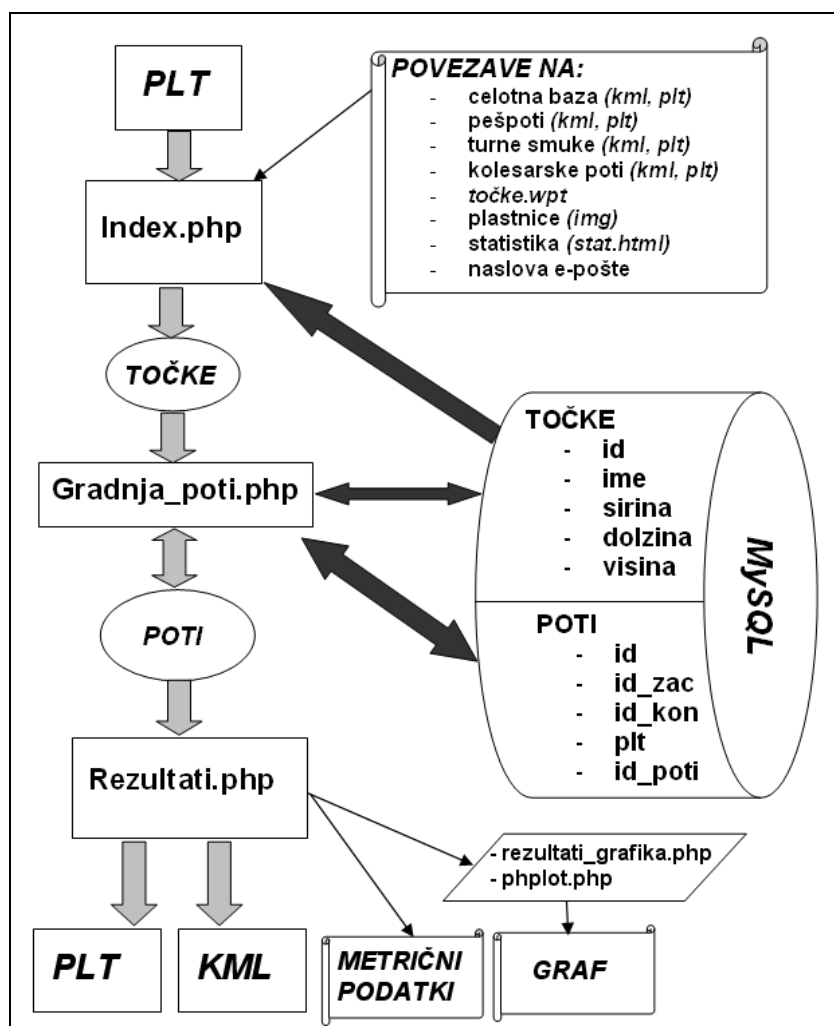
Obravnavano spletno mesto temelji na WAMP programskem svežnju Uniform Server različice 3.1.1. (<http://www.uniformserver.com>, 11.12.2007), katerega glavna odlika je kompaktnost. Sveženj zahteva majhno pomnilniško kapaciteto in vse datoteke so lahko shranjene v skupni mapi, kar pomeni, da lahko celoten strežnik zaženemo iz prenosnega pomnilniškega medija (npr. USB ključ). Sveženj podpira programske jezike PHP 5.0,

ActivePerl 5.8 in MySQL 4.0, kateremu pripada tudi uporabniški vmesnik PhpMyAdmin 2.6. Sveženj povezuje strežniški program Apache 2.0.

3.2 Struktura spletnega mesta

Spletno mesto »Planinske in kolesarske poti ter turni smuki z GPS podporo« sestavlja več spletnih strani, ki so napisane v jeziki HTML in PHP. Vsaka spletna stran, ki se izpiše v brskalniku, temelji na svoji osnovni datoteki, v katero so lahko vključene tudi ostale datoteke, bodisi kot hiperpovezave ali kot vgrajeni elementi (npr. slika za ozadje v JPG datoteki).

3.2.1 Informacijska arhitektura



Slika 3: Bločni diagram informacijske arhitekture spletnega mesta.

Dobro načrtovana informacijska arhitektura ali arhitektura spletnega mesta je bistven element za učinkovito delovanje in preprostost uporabe spletnega mesta. Informacijska arhitektura določa postavitev posameznih elementov in interakcijo med njimi. Za obravnavano spletno mesto je arhitektura opisana z bločnim diagramom v sliki 3. Delovanje posameznih datotek je podrobneje opisano v nadaljnjih podpoglavjih.

3.2.2 Mape in datoteke

Razporeditev map in datotek na strežniku je prilagojena naravi delovanja strežnika, vseeno pa ima razvijalec precej proste roke pri oblikovanju drevesne strukture map. Za obravnavano spletno mesto, ki po obsegu ni veliko, je razporeditev elementov na strežniku dokaj preprosta.

Spletno mesto se nahaja na strežniku s spletnim naslovom *http://pclakner.fgg.uni-lj.si*, ki poleg obravnavanega zajema še vrsto drugih podatkov in strani. Vse spletne strani se na strežniku nahajajo v mapi *~\UniServer3_2\diskw\www*, katero strežnik v brskalniku prikazuje pod zgornjim naslovom. Vse datoteke spletnega mesta »Planinske in kolesarke poti ter turni smuki z GPS podporo«, razen tistih, ki predstavljajo SQL bazo, so zbrane v podmapi *~\UniServer3_2\diskw\www\GPS* (na spletu se mapa nahaja na naslovu *http://pclakner.fgg.uni-lj.si/GPS*). Datoteke formata PLT (OziExplorer Plot File), ki služijo kot vhodni podatki, so shranjene v mapi *~\UniServer3_2\diskw\www\GPS\plt*. Datoteke SQL baze se na strežniku nahajajo v mapi *~\UniServer3_2\diskw\usr\local\mysql\data\gps*.

3.3 Vhodni podatki

Glavni vhodni podatki za spletno mesto so izhodni podatki iz GPS sprejemnika. To so datoteke formata GPX (GPS Exchange Format), v katerih so zapisane GPS sledi izletov. Datotečni format GPX je izmenjevalni format, ki temelji na standardu XML. Datoteke so nato pretvorjene v format PLT, ki je bolj primeren za obdelavo z osebnim računalnikom, podatki znotraj njih pa prilagojeni na način, opisan v 4. poglavju te diplomske naloge. Poleg datotek so vhodni podatki še priporočen način uporabe poti ter ime in lokacija v prostoru vsake začetne in končne točke posamezne sledi.

3.3.1 Primeren GPS sprejemnik

GPS sledi so zajete s pomočjo primerne ročnega GPS sprejemnika. To so sprejemniki, katerih osnovni namen je zajem podatkov o lokaciji z manjšo natančnostjo kot pri geodetskih meritvah. Večinoma se uporabljajo za navigacijo. Njihove pomembne značilnosti so, da so prenosni, priročni in enostavni za uporabo. Sprejemnik mora izpolnjevati naslednje pogoje:

1. ima možnost zajema sledi,
2. ima dovolj spomina za shranjevanje zajete sledi in
3. ima možnost prenosa zajete sledi na osebni računalnik (Golež, 2006).

3.4 Struktura SQL podatkovne baze

Urejene sledi so shranjene v mapi na strežniku, do katere s pomočjo PHP skript dostopamo na podlagi SQL baze, imenovane *GPS*. Baza *GPS* je sestavljena iz dveh tabel, *poti* in *tocke*. Povezovalni identifikatorji obeh tabel so polje *id* v tabeli *tocke* in polji *id_zac* ter *id_kon* v tabeli *poti*. Gre namreč za iste točke.

Strukturo tabele *poti* sestavljajo naslednja polja:

- identifikacijska številka sledi (*id*),
- začetna točka sledi (*id_zac*),
- končna točka sledi (*id_kon*),
- ime PLT datoteke (*plt*) in
- priporočen način uporabe poti (*id_poti*), ki nam pove ali je pot primerna za hojo, kolesarjenje, turno smuko ali neko kombinacijo naštetih načinov.

	Field	Type	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>id</u>	int(11)		No		auto_increment
<input type="checkbox"/>	id_zac	int(11)		Yes	NULL	
<input type="checkbox"/>	id_kon	int(11)		Yes	NULL	
<input type="checkbox"/>	plt	varchar(20)		Yes	NULL	
<input type="checkbox"/>	id_poti	int(1)		No	0	

Slika 4: Struktura tabele *poti* (izsek iz PhpMyAdmin).

Ostali vhodni podatki so vezani na posamezne začetne in končne točke sledi. Te točke so v isti podatkovni bazi shranjene v tabeli *tocke*. Polja v tej tabeli so:

- identifikacijska številka točke (*id*),
- skrajšano ime točke (*ime*),
- geografska širina na WGS84 elipsoidu (*sirina*),
- geografska dolžina na WGS84 elipsoidu (*dolzina*) in
- normalna ortometrična višina točke (*visina*).

	Field	Type	Attributes	Null	Default	Extra
<input type="checkbox"/>	id	int(11)		No		auto_increment
<input type="checkbox"/>	ime	varchar(30)		Yes	NULL	
<input type="checkbox"/>	sirina	float(8,6)		Yes	NULL	
<input type="checkbox"/>	dolzina	float(8,6)		Yes	NULL	
<input type="checkbox"/>	visina	float		Yes	NULL	

Slika 5: Struktura tabele *tocke* (izsek iz PhpMyAdmin).

3.5 Obdelava podatkov

Glavna posebnost baze je, da lahko uporabnik zgradi poljubno kombinacijo sledi, povezanih preko skupnih točk, ki so začetne in končne točke posamezno shranjenih sledi.

Na vstopni strani portala, imenovani *index.php*, si uporabnik iz tabele vseh točk izbere tisto, na kateri želi začeti gradnjo poti. Točke v tabeli so s pomočjo PHP skripte izbrane iz SQL baze. Ob izboru hiperpovezava preusmeri uporabnika na stran *gradnja_poti.php*, kjer so v tabeli razporejene možne poti iz izbrane točke. Ob izboru posamezne poti skripta iz SQL baze izbere vse tiste poti, ki imajo skupno končno točko prejšnje poti in jih ponudi uporabniku za nadaljnji izbor. Možnosti kombiniranja so poljubne v okviru povezanih poti.

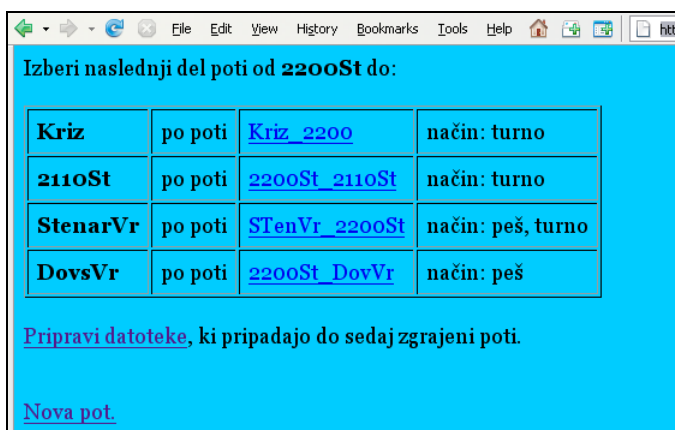
Uporabnik lahko začne gradnjo poti tudi na bolj pregleden način, z uporabo programa Google Earth (<http://earth.google.com>, 14.10.2007). V ta namen si mora predhodno iz strani *index.php* pridobiti KML (Keyhole Markup Language, http://en.wikipedia.org/wiki/Keyhole_Markup_Language, 14.10.2007) datoteko celotne baze poti (*baza.kml*) ali baze poti po posameznih načinih uporabe (peš, s kolesom ali turno). Ko odpre datoteko v Google Earth,

lahko grafično izbere začetno točko. S klikom nanjo se odpre pogovorno okno s hiperpovezavo na spletno mesto, na stran *gradnja_poti.php*.

3.5.1 Datoteka *index.php*

Ta datoteka je začetna (indeksna) stran spletnega mesta in se v brskalniku odpre ob navedbi spletnega naslova *http://pclakner.fgg.uni-lj.si/GPS* (slika 2). Strežniški program jo samodejno prepozna po imenu (beseda »index«), zato imena datoteke ni potrebno eksplicitno navajati v spletnem naslovu. Stran vsebuje naslov spletnega mesta, nagovor uporabniku, povezave na druge spletne strani in datoteke (opisane v sliki 3) ter tabelo točk. Tabela vsebuje polja z imeni vseh točk v bazi. Narejena je s pomočjo PHP skripte, ki črpa imena vseh točk iz SQL baze. Vsako polje je hkrati tudi hiperpovezava na datoteko *gradnja_poti.php*. Ob izboru točke se identifikacijska številka točke zapiše v spletni naslov datoteke *gradnja_poti.php*. Hkrati se na strežniku odpre nova seja, ki pripada le enemu trenutnemu uporabniku. Če strežnik hkrati uporablja več uporabnikov, se ti med seboj ne motijo, saj vsak izvaja gradnjo poti znotraj svoje seje.

3.5.2 Datoteka *gradnja_poti.php*



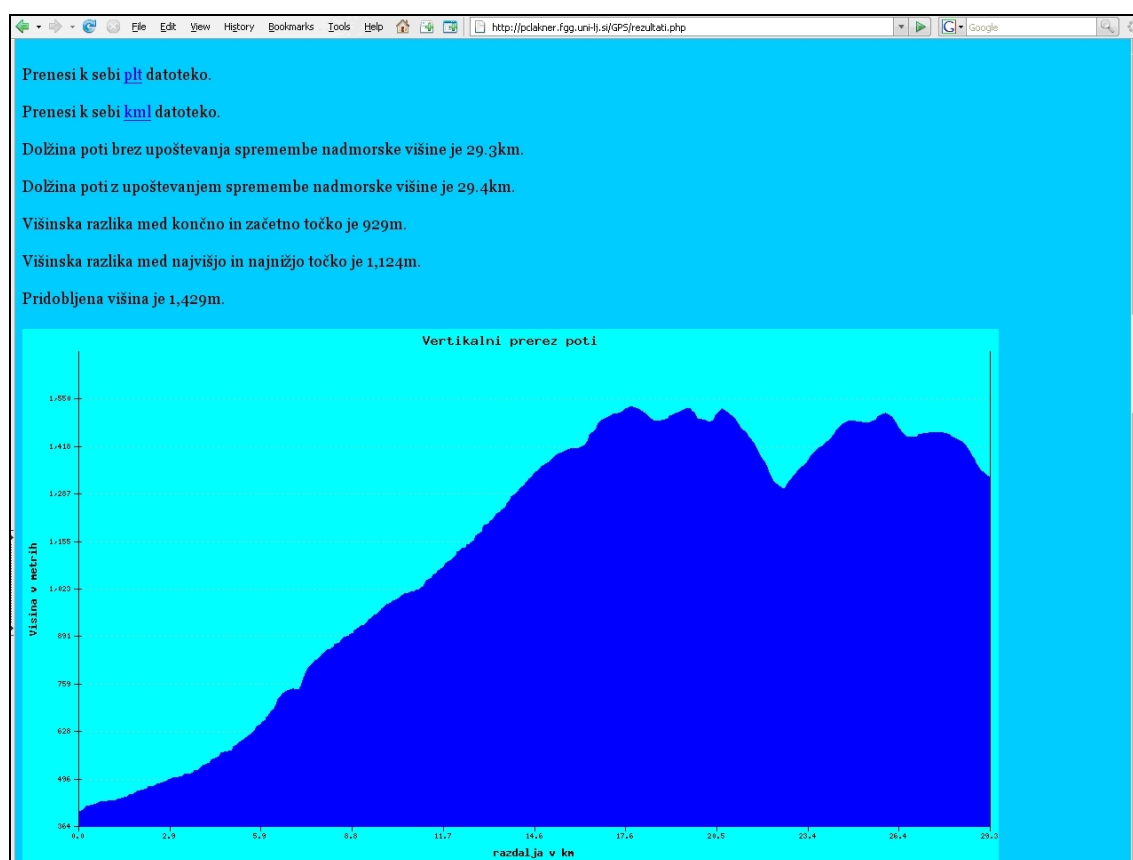
Slika 6: Prikaz strani *gradnja_poti.php* v brskalniku.

Datoteka *gradnja_poti.php* vsebuje v PHP vgrajeno polje `$_GET`, ki iz spletnega naslova potegne identifikacijsko številko točke, izbrane v tabeli datoteke *index.php*. Iz SQL baze nato izbere poti, katerih začetna ali končna točka sovпада z izbrano točko. Spletna stran, ki pripada

datoteki, nam torej izpiše izbrano začetno točko in v tabeli ponudi možne poti do posameznih končnih točk. Kot atribut poleg poti pripne v tabelo še priporočen način uporabe poti. Gradnjo poti lahko izvajamo teoretično v nedogled, če smo izbrali najmanj eno pot, pa lahko kadarkoli pripravimo datoteke, ki pripadajo do tedaj zgrajeni poti.

3.5.3 Datoteka *rezultati.php*

Zaključna datoteka v krogu gradnje poti nam ponudi izhodne podatke. Datoteke z izhodnimi podatki tvori na osnovi postopnega branja (funkcija `file_get_contents`) iz vhodnih PLT datotek in postopnega zapisovanja v izhodne datoteke. Zapisovanje poteka za vsako vrstico posebej (funkcija `fwrite`) po vrstnem redu, določenim z začetno in končno točko odseka poti. Ime izhodne datoteke je narejeno na podlagi IP naslova uporabnika, ki je v trenutni seji izgradil pot. Datoteka postreže tudi z metričnimi podatki o sledi in z grafičnim prikazom vertikalnega prereza zgrajene poti.



Slika 7: Prikaz strani *rezultati.php* v brskalniku.

3.6 Rezultati

Programske skripte v PHP jeziku omogočajo uporabniku združiti več sledi v eno samo povezano sled, ki jo lahko kot izhodni rezultat izvozi v datotekah formatov PLT ali KML. Poleg omenjenih datotek so na razpolago še metrični podatki, računsko pridobljeni iz vhodnih PLT datotek in grafični prikaz vertikalnega prereza zgrajene poti. Grafični prikaz je narejen s pomočjo programskega vtičnika PHPlot. Prikaz vsebuje graf višin v metrih glede na horizontalno razdaljo poti v kilometrih od začetne točke gradnje poti. Graf je opremljen z naslovom, poimenovanji obeh osi in številskimi vrednostmi na oseh. Metrični podatki so dolžina poti brez upoštevanja spremembe nadmorske višine, dolžina poti z upoštevanjem spremembe nadmorske višine, višinska razlika med končno in začetno točko, višinska razlika med najvišjo in najnižjo točko ter pridobljena višina na poti.

3.7 Uporaba

Možnosti uporabe izhodnih podatkov so praktično neomejene, v osnovi pa jo lahko glede na potencialne uporabnike razdelimo na dva sklopa.

Prvi in osnovni sklop zadovoljuje uporabnike, ki želijo sledi uporabiti na lastnih izletih s pomočjo svoje GPS naprave. Ti uporabniki lahko naredijo svojo izhodno datoteko formata PLT, ki jo nato s pomočjo primerne programa naložijo v spomin GPS sprejemnika, kateri jih nato na izletu vodi po naloženih sledih.

Drugi sklop nudi uporabnikom vrsto različnih informacij o samih poteh. Poleg omenjenih računsko pridobljenih metričnih podatkov lahko izhodne datoteke pregledujejo na svojih računalnikih. Datoteke vrste PLT lahko pregledujemo s programom OziExplorer (<http://www.ozieplorer.com>, 18.10.2007), datoteke KML pa s programom Google Earth.

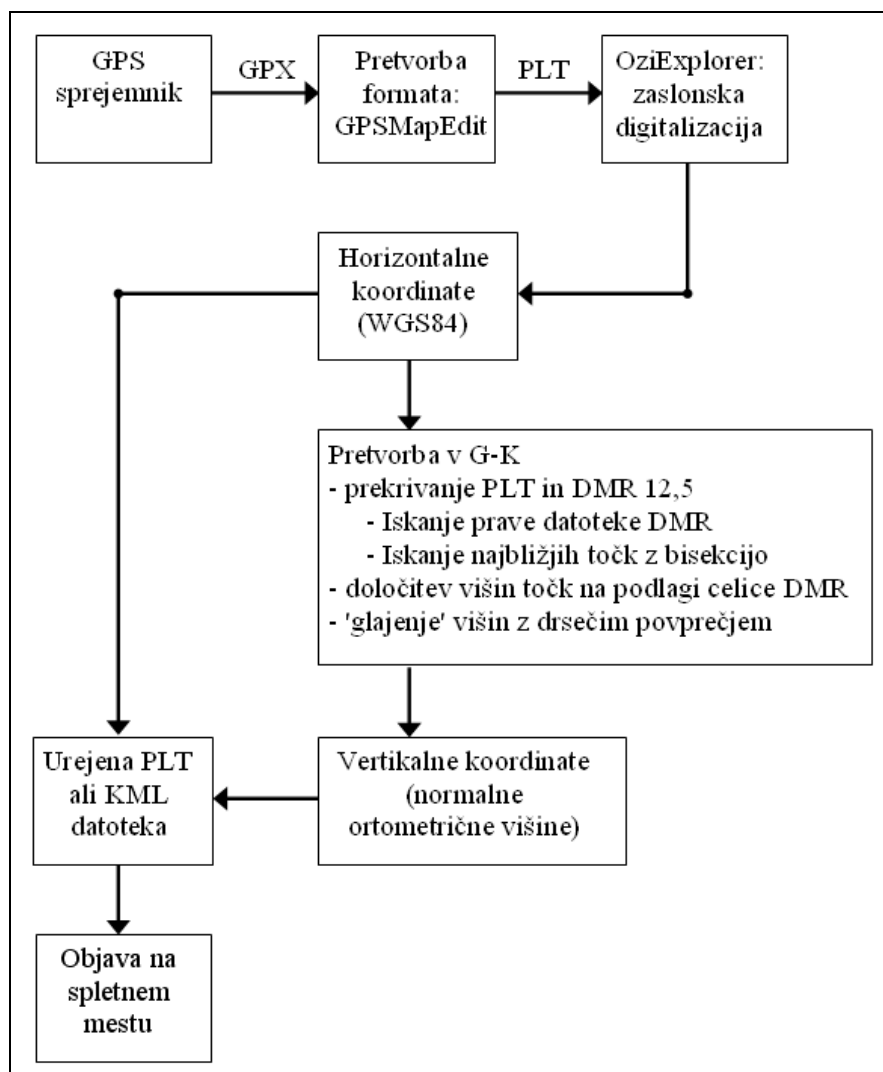
4. PRILAGAJANJE VHODNIH DATOTEK

GPS sledi, zajete z ročnimi GPS sprejemniki, bolj ali manj natančno ponazarjajo potek poti v naravi. Natančnost zajete poti je odvisna predvsem od ovir, ki onemogočajo neposreden dostop do signalov s satelitov GPS, in od konfiguracije terena (soteska, zaraščenost terena, visoke stavbe, itd) (Golež, 2006). Ovine zmanjšajo število satelitov, od katerih je mogoče s sprejemnikom zaznati signal. Število satelitov je odvisno tudi od časovnega termina zajema in geografske širine. Poleg števila satelitov na obzorju je pomembna tudi njihova geometrijska razporeditev, ki jo lahko predstavimo z mero PDOP (Positional Dilution Of Precision). Položajne napake zajetih točk se pojavljajo na posameznih točkah in v celih skupinah točk. Lahko se zgodi, da je signal moten do tolikšne mere, da je zajem točk povsem nemogoč. V takšnih primerih je zajeta GPS sled prekinjena.

Na kakovost zajema vpliva tudi tip sprejemnika in uporabniška nastavitvev načina zajema. Večina sprejemnikov omogoča nastavitvev načina zajema na časovni ali dolžinski interval. Z večjo gostoto točk lahko dosežemo boljšo aproksimacijo dejanske poti, vendar moramo pričakovati tudi večji raztros rezultatov.

Spletno mesto »Planinske in kolesarske poti ter turni smuki z GPS podporo« manipulira s shranjenimi GPS sledmi na več različnih načinov, zato morajo biti sledi kot vhodni podatki usklajene na več različnih nivojih. Za možnost kombiniranja poti morajo začetne in končne točke sosednjih sledi sovpadati. Za kakovostne računsko izvedene metrične podatke na podlagi sledi (poglavje 3.6) morajo le te kar najboljše ponazarjati pravi potek poti v naravi. Temu idealu se lahko približamo tako, da GPS sledi preuredimo na načine, opisane v naslednjih poglavjih.

GPS sledi gredo pred objavo na spletnem mestu skozi številne preureditve, ki so deloma avtomatizirane, deloma pa jih je treba izvesti ročno. Naslednja shema predstavlja potek obdelave sledi od zajema do objave na spletnem mestu.



Slika 8: Potek obdelave GPS sledi od zajema do objave na spletnem mestu.

4.1 Prenos na osebni računalnik, pretvorba formata in zaslonska vektorizacija

Na GPS napravi zajeto pot prenesemo na osebni računalnik. Prenos podatkov se razlikuje glede na sprejemnik, navadno se izvrši prek podatkovnega kabla ali brezžičnega vmesnika (Bluetooth ali IR). Ne glede na znamko naprave se lahko zajeti podatki vedno shranijo v format GPX (Golež, 2006). GPX je z XML standardom usklajen podatkovni format, namenjen izmenjevanju GPS podatkov (točk, sledi in poti) med aplikacijami in spletnimi storitvami. Shema GPX 1.1 je bila objavljena leta 2004 in je »de-facto« XML standard za lahko izmenjavo GPS podatkov že od prve objave GPX 1.0 v letu 2002 (<http://www.topografix.com/gpx.asp>, 23. 11. 2007).

```

1  <?xml version="1.0"?>
2  <gpx
3    version="1.0"
4    creator="ExpertGPS 1.1 - http://www.topografix.com"
5    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6    xmlns="http://www.topografix.com/GPX/1/0"
7    xsi:schemaLocation="http://www.topografix.com/GPX/1/0
8    http://www.topografix.com/GPX/1/0/gpx.xsd">
9    <time>2002-02-27T17:18:33Z</time>
10   <bounds minlat="42.401051" minlon="-71.126602" maxlat="42.468655"
11     maxlon="-71.102973"/>
12   <wpt lat="42.438878" lon="-71.119277">
13     <ele>44.586548</ele>
14     <time>2001-11-28T21:05:28Z</time>
15     <name>5066</name>
16     <desc><![CDATA[5066]]></desc>
17     <sym>Crossing</sym>
18     <type><![CDATA[Crossing]]></type>
19   </wpt>
20

```

Slika 9: Primer zapisa GPX 1.0.

Datoteko v formatu GPX moramo za nadaljnjo uporabo pretvoriti v format PLT, ki ga uporabljata program OziExplorer in obravnavano spletno mesto. Pretvorba je mogoča z veliko množico programov, eden izmed njih je program GPSMapEdit, ki poleg omenjenih formatov obvladuje še druge formate, ki se uporabljajo v različnih programskih paketih, povezanih s prostorsko informatiko.

Datoteka v formatu PLT še vedno vsebuje vse napake meritev. S programom OziExplorer izvedemo zaslonsko vektorizacijo sledi. Za boljšo preglednost v programu odpremo ustrezno kartografsko podlago (npr. DTK 25) in uvozimo zajeto sled. Novo sled ustvarimo povsem ročno, lomne točke nove sledi postavljamo na značilne lomne točke zajete sledi. Kartografska podlaga nam pri tem služi samo za orientacijo, prilagoditev sledi izvršimo zgolj na podlagi zajete sledi. S tem postopkom dosežemo, da razgibano in morda mestoma prekinjeno sled z veliko množico točk (količina je odvisna od nastavljenega intervala zajema položaja), zajetih z GPS sprejemnikom, naredimo zvezno in bolj točno. Z redukcijo količine podatkov tudi olajšamo nadaljnjo obdelavo. Namen vektorizacije je pridobitev urejenih horizontalnih koordinat sledi v smiselnem obsegu.

4.2 Določanje višin točk

Na območju Slovenije nadmorsko višino določamo glede na srednji nivo morske gladine v Trstu (normalna ortometrična višina). Tam je nivo morja približno 48 metrov višji od WGS84 elipsoida. Večina novejših prenosnih GPS sprejemnikov upošteva razliko med obema višinama in kaže pravilno (normalno ortometrično) višino, neodvisno od izbranega datuma ali projekcije. 2. maja 2000 je ameriška vlada odpravila selektivni dostop do GPS signala in natančnost GPS sprejemnikov se je znatno izboljšala, še posebej pri določanju višin. Če sprejemnik vidi dovolj satelitov, je točnost izmerjene višine v okviru nekaj metrov. Z odpravo selektivnega dostopa je GPS postal vsaj tako točen kot ročni višinomeri, ki delujejo na podlagi zračnega tlaka (http://www2.arnes.si/~gljsentvid10/GPS_sistemi.html, 24.11.07).

4.2.1 Digitalni model reliefa

Natančnost določitve višine nekaj metrov je zadovoljiva, vendar z metodo GPS nedosegljiva, če so prisotne ovire, ki zakrivajo in odbijajo signal. V gorskem svetu so ovire zelo pogoste, zato je smiselno GPS opazovanja kombinirati z boljšimi podatki o višinah in o samem reliefu. Najboljša urejena zbirka podatkov o reliefu v Sloveniji je DMR 5 (Digitalni model reliefa ločljivosti 5 metrov). Model je bil izdelan leta 2007, s predpisanim srednjim odklonom med 1 in 3 metri. (http://www.geodetski-vestnik.com/50/3/gv50-3_472-480.pdf, 19.1.2008). Nekoliko manj natančna tovrstna zbirka je DMR 12,5, ki je bil dokončan v letu 2006. V času nastajanja obravnavanega spletnega mesta DMR 5 še ni obstajal, zato sta avtorja pridobila pravico do uporabe DMR 12,5. Površina te zbirke obsega 2,7-kratno velikost Slovenije (55.000 km²). Ocenjena natančnost modela je 3,2 m za območje vse Slovenije, in sicer 1,1 m za ravnine, 2,3 m za gričevja, 3,8 m za hribovja in 7,0 m za gorovja, pri ločljivosti 12,5 m (<http://iaps.zrc-sazu.si/index.php?q=sl/node/70>, 25.11.07).

Digitalni model reliefa (DMR) razumemo kot digitalni opis oblikovanosti zemeljskega površja. Definicija ne vsebuje le prikaza površja samega, ampak tudi njegov opis z nakloni, ekspozicijo ter plastnicami, padnicami, točkami vrhov ter z drugimi značilnimi črtami in točkami. Nekoliko bolj skromen digitalen opis površja je zbirka DMV (Digitalni model višin), ki je enostaven zapis v dvorazsežno kvadratasto celično mrežo z višinami, zapisanimi kot

atributi. Pojem DMR pomeni precej več kot DMV, kljub razliki pa se je za izraza DMV in DMR med strokovnjaki in uporabniki udomačila poenostavljena kratica – DMR (<http://www.geodetski-vestnik.com/46/gv46-4.pdf>, 11.12.07). V nadaljnjem besedilu diplomske naloge bom tudi sam uporabljal ta izraz.

Podatke DMR lahko uporabimo za določitev višine poljubne točke znotraj obsega modela, torej na ozemlju celotne Slovenije. Ker obravnavano spletno mesto vsebuje GPS sledi na območju Slovenije, je zbirka DMR 12,5 primerna za določitev višin posameznih točk znotraj zbranih GPS sledi. Obdelavo podatkov za spletno mesto bi lahko nadgradili z uporabo DMR 5, vendar pa končni rezultat kljub večji gostoti in boljši natančnosti zbirke ne bi bil bistveno boljši.

Višino poljubne točke lahko določimo na podlagi DMR na več različnih načinov. Najbolj kompleksne metode so razvite z namenom določitve višin velike množice točk (oblaka točk) z veliko natančnostjo, torej z namenom aproksimacije zemeljskega površja na podlagi DMR. Ena izmed metod, ki to omogoča, je kriging.

4.2.2 Kriging

Kriging je skupina geostatističnih tehnik za interpolacijo vrednosti naključnega polja (npr.: višina točke na površju kot funkcija geografske lokacije) na neopazovani lokaciji, na podlagi znanih vrednosti polj za bližnje lokacije (<http://en.wikipedia.org/wiki/Kriging>, 29.11.2007). Geostatistične metode generirajo ploskve vključujoč statistične lastnosti meritev. Metode temeljijo na statistiki, z njihovo pomočjo lahko aproksimiramo ploskve in hkrati ocenimo natančnost izračunanih količin (<http://www.esri.com/library/whitepapers/pdfs/geostat.pdf>, 29.11.2007).

Metodo kriging lahko najdemo v aplicirani obliki v mnogih programskih paketih, namenjenih geostatistični obdelavi podatkov, kot so ESRI ArcGIS, Surfer (Golden Software) in dodatek za AutoCAD, Quicksurf.

Kriging temelji na konceptualnem razumevanju statističnih metod. Za namen diplomske naloge bi bila obravnava metode preveč obsežna, po osebnem pogovoru z viš. pred. mag. Daliborjem Radovanom sem se odločil, da je ne bom obravnaval. Nekaj preprostejših primerov za določitev višine na podlagi DMR pa sem analiziral v poglavju 4.4.

4.3 Prekrivanje datoteke PLT in baze DMR 12,5

Digitalni model reliefa DMR 12,5 je zbirka horizontalnih (x, y) in vertikalnih (z) kartezičnih koordinat, zapisanih v državnem koordinatnem sistemu D48 (Gauss – Kruegerjeva projekcija na lokalnem Besselovem elipsoidu), GPS sledi pa so zapisane v elipsoidnih (geografskih) koordinatah (φ, λ, h) na globalnem elipsoidu WGS 84. Ta sistem pogosto imenujemo GPS koordinatni sistem. Prekrivanje različnih slojev podatkov je mogoče, če so zapisani v istem koordinatnem sistemu (geodetskem datumu). Če hočemo GPS sledi združiti z DMR, jih moramo pretvoriti v državni koordinatni sistem.

4.3.1 Transformacija v državni koordinatni sistem

Obstaja več vrst transformacij tridimenzionalnih koordinatnih sistemov. Izbiro vrste transformacije določajo zahtevane lastnosti transformacije in lastnosti transformiranih koordinat. Lastnosti transformacije opredeljujejo transformacijski parametri med koordinatnima sistemoma.

Za namen transformacije rezultatov izmere GPS v državni koordinatni sistem se najpogosteje uporablja podobnostna transformacija, pri kateri je zveza med obema koordinatama podana s sedmimi transformacijskimi parametri:

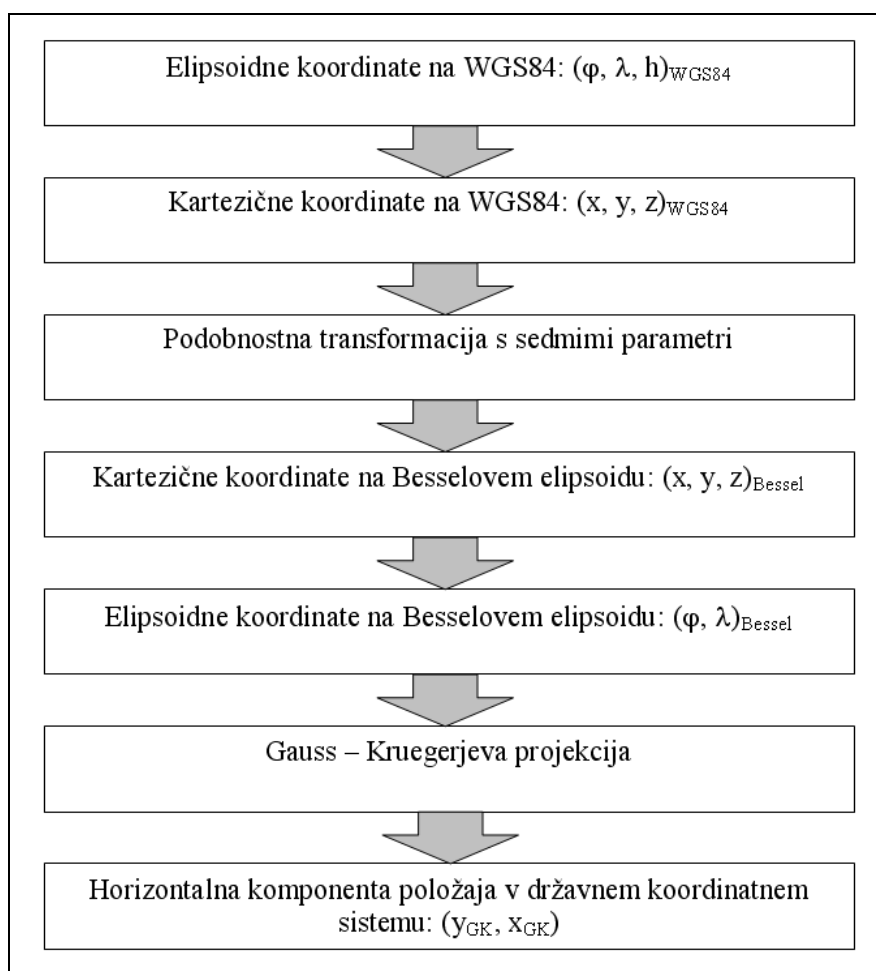
- trije premiki enega koordinatnega sistema glede na drugega,
- trije zasuki koordinatnega sistema glede na drugega in
- sprememba merila pri prehodu iz enega v drug koordinatni sistem.

Transformacija je zahtevna operacija, ki poteka v več posameznih korakih. Glede na zahtevano natančnost pri izvajanju meritev se prilagodita tudi postopek in način izvedbe

transformacije. Natančnost transformiranih koordinat je odvisna od natančnosti določitve transformacijskih parametrov in od natančnosti koordinat, ki se transformirajo (Golež, 2006).

V primeru zajema točk za potrebe prikazovanja poti na različnih kartografskih podlagah je zahtevana natančnost transformiranih koordinat v okviru nekaj metrov. Zato natančnost transformacije v državni koordinatni sistem zelo malo vpliva na končni rezultat.

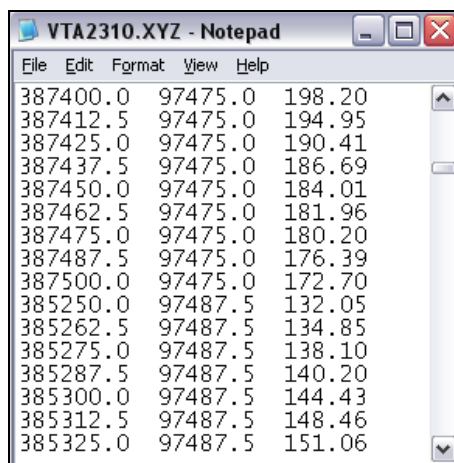
Iz sheme na sliki 8 je razvidno, da moramo v državni koordinatni sistem transformirati samo horizontalno komponento položaja točk v GPS sledi, za namen prekrivanja s slojem DMR. V spodnji shemi (slika 10) so predstavljeni glavni koraki transformacije.



Slika 10: Bločni diagram poteka transformacije koordinat iz GPS v državni koordinatni sistem.

4.3.2 Iskanje območja z bisekcijo

Prekrivanje podatkov temelji na podlagi iskanja celice DMR, ki predstavlja območje, v katerem se nahaja nova, poljubna točka. DMR 12,5 je zbirka podatkov, urejenih v datotekah tipa XYZ. Vsaka izmed 3258 datotek vsebuje tabelo koordinat, ki pokrivajo območje 2250 m krat 3000 m. Tabela v datoteki vsebuje 43621 vrstic, ki predstavljajo urejene trojice koordinat (y, x, z). Poljubni oz. testni točki iz transformirane datoteke PLT moramo v seznamu ustrezne datoteke XYZ najti ustrezna intervala v horizontalnih smereh x in y . V teh intervalih se bosta nahajali horizontalni koordinati testne točke. Iskanje take celice je mogoče programirati z metodo iskanja, imenovano bisekcija.



File	Edit	Format	View	Help
387400.0	97475.0	198.20		
387412.5	97475.0	194.95		
387425.0	97475.0	190.41		
387437.5	97475.0	186.69		
387450.0	97475.0	184.01		
387462.5	97475.0	181.96		
387475.0	97475.0	180.20		
387487.5	97475.0	176.39		
387500.0	97475.0	172.70		
385250.0	97487.5	132.05		
385262.5	97487.5	134.85		
385275.0	97487.5	138.10		
385287.5	97487.5	140.20		
385300.0	97487.5	144.43		
385312.5	97487.5	148.46		
385325.0	97487.5	151.06		

Slika 11: Primer zapisa DMR 12,5 v datoteki tipa XYZ.

Bisekcija je algoritem za iskanje podatka v urejenem seznamu (tabeli). Ustrezno datoteko najdemo tako, da izvajamo bisekcijo znotraj urejenega seznama datotek XYZ z navedenimi koordinatami začetnih točk v posamezni datoteki. Ko smo našli ustrezno datoteko, na podoben način poiščemo bližnje sosede testne točke.

V našem primeru iščemo območje (interval), na katerem se pojavi podatek y v urejenem seznamu $a_1, a_2, a_3, \dots, a_n$. Začetno iskalno območje je celoten seznam $a_1, a_2, a_3, \dots, a_n$. Nato v vsakem koraku prepолоvimo trenutno iskalno območje tako, da primerjamo y s srednjim elementom območja a_s , kjer je s celi del od $(i + j)/2$:

- če je $y < a_s$ nadaljujemo iskanje na območju $a_i, a_{i+1}, \dots, a_{s-1}$,
- če je $y = a_s$ smo našli x in vrnemo rezultat a_s ,
- če je $y > a_s$ nadaljujemo iskanje na območju $a_{s+1}, a_{s+2}, \dots, a_j$.

Postopek se ustavi, ko najdemo podatek $y = a_s$, ali pa postane širina območja $j - i$ manjša od 1 (v tem primeru se podatek y nahaja v intervalu med a_i in a_j , kjer velja $j = i + 1$) (<http://wiki.fmf.uni-lj.si/wiki/Bisekcija>, 4.12.2007).

4.4 Primerjava preprostih metod za določitev višine na podlagi DMR

Ko smo testni točki določili ustrezno celico DMR, na podlagi oglišč celice določimo višino točke. V naslednjih poglavjih predstavljam štiri metode določitve višine točke na podlagi DMR, ki so plod lastnega znanja. Poglavja vsebujejo izhodišča, izpeljave in končne enačbe za izračun višine na podlagi danih točk celice DMR. Narejeni so tudi praktični primeri uporabe enačb in obrazložitve rezultatov ter končna primerjava metod.

4.4.1 Metoda uteži težiščne točke

Metoda izračuna temelji na oddaljenosti poljubne točke od težiščne točke celice DMR. Metoda temelji na predpostavki, da če točka sovпада s težiščno točko celice, je njena višina izračunana kot srednja vrednost višin točk posameznih oglišč. Če točka ne sovпада, je njena višina izračunana kot srednja vrednost uteženih višin točk posameznih oglišč. Vrednosti uteži se spreminjajo z oddaljenostjo točke od težiščne točke. Simboli, uporabljeni v skici (slika 12) in enačbah so obrazloženi v naslednji preglednici.

Preglednica 1: Obrazložitev simbolov, uporabljenih v enačbah od (1) do (7).

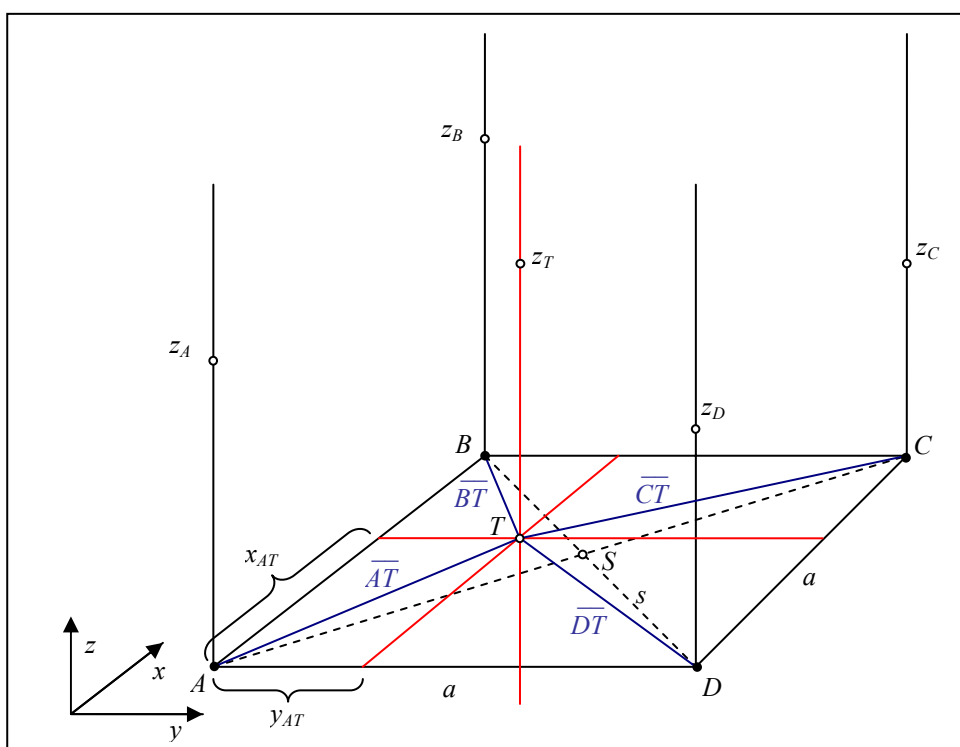
Simbol	Obrazložitev
A, B, C, D	Oglišča horizontalnega kvadrata, ki predstavlja celico DMR.
S	Težiščna točka celice DMR.
T	Testna točka znotraj celice DMR.

Se nadaljuje...

...nadaljevanje

x, y	Horizontalni komponenti koordinat točke.
z	Vertikalna komponenta koordinat (višina) točke.
P	Utež k posamezni višini oglišča.
a	Horizontalna dolžina stranice kvadrata celice DMR.
s	Horizontalna oddaljenost težiščne točke od oglišč kvadrata.

Indeksi simbolov so imena posameznih točk, ki nastopajo v izračunu in predstavljajo pripadnost simbola k določeni točki.



Slika 12: Skica za izračun metode uteži sredinske točke.

Osnovna enačba za izračun višine testne točke je osnovana na aritmetični sredini uteženih vrednosti višin ogliščnih točk.

$$z_T = \frac{1}{4}(z_A \cdot P_A + z_B \cdot P_B + z_C \cdot P_C + z_D \cdot P_D) \quad (1)$$

Utež višine na posamezni ogliščni točki je izračunana kot kvocient oddaljenosti težiščne točke in oddaljenosti testne točke od oglišča.

$$P_A = \frac{s}{AT} \quad P_B = \frac{s}{BT} \quad P_C = \frac{s}{CT} \quad P_D = \frac{s}{DT} \quad (2)$$

Horizontalno oddaljenost težiščne točke od oglišč kvadrata izračunamo s Pitagorovim izrekom.

$$s = \overline{AS} = \overline{BS} = \overline{CS} = \overline{DS} = \frac{a\sqrt{2}}{2} \quad (3)$$

$$a = \overline{AB} = \overline{BC} = \overline{CD} = \overline{DA} \quad (4)$$

Če uteži vstavimo v enačbo (1), dobimo splošno enačbo za izračun višine testne točke znotraj celice DMR z metodo uteži sredinske točke.

$$z_T = \frac{a\sqrt{2}}{8} \left(\frac{z_A}{AT} + \frac{z_B}{BT} + \frac{z_C}{CT} + \frac{z_D}{DT} \right) \quad (5)$$

Horizontalne dolžine razdalj med točkami izračunamo na podlagi koordinatnih razlik, s pomočjo Pitagorovega izreka.

$$\overline{AT} = \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2} \quad (6)$$

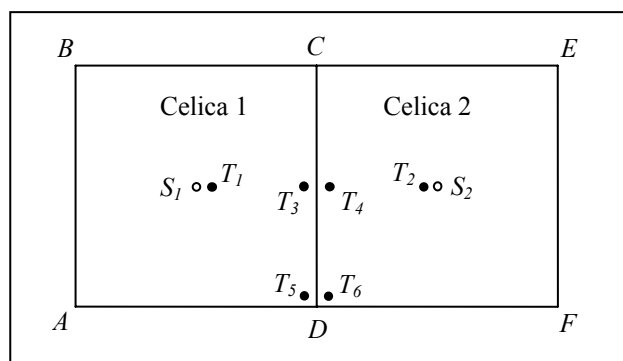
Če zgornji izraz, apliciran še na ostala tri oglišča, vstavimo v enačbo (5), dobimo končno enačbo za izračun višine točke na podlagi DMR.

$$z_T = \frac{a\sqrt{2}}{8} \left(\frac{z_A}{\sqrt{(x_T - x_A)^2 + (y_T - y_A)^2}} + \frac{z_B}{\sqrt{(x_T - x_B)^2 + (y_T - y_B)^2}} + \frac{z_C}{\sqrt{(x_T - x_C)^2 + (y_T - y_C)^2}} + \frac{z_D}{\sqrt{(x_T - x_D)^2 + (y_T - y_D)^2}} \right) \quad (7)$$

Primer

Primer numerične uporabe enačbe (7) sem naredil v programu MS Excel. Primer sem naredil za dve sosednji celici DMR-ja z ločljivostjo 12,5 m. Za dve sosednji celici sem se odločil zaradi narave točk, ki sestavljajo GPS sled. Točke so ponekod zelo blizu skupaj in v takšnem primeru tudi višine točk ne smejo biti preveč različne, saj bi to v naravi pomenilo velik skok v razgibanosti terena. Predvsem je potrebno preveriti pare točk, ki so blizu skupaj, pa vendar v različnih celicah DMR. Če je metoda ustrezna, je prehod med sosednjima celicama zvezen in ne »stopničast«.

Koordinate točk v ogliščih so izmišljene. Horizontalne komponente so nastavljene tako, da ustrezajo celični mreži DMR 12,5 m, vertikalne komponente pa so poljubne. Izračunal sem višine šestih točk, ki se nahajajo v dveh sosednjih celicah. Točki T_1 in T_2 sta pol metra oddaljeni od težiščnih točk, točki T_3 in T_4 sta pol metra od skupne stranice celic, T_5 in T_6 pa sta $\sqrt{2}/2$ m oddaljeni od skupnega oglišča celic. Točke z lihimi indeksi se nahajajo v levi, točke s sodimi pa v desni celici DMR (slika 13).



Slika 13: Razporeditev testnih točk v celici DMR.

Preglednica 2: Izračun višin točk z metodo uteži sredinske točke.

DMR a =		12,5		
	točke	X	Y	Z
Oglišča celice DMR	A	0	0	120
	B	12,5	0	150
	C	12,5	12,5	90
	D	0	12,5	105
	E	12,5	25	135
	F	0	25	45
Sredinski točki celice	S1	6,25	6,25	116,25
	S2	6,25	18,75	93,75
Poljubna točka v celici 1	T1	6,25	6,75	115,59
	T3	6,25	12	112,82
	T5	0,5	12	386,29
Poljubna točka v celici 2	T2	6,25	18,25	93,97
	T4	6,25	13	98,12
	T6	0,5	13	370,54

Rezultati izračuna očitno kažejo na slabo ustreznost metode, saj imajo sosednje točke na robovih celic (T_3 in T_4) zelo različne višine. Predvsem so neustrezne višine točk v bližini oglišč (T_5 in T_6), kjer njihova vrednost pade izven intervala najvišje in najnižje višine točk znotraj ene celice.

Metoda je torej ustrezna le za točke v bližini težišča celice DMR. Zanesljivost določitve višine točke pada z oddaljenostjo od težiščne točke celice. Za izračun višin točk GPS sledi metoda torej ni uporabna.

4.4.2 Metoda regresijske ravnine

Ravnino v prostoru lahko enolično definiramo, če imamo podane tri točke. Torej, če bi želeli s pomočjo mreže TIN (nepravilna trikotniška mreža) določiti višino poljubne točke z znanimi horizontalnimi koordinatami, bi skozi oglišča trikotnika, v katerega spada poljubna točka, napeli ravnino. Višinsko komponento poljubne točke bi dobili s presečiščem ravnine in vertikalne komponente točke.

Če imamo podane štiri točke v prostoru, kot nam jih podaja mreža DMR, v splošnem ne moremo enolično definirati ene ravnine, ki bi bila napeta na vse štiri točke. Lahko pa

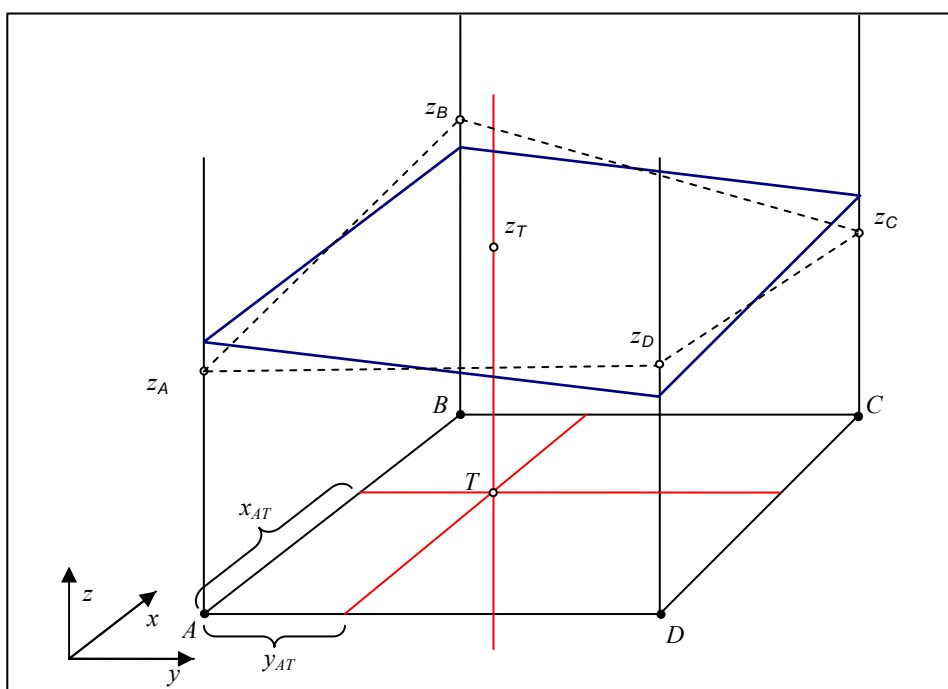
definiramo ravnino, ki se v najboljši možni meri prilega vsem štirim točkam. Takšno ravnino imenujemo regresijska ravnina, določimo pa jo s pomočjo metode najmanjših kvadratov. Z metodo dosežemo, da so kvadrati razdalj posameznih točk do ravnine minimalni.

Na sliki 14 je prikazana približna lega regresijske ravnine štirih točk celice DMR. Iz skice je razvidno, da na tak način pridobljene višine ne bodo ustrezne v bližini danih točk. Odstopanja naraščajo z višinsko razliko med posameznimi točkami celice DMR.

Simboli, uporabljeni na sliki 14 in enačbah od (8) do (16) so usklajeni s simboli, obrazloženimi v preglednici 1 v prejšnjem poglavju. Na novo uporabljeni simboli pa so obrazloženi v preglednici 3.

Preglednica 3: Obrazložitev simbolov, na novo uporabljenih v enačbah od (8) do (16).

Simbol	Obrazložitev
x	Normalni vektor ($x=[\alpha, \beta, \gamma]^T$).
A	Matrika koeficientov normalne enačbe.
b	Vektor konstant ($b=[z_A, z_B, z_C, z_D]^T$).



Slika 14: Skica za izračun višine točke na podlagi regresijske ravnine.

Eksplisitno enačbo ravnine v prostoru zapišemo v obliki:

$$\alpha \cdot x + \beta \cdot y + \gamma = z \quad (8)$$

Cilj zastavljene naloge je, da na podlagi štirih točk v prostoru izračunamo tri neznanke: α , β in γ , ki definirajo ravnino. Ko imamo definirano ravnino, ki gre blizu štirih takih točk, lahko višinsko komponento testne točke z_T izračunamo tako, da v enačbo ravnine vstavimo horizontalni koordinati x_T in y_T .

Nastavimo sistem štirih enačb s tremi neznankami.

$$\begin{aligned} \alpha \cdot x_A + \beta \cdot y_A + \gamma &= z_A \\ \alpha \cdot x_B + \beta \cdot y_B + \gamma &= z_B \\ \alpha \cdot x_C + \beta \cdot y_C + \gamma &= z_C \\ \alpha \cdot x_D + \beta \cdot y_D + \gamma &= z_D \end{aligned} \quad (9)$$

Sistem zapišemo v matrični obliki.

$$A \cdot x = b \quad (10)$$

$$\begin{bmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_C & y_C & 1 \\ x_D & y_D & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} z_A \\ z_B \\ z_C \\ z_D \end{bmatrix} \quad (11)$$

Sistem (10) je v splošnem nerešljiv. Če to enačbo pomnožimo z leve z matriko A^T , dobimo rešljivo matrično enačbo, imenovano normalna enačba.

$$A^T \cdot A \cdot x = A^T \cdot b \quad (12)$$

Enačbo lahko rešimo na več načinov, najbolj avtomatiziran način nam omogoča program Mathematica. Z njim lahko enačbe poenostavimo na ravni simbolov.

Oglišča celice DMR so elementi kvadrata, zato lahko njihove koordinate poenostavimo. S simbolom a označimo horizontalno dolžino stranice kvadrata celice DMR. Koordinate oglišč B , C in D lahko izrazimo glede na oglišče A .

$$\begin{aligned}
 x_B &= x_A + a & y_B &= y_A \\
 x_C &= x_A + a & y_C &= y_A + a \\
 x_D &= x_A & y_D &= y_A + a
 \end{aligned} \tag{13}$$

Normalni vektor x iz enačbe (12) izrazimo tako, da enačbo z leve strani pomnožimo z inverzno matriko $(A^T A)^{-1}$. Vektor x torej izračunamo z enačbo (14).

$$x = (A^T \cdot A)^{-1} \cdot A^T \cdot b \tag{14}$$

Program Mathematica nam na simbolni ravni poenostavi enačbo (14) in tako poda enačbo za izračun normalnega vektorja x (enačba (15)).

$$x = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \frac{-z_A + z_B + z_C - z_D}{2a} \\ \frac{-z_A - z_B + z_C + z_D}{2a} \\ \frac{d(3z_A + z_B - z_C + z_D) + 2(y_A(z_A + z_B - z_C - z_D) + x_A(z_A - z_B - z_C + z_D))}{4a} \end{bmatrix} \tag{15}$$

Primer

Višinsko koordinato z testne točke znotraj izbrane celice DMR izračunamo z enačbo (8). Primer izračuna sem naredil na isti način, kot v poglavju 4.4.1. Uporabil sem program MS Excel in iste testne točke. Na tak način bo mogoča kasnejša primerjava izračunanih višin po različnih metodah. Rezultati so zbrani v naslednji preglednici.

Preglednica 4: Izračun višin točk z metodo regresijske ravnine.

DMR $a =$		12,5		
	točke	X	Y	Z
Oglišča celice DMR	A	0	0	120
	B	12,5	0	150
	C	12,5	12,5	90
	D	0	12,5	105
	E	12,5	25	135
	F	0	25	45
Sredinski točki celice	S1	6,25	6,25	116,25
	S2	6,25	18,75	93,75
Poljubna točka v celici 1	T1	6,25	6,75	114,75
	T3	6,25	12	99,00
	T5	0,5	12	95,55
Poljubna točka v celici 2	T2	6,25	18,25	94,05
	T4	6,25	13	97,20
	T6	0,5	13	79,95

Preglednica 5: Koefficienti enačb ravnin, pripadajočih posameznima celicama.

CELICA 1	
alfa	0,6
beta	-3
gama	131,25
CELICA 2	
alfa	3
beta	-0,6
gama	86,25

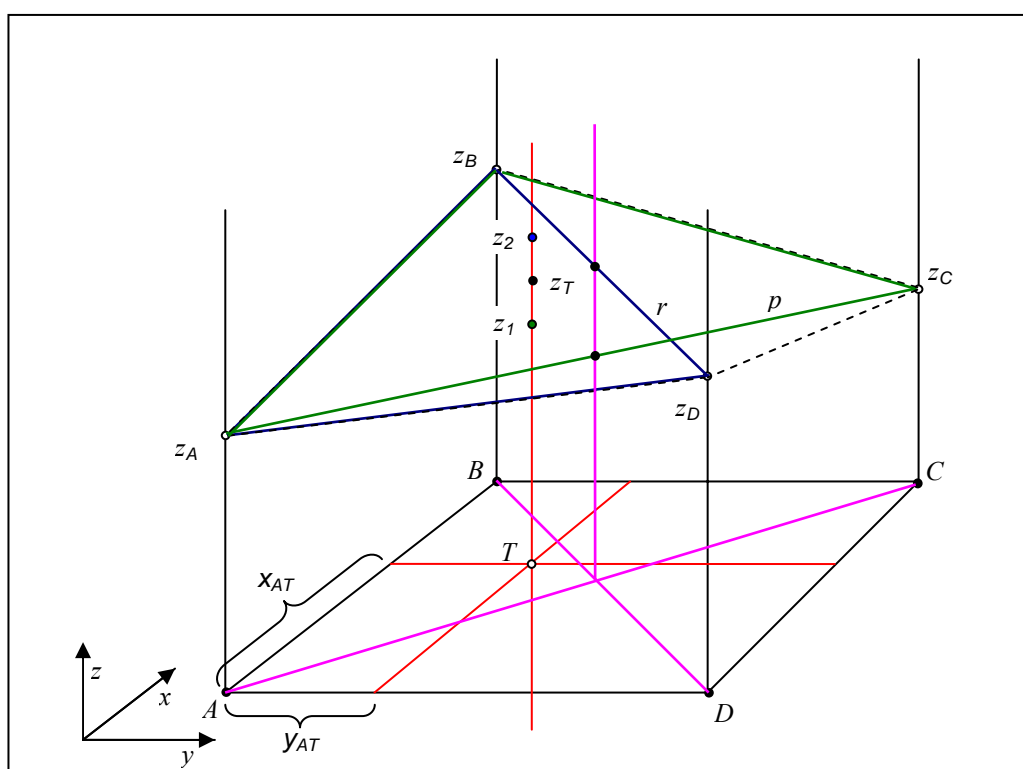
Izračunane višine točk nam kažejo na slabo ustreznost metode za izračun višin GPS sledi. Najbolj točno izračunana vrednost višine velja za točke blizu težišča kvadrata celice DMR. Bolj ko se približujemo robu in ogliščem celice, večje je odstopanje od pričakovane vrednosti višine. Regresijska ravnina se vrednostim višin v ogliščih zgolj približuje. V bližini oglišč

lahko izračunane višine padejo izven intervala največje in najmanjše vrednosti višine znotraj ene celice DMR. Metoda bi bila primerna za izračun višin na ravninskih območjih, kjer so višinske razlike med oglišči celice DMR majhne.

4.4.3 Metoda dveh ravnin

Določitev višine z metodo dveh ravnin je podobna metodi regresijske ravnine. Tam smo višino določili na podlagi presečišča vertikale skozi testno točko T in ene ravnine, v tem primeru pa določimo višino kot povprečno vrednost višin presečišč vertikale skozi točko T in dveh ravnin, ki jih lahko definiramo s točkami celice DMR.

Iz štirih točk v prostoru lahko definiramo štiri različne trojice točk. Na vsako trojico točk enolično napnemo ravnino. Če so omenjene točke dane točke celice DMR, lahko definiramo štiri trikotnike, vsak od njih je element ene ravnine (slika 15).



Slika 15: Skica za izračun višine točke z metodo dveh ravnin.

Preglednica 6: Razlage simbolov, uporabljenih v enačbah od (17) do (29).

Simbol	Obrazložitev
π, ρ, σ, τ	Ravnine v prostoru, definirane s točkami celice DMR.
z_1, z_2	Vertikalni koordinati presečišč dveh ustreznih ravnin in vertikale skozi točko T .
p, r	Diagonali, ki povezujeta nasprotno ležeča oglišča kvadratne celice DMR.
t, u	Premici, ki povezujeta točki A in B s testno točko T .
k	Naklonski koeficient premice.

Denimo, da trikotnik $\Delta z_A z_B z_D$ leži v ravnini π , $\Delta z_A z_B z_C$ leži v ravnini ρ , $\Delta z_B z_C z_D$ je v ravnini σ in $\Delta z_A z_C z_D$ je v ravnini τ . Če imajo dane točke različne višine, lahko definiramo zgornjo in spodnjo aproksimacijo površja. V primeru na sliki 15 zgornjo aproksimacijo enolično določata ravnini π in ρ , spodnjo pa σ in τ . Najbolj verjetna prava aproksimacija površja je aritmetična sredina zgornje in spodnje aproksimacije.

Testni točki T znotraj celice DMR določimo višino kot srednjo vrednost višin presečišč vertikale skozi T ter zgornje in spodnje aproksimacije površja. Vertikala skozi testno točko T vedno seka obe aproksimaciji površja.

Višino točke T izračunamo kot aritmetično sredino višin z_1 in z_2 .

$$z_T = \frac{z_1 + z_2}{2} \quad (16)$$

Višini z_1 in z_2 izračunamo kot presečišči vertikale skozi točko T in dveh ustreznih ravnin izmed ravnin π, ρ, σ in τ . Vsako izmed ravnin zapišemo v eksplicitni obliki.

$$\alpha \cdot x + \beta \cdot y + \gamma = z \quad (17)$$

Iskani vektor $[\alpha, \beta, \gamma]$ izračunamo tako, da v eksplicitno enačbo vsake ravnine vstavimo ustrezno urejeno trojico danih točk, ki so element posamezne ravnine. Tako za vsako od štirih

ravnin zapišemo sistem treh enačb s tremi neznankami. Npr. za ravnino π v enačbo (17) vstavimo koordinate točk A , B in D .

$$\begin{aligned}\alpha_{\pi} \cdot x_A + \beta_{\pi} \cdot y_A + \gamma_{\pi} &= z_A \\ \alpha_{\pi} \cdot x_B + \beta_{\pi} \cdot y_B + \gamma_{\pi} &= z_B \\ \alpha_{\pi} \cdot x_D + \beta_{\pi} \cdot y_D + \gamma_{\pi} &= z_D\end{aligned}\tag{18}$$

Na podoben način zapišemo sisteme še za ostale tri ravnine. Sisteme najlažje rešimo v matrični obliki.

$$A \cdot x = b\tag{19}$$

$$\begin{bmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_D & y_D & 1 \end{bmatrix} \cdot \begin{bmatrix} \alpha_{\pi} \\ \beta_{\pi} \\ \gamma_{\pi} \end{bmatrix} = \begin{bmatrix} z_A \\ z_B \\ z_D \end{bmatrix}\tag{20}$$

Enačbo (19) z leve strani pomnožimo z A^{-1} in tako iz enačbe izrazimo iskani vektor.

$$x = A^{-1} \cdot b\tag{21}$$

Člene v matriki A zapišemo na preprostejši način z upoštevanjem relacij med njimi, kot so zapisane v enačbah (13) v prejšnjem poglavju. Z upoštevanjem relacij rešimo matrični sistem iz enačbe (21). Avtomatiziran način za rešitev enačbe na simbolni ravni nam omogoča program Mathematica. Vektor x_{π} , ki določa ravnino π zapišemo v poenostavljeni obliki na simbolni ravni.

$$x_{\pi} = \left[\frac{z_B - z_A}{a}, \frac{z_D - z_A}{a}, \frac{dz_A + x_A z_A + y_A z_A - x_A z_B - y_A z_D}{a} \right]\tag{22}$$

Na enak način izrazimo še vektorje ostalih treh ravnin.

$$x_{\rho} = \left[\frac{z_B - z_A}{a}, \frac{z_C - z_B}{a}, \frac{dz_A + x_A z_A - x_A z_B + y_A z_B - y_A z_C}{a} \right] \quad (23)$$

$$x_{\sigma} = \left[\frac{z_C - z_D}{a}, \frac{z_C - z_B}{a}, \frac{(d + y_A)z_B - (d + x_A + y_A)z_C + (d + x_A)z_D}{a} \right] \quad (24)$$

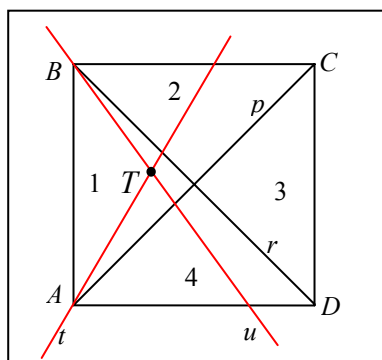
$$x_{\tau} = \left[\frac{z_C - z_D}{a}, \frac{z_D - z_A}{a}, \frac{dz_A + y_A z_A - x_A z_C + x_A z_D - y_A z_D}{a} \right] \quad (25)$$

Ob podanih horizontalnih koordinatah testne točke T lahko z eksplicitno enačbo ravnine in normalnim vektorjem izračunamo presečišče ravnine in vertikale skozi točko T . Enačbo (17) zapišemo v obliki skalarnega produkta dobljenega vektorja in vektorja testne točke T , v katerem smo neznano koordinato z nadomestili s skalarjem 1.

$$z = \alpha \cdot x + \beta \cdot y + \gamma = [\alpha, \beta, \gamma] \cdot [x, y, 1] \quad (26)$$

Za rešitev enačbe (16), ki nam izračuna končno višino točke T , moramo pravilno izračunati višini z_1 in z_2 , ki definirata presečišči vertikale skozi točko T z dvema ustreznima ravninama. Postopek za izbor dveh ustreznih ravnin lahko avtomatiziramo s programskimi jeziki, ki vsebujejo logične operacije (npr. PHP, jezik programa Mathematica, itd.). Ustrezen par ravnin izberemo glede na lego presečišča vertikale skozi točko T in horizontalnega kvadrata celice DMR.

Za izbor ustreznega para ravnin definiramo štiri območja znotraj kvadrata celice DMR, kot jih prikazuje slika 16. Območja so oštevilčena od 1 do 4, vsakemu od njih pripada par ravnin, kot je opredeljeno v preglednici 7.



Slika 16: Tloris celice DMR s prikazanimi območji za izbor ustreznih ravnin in premicami.

Preglednica 7: Območja za izbor ravnin z ustreznimi pari ravnin.

Območje	Prva ravnina	Druga ravnina
1	π	ρ
2	ρ	σ
3	σ	τ
4	τ	π

Vsako izmed štirih območij je omejeno s stranico celice DMR in dvema diagonalama kvadrata (p in r), ki povezujeta nasprotno ležeča oglišča celice. Na sliki 16 sta prikazani tudi premici t in u . Premico t definirajo horizontalne koordinate točk A in T , premico u pa horizontalne koordinate točk B in T . Če naklonska koeficienta premic t in u v horizontalni ravnini primerjamo z naklonskima koeficientoma diagonal p in r , lahko nedvoumno določimo eno od štirih območij, definiranih v preglednici 7, v katerem leži točka T . Ko smo določili območje, smo izbrali ustrezen par ravnin, katerih določujoče vektorje bomo uporabili v enačbi (26) za izračun višin z_1 in z_2 .

Naklonski koeficient diagonale p je k_p , njegova vrednost je 1. Za diagonalo r znaša vrednost $k_r = -1$.

Naklonska koeficienta premic t in u (k_t in k_u) sta spremenljiva z lego točke T , zato ju izračunavamo sproti.

$$k_t = \frac{x_T - x_A}{y_T - y_A} \quad (27)$$

$$k_u = \frac{x_T - x_B}{y_T - y_B} \quad (28)$$

Izbor ustreznega para ravnin lahko avtomatiziramo s programskimi jeziki. Ustrezna rešitev za avtomatizacijo je serija štirih pogojnih stavkov »else if«. Vsebina stavkov bi se v prevedenem pomenu glasila nekako takole:

Če $(k_t \geq 1 \text{ in } k_u < -1)$ $\{z_1 = x_\pi \cdot [x, y, 1]; z_2 = x_\pi \cdot [x, y, 1];\}$

sicer če $(k_t \geq 1 \text{ in } k_u \geq -1)$ $\{z_1 = x_\rho \cdot [x, y, 1]; z_2 = x_\sigma \cdot [x, y, 1];\}$

sicer če $(k_t < 1 \text{ in } k_u \geq -1)$ $\{z_1 = x_\sigma \cdot [x, y, 1]; z_2 = x_\tau \cdot [x, y, 1];\}$

sicer $(k_t < 1 \text{ in } k_u < -1)$ $\{z_1 = x_\tau \cdot [x, y, 1]; z_2 = x_\pi \cdot [x, y, 1];\}$.

Na ta način določeni vrednosti z_1 in z_2 vstavimo v enačbo (16) in tako pridobimo vrednost višine točke T .

Primer

Za primer izračuna sem uporabil isti pristop, kot pri ostalih metodah. Za dve celici sem izračunal osem normalnih vektorjev, končne višine testnih točk so povprečne vrednosti višin posameznih presečišč ustreznih ravnin in vertikal skozi točke.

Preglednica 8: Izračun višin točk z metodo dveh ravnin.

DMR a =		12,5		
	točke	X	Y	Z
Oglišča celice DMR	A	0	0	120
	B	12,5	0	150
	C	12,5	12,5	90
	D	0	12,5	105
	E	12,5	25	135
	F	0	25	45
Sredinski točki celice	S1	6,25	6,25	116,25
	S2	6,25	18,75	93,75

Se nadaljuje...

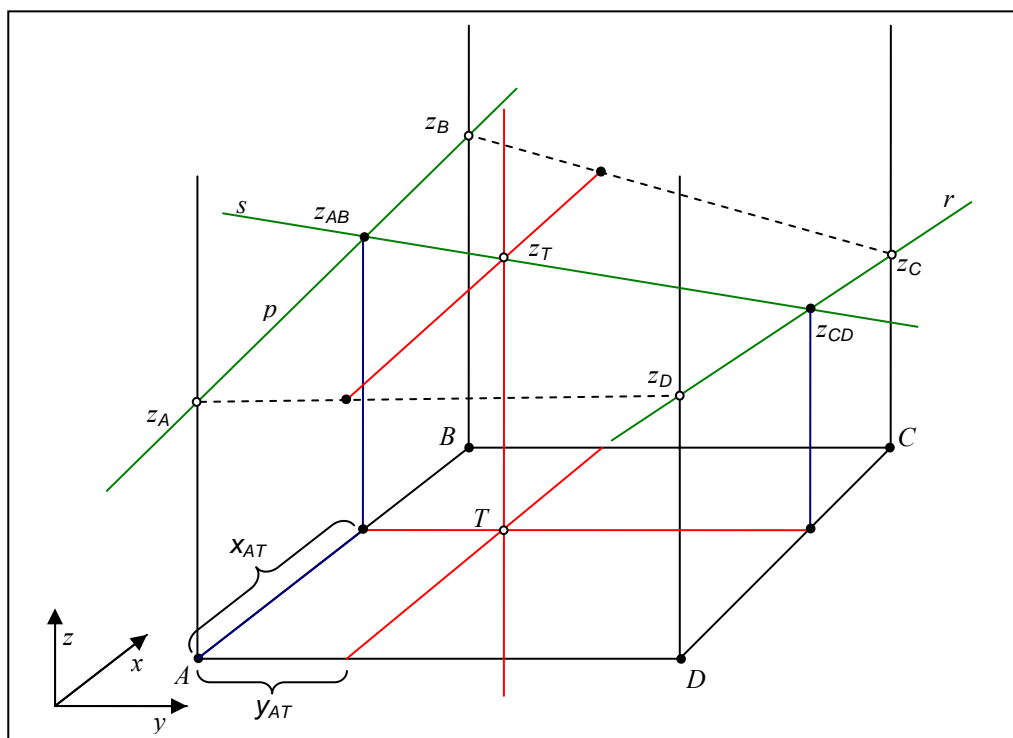
...nadaljevanje

Poljubna točka v celici 1	T1	6,25	6,75	114,75
	T3	6,25	12	99,00
	T5	0,5	12	105,90
Poljubna točka v celici 2	T2	6,25	18,25	94,05
	T4	6,25	13	97,20
	T6	0,5	13	104,10

Metoda je dokaj kompleksna, vendar pa ustreza za določitev višine katerekoli poljubne točke znotraj celice DMR. Če točka sovpada z ogliščem, je njena izračunana višina enaka podani višini oglišča. Na enak način sovpadajo višine točk na mejnem robu dveh sosednjih celic, izračunane vsaka na podlagi svoje celice. Zato je metoda primerna za določitev višin točk GPS sledi.

4.4.4 Linearna metoda

Izračun višine z linearno metodo temelji na presečiščih premic. Testno točko T znotraj celice DMR definira presečišče premice s in vertikale skozi točko T (slika 17).



Slika 17: Skica za izračun višine točke na podlagi presekov premic.

Premice p , r in s (na sliki 17 obarvane zeleno) lahko obravnavamo kot premice v ravninah, saj je njihova lega v prostoru vzporedna z ravninami, ki jih določajo koordinatne osi. Denimo, da premica p leži v ravnini π , premica r pa v ravnini ρ . Ravnini π in ρ sta obe vzporedni ravnini, ki jo definirata koordinatni osi x in z . Premica s pa je element ravnine σ , ki je vzporedna ravnini, ki jo definirata koordinatni osi y in z .

Metodo lahko izpeljemo na dva načina. Prvi način je opisan v zgornjem odstavku, za drugi način pa bi lahko postavili premici p in r vzporedno ravnini, ki jo definirata koordinatni osi y in z . Premica s bi bila vzporedna ravnini, ki jo definirata koordinatni osi x in z . Oba načina nam vrnete isti rezultat. Dokaz sem naredil s programom Mathematica (priloga A).

Razlage novih simbolov, ki se pojavijo v enačbah v tem poglavju, so zbrane v preglednici 9.

Preglednica 9: Razlage simbolov, uporabljenih v enačbah (29) do (35).

Simbol	Obrazložitev
k_p, k_r, k_s	Naklonski koeficienti premic p , r in s .
z_{AB}, z_{CD}	Vertikalni komponenti točk na presečiščih premice s s premicama p in r .

Eksplisitno enačbo premice p v ravnini π zapišemo v obliki:

$$z_{AB} = z_A + k_p(x_T - x_A) \quad (29)$$

$$k_p = \frac{z_B - z_A}{a} \quad (30)$$

Na podoben način zapišemo enačbo premice r v ravnini ρ .

$$z_{CD} = z_D + k_r(x_T - x_D) \quad (31)$$

$$k_r = \frac{z_C - z_D}{a} \quad (32)$$

Vertikalni komponenti točk na presečiščih premice s s premicama p in r smo tako izrazili z danimi količinami. Eksplicitno enačbo premice s v ravnini σ zapišemo podobno, kot za premici p in r (enačbi (29) in (31)).

$$z_T = z_{AB} + k_s(y_T - y_A) \quad (33)$$

$$k_s = \frac{z_{CD} - z_{AB}}{a} \quad (34)$$

V enačbo (33) vstavimo vse znane količine in izraze iz enačb (29) do (34). Izraz dodatno poenostavimo s skupino enačb (13) in dobimo enačbo (35) za izračun vertikalne komponente oz. višine točke T . Poenostavitve enačbe sem izvedel s pomočjo programa Mathematica.

$$z_T = \frac{1}{a^2} (a^2 z_A + a(x_A(z_A - z_B) + x_T(z_B - z_A) + (y_A - y_T)(z_A - z_D)) + (x_A - x_T)(y_A - y_T)(z_A - z_B + z_C - z_D)) \quad (35)$$

Primer

Primer izračuna sem naredil na isti način, kot pri ostalih metodah. Uporabil sem program MS Excel in iste testne točke. Rezultati so zbrani v preglednici 10.

Preglednica 10: Izračun višin točk z linearno metodo.

DMR a =		12,5		
	točke	X	Y	Z
Oglišča celice DMR	A	0	0	120
	B	12,5	0	150
	C	12,5	12,5	90
	D	0	12,5	105
	E	12,5	25	135
	F	0	25	45

Se nadaljuje...

...nadaljevanje

Sredinski točki celice	S1	6,25	6,25	116,25
	S2	6,25	18,75	93,75
Poljubna točka v celici 1	T1	6,25	6,75	114,75
	T3	6,25	12	99,00
	T5	0,5	12	105,07
Poljubna točka v celici 2	T2	6,25	18,25	94,05
	T4	6,25	13	97,20
	T6	0,5	13	102,17

Metoda vrne ustrezen rezultat za katerokoli točko znotraj celice DMR. Če točka sovпада z ogliščem, je njena izračunana višina enaka podani višini oglišča. Na enak način sovpadajo višine točk na mejnem robu dveh sosednjih celic, izračunane vsaka na podlagi svoje celice.

Posledica tega je, da je vsak prehod GPS sledi iz ene celice v drugo zvezen, ne glede na to, kje se nahajajo lomne točke sledi. Metoda je torej zelo primerna za določitev višin točk GPS sledi.

4.5 Primerjava rezultatov

V numeričnih primerih k posameznim metodam so izračunane višine šestih istih točk. Točke so razporejene v dve sosednji celici DMR, kot jih prikazuje slika 13. Omogočena je primerjava izračuna glede na uporabljeno metodo. V preglednici 11 so zbrani rezultati izračunov višin točk po vseh štirih metodah.

Preglednica 11: Primerjava rezultatov izračunov višin točk po posameznih metodah.

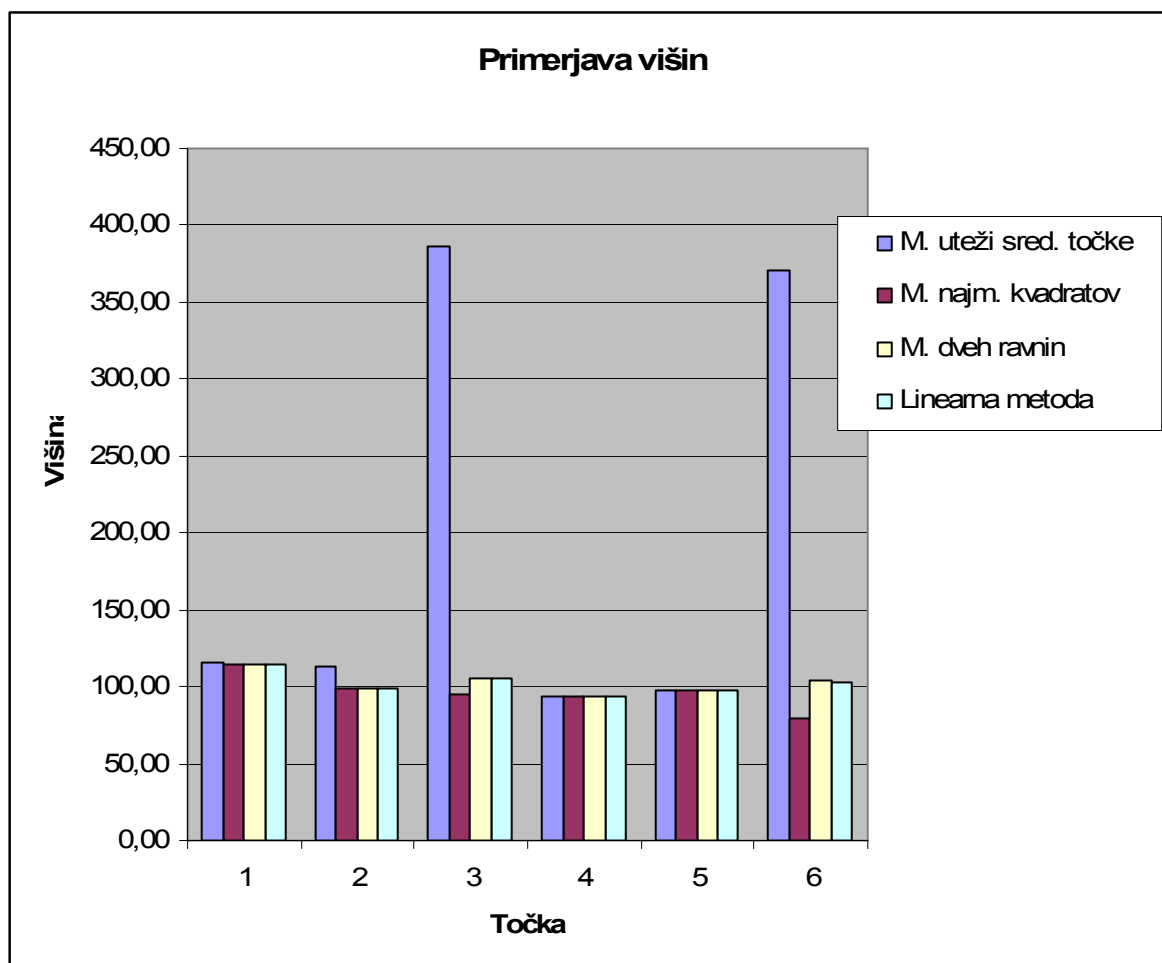
TOČKA	X	Y	Z			
			M. uteži sred. točke	M. najm. kvadratov	M. dveh ravnin	Linearna metoda
T1	6,25	6,75	115,59	114,75	114,75	114,75
T3	6,25	12	112,82	99,00	99,00	99,00
T5	0,5	12	386,29	95,55	105,90	105,07
T2	6,25	18,25	93,97	94,05	94,05	94,05
T4	6,25	13	98,12	97,20	97,20	97,20
T6	0,5	13	370,54	79,95	104,10	102,17

Grafičen prikaz rezultatov je prikazan na grafu v sliki 18. Zelo očitno izstopata višini točk T_5 in T_6 , izračunani z metodo uteži sredinske točke. Točki se nahajata blizu skupnega oglišča,

zato bi morali biti njihovi višini približno enaki med sabo in v primerjavi z ogliščem. Ker temu ni tako, lahko metodo označimo za neustrezno.

Metoda regresijske ravnine je prav tako sporna za določitev višine točke v bližini oglišča. Odstopanja niso tako velika kot pri metodi uteži sredinske točke, vendar pa se preostali dve metodi (metoda dveh ravnin in linearna metoda) obneseta veliko bolje. Lahko rečemo, da bi bila metoda regresijske ravnine sprejemljiva za izračun višin točk na ravninskih območjih, na razgibanem reliefu pa bi napake presegle sprejemljive meje.

Preostali dve metodi se obe izkazujeta kot primerni za določitev višin točk, saj so si izračunane vrednosti na posameznih točkah zelo podobne. Metodi določita ustrezne višine tudi točkama T_5 in T_6 .



Slika 18: Grafični prikaz primerjave izračunanih višin točk po posameznih metodah.

Metoda dveh ravnin in linearna metoda sta torej najbolj univerzalni in točni za določitev višine točke na podlag DMR. Glede na zahtevnost razvoja in aplikacijo metode pa je linearna metoda gotovo najboljša izbira za takšno nalogo. Metoda vrne najbolj točen rezultat in je preprosta za izpeljavo in avtomatizacijo.

4.6 Glajenje z drsečim povprečjem

Ko smo vsaki točki GPS sledi določili višino na podlagi DMR, je smiselno uporabiti postopek glajenja sledi z metodo drsečega povprečja. S tem postopkom se znebimo prevelikih skokov v višinah, ki nastopijo zaradi narave geometrije DMR v reliefno razgibanih predelih. Veliko sledi v bazi je posnetih na zelo strmih območjih, kjer majhen horizontalen premik v smeri največjega naklona pomeni veliko spremembo višine. Da bi se takšnim naključnim spremembam višine izognili, vsaki točki GPS sledi na novo določimo višino glede na višino točke za njo in pred njo. Nova višina točke je aritmetična sredina višin obravnavane točke ter točk pred njo in za njo. Če je i indeks vrstnega reda točke v nizu točk GPS sledi, izračunamo novo višino točke po enačbi (36).

$$z_{i(NOVA)} = \frac{z_{i-1} + z_i + z_{i+1}}{3} \quad (36)$$

Prva in zadnja točka v nizu točk GPS sledi nimata obeh sosedov, zato njunih višin ne popravljamo, ampak ohranimo dane vrednosti.

5. DOPOLNJEVANJE SPLETNEGA MESTA

5.1 SWOT analiza

SWOT (Strengths, Weaknesses, Opportunities, Threats) analiza je zelo učinkovito orodje za ugotavljanje stanja in možnosti za nadaljnji razvoj nekega projekta. Z analizo opredelimo prednosti, slabosti, priložnosti in nevarnosti, ki nam pomagajo pri odločanju o nadaljnjih

spremembah. Pomaga nam, da se osredotočimo na spreminjanje ključnih parametrov, ki vplivajo na dosego nekega cilja (http://en.wikipedia.org/wiki/SWOT_analysis, 2.10.2007).

Ključno začetno dejanje za dobro izvedeno analizo je definiranje cilja, ki ga želimo doseči. Za spletno mesto je cilj navadno povečanje števila obiskov. To dosežemo z dobrim oglaševanjem in s kakovostno ponudbo storitve.

V diplomski nalogi se osredotočam na kakovost storitve, zato si lahko za izdelavo SWOT analize zastavim vprašanje: Kaj je ključno za obstoj, povečanje uporabe in dobro uporabniško izkušnjo na spletnem mestu »Planinske in kolesarske poti ter turni smuki z GPS podporo«? Postavke, ki se dotikajo tega vprašanja, so zbrane v preglednici 12.

Preglednica 12: SWOT analiza spletnega mesta »Planinske in kolesarske poti ter turni smuki z GPS podporo«.

PREDNOSTI (Strengths)	Izviren način povezovanja GPS sledi. Prilagodljiva, nadgradljiva in povezljiva programska osnova. Hitrost delovanja. Zmožnost sprejema in obdelave velike količine podatkov. Zadovoljiva natančnost metričnih podatkov. Dostopnost kjerkoli in kadarkoli.
SLABOSTI (Weaknesses)	Nerodno izbiranje začetnih točk – preveč jih je. Uporabnik se ne znajde, kako spletno mesto uporabljati. Slaba ponudba atributnih podatkov v bazi. Pomanjkljiva baza – premalo sledi. Slaba estetika strani. Windows XP platforma je zelo ranljiva za vdore. GPS sledi so omejene na Slovenijo z bližnjo okolico. Morebitne napake v podatkih.
PRILOŽNOSTI (Opportunities)	Vedno več sprejemnikov v uporabi, število potencialnih uporabnikov narašča.

Se nadaljuje...

...nadaljevanje

	Prostorski podatki na spletu so vse pogostejši. Razvoj spletnih zemljevidov in globusov. Gorske poti so v naravi vedno slabše označene. Poti na uradnih kartah so zastarele in se ne obnavljajo.
GROŽNJE (Threats)	Konkurenca – stran ni avtorsko zaščitena, pojavi se lahko komercialni ponudnik, z boljšo ponudbo in podporo. Vdor v spletno mesto na Windows XP platformi. Kraja GPS sledi in objava na drugih spletnih mestih.

5.2 Možne dopolnitve in njihova racionalizacija

Možnosti za dopolnjevanje spletnega mesta so neomejene. Glede na dejstvo, da delo izvajam v sklopu diplomske naloge, si moram postaviti omejitve pri dopolnjevanju. Ideje za dopolnitve so delo doc. dr. Mitje Laknerja in mene. Osredotočila sva se na učinkovito in preprosto delovanje spletnega mesta in zadovoljstvo uporabnika s pridobljenimi informacijami. Tako pridobljene ideje za izboljšave so:

- Možnost izbiranja začetnih točk po atributih:
 - po regionalnem položaju,
 - po vrsti točke (izhodišče, vrh, planota, sedlo, ...),
 - po nadmorski višini točke,
 - po obstoju koč na točki,
 - po obstoju bivaka na točki,
 - po obstoju cerkve na točki,
 - po obstoju jezera na točki,
 - po obstoju slapa na točki.
- Možnost izbiranja poti po atributih:
 - po času hoje,
 - po višinski razliki,
 - po zahtevnosti poti.
- Vsako točko opremiti s povezavo na stran z opisom točke.
- Razvoj orodja za iskanje nedefiniranih presekov poti.

- Oznake pomembnih točk v grafičnem prikazu vertikalnega prereza poti.
- Odpraviti napako izrisa PLT datoteke v OziExplorerju.
- Znotraj KML datoteke nastaviti različne barve odsekov poti.
- Popravljanje višin iz GPS koordinatnega sistema v normalne ortometrične višine (poglavje 4.4).
- Popraviti skrajšana imena točk v bazi v popolna imena.
- Estetske dopolnitve strani.

Izmed naštetih možnih izboljšav so mnoge vsebinsko nesmiselne, za nekatere pa velja, da je časovni vložek veliko večji od pričakovanega učinka. Za odločitev o izvajanju posamezne dopolnitve je bil tako potreben tehten premislek. Z doc. dr. Mitjo Laknerjem sva izbor dopolnitev naredila glede na kriterij nujnosti in časovne zahtevnosti izvedbe. Časovno zelo zahtevna dopolnitev bi bila npr. razvoj orodja za iskanje nedefiniranih presekov poti, enostavna, vendar večini uporabnikov nepotrebna dopolnitev pa je npr. možnost izbiranja začetnih točk po atributu obstoja slapa na točki. Zamudna in zahtevna naloga bi bila tudi avtomatski izris različno obarvanih odsekov poti znotraj KML datoteke. Vsak odsek bi namreč moral biti definiran kot samostojen »placemark«, poleg barve bi moral imeti definirane še vse ostale attribute.

6. IZVEDENE DOPOLNITVE

V naslednjih podpoglavjih so opisane vse izvedene dopolnitve spletnega mesta »Planinske in kolesarske poti ter turni smuki z GPS podporo«. Dopolnitve so vsebinsko zelo raznolike, vsaka izmed njih terja eno ali več sprememb v zasnovi spletnega mesta. Spremembe zavzemajo položaje v različnih datotekah na strežniku.

V sklopu prilog k tej diplomski nalogi so priložene vse programske kode, ki so bile dopolnjene. Za ilustracijo podatkovne baze je v prilogah tudi nekaj reprezentativnih vrstic dopoljenih SQL tabel.

Dopolnjevanje SQL baze z atributnimi podatki je časovno zelo potratno delo. Podatke je potrebno najti in prilagoditi, nato pa ročno vnesti v SQL tabele. To pomeni definiranje novih polj v tabelah in njihovo izpolnjevanje. Za diplomsko nalogo sem uporabil obstoječi SQL tabeli poti in točk, iz katerih sem z namenom zmanjšanja količine podatkov izbral približno polovico poti in točk. Izbor je narejen tako, da se točke in poti ustrezno povezujejo in se nahajajo v različnih geografskih območjih.

Atributni podatki za dopolnjevanje so zbrani na različne načine in iz različnih virov. Spletni viri so v osnovi naslednja spletna mesta: www.hribi.net, www.pzs.si in sl.wikipedia.org. Veliko izletniških ciljev ima tudi svoje spletne strani z uporabnimi informacijami. Ostali viri so državne topografske karte, digitalni ortofoto in digitalni model reliefa, v povezavi z vektorskima slojema poti in točk iz obravnavane spletne baze. Za pregledovanje omenjenih kartografskih slojev sem uporabljal programa OziExplorer in Google Earth.

6.1 Možnost izbiranja začetnih točk po atributih

Slabost spletnega mesta je bil način izbiranja začetnih točk. Vse točke (približno 300 jih je) so bile zbrane v eni tabeli na začetni strani *index.php*. Uporabnik, ki ni vedel točno na kateri točki bi začel graditi pot, je imel nemalo težav iz tabele izluščiti sebi zanimivo točko. Namen izpopolnitve je torej olajšati uporabniku brskanje med začetnimi točkami na strani *index.php*. Izbira začetne točke v programu Google Earth je dovolj pregledna in ne rabi izpopolnitve.

Uporabniku je treba dati možnost, da lahko zoži nabor točk na neko pregledno in smiselno množico točk z istimi lastnostmi. V ta namen je treba predhodno v SQL tabeli opremiti točke z različnimi atributnimi podatki. Naslednji korak je izdelava orodja oz. grafičnega vmesnika, ki bo uporabniku omogočal opredelitev kriterijev iskanja in mu glede na izbor po atributih ponudil pripadajoče točke.

V izpopolnjeni različici lahko uporabnik izbira med točkami po naslednjih atributih:

- po regionalnem položaju (8 regij, ki predstavljajo gorovja in hribovja z okolicami),
- po vrsti točke (izhodišče, vrh, ostale točke),

- po obstoju koč na točki (je koč, ni koč),
- po nadmorski višini točke (13 razredov nadmorskih višin).

Atribut za višino točke je sestavni del že obstoječe SQL tabele *tocke*. Dopolnitev je potrebna le v izdelavi uporabniškega vmesnika.

6.1.1 Dopolnjevanje SQL tabele *tocke*

Tabela *tocke* se nahaja v SQL podatkovni bazi, poimenovani *GPS*. V predhodni različici je tabela vsebovala 5 polj (poglavje 3.4), dopolnjena tabela pa vsebuje 9 polj. Polje za navedbo nadmorske višine (*visina*) točke je že obstajalo.

	Field	Type	Attributes	Null	Default	Extra
<input type="checkbox"/>	id	int(11)		No		auto_increment
<input type="checkbox"/>	ime	varchar(20)		Yes	NULL	
<input type="checkbox"/>	sirina	float(8,6)		Yes	NULL	
<input type="checkbox"/>	dolzina	float(8,6)		Yes	NULL	
<input type="checkbox"/>	visina	float		Yes	NULL	
<input type="checkbox"/>	gorovje_kod	int(11)		No	0	
<input type="checkbox"/>	gorovje	varchar(50)		Yes	NULL	
<input type="checkbox"/>	vrsta_tocke	varchar(11)		No	karkoli	
<input type="checkbox"/>	koca	enum('N', 'Y')		No	N	

Slika 19: Struktura dopolnjene tabele *tocke* (izsek iz PhpMyAdmin).

6.1.1.1 Regionalni položaj

Za pripojitev atributa o regionalnem položaju sem se odločil dodati dve novi polji. Prvo polje *gorovje_kod* vsebuje števila tipa *integer*. Števila zavzemajo vrednosti od 0 (privzeta vrednost) do 8, vsako število večje od 0 kodira določeno regijo (preglednica 13). Polje *gorovje* vsebuje besedna poimenovanja regij, tip polja je *varchar*. Razlog dveh polj za definiranje regionalnega položaja je v tem, da je programiranje (X)HTML in PHP kode veliko preprostejše in preglednejše, če delamo s kratkimi atributi tipa *integer*, kot pa z dolgimi znakovnimi nizi. Slednje uporabljamo le pri izpisovanju poimenovanj na uporabniški vmesnik. Polje *gorovje* nima privzete vrednosti in ostane prazno, če ga ne izpolnimo.

Preglednica 13: Kodiranje regij, kot je uporabljeno v SQL tabeli *tocke*.

Koda regije (<i>gorovje_kod</i>)	Poimenovanje regije (<i>gorovje</i>)
1	Julijske Alpe
2	Karavanke
3	Kamniško Savinjske Alpe
4	Goriško, Notranjsko in Snežniško hribovje
5	Polhograjsko hribovje in Ljubljana
6	Škofjeloško, Cerkljansko hribovje in Jelovica
7	Pohorje in ostala severovzhodna Slovenija
8	Zasavsko hribovje, Dolenjska in Bela krajina

6.1.1.2 Vrsta točke

Z definirano vrsto točke lahko uporabnik izmed vseh točk izbere tiste, ki lahko služijo kot izhodišče, ali vse vrhove, ali pa vse tiste točke, ki niso ne vrh, niti izhodišče. V SQL tabeli sem definiral polje tipa *varchar*, ki lahko sprejme do 11 znakov. Ime polja je *vrsta_tocke*, privzeta vrednost pa je *karkoli*. Ta vrednost označuje točke, ki niso ne vrh, niti izhodišče, ostali dve vrednosti, ki nastopata v tem polju pa sta *vrh* in *izhodisce*.

6.1.1.3 Obstoje koč

Pri načrtovanju izleta v gore je pomemben podatek o obstoju planinskih koč na poti. Nekateri hribolazci si za cilj izleta zastavijo obisk koč, veliko pa je tudi takih, ki na svojem izletu ne želijo bližine koč. To je razlog, da sem v dopolnjeno SQL tabelo vključil tudi polje tipa *enum*, ki ima privzeto vrednost *N*, katera pomeni, da ni koč na točki. Obstoje koč označuje druga možna vrednost polja, *Y*. Privzeta vrednost polja je *N*.

6.1.2 Izdelava grafičnega vmesnika za izbiranje točk po atributih

Cilj zastavljene naloge je, da lahko uporabnik na čimbolj enostaven in udoben način zoži izbor med ponujenimi točkami. V ta namen sem dopolnil datoteko *index.php* in naredil novo datoteko *izbira.php*. Izbiranje točk poteka v treh korakih:

- Uporabnik v začetni datoteki *index.php* v obrazcu nastavi izbor. Obrazec sestavljajo spustni meniji in radijski gumbi. S klikom na gumb »Izberi« se izbrane vrednosti spremenljivk iz obrazca zapišejo v URL naslov.
- Zažene se druga datoteka *izbira.php*, ki iz URL naslova prebere izbrane spremenljivke s pripadajočimi vrednostmi in preko avtomatsko tvorjenega SQL stavka iz baze izbere ustrezne točke.
- Izbrane točke se nanizajo v tabelo. Ime točke je hkrati tudi hiperpovezava na datoteko *gradnja_poti.php*.

Za zastavljeno nalogo sta torej potrebni dve datoteki. Prva omogoča uporabniku nastavitve parametrov za izbor, druga pa prebere izbor in ga v berljivi obliki vrne uporabniku. Za prvo datoteko sem izkoristil možnosti neomejenega vdelovanja HTML kode v PHP in obratno. Tako sem v obstoječo datoteko *index.php* vdelal HTML kodo, ki tvori spustne menije in radijske gumbe.

6.1.2.1 Datoteka *index.php*

Začetna datoteka več ne zasuje uporabnika s tabelo vseh točk, ampak mu na mestu tabele ponudi obrazec za nastavitev izbire (slika 20). V obrazcu uporabnik preko grafičnega vmesnika določi vrednosti spremenljivk. Spustni meniji ponujajo izbor točke glede na gorovje, višino in obstoj koč, radijski gumbi pa glede na vrsto točke. Možni kriteriji za izbor točk glede na gorovje, obstoj koč in vrsto točke so navedeni v poglavju 6.1.1, 13 razredov višin pa je razvidnih iz kode v prilogi B. Privzete vrednosti spremenljivk v obrazcih so nastavljene na »vseeno« in če uporabnik ne opredeli nobenega kriterija, mu izbor vrne vse

točke, ki se nahajajo v bazi. Tak pristop torej omogoča, da lahko uporabnik izbira točke glede na nobenega, enega ali več kriterijev.

Izberi si točko, na kateri želiš začeti gradnjo poti:

Izberi točko glede na gorovje:

Izberi točko glede na višino: Od: do:

Izberi točko glede na obstoj koč:

Želiš začeti gradnjo poti na izhodišču, vrhu, ostalih točkah ali ti je vseeno?

Izhodišče:

Vrh:

Ostale točke:

Vseeno mi je:

Slika 20: Grafični vmesnik za izbor začetne točke po atributih.

Izbrane vrednosti na sliki 20 niso privzete vrednosti, ampak so izbrane za primer uporabe, ki ga navajam v nadaljevanju besedila. Izpis, kot je viden na sliki 20, omogoča spodnji izsek kode iz datoteke *index.php*. V datoteki je izsek vdolan med HTML značke, ki definirajo glavo in telo dokumenta (te značke so obrazložene v poglavju 6.1.2.2, ki obravnava datoteko *izbira.php*).

Preglednica 14: Razlaga uporabljenih HTML značk in ukazov v kodi obrazca za izbiranje (<http://www.w3schools.com/tags>, 24.10.2007).

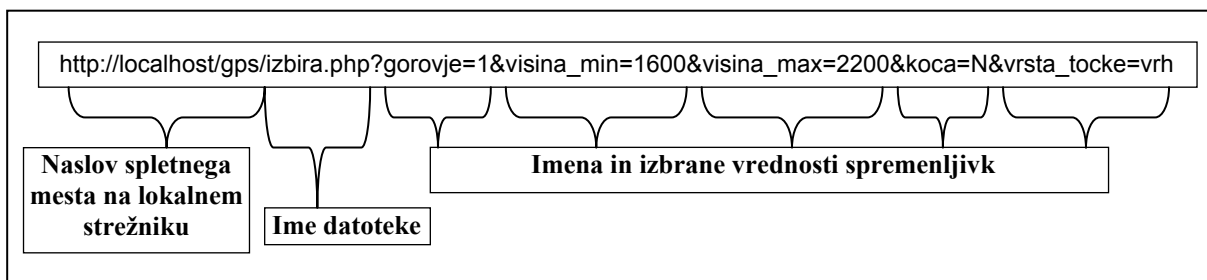
Značka	Razlaga
form	Tvori obrazec za uporabnikov vnos podatkov, ki lahko vsebuje tekstovna polja, spustne menije, radijske gumbе, potrditvena polja (kljukice), itd.
action	Je atribut k narejenemu obrazcu, ki za vrednost zahteva URL naslov ali datoteko, kamor bodo poslani podatki, ko uporabnik potrdi izbor. V mojem primeru so podatki poslani datoteki <i>izbira.php</i> .
method	Je atribut, ki določi način pošiljanja podatkov o izboru. Metoda <code>get</code> (polno ime metode je »HTTP GET«) pošlje izbrano vsebino obrazca preko URL naslova, metoda <code>post</code> (»HTTP POST«) pa preko telesa zahteve (veliko

Se nadaljuje...

...nadaljevanje

	brskalnikov ima težave s to metodo).
<code>select name</code>	Element naredi spustni meni, kateremu po potrebi definiramo tudi enolični
	identifikator, ki velja kot ime spremenljivke. Izbrana vrednost spremenljivke se preko identifikatorja prenese na strežnik.
<code>option value</code>	S to značko definiramo možno vrednost spremenljivke in ime, za katerim stoji vrednost v spustnem meniju. Vrednost označimo za privzeto z atributom <code>selected="selected"</code> .
<code>input type="radio"</code>	Definira polje za vnos, kamor lahko uporabnik vnese podatke. Tip polja je lahko tekstovno ali potrditveno polje, datoteka, itd. V mojem primeru je izbran tip polja radijski gumb. Polju določimo še ime (<code>name</code>) in vrednost, ki se skriva za izbiro (<code>value</code>)
<code>input type=submit</code>	Polje za vnos podatka, tip je nastavljen na <code>submit</code> (predloži). Tip polja oblikuje gumb, na katerega kliknemo, ko želimo podatke poslati strežniku.
<code>table</code>	Značka definira obstoj tabele.
<code>td</code>	Definira nov stolpec v tabeli.
<code>tr</code>	Definira novo vrstico v tabeli.
<code>br</code>	Definira novo vrstico v prostem tekstu.

Za boljšo preglednost obrazca sem v eni tabeli uredil spustne menije, v drugi pa radijske gumbe, skupaj s pripadajočimi razlagami. Razlage so čimbolj ilustrativne, da uporabniku neposredno povejo, kaj mu lahko določena izbira vrne. Ko uporabnik klikne gumb »Izberi«, obrazec pošlje podatke (izbrane spremenljivke s pripadajočimi vrednostmi) strežniku. Za moj primer in ob izbranih vrednostih, kot so prikazane na sliki 20, bo obrazec zapisal podatke v URL naslov datoteke *izbira.php*, kot je razvidno in razloženo na sliki 21. Imena spremenljivk in njihove vrednosti so vse definirane v obrazcu, definicije pa so prilagojene nadaljnji obdelavi podatkov v datoteki *izbira.php*.



Slika 21: Obrazložitev naslova datoteke *izbira.php*, ki vsebuje spremenljivke z definiranimi vrednostmi.

6.1.2.2 Datoteka *izbira.php*

Ko uporabnik nastavi kriterije izbire v datoteki *index.php* in klikne gumb »Izberi«, strežnik požene datoteko *izbira.php*. Datoteka je na spletnem mestu nova, parametri za določitev jezika in izgleda strani so poenoteni s celotnim spletnim mestom. Vsebina celotne datoteke je navedena v prilogi C, razlage funkcij pa so zbrane v dveh preglednicah. V prvi (preglednica 15) se nahajajo razlage uporabljenih HTML značk, v drugi (preglednica 16) so razlage PHP funkcij. Slika strani je prikazana v sliki 23, v poglavju 6.3.2, kjer je opisana dopolnitev strani s povezavami na opise točk.

Preglednica 15: Razlaga uporabljenih HTML značk in ukazov v kodi datoteke *izbira.php* (<http://www.w3schools.com/tags>, 24.10.2007).

Značka	Razlaga
!DOCTYPE	Deklaracija lastnosti dokumenta je prva stvar, ki jo definiramo v novi datoteki. Značka pove brskalniku, katera vrsta jezika je uporabljena. V mojem primeru je to »XHTML 1.0 Transitional«, katerega lastnost je, da lahko vsebuje tudi vse elemente jezika HTML.
html	Značka pove brskalniku, da gre za HTML dokument. Atribut <code>xmlns</code> definira podzvrst uporabljenega XML jezika.
head	Element označuje glavo dokumenta in lahko vsebuje osnovne informacije o dokumentu (npr. naslednji dve obrazloženi znački). Te vsebine brskalnik ne prikaže uporabniku.

se nadaljuje...

...nadaljevanje

title	Značka definira naslov dokumenta, ki se prikaže tudi v naslovni vrstici brskalnika.
style	Definira stil dokumenta, atribut <code>type</code> definira vrsto vsebine. Znotraj te značke določimo vrsto, barvo in velikost pisave, barvo ozadja, itd.
body	Označuje telo dokumenta, v katerem je zapisana vsa vsebina dokumenta, ki se v brskalniku prikaže kot spletna stran.
a	Značka definira sidrišče, na katerega navadno pritrdimo hiperpovezavo na drug dokument. To naredimo s pomočjo atributa <code>href</code> , ki mu definiramo ciljni naslov hiperpovezave.

Preglednica 16: Razlaga uporabljenih PHP funkcij in ukazov v kodi datoteke *izbira.php* (<http://si.php.net>, 24.10.2007).

Funkcija	Razlaga
<code>mysql_connect</code>	Funkcija odpre povezavo do MySQL strežnika. Argumenti funkcije so ime strežnika, uporabniško ime in geslo.
<code>die</code>	V primeru, da povezava z MySQL strežnikom ni mogoča (razlog je lahko nedelujoč strežnik ali napačno vneseni argumenti funkcije za povezavo), funkcija prekine delovanje PHP skripte. Argument funkcije je lahko poljubno besedilo, ki se izpiše ob izvedbi funkcije.
<code>mysql_select_db</code>	Funkcija izbere bazo podatkov, ki je aktivna na izbranem MySQL strežniku. Argument je ime baze, v mojem primeru je to <i>GPS</i> .
<code>\$_GET</code>	To je v PHP jezik vgrajena globalna spremenljivka, ki deluje kot funkcija za prenašanje podatkov med spletnimi stranmi preko metode »HTTP GET«. Z njeno pomočjo preberemo vrednosti spremenljivk iz URL naslova PHP datoteke (slika 21) in definiramo lokalno spremenljivko za uporabo v nadaljevanju skripte. Za argument ji navedemo ime spremenljivke, ki jo želimo pridobiti.
<code>if</code>	S pogojnim stavkom <code>if</code> uveljavimo določeno izjavo (v zavutih oklepajih), pod pogojem, da je argument stavka (v okroglih oklepajih) resničen.

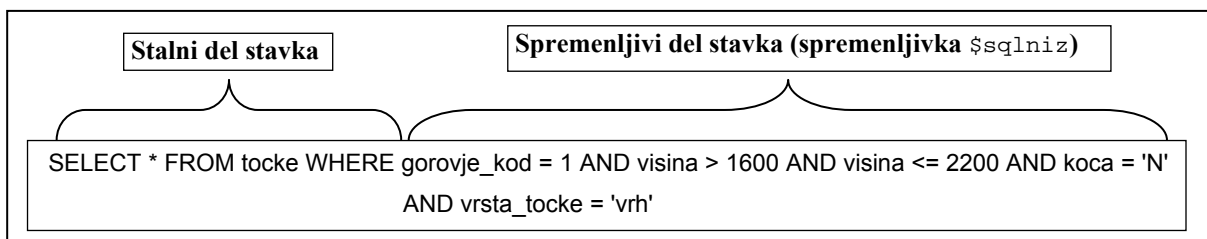
Se nadaljuje...

...nadaljevanje

else	Če argument stavka <code>if</code> ni resničen, lahko s stavkom <code>else</code> uveljavimo neko drugo izjavo.
<code>mysql_query</code>	Funkcija pošlje MySQL strežniku poizvedbo, opredeljeno z SQL stavkom v argumentu funkcije. Poizvedba se vrši na trenutno aktivni bazi, naenkrat je možna le ena poizvedba.
<code>mysql_fetch_array</code>	Funkcija od strežnika pridobi polja iz tabele, ki ustrezajo poizvedbi.
echo	Je neobičajna funkcija s posebnimi lastnostmi, njena osnovna naloga je, da vrne enega ali več nizov na uporabniški vmesnik.
while	Zanka uveljavlja določeno izjavo vse dokler je izjava v argumentu zanke resnična. Izvaja se iterativno, dokler je pogoj izpolnjen. Če pogoj ni izpolnjen v prvi iteraciji, se zanka ne izvede.
<code>printf</code>	Funkcija vrne oblikovan niz, kot je naveden v argumentu funkcije.
<code>mysql_close</code>	Zapre povezavo z MySQL strežnikom.

Spremenljivkam v datoteki *izbira.php* lokalno definiramo imena, neodvisno od imen iz datoteke *index.php*. Pomembna je le pravilna uporaba funkcije `$_GET` in pripadajočih elementov. Za boljšo preglednost nad programskimi skriptami lahko uporabljamo ista poimenovanja spremenljivk.

Glavna naloga datoteke je posredovati MySQL strežniku uporabnikovo zahtevo. Izbor se izvrši enkrat, zato je bilo potrebno oblikovati en SQL stavek, ki lahko izbira iz baze točk na podlagi enega ali več kriterijev. Pri tem sem izkoristil možnost povezovanja znakovnih nizov v PHP jeziku.



Slika 22: Sestava SQL stavka za izbor točk po uporabnikovih kriterijih.

SQL stavek je sestavljen iz dveh delov. Prvi del je neodvisen od uporabnika in pove SQL strežniku, naj iz tabele *tocke* izbere polja, ki bodo ustrezala kriterijem iz drugega dela stavka. Drugi del stavka je znakovni niz, ki je definiran s spremenljivko `$_sqlniz`. Ta se glede na uporabnikov izbor oblikuje s pomočjo `if` pogojnih stavkov. Če izbor zahteva nek kriterij, se ta pripiše v niz SQL stavka, sicer ne. Da spremenljivi del stavka ne bi ostal prazen (tako zahteva skladnja jezika SQL), se kriterij za izbiro gorovja pripiše v niz v vsakem primeru.

Izbrana polja iz tabele *tocke* so celotne vrstice vseh točk, ki ustrezajo kriterijem. Iz celotne vrstice posamezne točke potrebujemo le identifikator in ime točke. S pomočjo zanke `while` in funkcije `printf` oblikujemo tabelo vseh izbranih točk, v kateri so nanizana njihova imena. Zanka `while` za vsako točko iz tabele *tocke*, ki ustreza kriterijem uporabniškega izbora, ustvari v tabeli datoteke *izbira.php* novo vrstico, kamor zapiše ime točke in nanj pripne ustrezno hiperpovezavo na datoteko *gradnja_poti.php*. V hiperpovezavo se v URL naslov zapiše identifikator točke, tako da klik na ime točke preusmeri uporabnika na datoteko *gradnja_poti.php*, ki s funkcijo `$_GET` prebere identifikator začetne točke iz URL naslova in ponudi uporabniku gradnjo poti iz te točke na druge.

Na koncu strani *izbira.php* se nahaja še hiperpovezava na isto datoteko, kjer kriteriji za izbiro točk niso definirani, torej se v tabelo zapišejo vse točke iz tabele *tocke*.

6.2 Popravljanje skrajšanih imen točk v polna imena

Vse točke, uporabljene na spletnem mestu, imajo svoja enolična poimenovanja. V SQL tabeli *tocke* so navedena v stolpcu *ime*, pojavijo pa se še v datotekah *tocke.wpt*, *baza.kml*, *baza_pes.kml*, *baza_turno.kml* in *baza_kolo.kml*. Točke na spletnem mestu lahko v povezavi s krajevnimi poimenovanji grobo razdelimo na dva sklopa:

- Točke, ki krajevno sovpadajo z uradnim poimenovanjem kraja, kot ga lahko najdemo v evidenci Register zemljepisnih imen (REZI). Avtorja spletnega mesta sta točke poimenovala po uradnem poimenovanju, vendar so imena prilagojena dolžini največ 20 znakov. Vsa uradna poimenovanja dolžine več kot 20 znakov so skrajšana na različne načine, kar lahko zmede uporabnika pri brskanju med točkami.

- Točkam, ki so zajete na neimenovanih krajih, sta avtorja dala svoja poimenovanja. Nekatere so prejele ime po nadmorski višini, na kateri se nahajajo, nekatere pa po bližnjem uradno poimenovanem kraju. Vse take točke so v SQL tabeli *tocke* v polju *vrsta_tocke* rangirane kot karkoli, kar pomeni, da niso vrhovi ali izhodišča.

Vse točke v bazi so tudi brez sičnikov in šumnikov. Razlogi za okrnjena poimenovanja so v preprostejši avtomatski obdelavi podatkov. Večina točk je poimenovanih ob samem zajemu s pomočjo ročnega GPS sprejemnika. Večina teh naprav ima omejen prostor za poimenovanja in nima možnosti uporabe sičnikov in šumnikov.

Popolna poimenovanja točk so boljši nosilci informacij in pomagajo k boljši uporabniški izkušnji na spletnem mestu. Prednost krajših imen pa je preprostost uporabe v avtomatiziranju obdelave in lažji prikaz na natrpanih kartografskih podlagah. Uporaba imen točk v kombinaciji s kartografskimi prikazi majhnih meril v programih OziExplorer in Google Earth je velikokrat moteča zaradi prekrivanja imen na območjih gostega zajema točk. Popolna imena točk v primerjavi z okrajšanimi imeni na kartografskih podlagah zavzamejo več prostora.

Razlogov za in proti preimenovanju točk je torej veliko. Odločil sem se za uporabo popolnih imen le na spletnem mestu. Poimenovanja v vektorskih slojih tako ostanejo nespremenjena. Zaradi trenutnega ujemanja podatkov o imenih točk v zgoraj naštetih WPT in KML datotekah z imeni točk v SQL tabeli *tocke* sem se odločil dopolniti tabelo z dodatnim stolpcem, v katerem sem zbral popolna imena točk. Stolpec bo v izpisovanju na uporabniški vmesnik nadomestil stolpec z okrajšanimi imeni.

6.2.1 Dopolnjevanje SQL tabele *tocke*

V tabeli sem ustvaril nov stolpec z imenom *polno_ime*. Tip polja v stolpcu je varchar z največjo dolžino niza znakov 50 znakov, ki sem jo določil na osnovi najdaljšega uradnega poimenovanja točke v obstoječi bazi. Polje nima privzete vrednosti.

Viri za popolna poimenovanja točk so večinoma državne topografske karte DTK 50 in DTK 25. Za točke, zajete na manj pomembnih krajih sem kot vire uporabil topografske načrte TTN 5 in TTN 10, kjer so zajeta vsa imena iz Registra zemljepisnih imen.

Imena sem dopolnil tudi točkam, zajetim na lokacijah brez poimenovanj. Model poimenovanja sem poenotil za vse take točke. Tako sem na prvo mesto navajal nadmorske višine v metrih, za vezajem pa bližino pomembnejše točke (npr. »831m – pod Krimom«) ali pa sem navedel pojav v bližini točke (npr. »1780m – pastirska koč«).

6.2.2 Dopolnjevanje uporabniškega vmesnika

Popolna imena točk se na dopolnjenem spletnem mestu pojavijo v datotekah *izbira.php* in *gradnja_poti.php*. V prvi datoteki se imena izpišejo v tabelo izbranih točk. V argumentu uporabljene funkcije `printf` sem ustrezno zamenjal izpis imena iz SQL tabele *tocke*.

V drugi datoteki se ime točke pojavi na dveh mestih, kot ime začetne in kot ime možne končne točke. Identifikator začetne točke je z metodo `$_GET` prejet iz datoteke *izbira.php*, in vključen v SQL stavek, ki vrne popolno ime točke. Možne končne točke so ponujene v tabeli za gradnjo poti. Zanka `while` ugotovi, katere so možne končne točke iz izbrane začetne točke s pomočjo SQL stavka iz baze izbere njihova popolna imena in jih naniza v tabelo. Spremembe so bile torej potrebne le na SQL stavkih.

6.3 Povezava na stran z opisom točke

Da bi naredil izbiranje med točkami čim bolj udobno, sem se odločil izkoristiti še eno orodje, ki je preprosto, pa vendar zelo učinkovito. Uporabnik ima pri točki navedeno hiperpovezavo na stran, kjer se nahajajo dodatne informacije o točki, ki so lahko slike, opisi okolice in dostopa do točke in drugo. Seveda vse točke iz baze nimajo objavljenega opisa na spletu, najdemo pa lahko večino vrhov, planinskih koč in izhodišč. Kjer opis točke ne obstaja, hiperpovezava preusmeri uporabnika na stran z obrazložitvijo, da opisa ni na voljo.

Strani z opisi je za nekatere znane točke na pretek, za neznane pa jih ni ali pa jih je težko najti. Nahajajo se na različnih spletnih mestih, vendar ima najbolj popolno bazo točk z opisi spletno mesto www.hribi.net. Večina hiperpovezav za opise uporabnika preusmeri na to spletno mesto, zato sem se odločil zaprositi za dovoljenje za uporabo tam objavljenih informacij. Navkljub profitni zasnovi dejavnosti spletnega mesta so uporabo za namen diplomske naloge dovolili.

6.3.1 Dopolnjevanje SQL tabele *tocke*

Tabelo sem razširil za polje tipa *varchar*, in mu nadel ime *opis*. Največja možna dolžina niza v polju je 255 znakov, za privzeto vrednost polja pa je naveden URL naslov strani z razlago, da opisa ni na voljo (http://localhost/GPS/Slika_ni.html). V polja sem nato dodal spletne naslove strani z najboljšimi obstoječimi opisi točk.

6.3.2 Dopolnjevanje uporabniškega vmesnika

Hiperpovezave na strani z opisi sem dodal v tabelo na strani *izbira.php*. Tabela je tako razširjena za en stolpec, kjer je v vsaki vrstici pod besedo »Opis« nameščena hiperpovezava na opis točke iz iste vrstice (slika 23). V datoteki *izbira.php* sem v funkcijo `printf` dodal HTML značke za izpis opisanih celic tabele na uporabniški vmesnik. Hiperpovezava na stran z opisom se glede na uporabnikov izbor začetnih točk za gradnjo poti prepíše v tabelo iz SQL baze.

V spodnji tabeli se nahajajo točke po vašem izboru. Kliknite na željeno točko za začetek gradnje poti:

Možic	Opis
Viševnik	Opis
Šija	Opis
Sleme	Opis
Slatna	Opis
Kreda	Opis
Debela peč	Opis
Vrh nad Peski	Opis
1905m - Vrh nad Gracijo	Opis

[Išči po bazi vseh točk](#)

Slika 23: Prikaz strani *izbira.php* v brskalniku.

Stran z razlago, da opis ni na voljo, je samostojna stran na spletnem mestu, definira pa jo datoteka *slika_ni.html*. Oblikovanje strani je usklajeno s celotnim spletnim mestom, vsebina pa je le vrstica besedila »Opis točke ni na voljo.«, ki se v HTML datoteki nahaja v telesu besedila.

6.4 Opremljanje poti z atributi

Na obstoječem spletnem mestu »Planinske in kolesarske poti ter turni smuki z GPS podporo« je edina opisna informacija o poteh, ki jih predstavljajo zbrane GPS sledi, možen način uporabe poti. Informacija je v spletno mesto vključena z datoteko *gradnja_poti.php*. V tabeli na strani, ki jo datoteka izpiše, je informacija navedena opisno k vsaki ponujeni poti. Možnost izbiranja poti glede na način uporabe je ponujena povsem samostojno, na strani *index.php*, kjer lahko uporabnik sname celotne baze vseh pešpoti ali kolesarskih poti ali turnih smuk. Izbor je omejen na posamezne načine in ne ponuja možnosti kombiniranja.

Zelo dobra izpopolnitev spletnega mesta bi bila možnost izbiranja poti po več kriterijih in po posameznih kombinacijah kriterijev. Seveda bi morala biti ta možnost združena z možnostjo izbiranja točk po kriterijih. Žal bi bil tak model spletnega mesta zelo zahteven za vzpostavitev. Model bi grobo spremenil obstoječo informacijsko arhitekturo spletnega mesta, potrebna bi bila tudi dopolnitev uporabljene programske osnove, saj je obstoječa primerna le za enostavne modele izbiranja in kombiniranja podatkov iz SQL baze. Zahtevnejši modeli za izbiranje zahtevajo uporabo bolj namenskih programskih jezikov, kot je npr. javascript.

Ker je obravnavano spletno mesto brezplačne narave in ga dopolnjujem za namen diplomske naloge, moram stvari racionalizirati. Izvedba zgoraj omenjene ideje bi bila časovno preveč zahtevna. Zato sem se odločil, da atributne podatke o poteh vključim v spletno mesto le v opisni obliki, brez možnosti izbiranja po njih.

Opisni podatki o poteh se v dopolnjenem spletnem mestu pojavijo na dveh straneh. Na strani *gradnja_poti.php* sta v tabelo poleg podatka o možnem načinu uporabe posamezne poti dodani dve polji. Prvo navaja čas hoje, drugo pa zahtevnost posamezne poti. Na strani

rezultati.php se poleg metričnih podatkov, izračunanih na podlagi vhodnih PLT datotek izpiše še skupen čas hoje po zgrajeni poti.

6.4.1 Dopolnjevanje SQL tabele *poti*

Podatki za čas hoje in zahtevnost določenih odsekov poti so zaradi pomanjkanja virov za namen diplomske naloge izmišljeni. Pri dopolnjevanju tabel sem približne vrednosti določal na podlagi dolžine in višinskih razlik na posamezni poti.


	Field	Type	Attributes	Null	Default	Extra
<input type="checkbox"/>	id	int(11)		No		auto_increment
<input type="checkbox"/>	id_zac	int(11)		Yes	NULL	
<input type="checkbox"/>	id_kon	int(11)		Yes	NULL	
<input type="checkbox"/>	plt	varchar(20)		Yes	NULL	
<input type="checkbox"/>	id_poti	int(1)		No	0	
<input type="checkbox"/>	cas_hoje_min1	int(10)	UNSIGNED	No	0	
<input type="checkbox"/>	cas_hoje_min0	int(10)	UNSIGNED	No	0	
<input type="checkbox"/>	zahtevnost	varchar(40)		No	Ni podatka	

Slika 24: Struktura dopolnjene tabele *poti* (izsek iz PhpMyAdmin).

Tabelo *poti* sem razširil za tri polja. Prvi dve vsebujeta čas hoje v minutah, imenujeta se *cas_hoje_min1* in *cas_hoje_min0*. Prvo vsebuje čas hoje v smeri skladni s smerjo vhodne PLT datoteke, drugo pa čas hoje v obratni smeri. Obe polji sta tipa *integer* z največjo velikostjo niza 10. Polji imata definirani privzeti vrednosti 0 in sta označeni z atributom »unsigned«, kar pomeni, da ne moreta vsebovati negativnih vrednosti.

Tretje novo polje je namenjeno zapisu zahtevnosti. Polje tipa *varchar* se imenuje *zahtevnost*, največja dolžina niza je 40 znakov, privzeta vrednost pa je »Ni podatka«. Za opis zahtevnosti sem uporabil deset različnih razredov, ki so prikazani v preglednici 17. Sledijo si od najmanj proti najbolj zahtevnemu.

Preglednica 17: Razredi zahtevnosti poti, ki so uporabljeni v SQL tabeli *poti*.

	Poimenovanje zahtevnosti (polje <i>zahtevnost</i>)
Privzeta vrednost	Ni podatka
Najmanjša zahtevnost  Največja zahtevnost	Lahka označena pot
	Lahka neoznačena pot
	Lahko brezpotje
	Zahtevna označena pot
	Zahtevna neoznačena pot
	Zahtevno brezpotje
	Zelo zahtevna označena pot
	Zelo zahtevna neoznačena pot
	Zelo zahtevno brezpotje

6.4.2 Dopolnjevanje uporabniškega vmesnika

Čas hoje in zahtevnost odseka poti se na strani *gradnja_poti.php* pojavita v tabeli ob navedenih ostalih podatkih o poti (slika 24). V datoteki sem v funkcijo `printf`, ki vsebuje argumente za izpis tabele, na zadnji dve mesti dodal spremenljivki za izpis časa hoje in zahtevnosti poti. Vrednosti spremenljivk sta definirani ob črpanju podatkov iz baze z SQL stavkom. Koda izbere ustrezen čas hoje glede na smer hoje. Ustrezna izbira izmed polj *cas_hoje_min1* in *cas_hoje_min0* je izvedena s pomočjo `if` stavka.

Izberi naslednji del poti od Kreda do:					
Planina Dedno polje	po poti	PlDedno_Kreda	način: peš	čas hoje: 45 min	zahtevnost: Zahtevna označena pot
2045 - pod Slatno	po poti	Kreda_2045Sl	način: peš	čas hoje: 10 min	zahtevnost: Lahka neoznačena pot
Pripravi datoteke , ki pripadajo do sedaj zgrajeni poti.					
Nova pot.					

Slika 25: Prikaz strani *gradnja_poti.php* v brskalniku.

Na strani *rezultati.php* se poleg ostalih metričnih podatkov izpiše tudi skupen čas hoje v minutah po zgrajeni poti. V datoteki *rezultati.php* sem spremenljivki za čas definiral vrednost 0, znotraj `for` zanke pa se spremenljivki prišteje vrednost časa hoje za vsako izbrano pot. Glede na smer hoje se s pomočjo `if` stavka izbere prava vrednost časa hoje. Po izvedeni zanki `for` se spremenljivka kot vsota časov zapiše v funkcijo `printf`, katera jo s spremnim tekstom zapiše na stran v brskalniku.

7. SKLEP

V nalogi sem predstavil spletno mesto »Planinske in kolesarske poti ter turni smuki z GPS podporo«, ki sem ga v sklopu naloge dopolnil na več področjih. Dopolnjevanje je zahtevalo podrobno seznanitev s sestavo spletnega mesta in z uporabljenimi tehnologijami, na kateri temelji. Na kratko so opisane osnovne značilnosti programskih jezikov (X)HTML in PHP, podatkovne baze MySQL, katero krmili programski jezik SQL, ter strežniškega programa Apache.

Dopolnjevanja sem se kot laični programer lotil sistematsko in izvedel dopolnitve, ki so ključne za učinkovito delovanje spletnega mesta in niso preveč časovno potratne za učenje in izvedbo. Predvsem sem se osredotočil na prijaznost spletnega mesta uporabniku in na nabor informacij, ki jih uporabnik lahko pridobi. Na ta način sem se podrobno seznanil s postopki izdelave in optimizacije uporabniškega vmesnika na osnovi jezikov PHP in (X)HTML ter z izdelavo in vodenjem podatkovne baze MySQL.

Podatkovna baza, ki stoji za spletnim mestom, vsebuje veliko različnih tipov podatkov, ki so vsi vezani na lokacijo v prostoru. Glavni podatki, brez katerih spletno mesto ne bi obstajalo, so GPS sledi, ki so od zajema do objave podvržene različnim postopkom obdelave. Na tem mestu sem poleg opisov izvedenih prilagoditev sledi namenil del diplomske naloge tudi izpeljavi in primerjavi štirih različnih metod za določitev višine točke na podlagi DMR. Metode uporabljajo različne pristope k rešitvi problema, za najbolj primerno pa se je izkazala preprosta metoda na podlagi premic v prostoru.

Poleg metričnih podatkov so sestavni del baze tudi semantični podatki, katerih večino sem vključil v bazo v okviru dopolnjevanja. Izbor podatkov sem prilagodil dostopnosti le-teh in smiselnosti objave. Ti podatki so atributi GPS sledi in njihovih točk, večinoma pa sem jih pridobil iz obstoječih virov na medmrežju. Zaradi velikega obsega baze sem s podatki opremil približno polovico sledi, izbor sem prilagodil primernosti za predstavitev delovanja dopolnjene različice spletnega mesta. Poleg umestitve atributnih podatkov v bazo je bilo potrebno vsak podatek tudi primerno vključiti v uporabniški vmesnik.

Dopolnjena različica spletnega mesta še ni primerna za objavo na medmrežju, veliko dela bo še potrebno posvetiti dosledni in pravilni dopolnitvi podatkovne baze. Kljub temu je testna različica objavljena na spletnem mestu z naslovom <http://pclakner.fgg.uni-lj.si/GPSTest>.

Dopolnjeno spletno mesto je kljub konkurenci ponudnikov podobnih vsebin edinstveno in bi si zaslužilo dodatne dopolnitve, ki pa presegajo meje diplomske naloge. Predvsem bi bile dobrodošle dopolnitve z dodatnimi vsebinami in izpopolnitev oblikovanja strani. Nekoliko težje izvedljiva, vendar koristna dopolnitev bi bila tudi možnost, da uporabniki sami nalagajo GPS sledi na spletno mesto, kjer bi se te samodejno predelale in vključile v bazo.

Tehnologije, ki sem jih opisal v nalogi, nudijo odlično podporo avtomatizaciji postopkov. Programski jezik PHP podpira matematične in grafične operacije, zato se izkazuje kot primerno orodje za obdelavo prostorskih podatkov. Dejstvo, da je medmrežje najboljši medij za izmenjavo podatkov, me napeljuje k sklepu, da bi lahko v okviru študija prostorske informatike posvetili več pozornosti poznavanju programiranja in avtomatske obdelave podatkov. Predvsem bi bilo koristno spoznavanje brezplačnih spletnih tehnologij, v katero bi lahko študente vključili kot aktivne razvijalce programskih rešitev.

VIRI

ESRI. 2001. White Paper: ArcGIS Geostatistical Analyst. str. 8 – 10.
<http://www.esri.com/library/whitepapers/pdfs/geostat.pdf> (29.11.2007)

E-uspeh. Apache strežnik
<http://www.e-uspeh.com/baza-znanja/entry/12/61> (15.12.2007)

GNU Operating System
<http://www.gnu.org> (17.12.2007)

Golež, B. 2006. Diplomsko delo: Možnosti zajema in prikaza kolesarskih poti. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in Geodezijo: str. 9, 14.

Google Earth
<http://earth.google.com> (14.10.2007)

Hribi.net
<http://www.hribi.net> (8.10.2007)

MaFiRa-Wiki. Bisekcija
<http://wiki.fmf.uni-lj.si/wiki/Bisekcija> (4.12.2007)

OziExplorer
<http://www.oziexplorer.com> (18.10.2007)

PHP.net
<http://si.php.net> (24.10.2007)

PHPlot Project
<http://freshmeat.net/projects/phplot> (17.12.2007)

Planinska zveza Slovenije

<http://www.pzs.si> (8.10.2007)

Podobnikar, T. Digitalni model reliefa (DMR) Slovenije iz različnih podatkov. Ljubljana. Znanstvenoraziskovalni center SAZU.

<http://iaps.zrc-sazu.si/index.php?q=sl/node/70> (25.11.07)

Podobnikar, T. 2002. Model zemeljskega površja – DMR ali DMV?. Ljubljana. Geodetski vestnik 46/2002-4: 7 str.

Podobnikar, T., Mlinar, J. 2006. Izdelava in vzdrževanje digitalnega modela reliefa Slovenije z integracijo obstoječih virov. Ljubljana. Geodetski vestnik 50/2006-3: 8 str.

Pušaver, T. 2002. Kaj je HTML?. E-gradiva. Maribor. Srednja elektro-računalniška šola

<http://www.s-sers.mb.edus.si/gradiva/w3/html/index.html> (15.12.2007)

Radovan, D. 2002. Geoinformacijska terminologija: digitalni model reliefa, digitalni model višin, grid. Ljubljana. Geodetski vestnik 46/2002-4. 4 str.

Slovar informatike

http://www.islovar.org/iskanje_enostavno.asp (17.12.2007)

Štancar, M., Klemen S. 2002. PHP in MySQL na spletnem strežniku Apache. Ljubljana, Pasadena: 128 str.

The Uniform Server

<http://www.uniformserver.com> (11.12.2007)

The Web Standards Project

<http://www.webstandards.org/learn/articles/askw3c/oct2003> (15.12.2007)

Topografix. GPX

<http://www.topografix.com/gpx.asp> (23. 11. 2007)

Vilfan, I., Verbovšek, T. Uporaba GPS v Sloveniji.

http://www2.arnes.si/~gljsentvid10/GPS_sistemi.html (24.11.07)

Wamp Server

<http://www.wampserver.com/en> (17.12.2007)

Wikipedia, SQL

<http://en.wikipedia.org/wiki/SQL> (17.12.2007)

Wikipedia, KML

http://en.wikipedia.org/wiki/Keyhole_Markup_Language (14.10.2007)

Wikipedia, Kriging

<http://en.wikipedia.org/wiki/Kriging> (29.11.2007)

Wikipedia, SWOT Analysis

http://en.wikipedia.org/wiki/SWOT_analysis (2.10.2007)

Wikipedia, prosta enciklopedija

<http://sl.wikipedia.org> (8.10.2007)

World Wide Web Consortium. W3 Schools. Introduction to PHP.

http://www.w3schools.com/php/php_intro.asp (15.12.2007)

World Wide Web Consortium. W3 Schools. HTML 4.01 / XHTML 1.0 Reference.

<http://www.w3schools.com/tags> (24.10.2007)

PRILOGA A: Dokaz k linearni metodi v programu Mathematica

```
In[144]:= zAB = zA + k1 * (xT - xA) ; k1 = (zB - zA) / a ;  
zCD = zD + k2 * (xT - xD) ; k2 = (zC - zD) / a ;  
zAD = zA + k4 * (yT - yA) ; k4 = (zD - zA) / a ;  
zBC = zB + k5 * (yT - yB) ; k5 = (zC - zB) / a ;  
k3 = (zCD - zAB) / a ;  
k6 = (zBC - zAD) / a ;  
xB = xA + a ; yB = yA ;  
xC = xA + a ; yC = yA + a ;  
xD = xA ; yD = yA + a ;  
T1 = zAB + k3 * (yT - yA) // Simplify  
T2 = zAD + k6 * (xT - xA) // Simplify  
Out[153]=  $\frac{1}{a^2} (a^2 zA + a (xA (zA - zB) + xT (-zA + zB) + (yA - yT) (zA - zD)) +$   
 $(xA - xT) (yA - yT) (zA - zB + zC - zD))$   
Out[154]=  $\frac{1}{a^2} (a^2 zA + a (xA (zA - zB) + xT (-zA + zB) + (yA - yT) (zA - zD)) +$   
 $(xA - xT) (yA - yT) (zA - zB + zC - zD))$   
In[155]:= razlika = T1 - T2 // Simplify  
Out[155]= 0
```

PRILOGA B: Programska koda datoteke *index.php*

```

1  <?php
2  session_start();
3  $_SESSION['ppp']="";
4  ?>
5  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
6  <html xmlns="http://www.w3.org/1999/xhtml">
7  <head>
8  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
9  <link rel="shortcut icon" href="favicon.ico" >
10 <title>GPS planinske poti</title>
11 <style type="text/css">
12 <!--
13 body,td,th {
14 font-family: Georgia, Times New Roman, Times, serif;
15 font-size: 18px;
16 color: #000000;}
17 body {
18 background-color: #00CCFF;
19 background-image: url(p.jpg);
20 background-repeat: repeat-y;}
21 -->
22 </style></head>
23
24 <body>
25 <h1>Planinske in kolesarske poti ter turne smuke z GPS podporo (v nastajanju...) </h1>
26 <p>&nbsp;</p>
27 <p>Pot lahko zgradi&scaron; na dva na&#269;ina:</p>
28 <p>&nbsp;</p>
29 <ul>
30 <li>S pomo&#269;jo <a href="baza.kml">Google Earth datoteke(kml)</a>
   <a href="baza.plt">(plt)</a> celotne baze poti, kjer za&#269;ne&scaron;
   s klikom na za&#269;etno to&#269;ko in si zgradi&scaron; &#382;eljeno pot. Lahko pa si
   izbereš samo <a href="baza_pes.kml">peš poti (kml)</a> <a href="baza_pes.plt">(plt)</a>,
   samo <a href="baza_turno.kml">turne smuke (kml)</a> <a href="baza_turno.plt">(plt)</a> ali
   samo <a href="baza_kolo.kml">kolesarske ture (kml)</a> <a href="baza_kolo.plt">(plt)</a>.
   </li>
31 </ul>
32 <br>
33 <ul>
34 <li>S pomo&#269;jo spodnjega obrazca za izbiro to&#269;k po kriterijih. </li>
35 </ul>
36 <p><br>
37 Pregleden rezultat dobi&scaron;, &#269;e v Googlovi Zemlji odpre&scaron; zgrajeno pot (kml
   datoteka) preko celotne baze.<br>
38 Uporabna je tudi datoteka <a href="tocke.wpt">to&#269;k</a> (OziExplorer) in
39 <A href="plastnice">plastnice</A> (vir podatkov Geodetski zavod Slovenije).</p>
40 <p><br>
41 Statistika <a href="stat.html">baz</a>.</p>
42 <p>In seveda: ker podatki lahko vsebujejo napake, je pri njihovi uporabi v naravi koristno uporabiti
   vsaj malo zdrave pameti. Bolj natan&#269;no:</p>

```



```
91 <option value='600'>600</option>
92 <option value='800'>800</option>
93 <option value='1000'>1000</option>
94 <option value='1200'>1200</option>
95 <option value='1400'>1400</option>
96 <option value='1600'>1600</option>
97 <option value='1800'>1800</option>
98 <option value='2000'>2000</option>
99 <option value='2200'>2200</option>
100 <option value='2400'>2400</option>
101 <option value='2600'>2600</option>
102 <option value='3000'>3000</option>
103 </select></td></tr>
104
105 <tr><td>Izberi to&#269ko glede na obstoj ko&#269e:</td>
106 <td>
107 <select name="koca">
108 <option value="0" selected="selected">vseeno</option>
109 <option value='Y'>je ko&#269a</option>
110 <option value='N'>ni ko&#269e</option>
111 </select></td></tr>
112
113 </table><br>
114 ŹeliŹ za&#269eti gradnjo poti na izhodiŹ&#269u, vrhu, ostalih to&#269kah ali ti je vseeno?<br>
115 <table>
116 <tr><td>IzhodiŹ&#269e:</td>
117 <td><input type="radio" name="vrsta_tocke" value="izhodisce"></td></tr>
118 <tr><td>Vrh:</td>
119 <td><input type="radio" name="vrsta_tocke" value="vrh"></td></tr>
120 <tr><td>Ostale to&#269ke:</td>
121 <td><input type="radio" name="vrsta_tocke" value="karkoli"></td></tr>
122 <tr><td>Vseeno mi je:</td>
123 <td><input type="radio" name="vrsta_tocke" value=""></td></tr>
124 </table>
125 <br>
126
127 <input type=submit value="Izberi">
128 </form>
129 <br>
130 </body>
131 </html>
```

PRILOGA C: Programska koda datoteke *izbira.php*

```
1 <?php
2 session_start();
3 ?>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head>
7 <title>Izbira poti</title>
8 <style type="text/css">
9 <!--
10 body,td,th {
11 font-family: Georgia, Times New Roman, Times, serif;
12 font-size: 18px;
13 color: #000000;
14 }
15 body { background-color: #00CCFF;}
16 -->
17 </style></head>
18 <body>
19 <?php
20 mysql_connect("localhost","koren","pepe") or die ("Povezava ni možna");
21 mysql_select_db("GPS");
22
23 //spremenljivke, dolocene na podlagi vrednosti iz obrazca datoteke index.php
24 $gorovje = $_GET['gorovje'];
25 $visina_min = $_GET['visina_min'];
26 $visina_max = $_GET['visina_max'];
27 $koca = $_GET['koca'];
28 $vrsta_tocke = $_GET['vrsta_tocke'];
29
30 //pogojni stavki za izbiranje tock po kriterijih iz SQL tabele tocke
31 if ($gorovje=='0') {$g = "(gorovje_kod = 1 OR gorovje_kod = 2 OR gorovje_kod = 3 OR
   gorovje_kod = 4 OR gorovje_kod = 5 OR gorovje_kod = 6 OR gorovje_kod = 7 OR
   gorovje_kod = 8)";} else {$g = "gorovje_kod = $gorovje";}
32 if ($visina_min=='0') {$v_min = "";} else {$v_min = " AND visina > $visina_min";}
33 if ($visina_max=='0') {$v_max = "";} else {$v_max = " AND visina <= $visina_max";}
34 if ($koca=='0') {$k = "";} else {$k = " AND koca = '$koca'";}
35 if ($vrsta_tocke=="") {$v = "";} else {$v = " AND vrsta_tocke = '$vrsta_tocke'";}
36
37 //niz pogojev za izbiro iz SQL tabele
38 $sqlniz = "$g$v_min$v_max$k$v";
39 $tocke = mysql_query("SELECT * FROM tocke WHERE $sqlniz") or die ("Ne morem prebrati
   tock.");
40
41 echo "V spodnji tabeli se nahajajo to&#269;ke po vašem izboru. Kliknite na željeno to&#269;ko za
   za&#269;etek gradnje poti:";
42 echo '<br><br>';
43 echo "<table width=380 border=1>";
44 while($zapis=mysql_fetch_array($tocke)){
45 printf("<tr><td><a href='Gradnja_poti.php?idtocke=%s&idpotep=\\'>%s</a></td><td><a
```

```
        href=\"%s\"> Opis</a></td></tr>", $zapis["id"], $zapis["polno_ime"], $zapis["opis"]);
46 }
47 echo "</table>";
48 mysql_close();
49 ?>
50 <br><a href="izbira.php?gorovje=0&visina_min=0&visina_max=0&koca=0">Iš&#269;i po bazi
    vseh to&#269;k</a>
51 </body>
52 </html>
```

PRILOGA D: Programska koda datoteke *gradnja_poti.php*

```
1 <?php
2 session_start();
3 ?>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
8 <title>Gradnja poti</title>
9 <style type="text/css">
10 <!--
11 body,td,th {
12 font-family: Georgia, Times New Roman, Times, serif;
13 font-size: 18px;
14 color: #000000;
15 }
16 body {
17 background-color: #00CCFF;
18 }
19 -->
20 </style></head>
21
22 <body>
23 Izberi naslednji del poti od
24 <?php
25 $nacin[0]="peš";
26 $nacin[1]="turno";
27 $nacin[2]="kolo";
28 $nacin[3]="peš, kolo";
29 $nacin[4]="peš, turno";
30 $nacin[5]="kolo, turno";
31 $nacin[6]="peš, turno, kolo";
32
33 mysql_connect("localhost","koren","pepe") or die ("Povezava ni možna");
34 mysql_select_db("GPS");
35
36 $potep=$_GET['idpotep'];
37 if($potep=="") $_SESSION['ppp'] = ""; else
38 $_SESSION['ppp'] = $_SESSION['ppp'].$potep;
39
40 $idt=$_GET['idtocke'];
41
42 $im = mysql_query("SELECT polno_ime FROM tocke WHERE id = '$idt'");
43 $vrsta = mysql_fetch_row($im);
44 echo "<strong>".$vrsta[0].</strong> do:<br><br>";
45
46 $poti=mysql_query("SELECT * FROM poti WHERE ((id_zac = $idt)OR(id_kon = $idt))" or die
   ("Ne morem prebrati tock."));
47
48 echo "<table border=1 cellpadding=7>";
```

```
49 while($zapis=mysql_fetch_array($poti))
50 {
51 if ($zapis["id_zac"]==$idt) {$smer=1; $xx=$zapis["id_kon"]; $cas=$zapis["cas_hoje_min1"];}
52 else {$smer=0; $xx=$zapis["id_zac"]; $cas=$zapis["cas_hoje_min0"];}
53 $im = mysql_query("SELECT polno_ime FROM tocke WHERE id = '$xx'");
54 $vrsta = mysql_fetch_row($im);
55 $pp=substr('000'.$zapis["id"],-3).$smer;
56
57 $zaht=$zapis["zahtevnost"];
58
59 printf("<tr><td><strong>$vrsta[0]</strong></td><td>po poti </td><td><a
    href=\"Gradnja_poti.php?idtocke=%s&idpotep=%s\">%s</a></td><td>na&#269;in:
    %s</td><td>&#269;as hoje: %s
    min</td><td>zahtevnost: %s</td></tr>",
60 $xx, $pp, substr($zapis["plt"],0,-4),$nacin[$zapis["id_poti"]],$cas,$zaht);
61 }
62 echo "</table>";
63 mysql_close();
64
65 printf("<br><a href=\"rezultati.php\"> Pripravi datoteke</a>, ki pripadajo do sedaj zgrajeni
    poti.<br><br>");
66 printf("<br><a href=\"index.php\"> Nova pot.</a> <br>");
67 ?>
68
69 </body>
70 </html>
```


PRILOGA E: Programska koda datoteke *rezultati.php*

```
1 <?php
2 session_start();
3 $potep = $_SESSION['ppp'];
4 $_SESSION['ppp'] = "";
5 ?>
6 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
7 <html xmlns="http://www.w3.org/1999/xhtml">
8 <head>
9 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
10 <title>Rezultati</title>
11 <style type="text/css">
12 <!--
13 body,td,th {
14 font-family: Georgia, Times New Roman, Times, serif;
15 font-size: 18px;
16 color: #000000;
17 }
18 body {
19 background-color: #00CCFF;
20 }
21 -->
22 </style></head>
23
24 <body>
25 <?php
26 if(strlen($potep)==0){ echo "Na prej&scaron;nji strani izberi vsaj eno pot!<br><br>";} else
27 {
28 mysql_connect("localhost","koren","pepe") or die ("Povezava ni možna");
29 mysql_select_db("GPS");
30
31 $poti = str_split($potep, 4);
32 $n=count($poti)-1;
33
34 $ime="Temp/" . getenv('REMOTE_ADDR') . "_".substr(microtime(),2,3);
35 $datplt=fopen($ime.".plt","w");
36 $glavplt=
37 "OziExplorer Track Point File Version 2.1
38 WGS 84
39 Altitude is in Feet
40 Reserved 3
41 0,3,255,01.01.2006 00:00:00 ,0,0,2,8421376
42 1\r\n";
43 fwrite($datplt, $glavplt);
44 $cas=0;
45
46 for ($i=0; $i < $n+1; $i++)
47 {
48 $lok = str_split($poti[$i], 3);
```

```

49 $idplt=intval($lok[0]);
50 $smer=$lok[1];
51 $plt = mysql_query("SELECT plt FROM poti WHERE id = '$idplt'");
52
53 $array = mysql_fetch_array(mysql_query("SELECT * FROM poti WHERE id = '$idplt'"));
54
55 $vrsta = mysql_fetch_row($plt);
56 $vse =file_get_contents("Plt/".$vrsta[0]);
57 $vse =strstr ($vse, "\r\n "); //preskoci glavo
58
59 if($smer=='1') {
60 $b = str_split($vse, 70);
61 $k=count($b);
62 $cas_hoje = $array["cas_hoje_min1"];
63 for ($l=0; $l < $k+1; $l++) fwrite($datplt,$b[$l]);
64 } else
65 {
66 $a = str_split($vse, 70);
67 $m=count($a)-1;
68 $cas_hoje = $array["cas_hoje_min0"];
69 for ($j=0; $j < $m+1; $j++) fwrite($datplt,$a[$m-$j]);
70 }
71 $cas = $cas+$cas_hoje;
72 }
73
74 fclose($datplt); //plt datoteka narejena
75
76 $datkml=fopen($ime.".kml","w");
77 $glavakml=
78 "<?xml version=\\"1.0\\" standalone=\\"yes\\"?>
79 <kml creator=\\"Tone in Mitja http://www.mlakner.fgg.uni-lj.si/GPS/\\"
      xmlns=\\"http://earth.google.com/kml/2.0\\">
80 <Document>
81 <name>GPS</name>
82 <Folder>
83 <name>Tracks</name>
84 <Placemark>
85 <name>GPS</name>
86 <MultiGeometry>
87 <LineString>
88 <altitudeMode>clampedToGround</altitudeMode>
89 <coordinates>\n";
90 fwrite($datkml, $glavakml);
91
92 $plt =file_get_contents($ime.".plt");
93 $plt =strstr ($plt, "\n "); //preskoci glavo
94 $a=explode("\n",$plt);
95 $m=count($a)-1;
96
97 $st=3.14159265358979/180;
98 $ft=0.3048;
99 $R=6372.795;
100 $s=0;

```

```
101 $sc=0;
102 $sp=0;
103 $min=10000;
104 $max=0;
105
106 $f1=floatval(substr($a[1],2,10))*$st;
107 $l1=floatval(substr($a[1],14,9))*$st;
108 $p1=floatval(substr($a[1],26,7))*$ft;
109
110 for ($j=1; $j < $m+1; $j++)
111 {
112   $vrsta=$a[$j];
113   if(strlen($vrsta)>12){
114     $f2=substr($vrsta,2,10);
115     $l2=substr($vrsta,14,9);
116     $p2=floatval(substr($vrsta,26,7))*$ft;
117     fwrite($datkml,$l2.", ".$f2." 0.00000\r\n");
118     $f2=floatval($f2)*$st;
119     $l2=floatval($l2)*$st;
120     $c=sin($f1)*sin($f2)+cos($f1)*cos($f2)*cos($l1-$l2);
121     if($c < 1.0) $d=$R*acos($c); else $d=0;
122     $s=$s+$d;
123     $dh=$p2-$p1;
124     $sc=$sc+sqrt($d*$d+$dh*$dh/1000000);
125     if($dh>0) $sp=$sp+$dh;
126     if($p2<$min) $min=$p2;
127     if($p2>$max) $max=$p2;
128     $l1=$l2;
129     $f1=$f2;
130     $p1=$p2;
131   }
132 }
133
134 $repkml=
135 "</coordinates>
136 <tessellate>1</tessellate>
137 </LineString>
138 </MultiGeometry>
139 <Style>
140 <LineStyle>
141 <color>7FFF0000</color>
142 <width>4</width>
143 </LineStyle>
144 </Style>
145 </Placemark>
146 <visibility>1</visibility>
147 </Folder>
148 <visibility>1</visibility>
149 </Document>
150 </kml>";
151 fwrite($datkml, $repkml);
152 fclose($datkml); //kml datoteka narejena
153
```

```
154 mysql_close();
155
156 printf("<br>Prenesi k sebi <a href=\"%s\"> plt</a> datoteko.<br>",$ime.".plt");
157 printf("<br>Prenesi k sebi <a href=\"%s\"> kml</a> datoteko.<br>",$ime.".kml");
158 printf("<br>Dolžina poti brez upoštevanja spremembe nadmorske višine je
    ".number_format($s,1)."km.<br>");
159 printf("<br>Dolžina poti z upoštevanjem spremembe nadmorske višine je
    ".number_format($sc,1)."km.<br>");
160 printf("<br>Višinska razlika med kon&#269;no in za&#269;etno to&#269;ko je
    ".number_format(floatval(substr($a[$m],26,7))*$ft-floatval(substr($a[1],26,7))*$ft,0)."m.<br>");
161 printf("<br>Višinska razlika med najvišjo in najnižjo to&#269;ko je "
    ".number_format($max-$min,0)."m.<br>");
162 printf("<br>Pridobljena višina je ".number_format($sp,0)."m.<br>");
163 //cas hoje
164 printf("<br>Skupen &#269;as hoje je $cas minut.<br><br><br>");
165
166 printf("<IMG SRC=\"rezultati_grafika.php?ime=$ime\">");
167 printf("<br><br><br><a href=\"index.php\"> Nova pot.</a> <br>");
168 }
169 ?>
170 </body>
171 </html>
```

PRILOGA F: Izsek SQL tabele *tocke*

SQL result

Host: localhost
Database: gps
Generation Time: Dec 18, 2007 at 12:32 PM
Generated by: phpMyAdmin 2.6.0-beta2 / MySQL 4.0.20a
SQL-query: SELECT * FROM `tocke` LIMIT 0, 20;
Rows: 20

id	ime	polno_ime	sirina	dolzina	visina	gorovje_kod	gorovje	vrsta_tocke	koca	opis
1	Strahomer	Strahomer	45.942612	14.487844	311	4	Goriško, Notranjsko in Snežniško hribovje	izhodisce	N	http://localhost/GPS/Slika_ni.html
2	Krim1	831m - pod Krimom	45.920303	14.477378	831	4	Goriško, Notranjsko in Snežniško hribovje	karkoli	N	http://localhost/GPS/Slika_ni.html
3	Krim	Krim	45.928558	14.471043	1107	4	Goriško, Notranjsko in Snežniško hribovje	vrh	Y	http://www.hrbi.net/gora.asp?gorovjeid=26&id=178
4	Vintgar	Iški Vintgar	45.910519	14.497823	358	4	Goriško, Notranjsko in Snežniško hribovje	izhodisce	N	http://www.ljubljana.si/si/turizem/utrip_ljubljane...
9	Ambroz	Ambrož	46.275105	14.529147	1070	3	Kamniško Savinjske Alpe	izhodisce	N	http://www.hrbi.net/pot.asp?gorovjeid=3&id=351&po...
10	Kriška planina	Kriška Planina	46.292854	14.537225	1470	3	Kamniško Savinjske Alpe	izhodisce	N	http://localhost/GPS/Slika_ni.html
11	Zvoh	Zvoh	46.308323	14.540613	1970	3	Kamniško Savinjske Alpe	vrh	N	http://www.hrbi.net/gora.asp?gorovjeid=3&id=73
43	Gradisce	Gradliš#269e	46.299488	14.575035	778	4	Goriško, Notranjsko in Snežniško hribovje	vrh	N	http://localhost/GPS/Slika_ni.html
44	Zavrata	Zavrata	46.296314	14.552554	1125	3	Kamniško Savinjske Alpe	izhodisce	N	http://localhost/GPS/Slika_ni.html
45	Sneznik	Sneznik	45.589444	14.447386	1797	4	Goriško, Notranjsko in Snežniško hribovje	vrh	Y	http://www.hrbi.net/slika.asp?gora=1619
46	Leskova_dol	Leskova dolina	45.622154	14.458992	810	4	Goriško, Notranjsko in Snežniško hribovje	izhodisce	N	http://localhost/GPS/Slika_ni.html
92	Pariz_dol	Parizov dol	45.708408	14.011232	636	4	Goriško, Notranjsko in Snežniško hribovje	izhodisce	N	http://localhost/GPS/Slika_ni.html
93	Vremcsica	Vremš#269ica	45.688286	14.061617	1027	4	Goriško, Notranjsko in Snežniško hribovje	vrh	N	http://www.zaplana.net/1zleti/Vremcsica/
103	Poldov_Rovt	Poldov rovt	46.420242	13.861153	946	1	Julijske Alpe	izhodisce	N	http://www.hrbi.net/pot.asp?gorovjeid=1&id=22&pot...
104	Bivak2	Bivak 2	46.434032	13.837280	2120	1	Julijske Alpe	karkoli	N	http://www.hrbi.net/pot.asp?gorovjeid=1&id=22&pot...
114	MB_zel_post	Maribor - železniška postaja	46.553604	15.642547	275	7	Pohorje in ostala severovzhodna Slovenija	izhodisce	N	http://www.raifanurope.net/pix/si/station/Maribo...
115	Glazuta_odcep	Glazuta - odcep	46.514507	15.552838	921	7	Pohorje in ostala severovzhodna Slovenija	karkoli	N	http://localhost/GPS/Slika_ni.html
116	Bolfenk	Bolfenk	46.515896	15.577810	1044	7	Pohorje in ostala severovzhodna Slovenija	karkoli	Y	http://www.pohorje.org/
117	Ruska_odcep	Ruška ko#269a - odcep	46.494923	15.514165	1219	7	Pohorje in ostala severovzhodna Slovenija	karkoli	Y	http://www.ruskakoca.si/
118	Osankarica	Ošankarica	46.448994	15.419317	1187	7	Pohorje in ostala severovzhodna Slovenija	karkoli	Y	http://www.odmevgora.com/galerija/albums/users/pics/...

PRILOGA G: Izsek SQL tabele poti**SQL result**

Host: localhost
 Database: gps
 Generation Time: Dec 18, 2007 at 12:35 PM
 Generated by: phpMyAdmin 2.6.0-beta2 / MySQL 4.0.20a
 SQL-query: SELECT * FROM `poti` LIMIT 0, 20;
 Rows: 20

id	id_zac	id_kon	plt	id_poti	cas_hoje_min	zahtevnost	zaht_kod
1	1	281	Stra_335St.plt	0	5	2	Lahka oznaka#269;ena pot
2	2	282	Kr1_1068Kr.plt	0	25	3	Lahka neozna#269;ena pot
3	4	2	krim_4_2.plt	0	70	5	Zahtevna oznaka#269;ena pot
6	9	10	Ambroz_Kriska.plt	0	60	5	Zahtevna oznaka#269;ena pot
7	10	212	Kriska_1832Zvoh.plt	0	50	2	Lahka oznaka#269;ena pot
8	10	209	Kriska_1800.plt	0	60	2	Lahka oznaka#269;ena pot
31	43	44	Kriz_Zavrata.plt	3	60	2	Lahka oznaka#269;ena pot
32	44	209	Zavrata_1800.plt	1	50	5	Zelo zahtevna oznaka#269;ena pot
33	44	236	Zavrata_1350Ko.plt	4	40	8	Zelo zahtevna oznaka#269;ena pot
34	45	46	Sneznik.plt	1	160	5	Zahtevna oznaka#269;ena pot
80	92	93	Pariz_Vremsc.plt	0	120	3	Lahka neozna#269;ena pot
84	103	104	bivak2_zah.plt	0	240	9	Zelo zahtevna neozna#269;ena pot
101	114	115	MB_Glazuta.plt	2	300	2	Lahka oznaka#269;ena pot
102	115	116	Glaz_Bolfenk.plt	2	45	2	Lahka oznaka#269;ena pot
103	116	117	Bolfenk_Ruska.plt	2	110	3	Lahka neozna#269;ena pot
104	117	118	Ruska_Osank.plt	2	240	5	Zahtevna oznaka#269;ena pot
105	118	119	Osank_Masin.plt	2	240	5	Zahtevna oznaka#269;ena pot
106	119	120	Masin_Ribniska.plt	2	65	5	Zahtevna oznaka#269;ena pot
107	120	121	Ribnis_Grmov.plt	2	80	2	Lahka oznaka#269;ena pot
108	121	122	Grmov_SGrad.plt	2	100	3	Lahka neozna#269;ena pot