

Univerza  
v Ljubljani  
Fakulteta  
*za gradbeništvo  
in geodezijo*

*Janova 2  
1000 Ljubljana, Slovenija  
telefon (01) 47 68 500  
faks (01) 42 50 681  
fgg@fgg.uni-lj.si*



Univerzitetni študij geodezije,  
Geodezija

Kandidat:

**Matija Mlakar**

## **Uporaba odprte kode v okoljih GIS**

**Diplomska naloga št.: 823**

**Mentor:**

izr. prof. dr. Radoš Šumrada

Ljubljana, 2010

## **IZJAVA O AVTORSTVU**

Podpisani **Matija Mlakar** izjavljam, da sem avtor diplomske naloge z naslovom: »**UPORABA ODPRTE KODE V OKOLJIH GIS**«.

Izjavljam, da se odpovedujem vsem materialnim pravicam iz dela za potrebe elektronske separatoteke FGG.

Ljubljana, 20.9. 2009

---

(podpis avtorja)

**STRAN ZA POPRAVKE**

<b>Stran z napako</b>	<b>Vrstica z napako</b>	<b>Namesto</b>	<b>Naj bo</b>
-----------------------	-------------------------	----------------	---------------

## **BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK**

**UDK:** 004.42:004.6:659.2:91(043.2)  
**Avtor:** Matija Mlakar  
**Mentor:**izr. prof. dr. Radoš Šumrada  
**Naslov:** Uporaba odprte kode v okoljih GIS  
**Obseg in oprema:** 73 strani, 34 slik, 5 preglednic, 1 priloga  
**Ključne besede:** geografski operacijski sistemi, odprta koda, GRASS GIS

### **Izvelek:**

Geografski informacijski sistemi (GIS) oz. aplikacije, ki operirajo s prostorskimi podatki, so v vzponu, saj postajajo zelo zmogljive in učinkovite. Kot pri ostali programski opreми, se tudi na področju geografskih informacijskih sistemov pojavlja vedno več odprtokodnih aplikacij, ki jih lahko svobodno uporabljamo, spreminjamo, porazdeljujemo itd.

Diplomska naloga obravnava zamisel odprte kode ter povezavo le-te z okolji geografskih operacijskih sistemov. Predstavljene so različne odprtokodne aplikacije GIS, izmed katerih se diplomska naloga osredotoča na programski paket GRASS GIS. Predstavljene in opisane so različne možnosti uporabe slednjega programskega paketa, kjer največ pozornosti posveča osnovnim rastrskim in vektorskim operacijam, kot so npr. rastrske operacije z digitalnim modelom višin, reklasifikacija razredov, vektorizacija, območne in mrežne vektorske analize, interpolacijske metode idr. Operacije so bile izvedene za območje TTN lista 2027 (okolica Kopa), s katerimi so bile ugotovljene prednosti in slabosti odprtokodnih aplikacij (GIS).

## **BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION**

**UDC:** 004.42:004.6:659.2:91(043.2)  
**Author:** Matija Mlakar  
**Supervisor:** assoc. prof. dr. Radoš Šumrada  
**Title:** The use of open source GIS  
**Notes:** 73 pages, 34 pictures, 5 tables, 1 attachment  
**Key words:** geographic information system, open source, GRASS GIS

### **Abstract:**

Geographic Information Systems (GIS) or applications that operate with spatial data are on the rise, as they are becoming very powerful and efficient software tools. In the field of geographical information systems, like in any type of software, we can nowadays find several open source applications which can be used, modified, distributed, etc. without restrictions.

This graduation thesis deals with the concept of “open source” and its connection to geographical operating systems. The thesis describes different open source GIS applications, with emphasis on the software package GRASS GIS. The thesis also presents and describes various usage possibilities of GRASS GIS, focusing on the basic raster and vector operations such as: raster operations with digital elevation model, reclassification of classes, vectorization, buffer and network vector analysis, interpolation methods, etc. These operations were performed for the “TTN sheet” 2027 area (outskirts of Koper), with the goal to identify different strengths and weaknesses of open source (GIS) applications.

## **ZAHVALA**

Rad bi se zahvalil mentorju izr. prof. dr. Radošu Šumradi za pomoč pri diplomskem delu, moji družini ter vsem ostalim, ki so mi kakorkoli tekom študija pomagali in me pri delu spodbujali.

## KAZALO VSEBINE

<b>IZJAVA O AVTORSTVU.....</b>	<b>I</b>
<b>STRAN ZA POPRAVKE .....</b>	<b>II</b>
<b>BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK.....</b>	<b>III</b>
<b>BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION .....</b>	<b>IV</b>
<b>ZAHVALA.....</b>	<b>V</b>
<b>KAZALO VSEBINE.....</b>	<b>VI</b>
<b>KAZALO SLIK.....</b>	<b>VIII</b>
<b>KAZALO PREGLEDNIC.....</b>	<b>X</b>
<b>1 UVOD.....</b>	<b>1</b>
<b>2 GEOGRAFSKI INFORMACIJSKI SISTEM .....</b>	<b>2</b>
2.1 Definicija .....	2
2.2 Zgodovina.....	2
2.3 Podatki v GIS.....	4
2.4 Uporaba, prihodnost .....	6
<b>3 ODPRTA KODA .....</b>	<b>7</b>
3.1 Začetki .....	7
3.2 Zamisel .....	8
3.2.1 Prosta programska oprema.....	8
3.2.2 Odprtokodna programska oprema.....	9
3.3 GNU GPL .....	11
3.4 Odprtokodni projekti .....	11
<b>4 ODPRTA KODA IN GIS .....</b>	<b>12</b>
4.1 OSGeo.....	12
4.2 Odprtokodni GIS projekti.....	13
4.2.1 Spletni odprtokodni GIS projekti .....	13
4.2.1.1 MapServer.....	13
4.2.1.2 MapGuide .....	16
4.2.2 Namizni odprtokodni GIS projekti.....	18

4.2.2.1	Quantum GIS .....	18
4.2.2.2	gvSIG .....	20
4.2.3	"Prostorske knjižnice" .....	22
4.2.3.1	GDAL/OGR.....	22
4.2.3.2	GeoTools.....	23
4.2.4	GeoNetwork - katalog metapodatkov .....	24
<b>5</b>	<b>GRASS GIS .....</b>	<b>26</b>
5.1	Zgodovina .....	26
5.2	GRASS danes, specifikacije, lastnosti.....	27
5.3	Struktura programskega paketa GRASS .....	28
<b>6</b>	<b>DELO S PROGRAMSKIM PAKETOM GRASS .....</b>	<b>30</b>
6.1	Začetni koraki .....	30
6.2	Uvoz podatkov in prikaz.....	30
6.3	Rastrske operacije .....	33
6.3.1	Operacije z DMV-jem.....	33
6.3.2	Pretvorba vektorskega sloja v rastrski sloj, reklasifikacija, barvne tabele.....	38
6.3.3	Algebra karte .....	42
6.4	Delo z vektorskimi podatki.....	44
6.4.1	Opisni podatki .....	44
6.4.2	Vektorizacija in urejanje vektorskih slojev .....	48
6.4.3	Območne in mrežne analize .....	53
6.4.4	Interpolacijske metode s prostorskimi podatki.....	56
6.5	Vizualizacije podatkov .....	59
6.6	Programiranje v GRASS-u .....	62
6.7	Druge možnosti uporabe GRASS-a.....	64
<b>7</b>	<b>ZAKLJUČEK .....</b>	<b>66</b>
<b>VIRI</b>	<b>.....</b>	<b>71</b>
<b>PRILOGA</b>		



## KAZALO SLIK

Slika 1: Gilbertova reprodukcija Snowovega načrta razpršenosti kolere (History of GIS, 2008) ...	3
Slika 2: "Razslojevanje" podatkov v GIS (Geoservis, 2008) .....	5
Slika 3: Logotip 'Odprte kode' (Open source, 2009).....	7
Slika 4: Arhitektura MapServerja (The Open Source Geospatial Foundation, 2009) .....	14
Slika 5: Prikaz uporabe MapServerja (MapServer, 2009) .....	15
Slika 6: Prikaz pregledovanja rastrov z aplikacijo Internet Explorer (The Open Source Geospatial Foundation, 2009) .....	17
Slika 7: Prikaz uporabe aplikacije QGIS (The Open Source Geospatial Foundation, 2009).....	19
Slika 8: Prikaz uporabe aplikacije gvSIG (The Open Source Geospatial Foundation, 2009) .....	21
Slika 9: Logotip GRASS GIS (GRASS, 2009).....	26
Slika 10: DMV 12,5 v običajni barvni lestvici .....	31
Slika 11: Senčeni DMV .....	33
Slika 12: Prekrivanje ortofota in senčenega DMV-ja .....	34
Slika 13: Prikaz plastnic na podlagi ortofota .....	35
Slika 14: Usmerjenost terena z legendo .....	36
Slika 15: Naklon terena z legendo .....	37
Slika 16: Prikaz območij z nadmorsko višino od 30 do 110m.....	38
Slika 17: Rastrski prikaz rabe tal .....	39
Slika 18: Reklasificiran raster rabe tal .....	40
Slika 19: Prikaz rabe tal z asociativno določeno barvno tabelo.....	42
Slika 20: Prikaz travnikov in gozdov .....	43
Slika 21: Opisna tabela vektorskega sloja rabe tal.....	45
Slika 22: Seznam parcel manjših od 100m <sup>2</sup> ; od najmanjše do največje .....	46
Slika 23: Prikaz "zahtevnejše" poizvedbe SQL.....	47
Slika 24: Prikaz oljčnikov, vinogradov in njiv na podlagi ortofota .....	48
Slika 25: Prikaz vektorizacije linijskih objektov – cest .....	49
Slika 26: Primer urejanja topologije linijskih objektov (Neteler, 2008).....	50

Slika 27: Prikaz vektoriziranih cest na podlagi ortofota .....	50
Slika 28: Prikaz območij, ki so od glavne ceste oddaljena za največ 100m .....	53
Slika 29: Prikaz povezave točke s cestnim omrežjem .....	55
Slika 30: Prikaz najkrajše poti med izbranimi točkama (hišama) .....	56
Slika 31: Prikaz rezultatov interpolacij ter izvornega DMV-ja: Voronoi (zgoraj levo), IDW (zgoraj desno), RST (spodaj levo), izvorni DMV (spodaj desno) .....	59
Slika 32: RGB in IHS barvna modela (Neteler, 2008).....	60
Slika 33: Vizualizacija s pomočjo IHS barvnega modela.....	61
Slika 34: 3D prikaz območja izdelan v orodju NVIZ .....	62

## KAZALO PREGLEDNIC

Preglednica 1: Najbolj razširjeni rastrski in vektorski formati prostorske knjižnice GDAL/OGR (The Open Source Geospatial Foundation, 2009).....	23
Preglednica 2: Najbolj razširjeni rastrski in vektorski formati GeoTools (The Open Source Geospatial Foundation, 2009) .....	24
Preglednica 3: Pregled ukaznih predpon v GRASS-u (Neteler, 2008) .....	29
Preglednica 4: Gonilniki podatkovnih baz v aplikaciji GRASS GIS (Neteler, 2008) .....	44
Preglednica 5: Primerjava programskih paketov ArcGIS 9.0 ter GRASS 6.0 (Buchanan, 2005) .	69

## 1 UVOD

Dan za dnem se srečujemo s težavami oz. problemi, ki jih skušamo čim boljše rešiti. V času modernega, hitro razvijajočega sveta, moramo sprejeti veliko pomembnih odločitev, katerih precejšnji delež je tako ali drugače povezanih z lokacijo, na kateri se nahajamo. Tako so prostorski podatki in uporaba t.i. tehnologije GIS (Geografski Informacijski Sistem) postali del vsakdanjega življenja posameznika in ne več samo področje obravnave geografov, geodetov idr. Z razvojem računalništva, informacijske tehnologije in bliskovitim vzponom interneta so nastala številna podjetja, ki razvijajo spletne in "namizne" aplikacije GIS. Večina teh aplikacij nastane pod okriljem tržnih podjetij, vedno več pa jih nastaja po načelih odprte kode (open source), katere glavni namen je distribuiranje, uporabljanje ter spreminjanje izvorne kode po lastnih željah in interesih.

V diplomski nalogi se bom posvetil obravnavi koncepta odprte kode v okoljih GIS, predstavil obstoječe odprtokodne aplikacije GIS in podrobneje GRASS GIS, ki postaja vedno bolj celovita in dovršena odprtokodna aplikacija GIS (Neteler, 2008). Slednji programski paket bom uporabil tudi za razne rastrsko-vektorske prostorske analize in pokazal zmogljivosti aplikacije.

Na ta način se želim poglobiti v to perspektivno "vejo" okolij GIS, ki zaenkrat še ni v enaki meri zastopana kot "tržna", morda pa jo bo v prihodnosti s svojo "odprtostjo" in fleksibilnostjo prerasla in postala številka ena, bodisi na področju GIS-ov ali programske opreme na sploh. Prav gotovo gre za pomemben dejavnik razvoja aplikacij ter računalništva in s tem nezanemarljiv doprinos k bolj enostavnemu in kvalitetnemu načinu življenja posameznikov.

## **2 GEOGRAFSKI INFORMACIJSKI SISTEM**

### **2.1 Definicija**

Zaradi razmeroma novega znanstvenega in strokovnega področja je pojem geografskega informacijskega sistema (v nadaljevanju GIS, angl. Geographic(al) Information System) težko natančno opredeliti. Obstaja več različnih definicij GIS-a, od tehnoloških, procesno funkcijskih pa do vsebinskih definicij, kot tudi definicij geografske informacijske tehnologije. Največkrat je GIS definiran kot sistem za zajemanje, shranjevanje, vzdrževanje, obdelavo, povezovanje, analizo in predstavitev prostorskih geokodiranih podatkov (Drobne in Podobnikar, 1999). V ožjem pomenu lahko GIS predstavimo kot zbirko računalniških programov, namenjenih obdelavi podatkov s prostorsko komponento. V širšem pomenu pa lahko GIS razumemo kot sistem, ki ga sestavljajo programska oprema, strojna oprema, prostorski podatki, različni procesi, algoritmi, procedure ter nenazadnje človek kot uporabnik sistema.

### **2.2 Zgodovina**

Pred približno 15500 leti so naši predniki (t.i. kromanjonski človek) risali na stene jam (Lascaux; področje današnje Francije) podobe živali, ki so jih lovili za vir preživetja. Risbam oz. podobam so dodali nekakšne linije oz. črte, ki so ponazarjale poti preseljevanj živali. Z malo domišljije lahko v teh risbah vidimo simbolično podobnost s sodobnim GIS, saj lahko rečemo, da gre za povezavo podobe z neko informacijo (t.i. atributom).

Leta 1854 je britanski fizik John Snow odkril lokacijo izbruha kolere v Londonu s pomočjo ponazoritve položajev okuženih posameznikov. Študija razpršenosti le-teh v povezavi z možnimi izvori bolezni mu je pomagala določiti vodno pipo, ki je bila izvor kolere. Verjetno gre za eno prvih uporab nekega kartografskega izdelka, ne samo za prikaz nekega dela zemeljskega površja, temveč predvsem za izvedbo analize vzorcev geografsko odvisnega pojava (glej sliko 1).



**Slika 1:** Gilbertova reprodukcija Snowovega načrta razpršenosti kolere (History of GIS, 2008)

V začetku 20. stoletja se je razvila t.i. "foto litografija", v kateri karte (oz. kartografske pojave) razdelimo na posamezne plasti oz. sloje (angl. layers). Razvoj računalniške strojne opreme, ki so ga narekivale raziskave jedrskega orožja, pa je okrog leta 1960 posledično privedel tudi do računalniških kartografskih aplikacij.

Leta 1962 je dr. Roger Tomlinson v Ottawi (Kanada) na državnem ministrstvu za gozdarstvo in razvoj podeželja (Department of Forestry and Rural Development) razvil prvi operativni GIS, ki se je imenoval CGIS (Canada Geographic Information System). Uporabljal se je za shranjevanje,

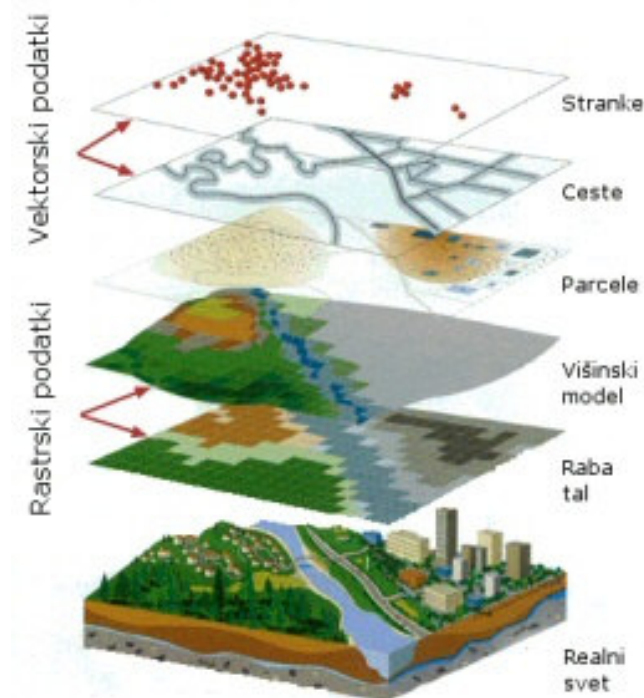
obdelavo ter vzdrževanje podatkov, ki so bili zbrani za CLI (Canada Land Inventory) z namenom določitve rabe tal in možnega razvoja kanadskega podeželja. Zbrani so bili podatki o vodovju, prsti, rastju, kmetijstvu, rekreaciji ter rabi tal v merilu 1: 50 000. Različne rabe tal so bile tudi klasificirane, tako da so lahko izvajali razne prostorske analize oz. poizvedovanja. CGIS je omogočal funkcije prekrivanja plasti (angl. overlay), merjenja, digitalizacije in skeniranja. Podpiral je uporabo državnega koordinatnega sistema, krivulje so imele urejeno topologijo, atributne in lokacijske informacije so bile shranjene v ločenih datotekah. Kot rezultat tega, je Tomlinson postal znan kot "oče GIS-a". CGIS so uporabljali do 90. let in zgradili veliko podatkovno bazo prostorskih podatkov Kanade, obravnavani sistem pa ni bil nikoli na voljo v komercialni obliki.

V zgodnjih 80. letih so bila podjetja M&S Computing (kasneje Intergraph), ESRI (Environmental Systems Research Institute) in CARIS (Computer Aided Resource Information System) glavni komercialni ponudniki programske opreme GIS. Uspešno so nadgradili prvogeneracijski pristop ločevanja prostorskih in 'atributnih' informacij v drugogeneracijski pristop organiziranja atributnih podatkov v podatkovne baze (DBMS). Vzporedno je potekal razvoj GIS-ov tudi v javnem sektorju. Gre za projekta MOSS (Map Overlay and Statistical System Project) ter projekt vojske Združenih držav Amerike (U.S. Army Corp of Engineering Research Laboratory), ki je razvijala GIS od leta 1982, z namenom prostorskega planiranja in upravljanja. V 90. letih se je uporaba GIS-ov množično razširila, saj se je pojavljalo vedno več uporabnikov na osebnih računalnikih in različnih operacijskih sistemih (npr. UNIX). Z razvojem interneta se je ob koncu 20. stoletja vedno več prostorskih podatkov prenašalo kar po medmrežju, kar je vzbudilo potrebo po enotnih podatkovnih formatih in standardizaciji prenosa podatkov. Zadnje čase pa se pojavlja vedno več 'prostih' in/ali odprtokodnih aplikacij GIS, ki se izvajajo na številnih sistemih in se jih lahko prilagodi za izvedbo specifičnega opravila/naloge.

### **2.3 Podatki v GIS**

Podatki v bazi GIS običajno predstavljajo realne objekte (ceste, rastje, reke ...) v digitalni obliki, ki jih lahko pridobimo na različne načine – bodisi z digitalizacijo papirnatih kart, s terensko

izmero (klasična/GPS), s fotogrametričnimi postopki (npr. lasersko skeniranje) idr. Objekte lahko predstavimo v diskretni – točkovni (npr. hiša), linijski (npr. cesta) ali ploskovni obliki (npr. jezero). V GIS-ih obstajata dva široka načina predstavitve le-teh, in sicer rastrski in vektorski. Vsak "pojav" (vektorski ali rastrski) lahko predstavimo s svojim slojem, kar nam da jasno in pregledno strukturo celotnega sistema (slika 2).



**Slika 2:** "Razslojevanje" podatkov v GIS (Geoservis, 2008)

V povezavi z opisnimi podatki, dobimo t.i. geopodatkovno bazo, ki omogoča razne prostorske poizvedbe. GIS orodja nam pa ne omogočajo samo poizvedb, temveč tudi analize ter nek rezultat le-teh, ki je lahko ali v obliki nekega kartografskega prikaza, ki nam nato služi za sprejemanje nekkih odločitev, ali pa v obliki opisnega podatka (npr. seznam naslovov v okolici izlitja strupov), ki ga lahko nato uporabimo za nadaljnje analize ali raziskave.



## 2.4 Uporaba, prihodnost

GIS-tehnologija se dandanes lahko uporablja za znanstvene raziskave, arheologijo, upravljanje z okoljem, upravljanje z nepremičninami, prostorsko planiranje, kartografijo, kriminologijo, upravljanje z energetskimi viri, marketing, logistiko in še za mnoge druge namene. Na primer, GIS nam lahko služi za določitev optimalne poti evakuacije v primeru naravne nesreče, za določitev vodovij, ki morajo biti zaščitena pred onesnaževanjem, ali celo za določitev nove poslovne lokacije, kjer bi lahko izkoristili zdajšnjo tržno neučinkovitost.

Razvoj GIS-ov poteka vzporedno z uporabo interneta. Vse komponente GIS-ov se ne nahajajo več na isti fizični lokaciji, temveč gre za nekakšen medmrežni sistem – podatkovna baza, procesiranje in uporabniški vmesnik sosedno delujejo prek interneta (primer: spletne aplikacije GIS, mobilni GIS ...). Uporaba interneta ter množična porazdeljenost podatkov je privedla tudi do potrebe po podatkovnih in prenosnih standardih, kar je ena izmed nalog ustanove OGC (Open Geospatial Consortium). Vedno več je spletnih kartografskih aplikacij (npr. Google maps), zmogljivih aplikacij GIS, ki nam z dodano "časovno komponento" prikažejo razne simulacije razvoja našega planeta ipd.

Kot vidimo, se uporaba širi na socialna področja, obenem pa se vztrajno in hitro razvija. Z vse večjo razširjenostjo uporabe GPS-a in vse bolj zmogljivimi mobilniki in dlančniki, nas GIS-i takorekoč spremljajo na vsakem koraku. Ob tem se pojavlja tudi socialno vprašanje naše zasebnosti oz. popolnega nadzora od "tistih zgoraj". Vsekakor pa se je bolje osredotočiti predvsem na prednosti GIS-ov, ki nam tako ali drugače lajšajo vsakdan in dvigujejo kvaliteto življenja.

### 3 ODPRTA KODA

Odrpta koda ("open source") je širok pojem, ki ga poznamo predvsem s področja računalništva oz. programske opreme ("open source software"). Čeprav lahko dandanes izraz 'odprta koda' zasledimo tudi na drugih področjih, se bom osredotočil na področje programske opreme. Na sliki 3 vidimo logotip 'Odrpte kode', ki poudarja predvsem njeno "odprtost".



Slika 3: Logotip 'Odrpte kode' (Open source, 2009)

#### 3.1 Začetki

V 80. letih 20. stoletja se je na področju programske opreme pojavila prav posebna novost. Univerza v Berkeleyu (ZDA) in Richard Stallman sta začela izdajati programsko opremo pod drugačnimi licencami, kot je bilo do tedaj v navadi. Gre za t.i. prosto programsko opremo (free software), ki je bila sprva domena akademskih raziskovalcev ter programerskih navdušencev, v drugi polovici 90. let pa so začela nastajati podjetja, ki so tudi v prosti programski opremi videla poslovne priložnosti (npr. nudenje pomoči uporabnikom, tehnična podpora, svetovanje, razvoj prijaznejših vmesnikov itd.). Termin odrpta koda (open source) se je pojavil šele leta 1998, ko so ga izbrali kot primernejšega, čeprav se Richard Stallman še vedno zavzema za prvotni termin (in tudi filozofijo pojma) 'free software'. Termina danes pomenita malce različni zadevi, saj sta plod dveh različnih ustanov, a se še vedno držita podobnih smernic in načel.

## 3.2 Zamisel

### 3.2.1 Prosta programska oprema

Leta 1984 je Richard Stallman začel s projektom GNU (gre za rekurzivni akronim: "Gnu's not Unix"), s katerim je želel ustvariti operacijski sistem podoben Unix-u, ki bi bil prosta programska oprema. Leta 1985 pa je ustanovil fundacijo FSF (Free Software Foundation), v okviru katere je definiral prosto programsko opremo s pomočjo štirih lastnosti (avtor namesto lastnosti uporablja izraz svobode oz. prostosti; angl. freedoms):

1. Programsko opremo lahko uporablja vsak, ne glede s kakšnim namenom. (freedom 0)
2. Vsak lahko preučuje zgradbo in delovanje programske opreme in jo po želji prilagaja svojim potrebam. (freedom 1)
3. Programsko opremo se lahko prosto porazdeljuje oz. kopira. (freedom 2)
4. Posameznik lahko programsko opremo spremeni, modificira, izboljša in svoje izboljšave deli z javnostjo in tako pripomore k splošni družbeni koristi. (freedom 3)

Glavna ideja take programske opreme je, da uporabniki in programerji berejo, porazdeljujejo in spreminjajo programsko kodo, na ta način pa se program razvija – se izboljša, napake se odpravijo, zmožnosti se povečajo ... Če se v ta proces vključi veliko število strokovnjakov oz. programerjev iz celega sveta, se lahko razvoj programske opreme odvija na neverjetno hiter in učinkovit način.

FSF bi rada poudarila, da gre pri prosti programski opremi predvsem za nekakšno svobodo pri uporabi programske opreme in ne zgolj za dostopnost do izvirne kode ter brezplačnost. V angleškem jeziku velja rek: *“Think of free speech, not free beer.”*

### 3.2.2 Odprtokodna programska oprema

Pobuda za odprto kodo OSI (Open Source Initiative) je neprofitna združba, ki obstaja od leta 1998 in skrbi za promocijo in licenciranje odprte kode (Open Source Initiative, 2009). Če se FSF zavzema predvsem za "svobodno" uporabo programske opreme, se na drugi strani OSI zavzema sicer za skorajda enake cilje, ampak z bolj pragmatičnega vidika. Odklanja stališča, da je lastniško-komercialna programska oprema apriori nemoralna in poudarja predvsem tehnične pomanjkljivosti v primerjavi z odprtokodnim pristopom.

V najširšem smislu je odprtokodna programska oprema tista, ki ima omogočen dostop do njene programske kode. V smislu OSI definicije odprte kode pa je to programska oprema, pri kateri pravila distribucije oz. licenciranja izpolnjujejo 10 kriterijev, zapisanih v definiciji odprte kode (Benčina, 2003):

1. *Svobodna distribucija* - licenca dovoljuje prosto porazdeljevanje in kopiranje programske opreme,
2. *Izvorna koda* - le-ta mora biti javno dostopna, samo končni 'binarni izdelek' ni dovolj,
3. *Izpeljana dela* – dovoljeno je spreminjanje in izdelava novih programskih rešitev, ki se morajo distribuirati pod enakimi licenčnimi pogoji kot to velja za originalno programsko opremo,
4. *Integriteta avtorja izvorne kode* - nedvoumno mora biti dovoljeno porazdeljevanje programske izvršne kode, lahko pa se zahteva, da so novonastali izdelki porazdeljeni pod novim imenom oz. spremenjeno oznako različice. Prepoved porazdeljevanja izvorne kode v spremenjeni obliki je sprejemljiva le v primeru, da licenca dovoljuje porazdeljevanje popravkov z namenom spreminjanja programa v času "prevajanja",
5. *Prepoved diskriminacije posameznikov in skupin* - licenca ne sme diskriminirati nikogar (posameznika ali skupine),

6. *Prepoved diskriminacije posameznih področij dejavnosti* - licenca ne sme nikogar omejevati pri uporabi programa na določenem področju (npr. na področju genetike, tržnih področjih...),

7. *Porazdeljevanje licence* - licenčne pravice se nanašajo na vsakogar, ki uporablja programsko opremo, brez dodatnih postopkov licenciranja,

8. *Licenca se ne nanaša na določen produkt* - pravice, vezane na programsko opremo, ne smejo biti odvisne od tega, ali je le-ta del določenega programskega porazdeljevanja. Posamezniki, ki jim je bil program porazdeljen, imajo enake pravice kot tisti, ki so jim bile dodeljene pravice za prvotno programsko distribucijo,

9. *Licenca ne sme omejevati druge programske opreme* - licenca ne sme postavljati omejitev nad programsko opremo, ki je porazdeljena skupaj z licencirano programsko opremo (npr. licenca ne sme zahtevati, da mora biti vsa programska oprema, ki je porazdeljena na istem mediju, prav tako odprtokodna programska oprema),

10. *Licenca mora biti nevtralna do tehnologije* - prevzemanje licenc ne sme biti omejeno na posamezne tehnološke rešitve.

Gibanji za prosto programsko opremo in odprto kodo torej ne gre tlačiti v isti koš. Imata različne poglede in cilje, čeprav na nekaterih projektih sodelujeta. Najpomembnejša razlika med njima je v njihovih vrednotah, načinu gledanja na svet ter v licenčnih zahtevah. Odprtokodna in prosta programska oprema se ne izdaja pod klasičnimi licencami, temveč pod posebnimi 'javnimi', od katerih je verjetno najbolj uporabljana in znana različica GNU GPL (General Public License).

V nadaljevanju bom uporabljal izraza odprta koda in prosto programsko opremo kot isti pojem, v smislu programske opreme, ki jo je mogoče svobodno uporabljati, spreminjati in porazdeljevati.

### **3.3 GNU GPL**

GNU GPL (General Public License) je licenca za prosto programsko opremo (Free Software Foundation, 2009), ki je bila osnovana v okviru GNU projekta pod okriljem že prej omenjenega Richarda Stallmana. Večina licenc prepoveduje prosto uporabo ter porazdeljevanje programske opreme, v nasprotju pa GNU GPL določa in obenem ščiti prosto uporabo, dostop in možnost spreminjanja izvorne programske kode. V GNU GPL je določeno, da mora v primeru spreminjanja programske kode ali distribuiranja programske opreme, imeti vsak naslednji uporabnik enake pravice kot predhodni – torej popolno svobodo nad uporabo in porazdeljevanjem, omogočen mu mora biti dostop do programske kode in možnost spreminjanje le-te. Obstaja tudi modificirana različica licence imenovana GNU LGPL (Lesser General Public License), ki ni tako stroga in je bila prvotno osnovana kot licenca za programske knjižnice. Prva različica GNU GPL je izšla leta 1989, trenutno pa je aktualna tretja različica (v3), ki je izšla leta 2007.

### **3.4 Odprtokodni projekti**

Projekte odprtokodne programske opreme postavljajo in vzdržujejo prostovoljni programerji. Med najbolj znane projekte sodijo Apache HTTP Server, Internet Protocol in internetni brskalnik Mozilla Firefox, vsem poznan je tudi operacijski sistem Linux - odprtokodni operacijski sistem Unix-like. Do danes je nastalo ogromno odprtokodne programske opreme in lahko trdimo, da kar uspešno konkurira običajni komercialni različici.

Na področju geodezije kot tudi GIS tehnologije obstaja prav tako veliko odprtokodnih projektov, kot so npr. Mapserver, GRASS GIS, gvSIG idr., ki pa jih bom podrobneje predstavil v naslednjem poglavju.

## 4 ODPRTA KODA IN GIS

Popoln dostop do izvorne programske kode je še posebno pomemben v okoljih GIS, saj v ozadju aplikacij ležijo številni algoritmi, ki so lahko zelo kompleksni in lahko znatno vplivajo na končne rezultate prostorskih analiz in modeliranj. Če želimo razumeti delovanje nekega sistema, ki ni tako enostaven kot npr. pri programski opremi urejevalnikov besedil, je pomembno, da lahko pregledujemo in preverjamo implementacije posameznih funkcij. Čeprav povprečni uporabniki težka najdejo napake v programski kodi, se s pomočjo današnje 'internetne povezanosti' zlahka najdejo številni strokovnjaki, ki so pripravljeni testirati, analizirati in popraviti programsko kodo. Glavna organizacija, v okviru katere se odvijajo številni projekti na področju odprtokodne tehnologije GIS, je nastala leta 2006 in se imenuje OSGeo (Open Source Geospatial Foundation).

### 4.1 OSGeo

Open Source Geospatial Foundation je neprofitna organizacija (The Open Source Geospatial Foundation, 2009), katere cilj je spodbujati in promovirati razvoj odprtokodnih tehnologij GIS ter prostorskih podatkov. V OSGeo so vključeni strokovnjaki, programerji, prostovoljci idr. s celega sveta, ki skušajo izboljšati in razširiti uporabo odprtokodnih aplikacij GIS ter na ta način doprinesti k splošnim družbenim koristim. Cilji OSGeo so:

- zagotoviti sredstva za projekte organizacije,
- promovirati prosto dostopne prostorske podatke,
- promovirati uporabo odprtokodne programske opreme na področju tehnologije GIS,
- spodbujati implementacijo 'odprtih' standardov,
- zagotoviti visoko kvaliteto projektov OSGeo,
- približati programsko opremo končnim uporabnikom,
- poučevati in nuditi podporo uporabnikom odprtokodne programske opreme GIS,
- spodbujati sodelovanje med skupnostmi OSGeo, ki uporabljajo različne programske jezike in operacijske sisteme,
- prirejati letne konference (FOSS4G – Free and Open Source Software for Geospatial),

- podeljevati 'Sol Katzovo' nagrado za usluge skupnosti OSGeo.

## **4.2 Odprtokodni GIS projekti**

V nadaljevanju bom opisal nekatere projekte oz. aplikacije, ki so bile razvite v okviru OSGeo in so po mojem mnenju najbolj razširjene in najpomembnejše. Naj povem, da so pomembna dela omenjene organizacije poleg razvijanja aplikacij tudi povezovanje razvijalcev in uporabnikov odprtokodnih GIS aplikacij, poučevanje novih uporabnikov itd., saj bi bili drugače projekti neuresničljivi. Omenjeno povezovanje poteka globalno prek spleta, in sicer večinoma s pomočjo elektronske pošte ("mailing lists") ter spletnih klepetalnic (npr. IRC – Internet Relay Chat).

### **4.2.1 Spletni odprtokodni GIS projekti**

#### **4.2.1.1 MapServer**

Programski paket MapServer je odprtokodno razvojno okolje (The Open Source Geospatial Foundation, 2009), ki se uporablja za izgradnjo spletnih prostorskih aplikacij in storitev. Gre za hitro, prilagodljivo in zanesljivo orodje, ki ga lahko integriramo v skoraj vsako okolje GIS. MapServer so razvili na Univerzi Minnesota (University of Minnesota), dandanes pa pri razvoju sodelujejo programerji s celega sveta.

MapServer deluje na vseh glavnih operacijskih sistemih in skoraj na vsakem spletnem strežniku. Vsebuje t.i. MapScript - zmogljivo okolje, ki podpira programske jezike PHP, Python, Perl, C# in Java. S pomočjo MapScript-a, lahko hitro in enostavno razvijamo kompleksne prostorske spletne aplikacije.

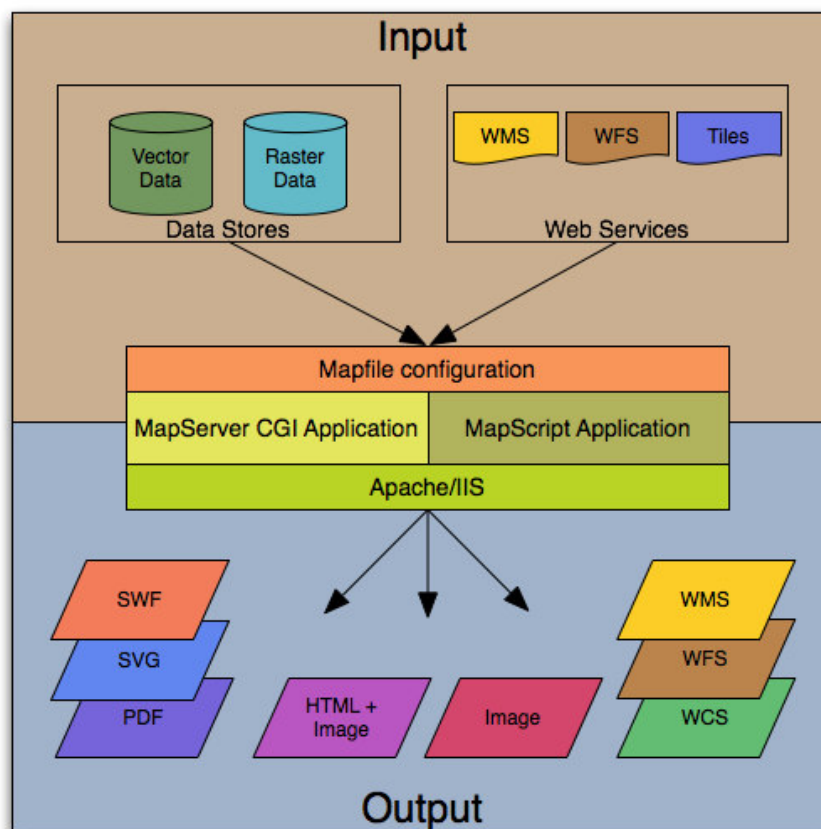
Za uporabnike MapServer-ja je na voljo tudi dobra uporabniška podpora, ki pomaga reševati kakršnekoli probleme pri uporabi.



Kot rečeno, gre za odprtokodno okolje, torej je programska koda prosto dostopna. Lahko jo spreminjamo in razvijamo, da bo aplikacija služila našim prav posebnim zahtevam in potrebam.

MapServer omogoča oz. podpira:

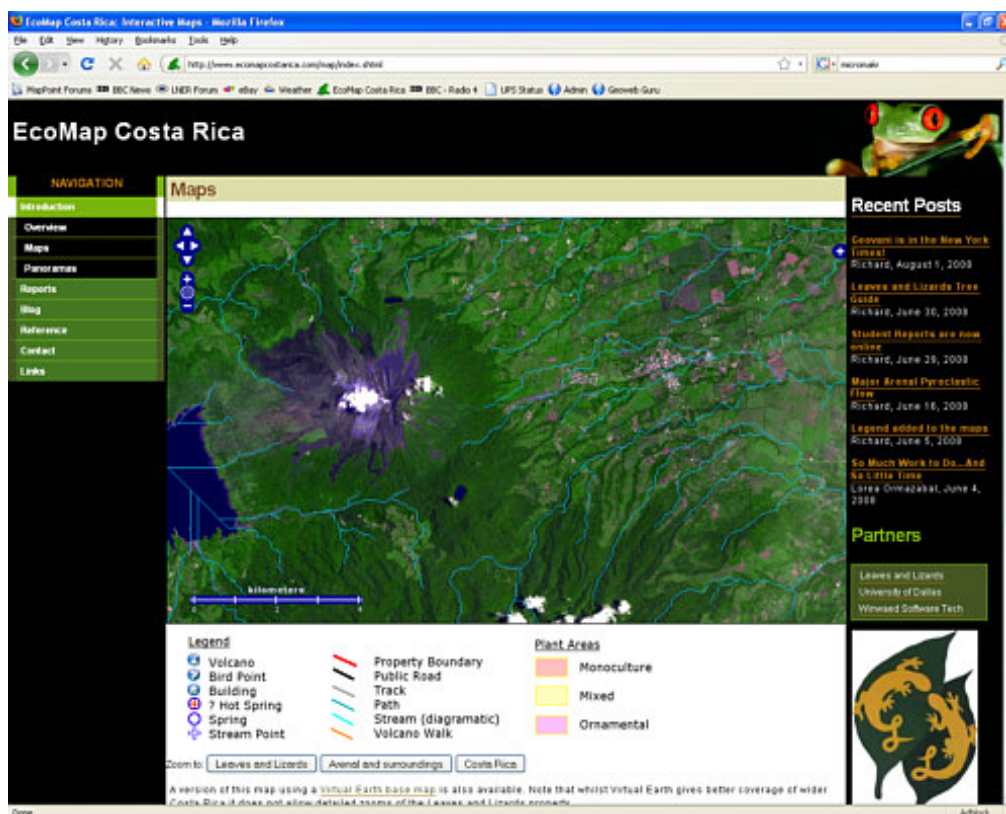
- standardne podatkovne formate in prostorske podatkovne baze,
- klasifikacijo pojavov/objektov v realnem času (angl. on the fly),
- projiciranje rastrskih in vektorskih podatkov v realnem času,
- široko paleto prostorsko atributnih poizvedb,
- razširjene standarde kot so npr. WMS, WFS in WCS,
- najboljše odprto kodne tehnologije GIS kot so npr. GDAL/OGR, PostGIS in PROJ.4,
- povezavo s priljubljenimi okolji kot so ka-Map, Chameleon, Mapbender, MapBuilder in Cartoweb.



**Slika 4:** Arhitektura MapServerja (The Open Source Geospatial Foundation, 2009)

Slika 4 prikazuje arhitekturo programskega paketa MapServer, ki ponazarja možne vhodne podatke, ki jih organiziramo v t.i. 'kartne datoteke' (angl. mapfiles), ki so srce programa. V teh datotekah se definirajo odnosi med objekti, povezave med podatki ter način izrisa izhodnih podatkov (output). Poleg razširjenih podatkovnih formatov na sliki opazimo tudi WMS (Web Map Service), WFS (Web Feature Service) ter WCS (Web Coverage Service), ki so t.i. odprti standardi, ki določajo prenos, uporabo ter prikazovanje prostorskih podatkov na internetu. Tako kot GML (Geography Markup Language) so bili tudi ti trije standardi razviti na OGC-ju (Open Geospatial Consortium).

Na sliki 5 vidimo primer uporabe aplikacije MapServer.



Slika 5: Prikaz uporabe MapServerja (MapServer, 2009)

#### 4.2.1.2 MapGuide

MapGuide Open Source je spletna platforma, ki omogoča uporabnikom hiter razvoj in uporabo spletnih prostorskih aplikacij ter prostorskih spletnih storitev (The Open Source Geospatial Foundation, 2009). Vsebuje interaktivni pregledovalnik, ki omogoča uporabo funkcij kot so: izbiranje pojavov/objektov, pregledovanje lastništev, območne operacije (angl. buffer), merjenje razdalj in druga poizvedovanja. Med drugim ima tudi XML (Extensible Markup Language) podatkovno bazo za upravljanje vsebine in podpira najbolj razširjene prostorske formate datotek, podatkovnih baz ter standardov.

Uporabljamo ga lahko na Linux ali Windows operacijskem sistemu, na Apache in IIS spletnih strežnikih, ponuja pa tudi obsežne PHP, .NET, Java in JavaScript API-je (Application Programming Interface) za razvijanje aplikacij. Aplikacija je izdana pod licenco LGPL (Lesser General Public License).

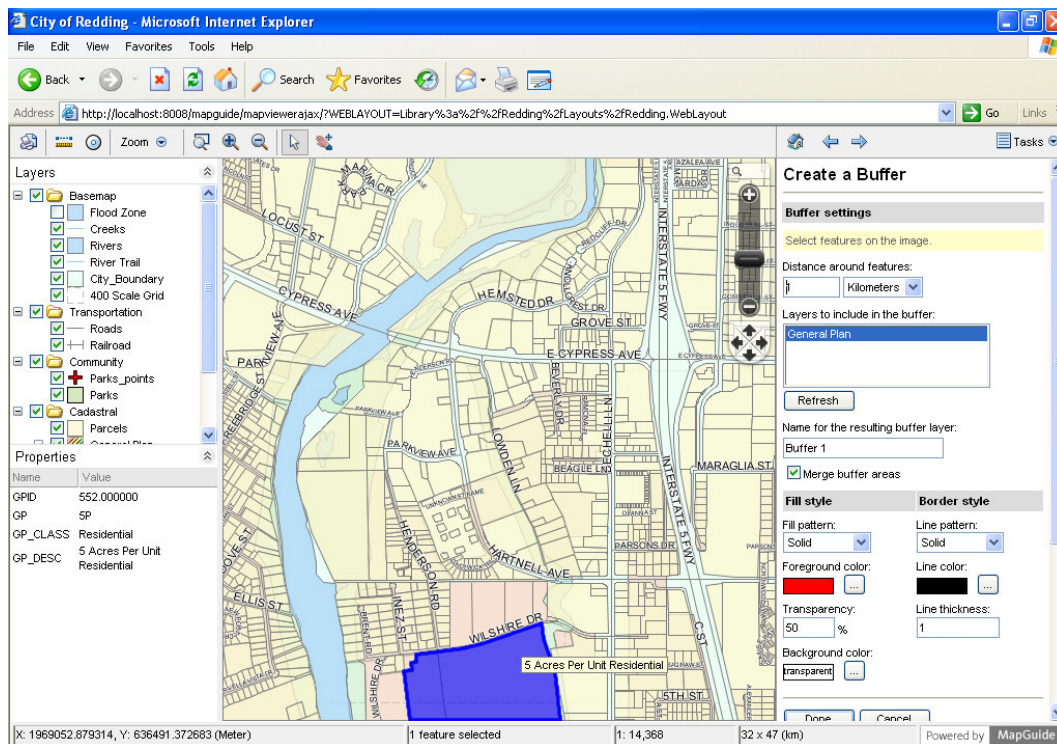
MapGuide omogoča oz. podpira:

- MS Windows, Linux, Apache, IIS, različne brskalnike,
- interaktivno pregledovanje kart,
- kvalitetne kartografske prikaze,
- vgrajeno hranjenje XML dokumentov,
- dostop do rastrskih/vektorskih podatkov z uporabo FDO (Feature Data Objects) API-ja,
- fleksibilen razvoj aplikacij - PHP, .NET, Java,
- hitro in varno strežniško platformo.

MapGuide lahko torej uporabljamo za izdelavo tematskih kart, izvedbo prostorskih analiz ter izdelave poročil.

S pregledovalnikom AJAX lahko rastrske karte pregledujemo na skoraj vsakem brskalniku. Tako lahko vsak uporabnik na poljubni platformi dostopa do prostorskih prikazov in kart, ne da bi zato

potreboval posebni brskalnik. Na sliki 6 lahko vidimo primer pregledovanja rastrskih kart s pomočjo brskalnika Internet Explorer.



**Slika 6:** Prikaz pregledovanja rastrov z aplikacijo Internet Explorer (The Open Source Geospatial Foundation, 2009)

Poleg opisanih odprtokodnih spletnih aplikacij, najdemo tudi druge kot so npr.:

- Mapnik - C++/Python 'knjižnica', ki se uporablja v aplikaciji OpenStreetMap,
- GeoServer – odprtokodni strežnik, ki omogoča uporabnikom dodajanje in spreminjanje prostorskih podatkov,
- ObjectFX SpatialFX Web Mapping Tools, OGC Web Map Service.

## 4.2.2 Namizni odprtokodni GIS projekti

### 4.2.2.1 Quantum GIS

Quantum GIS (QGIS) je uporabniku prijazen odprtokodni geografski informacijski sistem (The Open Source Geospatial Foundation, 2009), ki deluje na operacijskih sistemih GNU/Linux, Unix, Mac OSX in MS Windows. Podpira vektorske, rastrske ter različne formate podatkovnih baz in je izdan pod GNU GPL.

S QGIS lahko brskamo, urejamo in ustvarjamo različne vektorske in rastrske formate, kar vključuje ESRI 'shape' datoteke, prostorske podatke v PostgreSQL/PostGIS, GeoTiff ter rastre in vektorje aplikacije GRASS. Možno je izdelati tudi posebne aplikacije s pomočjo programskih jezikov Python in C++, karte pa lahko pripravimo za tisk, s pomočjo posebne aplikacije 'print composer'.

Posebne možnosti QGIS so tudi npr. vnos tekstovnih podatkov, prenos poti ali točk z GPS sprejemnika, pa tudi vizualizacija WMS in WFS slojev.

QGIS omogoča oz. podpira:

- rastrske in vektorske formate,
- povezavo z programskim paketom GRASS GIS,
- orodja za digitalizacijo,
- aplikacijo 'print composer',
- prostorske zaznamke,
- pregledovanje in urejanje atributnih podatkov,
- označevanje, urejanje, identificiranje pojavov/objektov,
- razširjene OGC standarde (npr. WMS, WFS),
- projiciranje kart v realnem času.

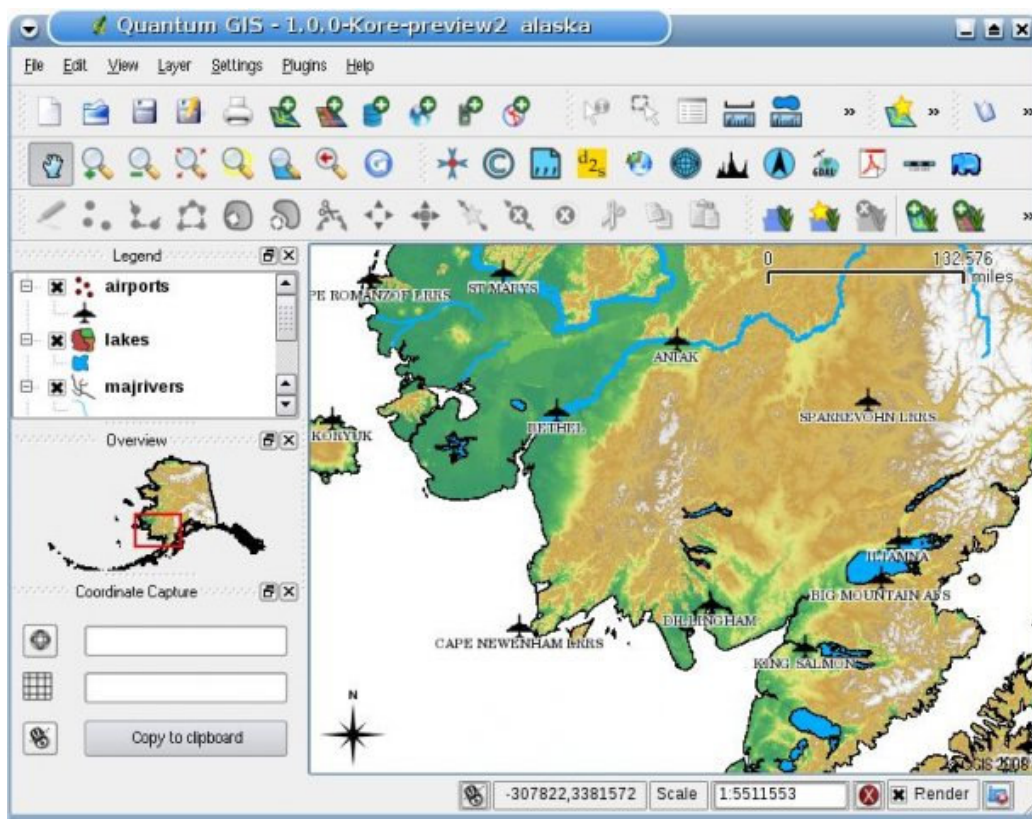
Za QGIS je na voljo tudi veliko dodatnih vgradnih aplikacij (t.i. vsadkov) s katerimi lahko na karto dodamo:

- oznako avtorja oz. avtorskih pravic (copyright),
- 'puščico severa',
- merilo,
- stavbe,

ali pa nam ti dodatni programi omogočajo:

- shranitev podatkov v obliki, ki jo lahko uporabimo v Mapserver-ju,
- orodja za uporabo GPS-a,
- georeferenciranje rastrskih podatkov,
- povezavo s programskim paketom GRASS itd.

Na sliki 7 je prikazan grafični uporabniški vmesnik aplikacije QGIS.



Slika 7: Prikaz uporabe aplikacije QGIS (The Open Source Geospatial Foundation, 2009)

#### 4.2.2.2 gvSIG

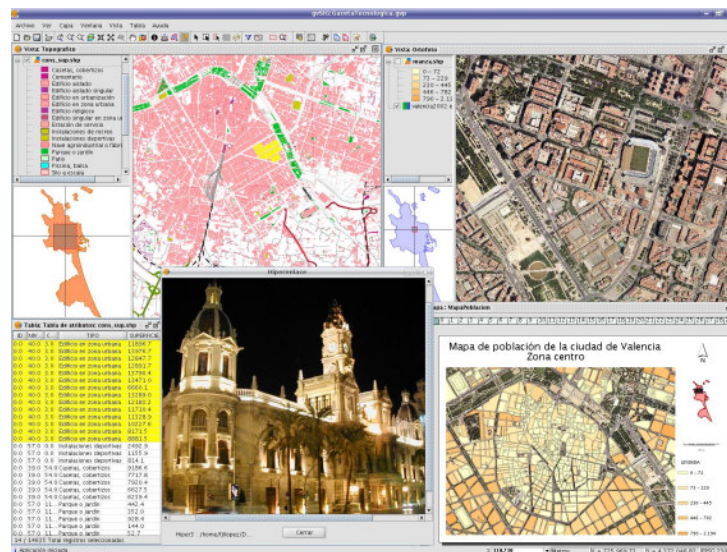
Aplikacija gvSIG je geografski informacijski sistem (The Open Source Geospatial Foundation, 2009), ki je kot namizna aplikacija namenjena zajemanju, shranjevanju, urejanju, analiziranju in porazdeljevanju raznovrstnih prostorskih podatkov, z namenom reševanja kompleksnih problemov načrtovanja in upravljanja. gvSIG je znan po tem, da ima uporabniku prijazen vmesnik ter da lahko uporablja običajne formate datotek – rastrske in vektorske. Obsega široko paleto orodij za delo s prostorskimi podatki (poizvedovanja, izdelava predlog, geoprocesiranje, omrežja ...), ki je potrebna za uporabnike, ki se kakorkoli ukvarjajo s prostorsko povezanimi problemi.

Aplikacijo gvSIG lahko enostavno širimo oz. razvijamo po lastnih željah, saj kot odprtokodna programska oprema, izdana pod licenco GNU/GPL, dopušča svobodno uporabo, distribucijo in nadgradnjo. gvSIG je bil razvit v programskem jeziku Java, uporabljamo pa ga lahko na operacijskih sistemih Linux, Windows ter Mac OS X, in sicer že v številnih jezikih (angleški, španski, francoski, nemški, italijanski ...).

gvSIG omogoča oz. podpira:

- večino rastrskih, vektorskih in formatov podatkovnih baz,
- razširjene OGC standarde (WMS, WFS, WCS, WMC),
- široko paleto standardnih GIS orodij (poizvedbe, merjenje razdalj, legenda, opisni podatki, območne operacije, prekrivanje slojev ...),
- napredna CAD orodja (delo z vektorskimi podatki, topologija, mreža ...),
- napredna rastrska orodja (georeferenciranje, transparentnost, kontrast ...),
- mrežne analize, 3D vizualizacije, klasifikacija razredov ...,
- izdelavo poročil, mobilni gvSIG itd.

Na sliki 8 je prikazan primer uporabe aplikacije gvSIG.



**Slika 8:** Prikaz uporabe aplikacije gvSIG (The Open Source Geospatial Foundation, 2009)

Druge odprtokodne namizne aplikacije GIS:

- GRASS GIS – celovita aplikacija GIS, ki je podrobneje predstavljena v naslednjih poglavjih,
- SAGA GIS (System for Automated Geoscientific Analyses) - sistem za samodejne geoznanstvene analize,
- MapWindow GIS – odprtokodna namizna aplikacija GIS ter komponenta za programiranje,
- ILWIS (Integrated Land and Water Information System) – sistem, ki združuje podobe, vektorske ter tematske podatke,
- uDig – celovit odprtokodni GIS napisan v Javi,
- JUMP GIS (Java Unified Mapping Platform) – GIS izdelan v Javi, iz katerega so nastali še OpenJUMP, SkyJUMP, deeJUMP in Kosmo,
- Kalypso – odprtokodni GIS, ki se posveča numeričnim simulacijam pri upravljanju voda,
- TerraView – namizni GIS, ki upravlja vektorske in rastrske podatke v relacijski podatkovni bazi,
- Capaware – izjemno hiter C++ 3D GIS, s številnimi vsadki za prostorske analize in vizualizacije.



### 4.2.3 "Prostorske knjižnice"

#### 4.2.3.1 GDAL/OGR

GDAL/OGR (Geospatial Data Abstraction Library) je "prevajalna knjižnica" za rastrske in vektorske formate prostorskih podatkov (The Open Source Geospatial Foundation, 2009), ki je izdana pod X/MIT odprtokodno licenco organizacije OSGeo. Služi kot dodatek prostorskim aplikacijam, da le-te lahko 'preberejo' različne podatkovne formate, pa tudi za razne pretvorbe in obdelave prostorskih podatkov.

GDAL podpira preko 50 rastrskih formatov, OGR pa preko 20 vektorskih formatov.

Omogoča 'dostop' do podatkov za mnoge aplikacije kot so MapServer, GRASS, QGIS in OpenEV, uporablja pa se tudi v programskih paketih kot so OSSIM, Cadcorp SIS, FME, Google Earth, VTP, Thuban, ILWIS, MapGuide in ArcGIS. GDAL/OGR je torej najbolj razširjena in uporabljana "prostorska knjižnica."

GDAL/OGR omogoča oz. podpira:

- dostop iz različnih programskih jezikov kot so Python, Java, C#, Ruby, VB6 in Perl,
- vektorski podatkovni model tesno povezan z aplikacijami OGC (OGC Simple Features),
- uporabo koordinatnih sistemov (PROJ.4) in tekstovnih opisov le-teh,
- orodja za podatkovne translacije, obdelavo podob, t.i. "subsetting" itd.,
- zelo učinkovit dostop do rastrskih podatkov (različni pogledi ipd.),
- uporabo velikih datotek (večjih od 4GB).

V preglednici 1 so zapisani najbolj razširjeni rastrski in vektorski formati "prostorske knjižnice" GDAL/OGR.

**Preglednica 1:** Najbolj razširjeni rastrski in vektorski formati prostorske knjižnice GDAL/OGR  
(The Open Source Geospatial Foundation, 2009)

<b>Raster</b>	<b>Vektor</b>
GeoTIFF	MapInfo (tab and mid/mif)
Erdas Imagine	ESRI Shapefile
SDTS	ESRI Coverages
ESRI Grids	ESRI Personal Geodatabase
ECW	DGN
MrSID	GML
JPEG2000	PostGIS
DTED	Oracle Spatial

#### 4.2.3.2 GeoTools

GeoTools je odprtokodni programski paket orodij GIS (The Open Source Geospatial Foundation, 2009), razvit v okolju Java in vsebuje številne razvijajoče se aplikacije OGC-ja. GeoTools poznamo tudi v povezavi s projektom GeoAPI, ki razvija prostorske Java vmesnike.

Programska koda paketa GeoTools je zgrajena s pomočjo najnovejših Java aplikacij/okolij, njena arhitektura pa omogoča dodatno funkcionalnost za enostavno nadgradnjo. GeoTools oz. izvorna koda je izdana pod GNU LGPL licenco.

GeoTools podpira oz. omogoča:

- OGC mrežna prekrivanja,
- koordinatne referenčne sisteme in transformacije,
- 'simbologijo' z uporabo OGC SLD (Styled Layer Descriptor) specifikacije,
- atributne in prostorske filtre z uporabo "OGC Filter Encoding" specifikacije,
- grafe in mrežne prikaze,
- topološke kontrole vektorskih objektov – JTS (Java Topology Suite),
- dva načina "izdelave grafičnih podob" (angl. rendering):
  - 1) LiteRenderer (za omrežne prikaze, ki ne zahtevajo veliko pomnilnika),
  - 2) J2D – 'renderer različnih stanj', primeren za interaktivne prikaze.

V preglednici 2 so zapisani najbolj razširjeni rastrski in vektorski formati programskega paketa orodij GeoTools.

**Preglednica 2:** Najbolj razširjeni rastrski in vektorski formati GeoTools (The Open Source Geospatial Foundation, 2009)

<b>Raster</b>	<b>Vektor</b>
OGC Web Mapping Server (WMS)	ESRI Shapefile
ArcGrid	OGC Geography Markup Language (GML)
GeoTIFF	OGC Web Feature Server (WFS)
podobe georeferencirane z "world datoteko"	PostGIS
	Oracle Spatial
	ArcSDE
	MySQL

Na spletu najdemo tudi druge 'prostorske knjižnice', kot sta npr.:

- Proj.4 – knjižnica kartografskih projekcij in
- OpenLayers – odprtokodna knjižnica AJAX, ki služi za dostopanje do različnih (prostorskih) podatkovnih slojev.

#### 4.2.4 GeoNetwork - katalog metapodatkov

GeoNetwork je odprtokodna aplikacija, ki služi kot katalog prostorskih virov (The Open Source Geospatial Foundation, 2009). Omogoča zmogljivo urejanje metapodatkov, orodja za iskanje ter interaktivno pregledovanje spletnih kart. Aplikacija je bila razvita z namenom, da bi povezali ustanove, ki se ukvarjajo s prostorskimi podatki in tako ustvarili sodobno arhitekturo prostorskih podatkov. Na ta način dobimo zmogljivo orodje ob nizkih stroških, kar se sklada s koncepti odprtokodne organizacije FOSS ter odprtih standardov.

GeoNetwork omogoča oz. podpira:

- neposredno iskanje po "prostorskih katalogih",
- prenos podatkov (podob, dokumentov ipd.),
- interaktivni spletni pregledovalnik kart, ki povezuje različne spletne GIS aplikacije,

- aktualne posodobitve, na voljo tudi preko RSS tehnologije,
- spletno urejanje metapodatkov z integriranimi številnimi predlogami,
- podporo za ISO19115/ISO19119/ISO19139, FGDC idr.,
- sinhronizirano shranjevanje metapodatkov med posameznimi katalogi,
- skupinski in uporabniški management,
- uporabniški vmesnik v več jezikih.

Razvoj programske opreme poteka zelo hitro, tako da je nemogoče zajeti vse obstoječe odprtokodne aplikacije GIS v obravnavo. Popolnejši seznam ter pregled proste programske opreme na področju tehnologije GIS najdemo na spletni strani <http://freegis.org>.

## 5 GRASS GIS

V nadaljevanju bom podrobneje predstavil programski paket GRASS GIS (Geographic Resources Analysis Support System, 2009), saj se mi zdi med vsemi predhodno opisanimi najpomembnejši. Programski paket GRASS GIS (v nadaljevanju GRASS) sem uporabil tudi za reševanje nekaterih GIS oz. prostorskih problemov, ki jih bom v nadaljevanju opisal. Na sliki 9 vidimo logotip aplikacije GRASS:



**Slika 9:** Logotip GRASS GIS (GRASS, 2009)

### 5.1 Zgodovina

Programski paket GRASS (Geographic Resources Analysis Support System) je nastal v zgodnjih 80. letih prejšnjega stoletja, in sicer pod okriljem ameriške vojske oz. njenega raziskovalnega laboratorija - CERL (Construction Engineering Research Laboratory). Prvotno je bil razvit za potrebe upravljanja in gospodarjenja z zemljišči vojaških vadbenih območij, dandanes pa je postal eden izmed najbolj celovitih, večnamensko uporabnih ter zmogljivih okolij GIS. Zmogljivosti računalniške strojne opreme so bile v 80-ih letih precej klavrne, toda s pomočjo nadebudnih mladih programerjev in raziskovalcev se je 'rodila' prva delujoča različica programa GRASS, ki je bila napisana v programskem jeziku C. S pojavom interneta se je povezovanje različnih raziskovalnih organizacij ter širjenje programske opreme zelo poenostavilo. Tako je pospešeno stekel razvoj GRASS-a oz. CERL-a in tako je slednji v zgodnjih 90-ih letih prejšnjega stoletja ustanovil fundacijo "Open GRASS Foundation", ki se je nato preoblikovala v Open GIS

Consortium (OGC, danes znan kot Open Geospatial Consortium). V letu 1996 se je končal razvoj GRASS-a s strani CERL-a, kar je dve leti kasneje privedlo do nastanka t.i. "razvojne ekipe GRASS-a" (GRASS Development Team). Oktobra 1999 je bil programski paket GRASS prvič izdan kot prosta programska oprema, in sicer pod licenco GNU GPL. Dandanes je GRASS rezultat dela številnih znanstvenikov, programerjev, zahtevnih uporabnikov in entuziastov iz celega sveta. Projekt je privlačen predvsem za uporabnike nove generacije zaradi posodabljanja programske opreme, ki poteka s pomočjo sistema za nadzor izvorne kode CVS (angl. Concurrent Version Control System, sistem za kontrolo hkratnih različic) ter takojšnjim strokovnim pregledom sprememb izvorne kode, ki se izvaja kar prek elektronske pošte. V letu 2006 je nastal OSGeo, ki je združil številne "FOSS4G" projekte, med drugim tudi GRASS, pod okrilje ene organizacije. Tako si danes OSGeo prizadeva, da bi GRASS postal celovita programska oprema za upravljanje z okoljem/zemljišči, z 2D/3D rastrskimi in vektorskimi orodji, z možnostjo upravljanja opisnih podatkov (SQL) ter možnostjo povezovanja z drugimi "FOSS4G" aplikacijami. Kot rečeno, pri razvijanju aplikacij sodelujejo strokovnjaki iz celega sveta, t.i. razvojna ekipa GRASS-a pa je trenutno situirana v raziskovalnem centru ITC-irst (Centro per la Ricerca Scientifica e Tecnologica) v Trentu v Italiji. GRASS je na voljo na spletni strani <http://grass.itc.it/>, v 'izvršljivih' binarnih oblikah za različne UNIX, MacOS X in MS-Windows platforme skupaj s celotno C-izvorno programsko kodo.

## 5.2 GRASS danes, specifikacije, lastnosti

GRASS je rastrsko/vektorski geografski informacijski sistem, ki vključuje orodja za obdelavo podob ter vizualizacije prostorskih podatkov. Vsebuje edinstvene algoritme in metode za obdelavo podatkov in je eden izmed redkih GIS, ki deluje na različnih operacijskih sistemih in platformah. V osnovi gre za namizni GIS, zadnje čase pa skušajo povezati in vgraditi različne spletne prostorske aplikacije, kar bi GRASS naredilo še bolj celovit in funkcionalen programski paket. Napisan je v programskem jeziku C, deluje pa na sledečih operacijskih sistemih:

- GNU/Linux (Intel, PowerPC, Sun, ...),
- Solaris (SPARC, i86),
- SGI IRIX,

- HP UX,
- Mac OS X (Darwin),
- IBM AIX,
- BSD-Unix različice,
- FreeBSD,
- CRAY Unicos,
- iPAQ/Linux,
- MS-Windows (neposredno ali s pomočjo aplikacije Cygwin).

GRASS podpira številne rastrske in vektorske formate (z uporabo knjižnic GDAL in OGR):

Raster: ASCII, ARC/GRID, E00, GIF, GMT, TIF, PNG, ERDAS LAN, Vis5D, SURFER (.grd), CEOS (SAR, SRTM, LANDSAT7 itd.), ERDAS LAN, HDF, LANDSAT TM/MSS, NHAP posnetki iz zraka, SAR, SPOT ...

Vektor: ASCII, ARC/INF, ARC/INFO E00, ArcView SHAPE (s topološkimi popravki), BIL, DLG (U.S.), DXF, DXF3D, GMT, GPS-ASCII, USGS-DEM, IDRISI, MOSS, MapInfo MIF, TIGER, VRML, CSV, dBase ...

Celoten seznam je na voljo na spletnih straneh:

[http://www.gdal.org/formats\\_list.html](http://www.gdal.org/formats_list.html)

[http://www.gdal.org/ogr/ogr\\_formats.html](http://www.gdal.org/ogr/ogr_formats.html)

### **5.3 Struktura programskega paketa GRASS**

GRASS je sestavljen iz številnih modulov, ki služijo za hitro in učinkovito analizo oz. uporabo rastrskih, vektorskih in opisnih podatkov. Ti moduli so zgrajeni tako, da so čim bolj optimalni pri opravljanju specifične naloge. Kot pri sorodni 'UNIX metodologiji', ki povezuje številne manjše module/aplikacije za izvedbo kompleksnejše naloge, se prostorske analize v GRASS-u izvajajo na podoben način. Moduli se med seboj povezujejo ali neposredno prek t.i. pip (konstrukt

operacijskega sistema, ki se uporablja za prenos podatkov med različnimi procesi) ali posredno prek vmesnih datotek. Vsak modul obdela manjše število argumentov in vrne novo datoteko prostorskih podatkov (raster/vektor ipd.) ali pa kar modificira obstoječo karto oz. opisno tabelo. Posamezni ukazi so pregledno urejeni po funkcijskih razredih prek ukazne predpone v preglednici 3.

**Preglednica 3:** Pregled ukaznih predpon v GRASS-u (Neteler, 2008)

Predpona	Funkcijski razred	Vrsta ukaza
d.*	prikaz (angl. display)	grafični prikaz
db.*	podatkovne baze (angl. database)	upravljanje s podatkovnimi bazami
g.*	splošno (angl. general)	splošne datotečne operacije
i.*	podobe (angl. imagery)	obdelava podob
m.*	razno (angl. miscellaneous)	drugi ukazi
ps.*	“založništvo” (angl. postscript)	izdelava karte v ‘Postscript’ formatu
r.*	raster	obdelava 2D rastrov
r3.*	3D raster	obdelava 3D rastrov
v.*	vektor	obdelava vektorskih podatkov

Čeprav ta način daje izjemno moč uporabnikom, ki osvojijo celoten sklop GRASS modulov, pa ni optimalen, saj je večina uporabnikov navajena grafičnega uporabniškega vmesnika. Lahko bi rekli, da je jedro GRASS-a pravzaprav profesionalno analitično orodje, ne pa namizno orodje GIS, vendar so bili tekom let razviti številni grafični uporabniški vmesniki z uporabo TclTk orodij, zadnje čase pa tudi z orodji wxWidgets Phyton. GRASS vsebuje številne knjižnice (“rowio”, “segments” idr. knjižnice), namenjene poenostavitvi upravljanja in dostopanja do zelo velikih datotek. Nedavno je bila k izvorni kodi dodana še t.i. numerična knjižnica, ki služi kot pomoč pri pretvorbah kompleksnih algoritmov v drugačne oblike, ki lahko optimalno izkoristijo zmogljivosti računalniške strojne opreme.



## 6 DELO S PROGRAMSKIM PAKETOM GRASS

### 6.1 Začetni koraki

Programski paket GRASS sem instaliral kot program za UNIX, ki deluje v MS-Windows okolju s pomočjo programa Cygwin, nedavno pa je izšla različica WinGRASS 6.4., ki deluje samostojno tudi v Microsoftovih "oknih". Po uspešni instalaciji sledi prvi zagon programa, pri katerem je potrebno najprej definirati mapo, v kateri so shranjeni podatki (npr. Grassdata), nato pa ustvariti novo 'lokacijo', znotraj katere lahko imamo številna delovna področja (angl. mapsets). Sledi izbira koordinatnega sistema, kar lahko naredimo na več načinov:

- izberemo ga s spustnega seznama,
- izberemo t. i. EPSG (European Petroleum Survey Group) kodo,
- uporabimo koordinatni sistem georeferencirane datoteke,
- uporabimo koordinatni sistem iz WTK ali PRJ datotek,
- ustvarimo svoje parametre koordinatnega sistema (PROJ.4),
- uporabimo lokalni xy koordinatni sistem.

Ustvaril sem novo lokacijo z imenom *Koper*, definiral državni koordinatni sistem ter dodal delovno področje z imenom *vaja*. Po teh začetnih korakih se delo lahko prične.

### 6.2 Uvoz podatkov in prikaz

S pomočjo orodij GDAL/OGR sem uvozil podatke za list 2027 TTN5 (območje v bližini Kopra):

- DMV 12,5 (digitalni model višin z resolucijo 12.5m) v formatu .grd, ki sem ga shranil kot rastrski sloj *dem*,
- ortofoto posnetek območja, shranil sem ga kot RGB podobo – 3 rastrski sloji: *ortofoto.red*, *ortofoto.green*, *ortofoto.blue*,
- sloj rabe tal izbranega območja v formatu .shp, ki sem ga shranil kot vektorski sloj *raba*.

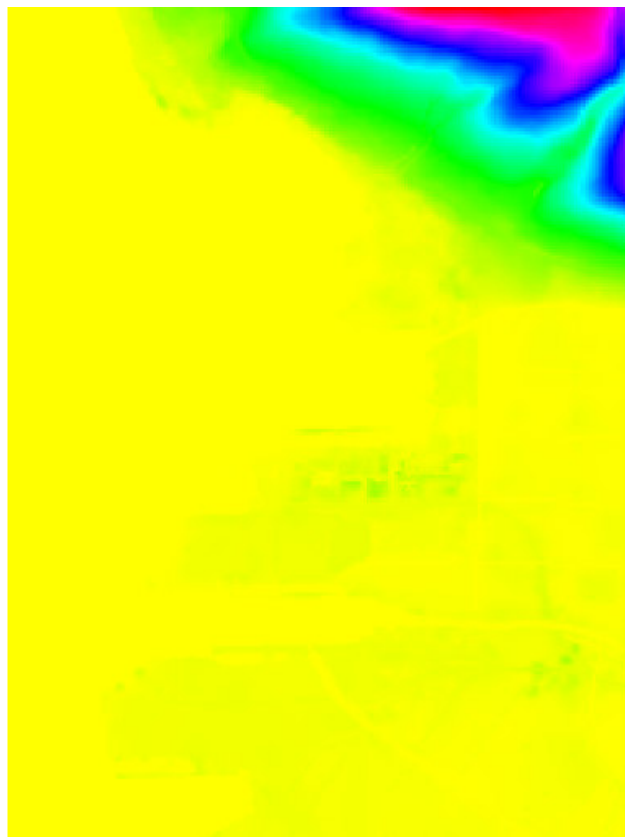
Pri uvozu ortofota izbranega območja so nastali trije različni rastrski sloji, saj gre za t.i. 'RGB' podobo. Nastali so torej rastrski sloji za tri 'barvne kanale' – rdeč (R), zelen (G) ter moder (B).

Delo v GRASS-u lahko poteka prek grafičnega uporabniškega vmesnika ali s pomočjo t.i. ukazne vrstice. Za prikaz uvoženega DMV-ja lahko torej enostavno kliknemo na ikono za prikaz izbranih slojev, ali pa v ukazno vrstico vnesemo ukaz:

```
d.rast map=dem@vaja
```

kjer je 'd' predpona za prikaz, 'rast' nam pove, da gre za rastrski sloj, map=dem@vaja pa nam pove, da gre za sloj z imenom *dem* v delovnem področju *vaja*.

Na sliki 10 je prikazan DMV 12,5 v običajni barvni lestvici.



**Slika 10:** DMV 12,5 v običajni barvni lestvici

Če nas zanimajo splošne informacije o tem sloju oz. t.i. metapodatki (t.j. 'podatki o podatkih'), jih prikažemo z ukazom `r.info dem`:

```
+-----+
| Layer:      dem                               Date: Tue Jul 14 18:02:46 2009 |
| Mapset:     vaja                             Login of Creator: Matija  |
| Location:   Koper                            |
| DataBase:   C:/Documents and Settings/Matija/My Documents/GIS DataBase |
| Title:      ?? ( dem )                      |
| Timestamp:  none                            |
+-----+
|
| Type of Map: raster                          Number of Categories: ?? |
| Data Type:   FCELL                           |
| Rows:        241                             |
| Columns:     181                             |
| Total Cells: 43621                           |
| Projection:  x,y                             |
|             N: 49006.25   S: 45993.75   Res: 12.5 |
|             E: 403256.25  W: 400993.75   Res: 12.5 |
| Range of data: min = 0.000000 max = 109.480003 |
|
| Data Description:                             |
| generated by r.in.arc                         |
|
| Comments:                                     |
| r.in.arc input="C:\Documents and Settings\Matija\My Documents\GIS Da\ |
| taBase\VTB2027.GRD" output="dem" type="FCELL" mult=1.0 |
+-----+
```

Iz dobljenega 'poročila' lahko razberemo osnovne podatke o rastrskem sloju *dem*, nekatere od njih bomo kasneje uporabili pri obdelavi podatkov.

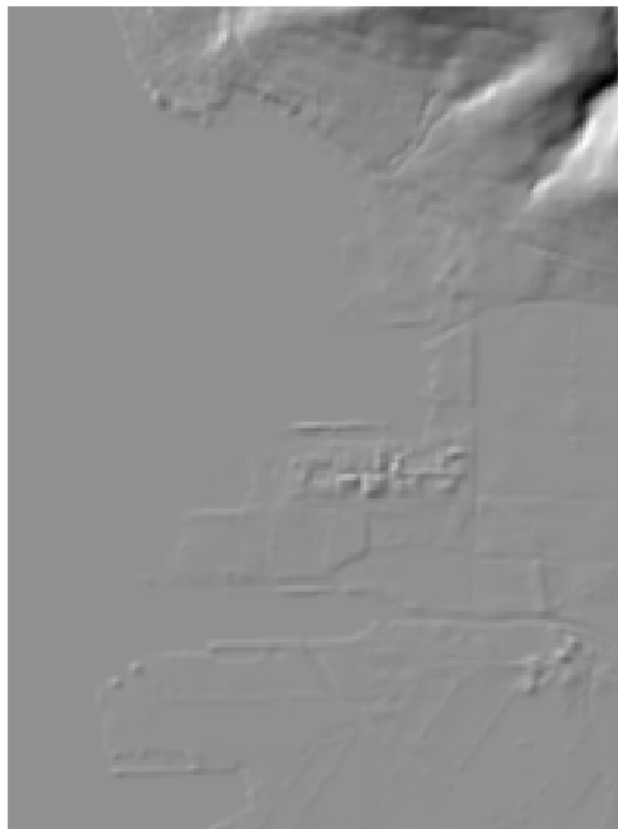
## 6.3 Rastrske operacije

### 6.3.1 Operacije z DMV-jem

Dobljeni DMV lahko prikažemo s senčenjem (siva barvna lestvica, parametri "sonca" oz. osvetlitve: azimut  $315^\circ$ , zenitna razdalja  $45^\circ$ ):

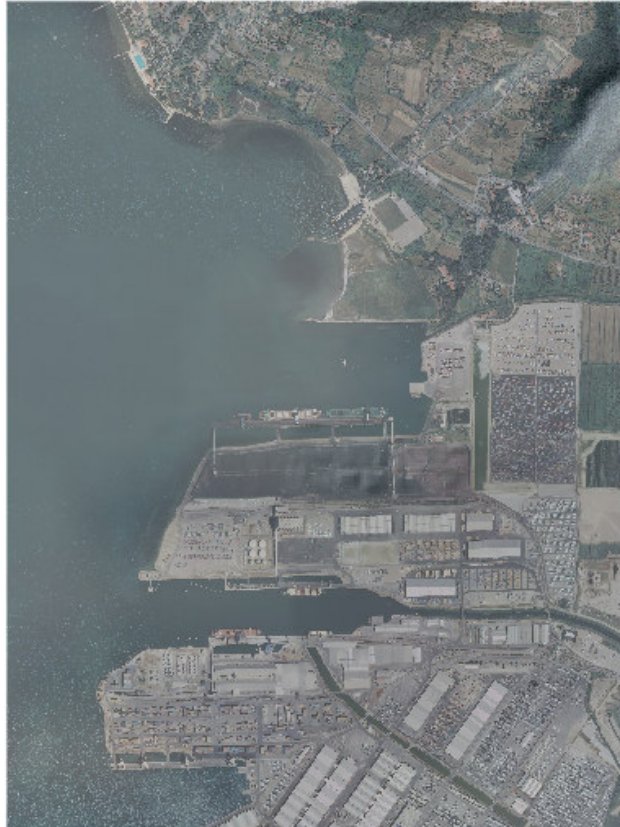
```
r.shaded.relief map=dem@vaja shadedmap=sencenje altitude=45 ...  
azimuth=315
```

Slika 11 prikazuje senčeni DMV, na katerem se jasno vidi, kje so ravnine in kje "vzpetine".



**Slika 11:** Senčeni DMV

Nato sem uporabil rastrsko operacijo *overlay* – prekrivanje slojev *sencenje* ter *ortofoto* (*ortofoto.red*, *ortofoto.green*, *ortofoto.blue*). Uporabil sem 50% prosojnost sloja *ortofoto*, izdelek (slika 12) pa žal ni najboljši, saj gre za relativno ravninsko območje.

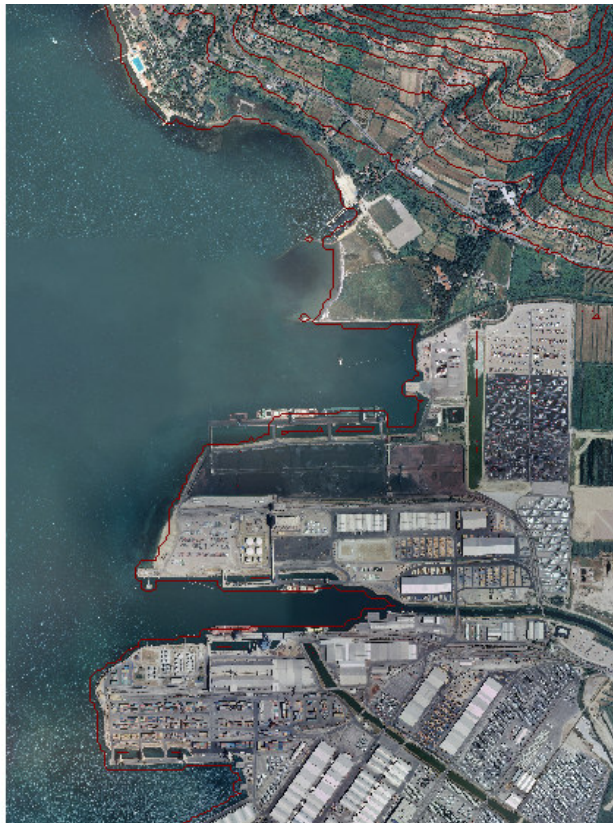


**Slika 12:** Prekrivanje ortofota in senčenega DMV-ja

Iz DMV-ja lahko ustvarimo tudi vektorski sloj plastnic z ukazom (parametri: vhodni rastrski sloj 'dem', izhodni vektorski sloj 'plastnice', ekvidistanca 10m):

```
r.contour input=dem@vaja output=plastnice step=10
```

Če zopet prikažemo ortofoto, na katerega tokrat položimo plastnice v rjavi barvi, dobimo sliko 13:



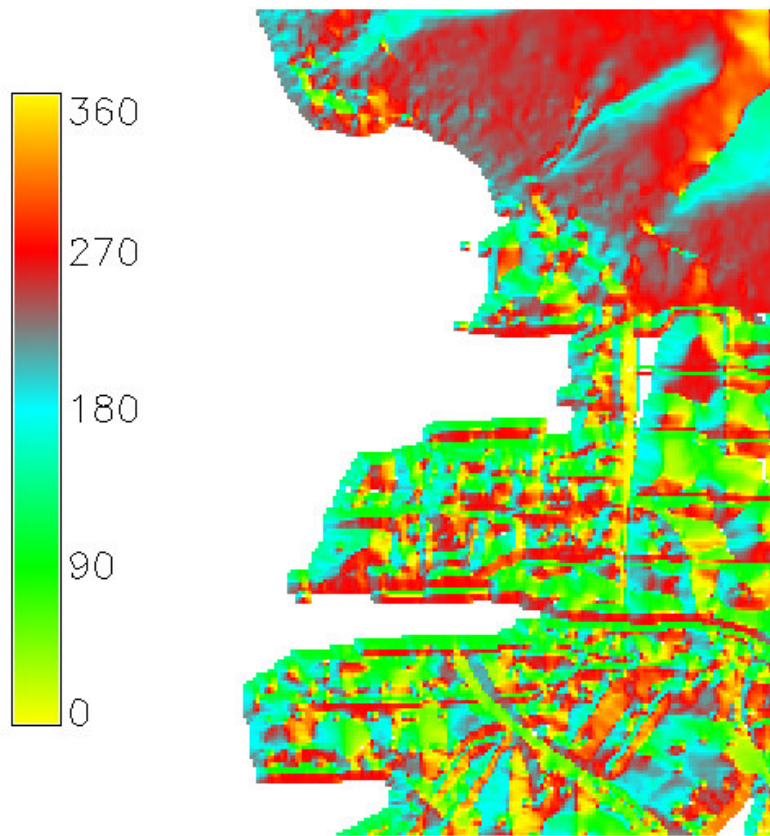
**Slika 13:** Prikaz plastnic na podlagi ortofota

S pomočjo DMV-ja lahko določimo tudi naklon in usmerjenost površja. Naklon se meri s spremembo z-koordinate (višine) na določeni razdalji, usmerjenost pa pomeni orientacijo celic glede na smer severa oz. azimut naklona terena.

```
r.slope.aspect elevation=dem@vaja slope=naklon aspect=usmerjenost
```

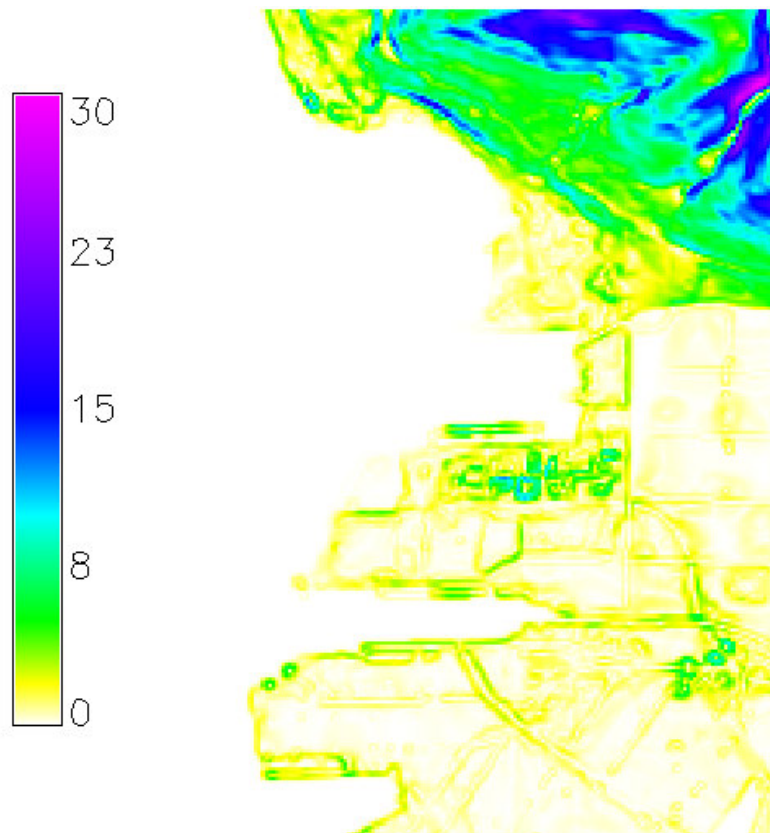
Za lažje razumevanje dobljenih slojev prikazu dodamo še legendo z ukazom `d.legend`.

Na sliki 14 je prikazan rastrski sloj usmerjenosti terena, legenda je v 'kotnih stopinjah'.



**Slika 14:** Usmerjenost terena z legendo

Na sliki 15 pa je prikazan rastrski sloj naklona terena, ki je tako kot usmerjenost, prikazan v kotnih stopinjah.



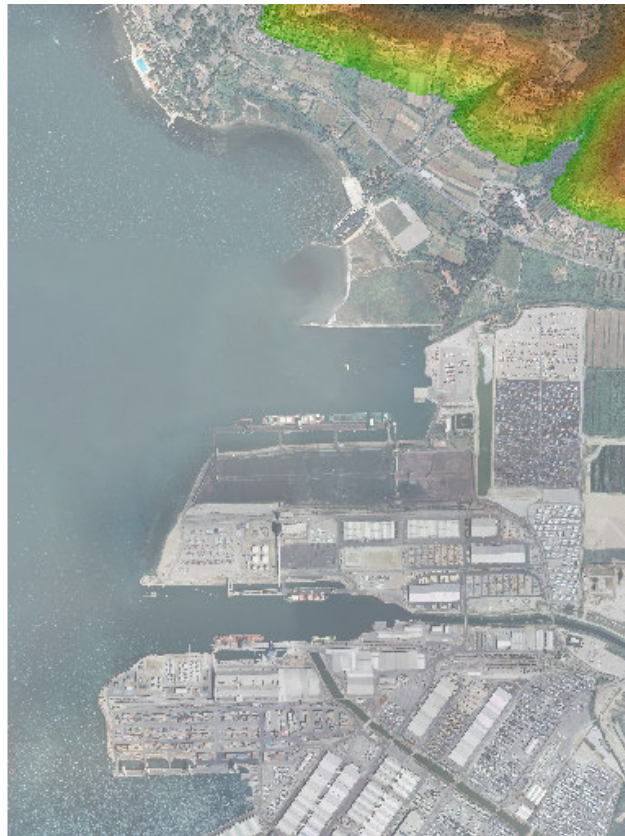
**Slika 15:** Naklon terena z legendo

Če sedaj pogledamo v poročilo o sloju *dem*, lahko vidimo, da je razpon nadmorskih višin DMV-ja od 0 do 109 m. Če želimo prikazati območja, ki so višja od 30 m nadmorske višine, to storimo z ukazom:

```
d.rast dem val=30-110
```

Dobljena območja prikažemo na sliki 16 na podlagi ortofota:





**Slika 16:** Prikaz območij z nadmorsko višino od 30 do 110m

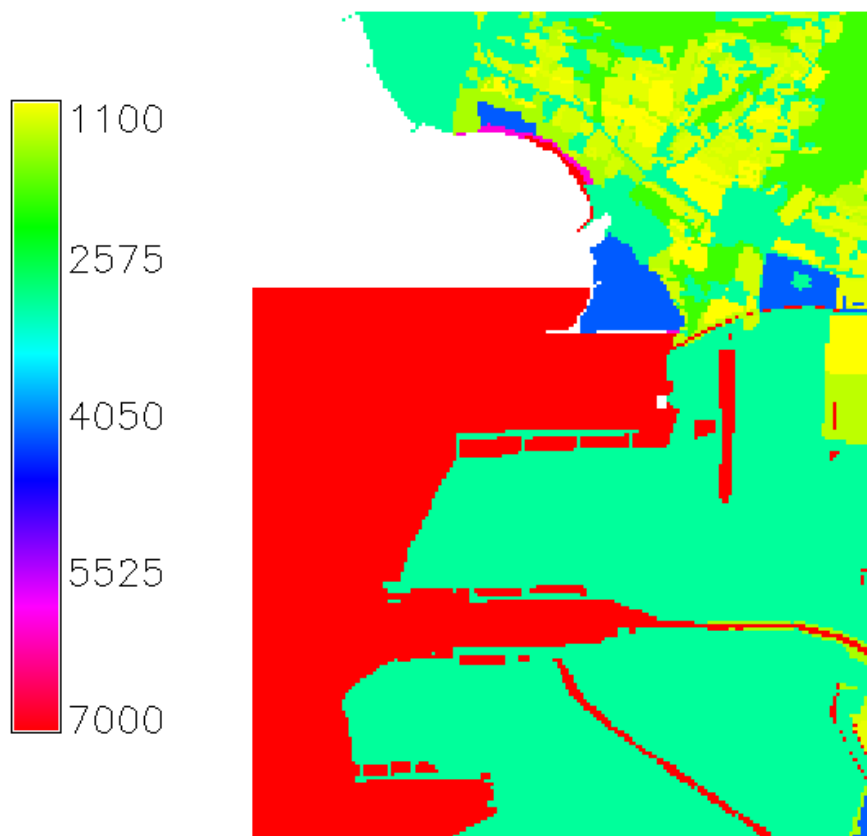
### **6.3.2 Pretvorba vektorskega sloja v rastrski sloj, reklasifikacija, barvne tabele**

Vektorske sloje lahko pretvorimo v rastre. Pretvorba vektorskih elementov v rastre je odvisna od nastavljene resolucije. Za 'prenos' vektorskih podatkovnih vrednosti oz. opisnih podatkov na novonastali raster, imamo več možnosti:

- `attr`: vrednosti se "preberejo" iz atributne tabele,
- `cat`: uporabimo vrednosti posameznih 'kategorij',
- `val`: uporabimo posebne vrednosti (s pripadajočimi parametri),
- `z`: uporabimo `z` koordinato (pri točkah in plastnicah),
- `dir`: uporabimo smeri – to je mogoče samo pri linijah.

Če sedaj pretvorimo vektorski sloj rabe tal v raster, vrednosti "vzamemo" iz atributne tabele, za določitev kategorij rastra pa uporabimo stolpec RABA\_ID.

```
v.to.rast input=raba@vaja output=raba_rast@vaja use=attr ...  
column=RABA_ID labelcolumn=RABA_ID
```



**Slika 17:** Rastrski prikaz rabe tal

Na sliki 17 je raster rabe tal prikazan v barvni lestvici mavrice. Če pogledamo *Šifrant in opis vrst dejanske rabe kmetijskih in gozdnih zemljišč* (glej prilogo A) lahko vidimo, da gre pri rumeno-zelenih območjih predvsem za kmetijska zemljišča, pri modro-rdečih pa za nekmetijska. V severozahodnem delu območja ima očitno vektorski sloj rabe tal manjkajoče podatke, oz. nepopolno topologijo, saj je v rastrskem sloju nastalo območje brez vrednosti/podatkov.

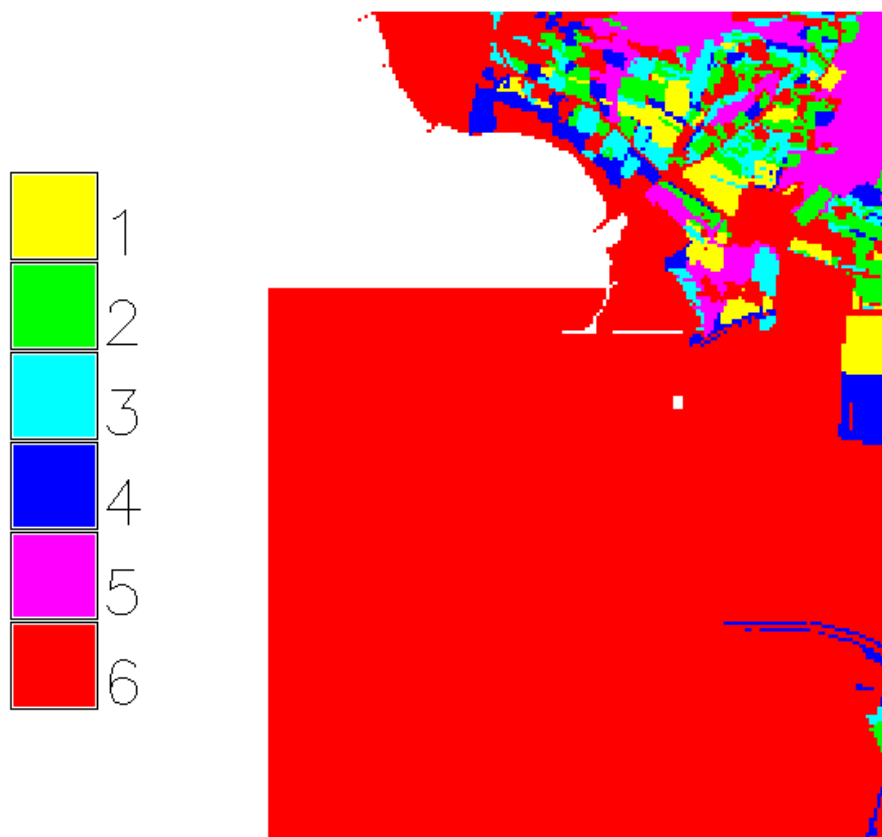
Če želimo dobiti bolj pregleden izdelek, lahko izvedemo reklasifikacijo razredov. Zopet si pomagamo s *Šifrantom in opisom vrst dejanske rabe kmetijskih in gozdnih zemljišč*, da ustvarimo npr. šest smiselnih razredov, kar zapišemo v datoteko (npr. *reclass.txt*) v obliki:

```
1100 1160 1180 1190 = 1 NJIVE  
1211 1212 1221 1222 1230 1240 = 2 NASADI  
1300 1321 1800 = 3 TRAVNIKI  
1410 1420 1500 1600 = 4 DRUGA_KMETIJSKA  
2000 = 5 GOZD  
3000 4100 4210 4220 5000 6000 7000 = 6 NEKMETIJSKA
```

Reklasifikacijo izvedemo z ukazom:

```
r.reclass input=raba_rast output=raba_reclass rules=reclass.txt
```

Na sliki 18 ima nov raster le šest danih kategorij:



**Slika 18:** Reklasificiran raster rabe tal

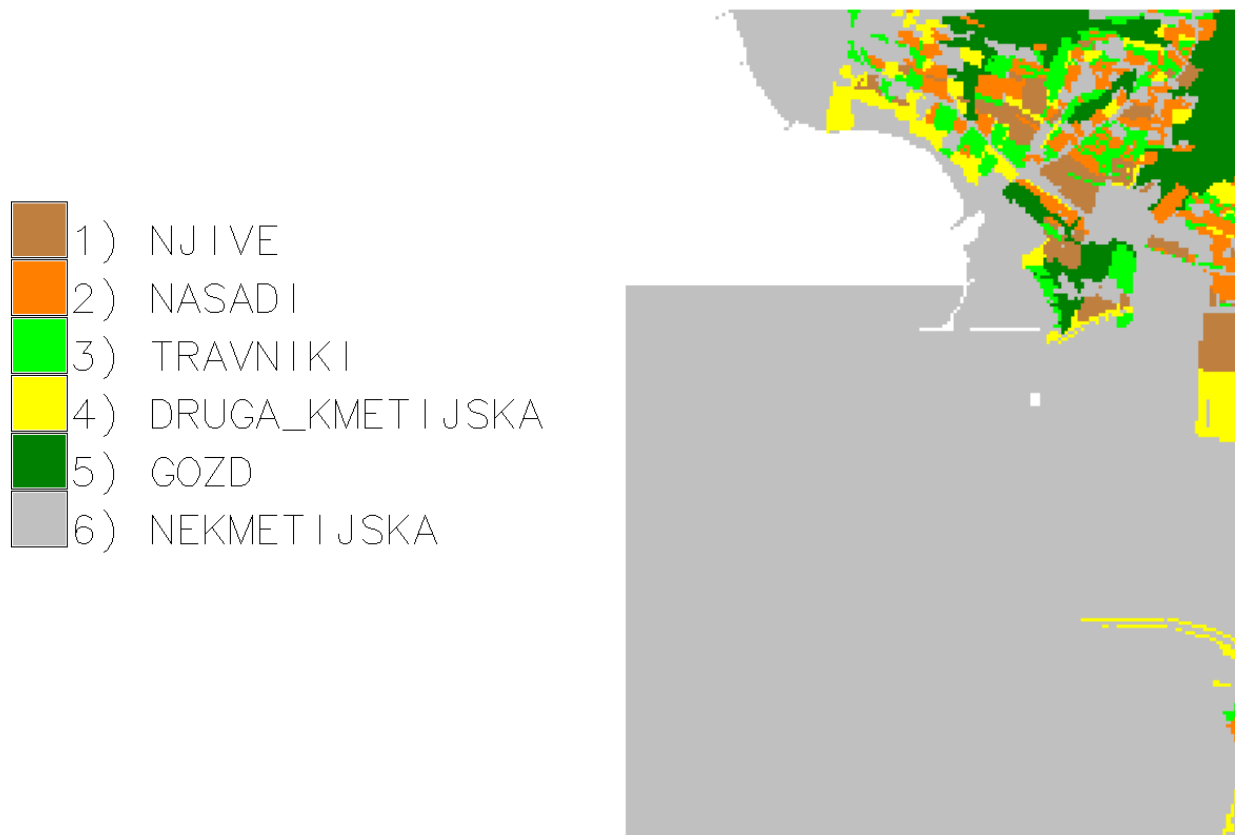
Da bo prikaz malce bolj razumljiv in asociativen, lahko spremenimo še barvno tabelo prikaza. Podobno kot za reklasifikacijo rabe tal, tudi tokrat ustvarimo datoteko (npr. *colors.txt*), v katero vpišemo nove barve pripadajočim vrednostim. Barve lahko definiramo na več načinov, in sicer:

- z imenom barve (*red, blue, yellow, black ...*),
- kot skupek zelene, rdeče in modre (RGB) barve (od 0 do 255) v obliki *rdeča:zelena:modra* (npr. *128:255:0*),
- odstotkovno – določimo pri katerem odstotku najvišje vrednosti rastra se barva spremeni (npr. *0% black 20% blue 30% yellow* itn.).

Za sestavo datoteke sem uporabil prva dva načina:

```
1 brown
2 orange
3 green
4 yellow
5 0:128:0
6 192:192:192
nv white
```

Zadnja vrstica nam pove, da so območja brez podatka ("nv = no value", oz. NULL) prikazana v beli barvi. Tako sedaj dobimo bolj jasen prikaz na sliki 19:



**Slika 19:** Prikaz rabe tal z asociativno določeno barvno tabelo

### 6.3.3 Algebra karte

Algebra karte oz. "kartno računalno" (angl. map calculator) je pripomoček, s katerim lahko izvajamo številne analize na rastrskih slojih. Te analize temeljijo na uporabi matematičnih funkcij in logičnih/relacijskih operatorjev.

Če iz prejšnje karte rabe tal prikažemo gozdne površine, je potrebno v orodje *map calculator* vpisati izraz:

```
raba_reclass@test == 5
```

Za prikaz območij višjih od 50 m, bi vpisali:

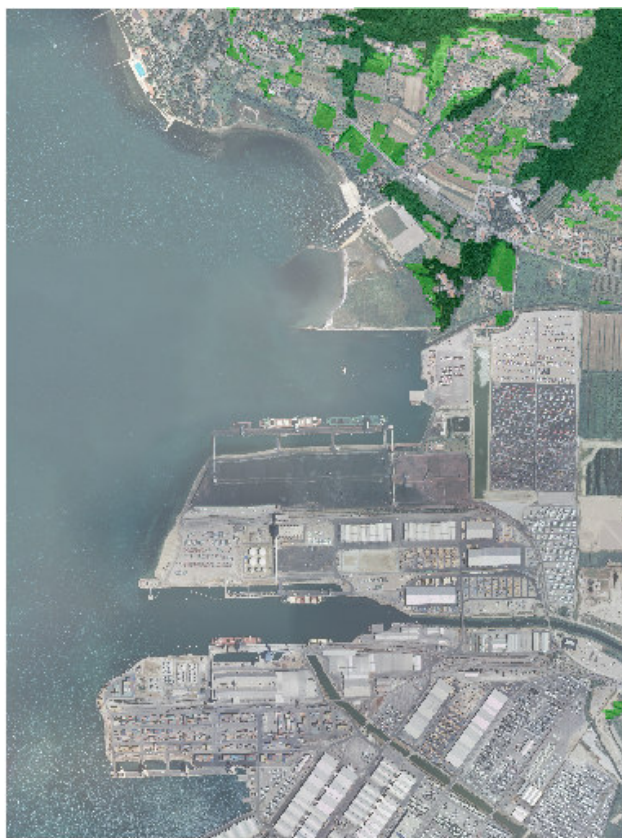
```
dem@test > 50
```

Podobno bi lahko s pomočjo relacijskih operatorjev izvedli še številne druge analize.

Funkcijo *mapcalc* lahko uporabljamo tudi neposredno iz ukazne vrstice:

```
r.mapcalc "gozd_in_travniki = if(raba_reclass==5,5,0) + \
if(raba_reclass==3,3,0)"
r.colors map=gozd_in_travniki@vaja raster=raba_reclass@vaja
```

Zgornja funkcija nam ustvari rastrski sloj *gozd\_in\_travniki*, in sicer tako, da iz sloja *raba\_reclass* vzame območja z vrednostma 3 in 5 in jim ohrani enako vrednost, drugim območjem pa pripiše vrednost 0. Dobljenemu sloju nato še pripišemo tako barvno tabelo kot jo ima sloj *raba\_reclass* in ga prikažemo na podlagi ortofota na sliki 20:



**Slika 20:** Prikaz travnikov in gozdov

## 6.4 Delo z vektorskimi podatki

### 6.4.1 Opisni podatki

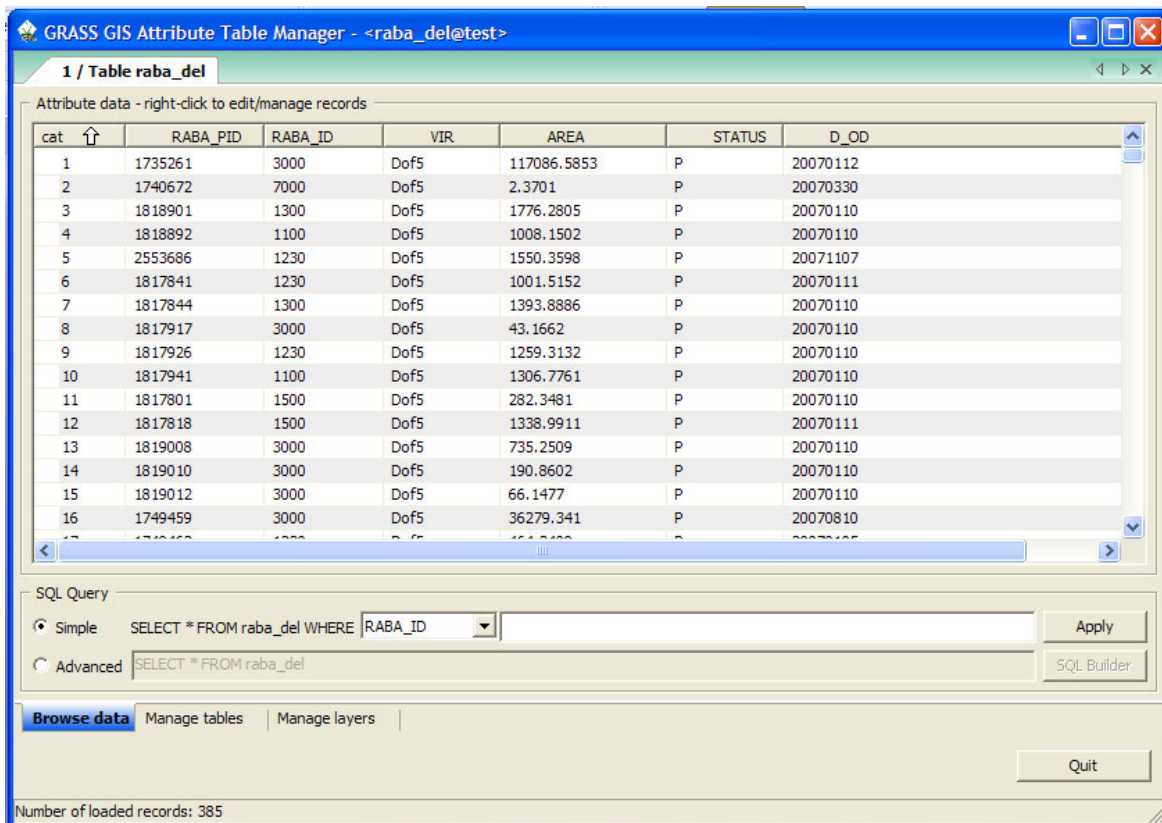
Vektorskim slojem lahko v t.i. atributno tabelo dodamo številne opisne podatke oz. t.i. attribute, ki so neposredno povezani z geometrijo vektorjev. Za urejanje, poizvedovanje in obdelavo opisnih podatkov, mora biti omogočeno delo s podatkovnimi bazami.

GRASS lahko povežemo s številnimi relacijskimi podatkovnimi bazami (angl. RDBMS), poleg tega pa združuje različne gonilnike podatkovnih baz v vmesniku DBMI (angl. database management interface). Od tipa RDBMS in gonilnika podatkovne baze je odvisno, katere SQL (angl. Structured Query Language) poizvedbe oz. ukaze lahko izvršujemo. V preglednici 4 lahko vidimo, katere gonilnike podatkovnih baz podpira GRASS.

**Preglednica 4:** Gonilniki podatkovnih baz v aplikaciji GRASS GIS (Neteler, 2008)

gonilnik	
dbf	DBF datoteke (angl. database files)
sqlite	SQLite vgrajena podatkovna baza
pg	PostgreSQL RDBMS
mysql	MySQL RDBMS
mesql	MySQL vgrajena podatkovna baza
odbc	UnixODBC (PostgreSQL, Oracle, itd.)

Če si sedaj ogledamo opisno tabelo vektorskega sloja rabe tal, ki uporablja gonilnik *dbf*, dobimo sliko 21:



The screenshot shows the 'GRASS GIS Attribute Table Manager' window for a table named 'raba\_del'. The table contains 17 rows of data with columns: cat, RABA\_PID, RABA\_ID, VIR, AREA, STATUS, and D\_OD. Below the table is an SQL Query section with 'Simple' selected and a query: 'SELECT \* FROM raba\_del WHERE RABA\_ID'. The status bar at the bottom indicates 'Number of loaded records: 385'.

cat	RABA_PID	RABA_ID	VIR	AREA	STATUS	D_OD
1	1735261	3000	Dof5	117086.5853	P	20070112
2	1740672	7000	Dof5	2.3701	P	20070330
3	1818901	1300	Dof5	1776.2805	P	20070110
4	1818892	1100	Dof5	1008.1502	P	20070110
5	2553686	1230	Dof5	1550.3598	P	20071107
6	1817841	1230	Dof5	1001.5152	P	20070111
7	1817844	1300	Dof5	1393.8886	P	20070110
8	1817917	3000	Dof5	43.1662	P	20070110
9	1817926	1230	Dof5	1259.3132	P	20070110
10	1817941	1100	Dof5	1306.7761	P	20070110
11	1817801	1500	Dof5	282.3481	P	20070110
12	1817818	1500	Dof5	1338.9911	P	20070111
13	1819008	3000	Dof5	735.2509	P	20070110
14	1819010	3000	Dof5	190.8602	P	20070110
15	1819012	3000	Dof5	66.1477	P	20070110
16	1749459	3000	Dof5	36279.341	P	20070810
17	1749460	3000	Dof5	101.8400	P	20070810

Slika 21: Opisna tabela vektorskega sloja rabe tal

kjer so posamezni opisni podatki:

cat	ID posameznega vektorskega objekta (angl. category number)
RABA_PID	neponovljiva številka poligona rabe tal
RABA_ID	šifra dejanske rabe zemljišč
VIR	vir zajema podatkov
AREA	površina poligona (m <sup>2</sup> )
STATUS	P - veljavno stanje
D_OD	datum zadnje spremembe

Če nas zanimajo npr. samo opisni podatki (brez grafičnega prikaza), lahko po njih poizvedujemo kar v prikazu atributne table z uporabo 'SQL' poizvedb. Za npr. prikaz seznama vseh parcel, ki so manjše od 100m<sup>2</sup>, vtipkamo preprosto poizvedbo:



```
SELECT * FROM raba_del WHERE AREA < 100
```

Rezultat je prikazan na sliki 22:

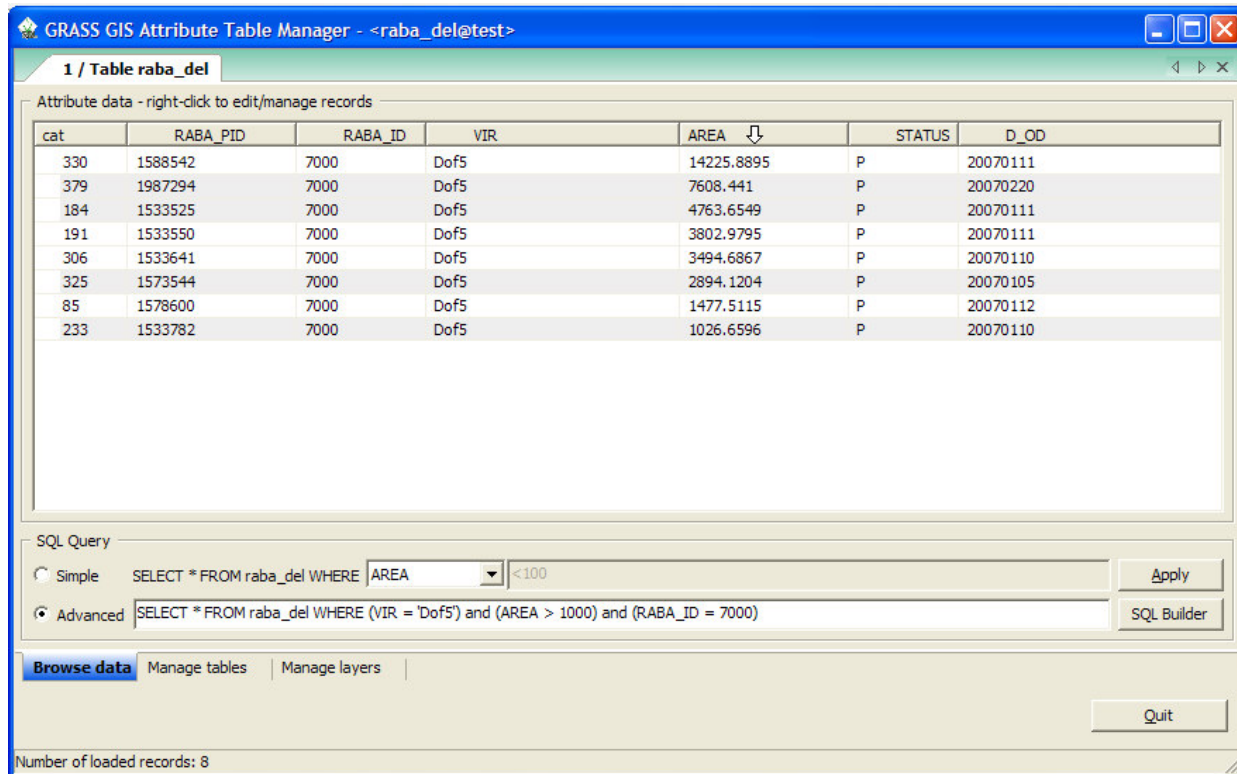
cat	RABA_PID	RA...	VIR	AREA	STATUS	D_OD
2	1740672	7000	Dof5	2.3701	P	20070330
72	14985	1500	Baseline_2	2.4059	P	20070330
168	1020394	3000	Dof5b	21.193	P	20070111
382	2567818	1300	Dof5	22.2725	P	20080212
264	1819952	3000	Dof5	32.9593	P	20070111
8	1817917	3000	Dof5	43.1662	P	20070110
174	1020392	3000	Dof5b	49.8534	P	20070220
135	1020557	3000	Dof5b	52.3833	P	20070110
50	1531990	3000	Dof5	53.2412	P	20070907
180	1020715	3000	Dof5b	56.5497	P	20070111
195	1803996	1230	Dof5	61.2909	P	20070110
15	1819012	3000	Dof5	66.1477	P	20070110
269	1300640	3000	Dof5	70.6726	P	20070110
178	1020840	3000	Dof5b	72.2155	P	20070111
52	15618	3000	Baseline_2	74.5276	P	20070110
172	1020004	3000	Dof5b	75.5545	P	20070110
245	1531750	3000	Dof5	77.4814	P	20070111
164	1020820	3000	Dof5b	78.9124	P	20070111
177	1020838	3000	Dof5b	79.6521	P	20070111
266	1821719	1500	Dof5	81.9045	P	20070111
268	1820401	3000	Dof5	83.4584	P	20070111

**Slika 22:** Seznam parcel manjših od 100m<sup>2</sup>; od najmanjše do največje

Če so poizvedbe kompleksnejše se poslužimo vrstice "napredno" in npr. vtipkamo:

```
SELECT * FROM raba_del WHERE (VIR = 'Dof5') and (AREA > 1000) and  
(RABA_ID = 7000)
```

kar nam da seznam vseh parcel vodnih površin (šifra rabe 7000), ki so bile zajete s pomočjo digitalnega ortofota (v merilu 1 : 5000) ter imajo površino večjo od 1000m<sup>2</sup>. Rezultat je prikazan na sliki 23.



cat	RABA_PID	RABA_ID	VIR	AREA	STATUS	D_OD
330	1588542	7000	Dof5	14225.8895	P	20070111
379	1987294	7000	Dof5	7608.441	P	20070220
184	1533525	7000	Dof5	4763.6549	P	20070111
191	1533550	7000	Dof5	3802.9795	P	20070111
306	1533641	7000	Dof5	3494.6867	P	20070110
325	1573544	7000	Dof5	2894.1204	P	20070105
85	1578600	7000	Dof5	1477.5115	P	20070112
233	1533782	7000	Dof5	1026.6596	P	20070110

Slika 23: Prikaz "zahtevnejše" poizvedbe SQL

Če pa želimo grafično prikazati npr. vse njive (šifra rabe 1100) na obravnavanem območju, to storimo z ukazom `v.extract`:

```
v.extract input=raba_del output=njive where=RABA_ID=1100
```

Podobno storimo še za vinograde (šifra rabe 1211) ter oljčnike (šifra rabe 1230) in vse skupaj prikažemo s 60% prosojnostjo na podlagi ortofota (slika 15). Njive so prikazane z rjavo barvo, vinogradi z zeleno, oljčniki pa z rumeno.



**Slika 24:** Prikaz oljčnikov, vinogradov in njiv na podlagi ortofota

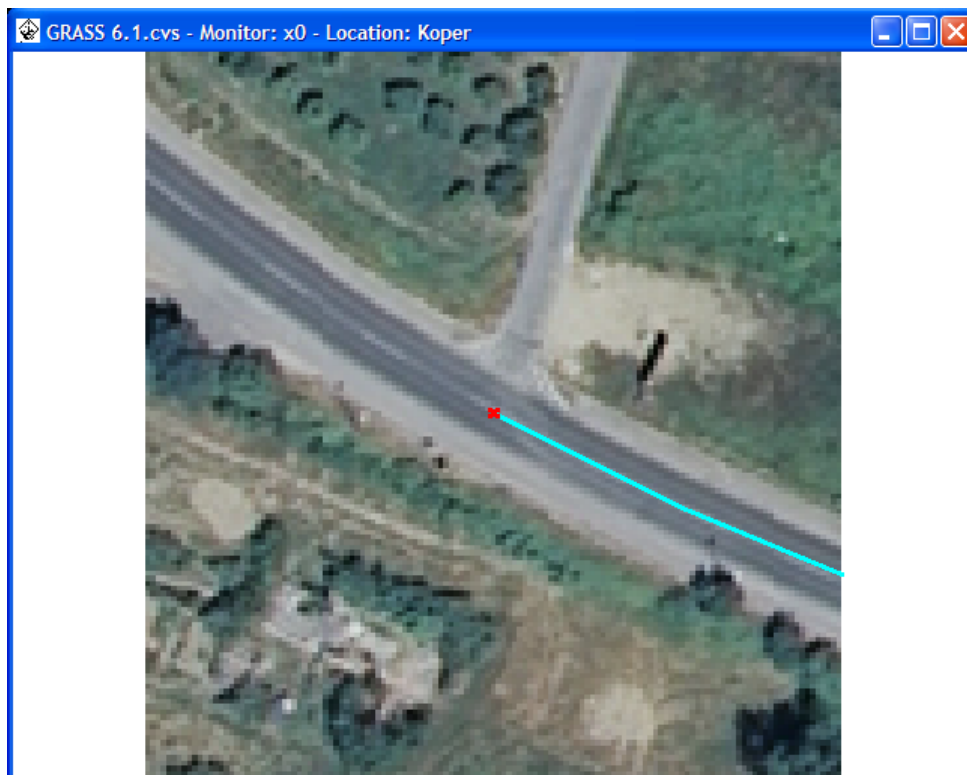
#### 6.4.2 Vektorizacija in urejanje vektorskih slojev

Vektorizacija je pretvarjanje rastrskih podatkov v vektorske. To lahko storimo 'samodejno' z ukazom *r.to.vect*, kar pa nam običajno ne da zelo natančnega izdelka. Če želimo izdelati npr. vektorski sloj cest, je to bolje storiti ročno z ukazom *v.digit*. Vektorizacija oz. digitalizacija (če gre za karto na papirju ali skenirano karto) poteka kar z uporabo miške (oz. digitalizatorja).

Ustvaril sem vektorski sloj *ceste*, ki sem mu 'pripisal' atributno tabelo s stolpcema *cat* ter *kategorija*. Vektorizirane ceste sem razdelil v 3 kategorije, in sicer:

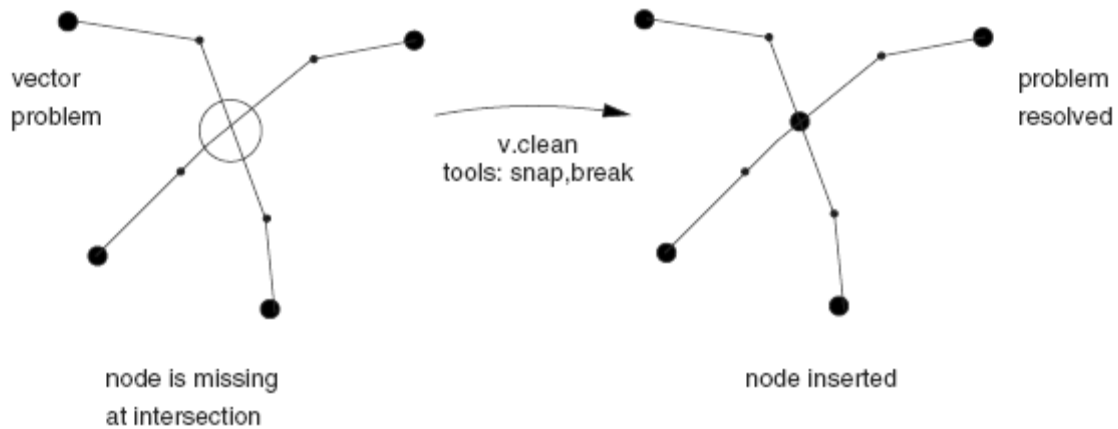
1. kategorija – regionalne, glavne medkrajevne ceste,
2. kategorija – lokalne, območne ceste,
3. kategorija – dovozi do hiš, makadamske ceste.

Vektorizacija linijskih objektov (npr. cest) poteka po sredini objekta, kar prikazuje slika 25.



**Slika 25:** Prikaz vektorizacije linijskih objektov – cest

Pri vektorskih slojih je še posebno pomembna urejena topologija, saj so v nasprotnem primeru prostorske analize neizvršljive. Pozorni moramo biti na ustrezna vozlišča linij, kar prikazuje tudi slika 26.



**Slika 26:** Primer urejanja topologije linijskih objektov (Neteler, 2008)

Topološke nepravilnosti odpravimo z ukazom *v.clean*, nato pa po potrebi 'zgradimo' končno topologijo z ukazom *v.build*. Končni vektorski sloj cest je prikazan na sliki 27.



**Slika 27:** Prikaz vektoriziranih cest na podlagi ortofota

Če pogledamo poročilo sloja *ceste* z dolžinami posameznih cest, dobimo:

```
v.report ceste option=length units=meters
```

```
Displaying column types/names for database connection of layer 1:
```

```
cat|kategorija|length
```

```
1|1|2233.91750658673  
2|2|999.053999322269  
3|2|885.815535341935  
4|3|703.769240361714  
5|2|8035.50107011093  
6|2|498.377518537969  
7|2|50.1165986558953  
8|3|66.7023845919926  
9|2|186.600950161409  
10|3|274.754328350145  
11|2|229.745382639357  
12|3|36.2659025614206  
13|3|30.545444784197  
14|3|321.702514134373  
15|2|154.016878740731  
16|3|440.495287735019  
17|2|27636.0252669177  
18|2|1011.08921460984  
19|3|234.21957650484  
20|2|1325.93778723697  
21|2|380.052519640169  
22|3|99.3855507749263  
23|3|30.2044749977967  
24|3|51.2344520448491  
25|3|38.2311209168499  
26|3|247.584245125468  
27|2|129.349877393488  
28|2|2630.26243509555  
29|2|75.9906944310422  
30|2|287.116949262305  
31|3|68.3011394665061  
32|3|560.9403985077  
33|3|38.7784407237985  
34|3|60.4571604918172  
35|3|80.7935457268734  
36|3|76.2298154755924  
37|3|447.954092644258
```

```
...
```

Metapodatke sloja uredimo z ukazom *v.support* ter jih nato prikažemo:

v.info ceste

```
+-----+
| Layer:          ceste                                     |
| Mapset:         vaja                                     |
| Location:       Koper                                    |
| Database:       C:/Documents and Settings/Matija/My Documents/GIS DataBas |
| Title:         Vektorski sloj cest za list 2027         |
| Map scale:     1:1                                       |
| Map format:    native                                    |
| Name of creator: Matija Mlakar                          |
| Organization:  FGG                                       |
| Source date:   Neznani (DOF5)                           |
|-----|
|   Type of Map:  vector (level: 2)                        |
|               |                                         |
|   Number of points: 0                               Number of areas: 0 |
|   Number of lines: 390                               Number of islands: 0 |
|   Number of boundaries: 0                           Number of faces: 0 |
|   Number of centroids: 0                             Number of kernels: 0 |
|               |                                         |
|   Map is 3D:      No                                   |
|   Number of dblinks: 1                                 |
|               |                                         |
|   Projection: x,y                                     |
|       N: 49000   S: 46000                             |
|       E: 403250  W: 401000                             |
|               |                                         |
|   Digitization threshold: 0                           |
|   Comments: /                                          |
+-----+
```

### 6.4.3 Območne in mrežne analize

Če nas npr. zanimajo območja, ki so od glavne ceste oddaljena za največ 100m, to storimo s t.i. "območnimi" operacijami. Za omenjeno nalogo, moramo iz vektorskega sloja *ceste* najprej 'izluščiti' glavno cesto, ki sem ji pri vektorizaciji določil kategorijo 1. Torej se ukaz glasi:

```
v.extract ceste out=glavna_cesta where=kategorija=1
```

Nato dobljenemu vektorskemu sloju določimo t.i. "bafer" 100-tih metrov (glej sliko 28):

```
v.buffer glavna_cesta out=cestabuffer100 distance=100
```



**Slika 28:** Prikaz območij, ki so od glavne ceste oddaljena za največ 100m



Če bi želeli prikazati območja samo 'v eno stran', bi namesto `v.buffer` uporabili ukaz `v.parallel` ali `v.segment`.

Med običajne mrežne analize spada iskanje najkrajše poti med dvema točkama. Izbrani točki je potrebno najprej vektorizirati, kar lahko storimo interaktivno (kot v prejšnjem primeru cest) ali pa kar z ukazom v izvršni vrstici, če so nam koordinate točk znane:

```
echo "403056.18|48293.27|1" | v.in.ascii out=hisa
```

Ko imamo začetno in končno točko v vektorski obliki, določimo razdaljo le-teh od obstoječega cestnega omrežja:

```
v.distance -p from=hisa to=ceste upload=dist column=dist  
from_cat|dist  
1|20.488887
```

Razdaljo uporabimo pri povezovanju točke s cestnim omrežjem, saj moramo določiti t.i. prag (angl. *threshold*), do katerega lahko algoritem išče povezavo z vozliščem v omrežju. Kot vidimo je razdalja točke *hisa* do sloja *ceste* malce več kot 20m, tako da je dovolj če nastavimo prag na 30m:

```
v.net ceste points=hise out=ceste_hise operation=connect  
thresh=30
```



**Slika 29:** Prikaz povezave točke s cestnim omrežjem

Po uspešni 'vključitvi' točk v cestno omrežje (slika 29), lahko določimo najkrajšo pot med njima:

```
v.net.path ceste_hise out=pot_do_cilja file=startend.txt
```

V datoteki *startend.txt* zapišemo ID poizvedbe ter ID začetne in končne točke (*ID cat1 cat2*):

```
1 1 2
```

Na sliki 30 je prikazana najkrajša pot med izbranimi točkama.



**Slika 30:** Prikaz najkrajše poti med izbranimi točkama (hišama)

#### **6.4.4 Interpolacijske metode s prostorskimi podatki**

'Prostorska interpolacija' pretvori vektorske točke ali izolinije, ki predstavljajo nek "zvezni" pojav, v rastrsko podobo z uporabo funkcije, ki poteka skozi ali blizu danih točk (Neteler, 2008). Če funkcija poteka skozi dane točke, govorimo o interpolaciji, drugače pa o približkih oz. t.i. aproksimaciji. Obstaja mnogo funkcij, s katerimi lahko interpoliramo vrednosti danih točk, jaz pa sem uporabil sledeče metode:

1. *Voronoijevi poligoni* – vektorske točke pretvorimo v vektorske poligone, tako da vsak poligon pridobi vrednost atributa točke, poligone pa lahko nato pretvorimo v raster. Ta varianta interpolacije običajno ni optimalna, saj kot rezultat ne dobimo 'gladke površine', pač pa nekakšen mozaik manjših ploskev. Uporablja se v primerih, kjer ne potrebujemo gladke ploskve, ampak stopničasto – diskretne vrednosti.

2. *Interpolacija z utežno inverzno razdaljo (IDW)* – način interpolacije, ki predpostavlja, da imajo vrednosti na bližnjih lokacijah večji vpliv na iskanje novih vrednosti, kot jih imajo vrednosti na bolj oddaljenih lokacijah (Bobnar, 2005). Torej se točkam, ki so bliže celici, katere vrednost se interpolira, dodeli večja utež kot bolj oddaljenim točkam. Interpolacija oz. metoda prilagajanja ploskve z inverzno razdaljo interpolira samo vrednosti med danimi točkami, kar pomeni, da ni mogoče dobiti vertikalnih ekstremov, če le-ti niso podani. Najboljše rezultate interpolacije dobimo torej takrat, ko je vzorec danih točk gost, ploskev relativno gladka, ekstremi ploskve pa so zajeti v danih točkah.

3. *Polinomska interpolacija oz. metoda zlepkov* (angl. regularized spline with tension) – izračuna vrednosti mrežnih točk s pomočjo funkcije, ki simulira tanko in fleksibilno ploskev, ki poteka skozi oz. blizu danih točk. Za doseganje želene natančnosti je potrebna pravilna določitev parametrov kot sta npr. gladkost in napetost. Za razliko od interpolacije z utežno inverzno razdaljo lahko metoda zlepkov predvidi ekstreme ploskve. V GRASS-u lahko polinomsko interpolacijo izvedemo iz vektorskih točk ali izolinij ter lahko izračunamo odstopanja od danih vrednosti.

Za interpolacijo sem iz rastrskega sloja dem izbral 1000 naključno razporejenih točk, ki sem jih shranil v vektorski sloj točk *dmvtocke*:

```
r.random input=dem n=1000 vector_output=dmvtocke
```

Nato sem ustvaril voronoijske diagrame in jih pretvoril v raster:

```
v.voronoi in=dmvtocke out=voronoidmv  
v.to.rast voronoidmv out=voronoirast
```

Sledil je izračun interpolacije z uteženimi inverznimi razdaljami (IDW):

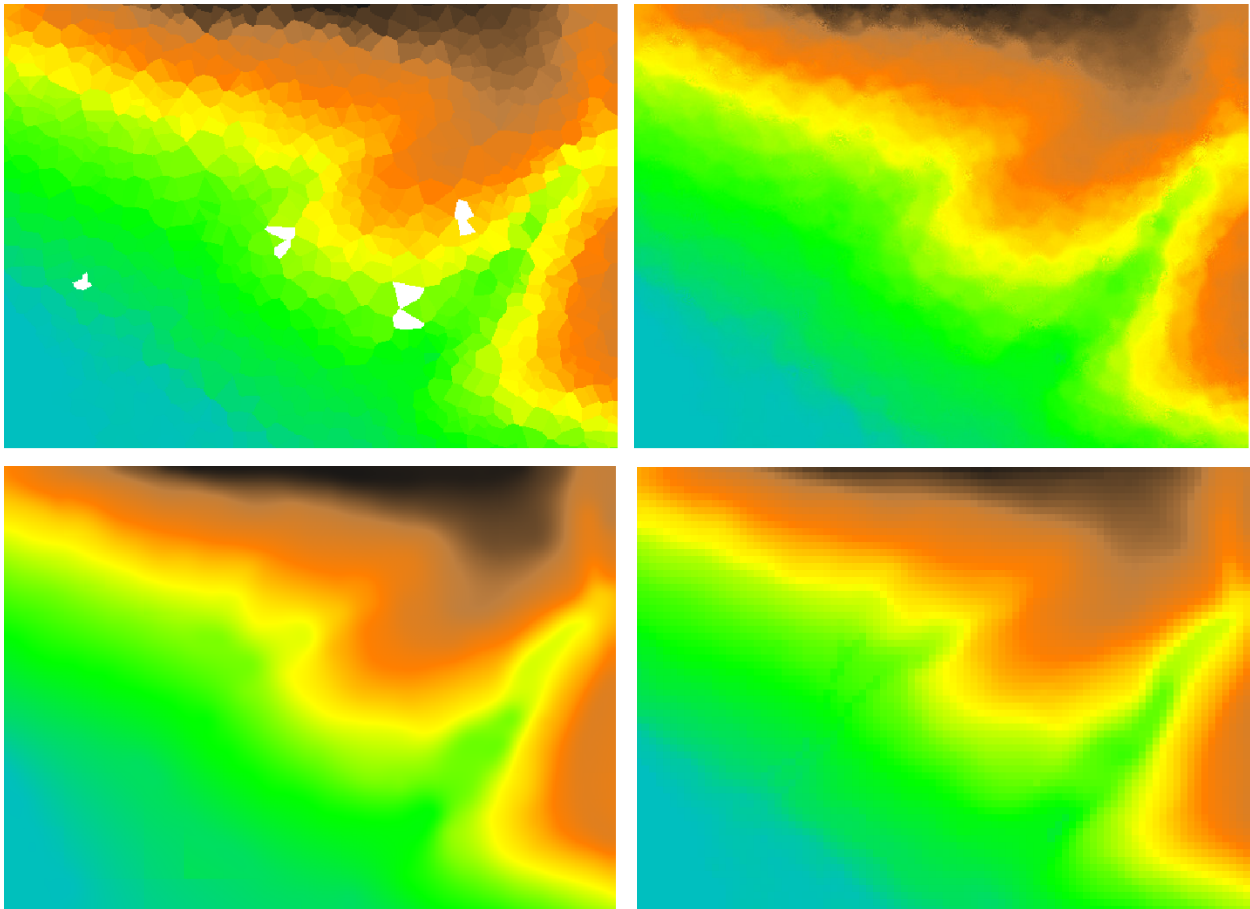
```
v.surf.idw in=dmvtocke out=idw column=value
```

In še polinomska interpolacija (RST):

```
v.surf.rst input=plastnice elev=rst
```

Če sedaj prikažemo vse tri končne izdelke interpolacije skupaj z originalnim DMV-jem, lahko vidimo, da je originalu najbolj podobna ploskev nastala iz metode zlepkov. Glavni razlog za to je v vhodnih podatkih, saj je bila interpolacija oz. metoda zlepkov izvedena iz plastnic, drugi dve pa iz manjšega vzorca vektorskih točk. Če bi želeli izboljšati interpolacijo z utežnimi inverznimi razdaljami, bi morali izbrati večje število točk vhodnih točk, kar pa seveda zahteva tudi daljše procesiranje podatkov. Voronoijevi diagrami v tem primeru niso primeren način interpolacije, saj želimo dobiti čim bolj gladko ploskev, kar pa ni mogoče pri tej metodi. Bele lise na prikazu Voronoijevih diagramov so rezultat neurejene topologije, ki je "nastala" pri izračunu diagramov iz točk.

Na sliki 31 lahko vidimo končne rezultate posameznih interpolacij v primerjavi z izvornim DMV-jem:



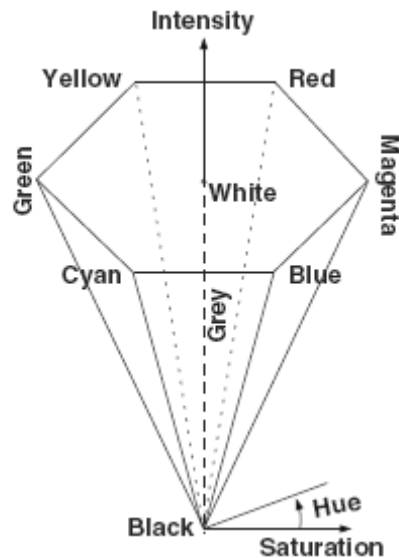
**Slika 31:** Prikaz rezultatov interpolacij ter izvornega DMV-ja: Voronoi (zgoraj levo), IDW (zgoraj desno), RST (spodaj levo), izvorni DMV (spodaj desno)

## 6.5 Vizualizacije podatkov

V GRASS-u najdemo tudi nekaj orodij za prikazovanje oz. vizualizacije prostorskih podatkov. Izbrane karte enostavno prikažemo v prikaznem oknu, v katerem lahko karti dodamo še npr. puščico severa, poljubno besedilo ali kot smo že lahko videli – legendo. Prikaze lahko za zunanjo uporabo po potrebi izvozimo v podobe formatov .png, .gif in .tiff.

Če želimo izboljšati percepcijo prikaza reliefa, iz DMV-ja ustvarimo senčeno karto, nato pa uporabimo t.i. barvno transformacijo, ki temelji na IHS barvnem modelu. Medtem ko RGB barvni model temelji na 'aditivnosti' rdeče, zelene in modre barve, gre pri IHS barvnem modelu

za določanje intenzitete oz. svetlosti (intensity), valovne dolžine oz. barve (hue) ter stopnjo nasičenosti (saturation). Na sliki 32 je prikazana zamisel barvnih modelov RGB, CMYK ter IHS.

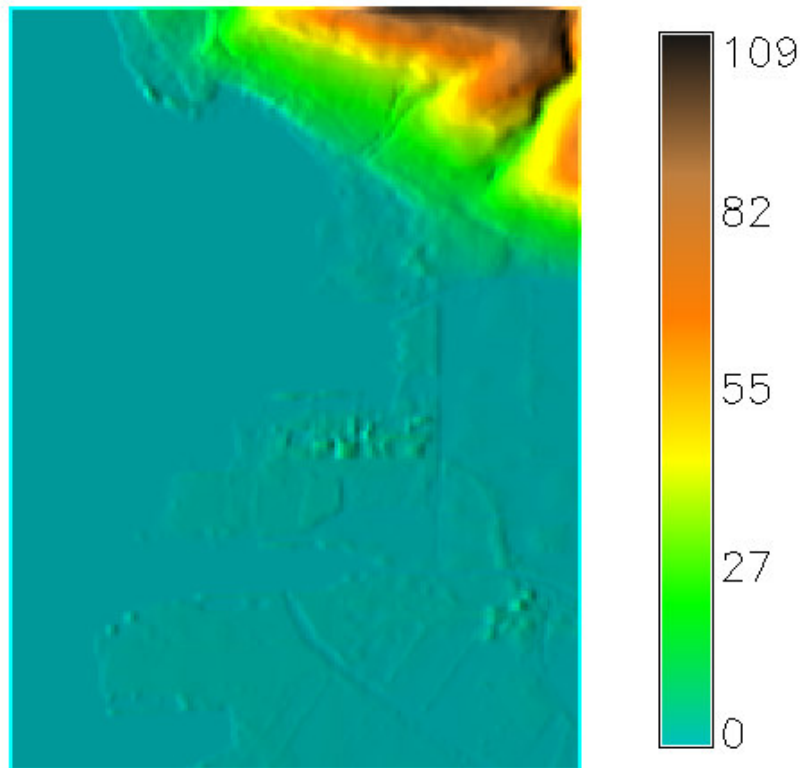


**Slika 32:** RGB in IHS barvna modela (Neteler, 2008)

Pri transformaciji kot "barvo" (parameter hue) uporabimo karto DMV-ja, kot intenziteto pa senčeno karto. Vse skupaj še malce posvetlimo in prikažemo z legendo in naslovom (slika 33):

```
d.his h_map=dem i_map=sencenje brighten=60
```

## DMV 12,5 s senčenjem



**Slika 33:** Vizualizacija s pomočjo IHS barvnega modela

Zmogljujejše interaktivno orodje znotraj GRASS GIS, ki služi predvsem za 3D-vizualizacije, se imenuje NVIZ.

Omogoča:

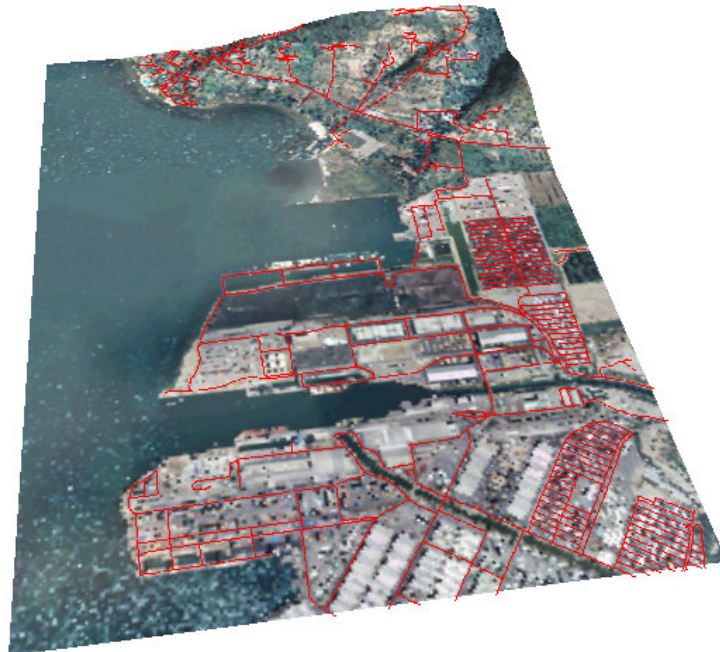
- vizualizacije v 3D prostoru z uporabo rastrov za relief, barvne podlage in prosojnost,
- interaktivno pozicioniranje, 'zoomiranje' ter 'povečevanje' višin za določen faktor
- interaktivno osvetlitev z določanjem barve, jakosti ter položaja vira osvetlitve,
- prikaz vektorskih slojev na reliefu ali drugi določeni višini, prikaz opisnih podatkov,
- ustvarjanje animacij s pomočjo glavnih pogledov (prelet) ali s pomočjo posebne skripte,
- interaktivno poizvedovanje rastrskih podatkov,
- interaktivni prerez več površin,
- prikaz 'volumskih' podatkov s pomočjo 'izopovršin'.



Sedaj ustvarimo 3D-prikaz območja, kjer uporabimo:

- DMV kot površino reliefa,
- ortofoto kot barvno podlago,
- vektorski sloj cest, dvignjenih na višino DMV-ja,
- povečanje višin s faktorjem 1.5,
- osvetlitev z belo svetlobo z jugozahodne smeri.

Končna 3D vizualizacija je prikazana na sliki 34:



**Slika 34:** 3D prikaz območja izdelan v orodju NVIZ

## 6.6 Programiranje v GRASS-u

GRASS je odprtokodna aplikacija, ki ima prosto dostopno celotno programsko kodo, ki jo lahko po želji porazdeljujemo in spreminjamo. Gre za približno 600 tisoč (in še raste) vrstic C, Makefiles, Tcl/Tk in Python kode ter 'shell script'.

Izkušeni uporabniki, ki želijo sodelovati pri razvoju aplikacije, morajo najprej dobiti dovoljenje od GRASS-PSC (GRASS Project Steering Committee), komunikacija med razvijalci pa v glavnem poteka prek elektronske pošte (t.i. GRASS developers mailing list).

Poznamo dva osnovna nivoja programiranja v GRASS-u. Povprečni uporabniki uporabljajo t.i. "skriptno programiranje", s katerim avtomatizirajo ponavljajoče se procese. Izkušeni uporabniki pa razvijajo programsko kodo na osnovi SWIG (Simplified Wrapper and Interface Generator) ali C-API (Application Programming Interface).

Če npr. pri delu z DMV-jem vedno uporabljamo enake ukaze, lahko sestavimo preprosto "skripto", ki nam v bodoče olajša oz. skrajša delo:

```
#!/bin/sh -x
# Ta program je prosta programska oprema izdana pod GNU GPL
# Prikaz DMVja in izračun senčenja, naklona in usmerjenosti

if [ -z "$GISBASE" ]
then
    echo ""
    echo "Zagon tega programa je mogoč le znotraj GRASS-a!"
    echo ""
    exit 1
fi

# Prvi parameter je ime karte:
MAP=$1

# Nastavitev zooma na karto:
g.region rast=$MAP

# Prikaz kart:
d.erase
r.colors $MAP color=elevation
d.rast $MAP

# Izračun senčenja
r.shaded.relief $MAP sencenje altitude=45 azimuth=315

# Izračun naklona in usmerjenosti
r.slope.aspect $MAP slope=naklon aspect=usmerjenost
```

Kot vidimo, z zgornjim programom prikažemo DMV v višinski barvni lestvici, nato pa izračunamo karte senčenja, naklona in usmerjenosti. Ime karte DMV-ja je shranjena kot spremenljivka MAP, ki je uporabljena pri nadaljnjih izračunih in prikazih. Prva vrstica (`#!/bin/sh -x`) je obvezna za UNIX shell skripte, kjer nam '-x' omogoča iskanje programerskih napak v datoteki, saj se program izvaja postopoma in vsak ukaz izpiše na zaslon. Komentarje pišemo za znakom #.

Program shranimo kot ANSI (angl. American National Standards Institute) datoteko (npr. *r.dmv*) v mapo GRASS\Scripts\ ali drugo prej določeno mapo. Datoteki moramo določiti še posebna 'dovoljenja' z ukazom:

```
chmod a+x r.dmv
```

Končni program nato uporabljamo enako kot druge vgrajene module, in sicer:

```
r.dmv ime_dmvja
```

V "skriptnem programiranju" lahko uporabljamo tudi UNIX ukaze kot so *awk*, *sed* in *cut* ter 'klicanje' funkcij GRASS-a, kar nam omogoča sestavo precej kompleksnih programov. Za razvijanje in spreminjanje programske kode po lastnih željah, pa je potrebno osvojiti še nekaj znanja programskega jezika C idr.

## 6.7 Druge možnosti uporabe GRASS-a

Čeprav sem se pri svojem delu osredotočil na osnovne rastrsko-vektorske operacije, lahko v programskem paketu GRASS najdemo še številne druge možnosti uporabe. Gre za kompleksnejše module, ki so namenjeni zahtevnejšim uporabnikom (npr. delo z volumskimi objekti, obdelava podatkov laserskega skeniranja idr.) pa tudi za široko paleto aplikacij na področju obdelave podob daljinskega zaznavanja, kjer GRASS omogoča:

- uvoz podob daljinskega zaznavanja (npr. satelitov LANDSAT),
- predhodno obdelavo podatkov (kalibracija podob, atmosferski popravki),
- izdelava temperaturne karte iz 'toplotnega kanala' satelita,
- radiometrične transformacije in izboljšave podob ('ratio', PCT),
- geometrične analize z matričnimi filtri,
- "fuzija" podob (izboljšava geometrične resolucije s pomočjo združitve multispektralnih kanalov s pankromatsko podobo) – IHS ter Broveyjeva transformacija,
- tematsko klasifikacijo satelitskih podatkov,
- 'časovne' analize,
- segmentacijo in prepoznavo vzorcev.

GRASS omogoča tudi povezavo z drugimi programskimi paketi kot so npr.:

- gstat, R (t.j. programska oprema za področje statistike),
- GPS Manager, GPStrans, GPSBabel,
- MapServer.

Ker je GRASS v fazi stalnega izboljševanja in razvijanja, se število možnosti ter zmogljivost in učinkovitost aplikacij nenehno spreminja. Pri izdelavi diplomske naloge sem uporabljal različici GRASS 6.1. ter 6.4., kmalu pa lahko pričakujemo različico 7.

## 7 ZAKLJUČEK

Današnjega življenja si brez računalniške tehnologije skorajda ne moremo predstavljati, zato je tudi razvijanje programske opreme v razcvetu. Čeprav je večina aplikacij še vedno v komercialno-tržni oz. "lastniški" obliki, se jim tudi odprtokodne 'približujejo' v vseh pomenih, saj postajajo vedno bolj učinkovite, celovite, dovršene in tudi uporabnikom prijazne. Če si najprej pobliže pogledamo prednosti slednjih, lahko rečemo, da so:

1. *Zanesljivost* – Načelo odprte kode je "izdajaj zgodaj in pogosto" in zato ima vsaka aplikacija številne različice, ki omogočajo hitro pridobivanje povratnih informacij kot tudi odkrivanje napak s strani številnih uporabnikov. Dostopnost izvorne kode omogoča, da aplikacije popravijo ali izboljšajo kar uporabniki sami in tako skrbijo za zanesljiv in hiter razvoj.

2. *Stabilnost* – Odprtokodna programska oprema je na voljo brezplačno in nikogar ne sili h kupovanju novih licenc oz. prehajanju na novejšo različico aplikacije. Prilagojena je odprtim standardom, tako da tudi 'kompatibilnost' ne predstavlja hujših ovir.

3. *Transparentnost* – Popolnoma dostopna izvorna koda omogoča uporabnikom jasen vpogled v strukturo in delovanje aplikacij, kar vliva zaupanje med uporabniki in razvijalci, saj so podatki o varnosti, prilagodljivosti ipd. jasno podani.

4. *Ekonomičnost* – Kot rečeno je izvorna koda na voljo brezplačno, aplikacije lahko brezplačno kopiramo, v večini primerov lahko aplikacije tudi brezplačno nadgradimo. Posredno imamo manj stroškov tudi zaradi upravljanja in varovanja, saj gre za varnejšo (v smislu računalniških virusov in vdorov neželenih uporabnikov) in stabilnejšo programsko opremo v primerjavo z lastniško.

5. *Svoboda, prilagodljivost* – Izvorno kodo lahko po svojih željah spreminjamo in tako aplikacijo popolnoma prilagodimo lastnim potrebam. Odprtokodne aplikacije običajno uporabljajo odprte standarde in se izogibajo kakršnikoli odvisnosti od drugih proizvajalcev.

6. *Podpora* – Običajno programerji razvijajo aplikacije za skupne interese ali potrebe in so pripravljeni nuditi pomoč uporabnikom, saj drugače njihov produkt ne bi koristil svojemu namenu. Uporabniki dobijo odgovore na svoja vprašanja na različne načine - npr. prek elektronske pošte, spletnih klepetalnic itd.

Seveda tudi pri odprtokodni programski opremi najdemo kakšne pomanjklivosti oz. slabosti, ki pa so:

1. *Uporabniški vmesniki* – V večini primerov so grafični uporabniški vmesniki odprtokodnih aplikacij manj asociativni in uporabnikom neprijazni, saj se razvijalci bolj osredotočajo na funkcionalnost kot na izgled. Lahko rečemo da so aplikacije običajno razvite 'od inženirjev za inženirje', ki dobro razumejo strukturo aplikacij, preprostim uporabnikom pa lahko neprijazen uporabniški vmesnik predstavlja veliko oviro.

2. *Zahtevnost uporabe* – Kot rečeno je odprtokodna programska oprema bolj programersko naravnana in za uporabo lahko zahteva precej znanja. Nekaterim uporabnikom lahko težave povzroča že sama namestitev aplikacij, saj po večini niso tipa "vstavi in poženi" (angl. plug and play). Prav tako je pomoč uporabnikom včasih težko razumljiva, saj gre običajno za podrobno tehnično dokumentacijo, ki je predvsem namenjena razvijalcem in naprednim uporabnikom.

3. *Lastništvo* – Čeprav se lepo sliši, da je nekaj prosto dostopno ali pa npr. družbena lastnina, je slaba stran tega pomanjkanje odgovornosti. Če se nam nekaj zalomi pri lastniški programski opremi, lahko podjetje tožimo oz. pričakujemo pojasnila odgovornih, pri odprtokodnih aplikacijah pa pravzaprav temeljimo na zaupanju.

4. *Združljivost oz. 'kompatibilnost'* – Nekatere odprtokodne aplikacije ne funkcionirajo najbolje z drugimi aplikacijami, tipičen primer so aplikacije, ki so razvite v UNIX-u in ne delujejo na razširjenih Microsoftovih "oknih".

*5. Težak začetek in razvoj* – Odprtokodni projekt zaživi le v primeru, da privabi v skupnost dovolj izurjenih razvijalcev, ki so usmerjeni v rešitev nekega problema - imajo torej skupen cilj. Tako je težko preiti iz beta faze razvoja v stabilno stanje, saj lahko zanimanje za projekt hitro mine.

Večina opisanih lastnosti velja tudi za aplikacijo GRASS GIS. Na voljo je brezplačno na spletu. Primarno je razvita za UNIX operacijski sistem in do izida aktualne različice ni delovala v okolju Windows brez dodatne posebne aplikacije. Instalacija je relativno zahtevna, prav tako kot tudi sama uporaba aplikacije. Aktualna različica, ki ima grafični uporabniški vmesnik razvit na osnovi Python/wxWidgets, je veliko boljša in uporabniku prijaznejša od prejšnjih različic, ki so imele grafični vmesnik razvit na osnovi Tcl/Tk. Če pri delu potrebujemo pomoč, nam jo razvijalci nudijo neposredno prek elektronske pošte ali prek spletne klepetalnice IRC.

Točen podatek o tem, koliko uporabnikov danes uporablja GRASS, ni znan. Ocena M. Netelerja iz leta 2004 je, da je takrat GRASS uporabljalo med 10 in 30 tisoč uporabnikov. Verjetno je danes ta številka še nekoliko večja, a vseeno lahko z gotovostjo trdimo, da je število uporabnikov ESRI-jevega "komercialnega konkurenta" ArcGIS veliko večje. Na spletni strani podjetja ESRI najdemo podatek, da njihovo programsko opremo (katere glavni produkt je ArcGIS) vsak dan uporablja več kot milijon uporabnikov.

Če želimo primerjati GRASS in ArcGIS še s stališča funkcionalnosti, možnosti uporabe ipd., lahko navedemo ugotovitve T. Buchanana, ki je v svojem doktoratu (Comparison of Geographic Information System software (ARCGIS 9.0 and GRASS 6.0), Implementation and case study) primerjal programska paketa GRASS 6.0 ter ArcGIS 9.0. Njegove ugotovitve, ki se skladajo tudi z mojimi ugotovitvami, so navedene v preglednici 5:

**Preglednica 5:** Primerjava programskih paketov ArcGIS 9.0 ter GRASS 6.0 (Buchanan, 2005)

<b>Kategorija</b>	<b>ArcGIS</b>	<b>GRASS</b>
stroški	Lahko so zelo visoki, odvisno od licenc ter razširitev, ki jih potrebujemo.	Stroškov praktično ni, saj je odprtokodna programska oprema na voljo brezplačno.
namestitve	Zelo enostavna.	Malce zahtevnejša kot pri "običajni" programski opremi za Microsoftova "okna".
učenje in tehnična podpora	Na voljo so številni učbeniki ter dobra tehnična podpora.	Zelo malo pripomočkov za učenje ter skromna tehnična podpora.
podatkovna baza	Enostavna za uporabo ter razvoj, s številnimi možnostmi.	Zahtevnejša uporaba, uporabnik nima toliko svobode kot pri ArcGIS-u.
uvoz in izvoz podatkov	Možno je delati z večino formatov datotek, ki jih najdemo na spletu.	Uvozimo ter izvozimo lahko zelo veliko različnih formatov datotek.
prikaz	Odličen prikaz podatkov, projiciranje le-teh v realnem času.	Omogoča prikaz podatkov, vendar ne omogoča projiciranja le-teh v realnem času (še le v različici 6.4).
vizualizacije	Omogoča izdelavo številnih profesionalnih kart.	Nima veliko možnosti za kartografsko oblikovanje.
splošna funkcionalnost	Omogoča široko paleto orodij, vendar ne omogoča obdelave podob ter statističnih analiz brez dodatnih razširitev (npr. ENVI).	Omogoča številna analitična orodja, vključno z obdelavo podob ter statističnimi orodji.
splošna zahtevnost uporabe	Enostaven in uporabniku prijazen način uporabe.	Zahtevnejši način uporabe in uporabniku ne najbolj prijazen.



Pri mojem delu z GRASS-om mi je sprva povzročal težave uporabniški vmesnik ter programerska naravnost aplikacije – izvrševanje ukazov prek ukazne vrstice, določevanje parametrov in spremenljivk ... Ko sem osvojil osnove programa in se navadil na način dela, je delo steklo. Lahko rečem, da je GRASS GIS celovito orodje GIS, saj nudi široko paleto modulov, ki zadovoljijo tudi zahtevne uporabnike. Ker gre za odprtokodno aplikacijo, je struktura modulov prosto dostopna, kar je za inženirje še kako pomembno. Za doseganje optimalnih rezultatov je poznavanje algoritmov in načina obdelave podatkov ključnega pomena.

V diplomski nalogi sem predstavil zamisel odprte kode ter povezavo le-te z geografskimi informacijskimi sistemi. Opisal sem nekatere odprtokodne aplikacije GIS, ki so večinoma plod organizacije OSGeo, seveda pa najdemo na spletu še številne druge.

Osredotočil sem se na programski paket GRASS GIS, s pomočjo katerega sem prikazal uporabo osnovnih rastrskih in vektorskih operacij. Vhodni podatki bi lahko bili izbrani boljše, saj gre za relativno ravninsko območje, na katerem operacije z DMV-jem niso dale najbolj zanimivih rezultatov. Prav tako del območja prekriva morje, ki je za prostorske analize skorajda neuporabno. Na območju morja je prišlo tudi do topoloških nepravilnosti, in sicer pri analizah, ki so vsebovale sloj rabe tal.

Vseeno pa menim, da diplomska naloga daje dober vpogled v odprtokodne aplikacije GIS, ki so trenutno na voljo in lepo predstavi zmogljivosti ter način uporabe programskega paketa GRASS GIS. Predstavljene so tudi prednosti in slabosti odprtokodnih aplikacij, ki so bile ugotovljene med praktičnim delom. Nad preizkušeno programsko opremo sem navdušen in jo priporočam tudi drugim nadobudnim uporabnikom tehnologije GIS.

## VIRI

Benčina, J. 2003. V pričakovanju hitre rasti uveljavljanja odprte kode, INDO 2003 v: Posvetovanje informatikov v državni upravi, Zbornik referatov. Ljubljana, Vlada Republike Slovenije, Center za informatiko: str. 456-458.

Bobnar, S., Drobne, S., Šumrada, R., 2005. Vaje iz prostorskih analiz v ArcGIS-u. Ljubljana: str. 105-112.

Buchanan, T., 2005, Comparison of Geographic Information System software (ARCGIS 9.0 and GRASS 6.0), Implementation and case study. Hardin – Simmons University: str. 34-40, 73.

Drobne, S., Podobnikar, T. 1999. Osnovni pojmi v geografskih informacijskih sistemih.  
[http://www.fgg.uni-lj.si/sdrobne/GIS\\_Pojm/Index.htm](http://www.fgg.uni-lj.si/sdrobne/GIS_Pojm/Index.htm) (2.12.2008).

FreeGIS Project. 2009.  
<http://freegis.org/> (2.1.2010)

Free Software Foundation. 2009. GNU General Public License.  
<http://www.gnu.org/licenses/gpl.html> (7.3.2009).

Geografski informacijski sistemi.  
<http://www.geoservis.si/uporabno/gis/gis.htm> (10.2.2009).

Geographic information system, Wikipedia, The free encyclopedia.  
[http://en.wikipedia.org/wiki/Geographic\\_information\\_system](http://en.wikipedia.org/wiki/Geographic_information_system) (13.11.2008).

Geografski informacijski sistem. 2009. Wikipedija, Prosta enciklopedija.  
[http://sl.wikipedia.org/wiki/Geografski\\_informacijski\\_sistem](http://sl.wikipedia.org/wiki/Geografski_informacijski_sistem) (20.11.2008).

Geographic Resources Analysis Support System. 2009.

<http://grass.itc.it/> (14.4.2009).

Ministrstvo za kmetijstvo, gozdarstvo in prehrano, Grafični podatki RABA.

<http://rkg.gov.si/GERK/> (10.4.2009).

Neteler, M., Mitasova, H. 2008. OPEN SOURCE GIS, A GRASS GIS Approach, 3rd edition.  
Boston, Springer: 417 str.

Neteler, M. 2004. [GRASSLIST:4119] Some GRASS user statistics,

<http://lists.osgeo.org/pipermail/grass-user/2004-August/026305.html> (3.1.2010)

Odperta koda.

<http://www.odprtakoda.org> (10.1.2009).

Odperta koda. 2009. Wikipedija, Prosta enciklopedija.

[http://sl.wikipedia.org/wiki/Odperta\\_koda](http://sl.wikipedia.org/wiki/Odperta_koda) (25.1.2009).

Open Source Initiative.

<http://www.opensource.org/> (9.3.2009).

Primožič, P. 2005. Uporaba odprte kode kot osnova za razvoj programske opreme, Diplomaska naloga. Ljubljana, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko: str. 7-34.

Stallman, R. 2007. Why Open Source misses the point of Free Software.

<http://www.gnu.org/philosophy/open-source-misses-the-point.html> (26.2.2009).

The Open Source Geospatial Foundation, OSGeo Projects.

<http://www.osgeo.org/> (15.3.2009).

Vasudevan, A. 2001. CVS-RCS-HOW-TO za Linux (sistem za kontrolo izvirne kode).  
<http://www.lugos.si/delo/slo/HOWTO-sl/ CVS-RCS-HOWTO-sl-1.html> (5.4.2009).

## PRILOGA

### Priloga A:

### Šifrant in opis vrst dejanske rabe kmetijskih in gozdnih zemljišč

#### ŠIFRANT IN OPIS VRST DEJANSKE RABE KMETIJSKIH IN GOZDNIH ZEMLJIŠČ

SKUPINA DEJANSKE RABE	ŠIFRA	VRSTA DEJANSKE RABE (najmanjša površina zajema)	OPIS DEJANSKE RABE
NJIVE IN VRTOVI	1100	Njiva (1000 m <sup>2</sup> )	Površina, ki jo orjemo ali drugače obdelujemo in obračališča, namenjena obdelavi te površine (širine do 2 m). Na tej površini pridelujemo enoletne in nekatere večletne kmetijske rastline (žita, krompir, krmne rastline, oljnice, predivnice, sladkorna pesa, zelenjadnice, vrtnine, okrasne rastline, zelišča, jagode itd.). Sem sodi tudi zemljišče v prahi in ukorenitšče hmeljnih sadik. V ta razred uvrščamo tudi zemljišče, ki je začasno zasejano s travo ali drugimi krmnimi rastlinami (za obdobje manj kot 5 let) in se uporablja za košnjo ali pašo večkrat na leto. Če je površina porasla s travno rušo in ni preorana v obdobju pet ali več let, jo uvrstimo v trajni travnik.
	1160	Hmeljišče (500 m <sup>2</sup> )	Površina, na kateri so žičnica ter obračališča in poti, potrebne za obdelavo hmeljišča. Vključuje površino hmeljišča v obdelavi oziroma v premeni.
	1180	Trajne rastline na njivskih površinah (1000 m <sup>2</sup> )	Drevesnica, trsnica, zarišče podlag, nasad matičnih rastlin, nasad okrasnih trajnih rastlin za vzgojo rezanega cvetja, trajna zelišča, trajne zelenjadnice.
	1190	Rastlinjak (25 m <sup>2</sup> )	Steklenjaki in plastenjaki z močnejšo konstrukcijo in daljšo življenjsko dobo, v katerih se vzgajajo okrasne rastline, rezano cvetje, zelenjava, zelišča, matične rastline, podlage, sadike, jagode itd.
TRAJNI NASADI	1211	Vinograd (500 m <sup>2</sup> )	Površina, zasajena z vinsko trto ( <i>Vitis vinifera</i> ), vključno z obračališči in potmi v vinogradu ter brežinami pri vinogradu na terasah.
	1212	Matičnjak (500 m <sup>2</sup> )	Površina, zasajena z matičnimi rastlinami, namenjena za pridelavo ključev podlag vinske trte, vključno z obračališči in potmi.
	1221	Intenzivni sadovnjak (1000 m <sup>2</sup> )	Površina, zasajena s sadnimi vrstami, pri obdelavi katere se uporabljajo sodobne intenzivne tehnologije. Intenzivni sadovnjak zajema površino nasada skupaj z obračališči in potmi ter brežinami, če je nasad zasajen v terasah. Nasade jagod uvrščamo v vrsto dejanske rabe njiva.
	1222	Ekstenzivni oziroma travniški sadovnjak (1000 m <sup>2</sup> )	Sadovnjak, ki ni primeren za intenzivno pridelavo. To je običajno nasad visokodebelnih sadnih dreves, vzgojenih na bujni podlagi ali iz semena, z gostoto več kot 50 dreves na hektar. V ekstenzivnem oziroma travniškem sadovnjaku lahko raste ena ali več različnih sadnih vrst.
	1230	Oljčnik (500 m <sup>2</sup> )	Površina, zasajena z oljkami, ki so med seboj oddaljene največ 20 metrov, povečana za širino oziroma dolžino največ 10-ih metrov od debel oljk, zasajenih na zunanjih robovih oljčnika, ki je namenjena za obračališča in pomožne poti.
	1240	Ostali trajni nasadi (500 m <sup>2</sup> )	Površina zasajena z eno ali več različnimi vrstami trajnih rastlin.
TRAVNIŠKE POVRŠINE	1300	Trajni travnik (1000 m <sup>2</sup> )	Površina porasla s travo, deteljami in drugimi krmnimi zelmi, ki se jo redno kosi oziroma pase. Takšna površina ni v kolobarju in se ne orje. Kot trajni travnik se šteje tudi površina, porasla s posameznimi drevesi, kjer gostota dreves ne presega 50 dreves/hektar.
	1321	Barjanski travnik (1000 m <sup>2</sup> )	S travinjem, šašem in močvirsko preslico poraslo zemljišče na organskih ali mineralno-organskih tleh, na katerem nivo talne vode med letom pogosto doseže površino tal.
	1800	Kmetijsko zemljišče, poraslo z gozdnim drevjem (1000 m <sup>2</sup> )	Površina, porasla s travinjem, na kateri rastejo posamična gozdna drevesa oziroma grmi in se redno, vsaj enkrat letno popase oziroma pokosi. Pokrovnost travinja je vsaj 80 %, pokrovnost drevesnih krošenj oziroma grmov pa je manjša od 75 %.

Mlakar, M. 2009. Uporaba odprte kode v okoljih GIS.  
 Dipl. nal. - UNI. Ljubljana, UL, FGG, Odd. za geodezijo, Geodetska smer.

DRUGE KMETIJSKE POVRŠINE	1410	Kmetijsko zemljišče v zaraščanju (1000 m <sup>2</sup> )	Zemljišče, ki se zarašča zaradi opustitve kmetovanja ali preskromne kmetijske rabe. Pokravnost dreves je 20–75 %. Če se takšno zemljišče 20 ali več let ne uporablja za kmetijske namene, če pokravnost dreves preseže 75 % in če je premer debel več kot 10 cm, preide v gozd.
	1420	Plantaza gozdnega drevja (1000 m <sup>2</sup> )	Plantaza gozdnega drevja je nasad gozdnega drevja, ki je namenjen izključno pridelavi lesa, okrasnih dreves ali plodov oziroma drugih delov drevja in pri katerih so razdalje med drevjem že ob zasaditvi takšne kot ob predvidenem končnem razvojnem stanju sestoja.
	1500	Drevesa in grmičevje (1000 m <sup>2</sup> )	Površina, porasla z drevesi in grmičevjem. Sem uvrščamo tudi obvodno zarast, če so obrečni pasovi porasli z drevjem oziroma grmovjem, ter mejice iz gozdnih dreves oziroma grmičevja
	1600	Neobdelano kmetijsko zemljišče (1000 m <sup>2</sup> )	Površina, ki je npr. rigolana in pripravljena za zasaditev novih trajnih nasadov. Kmetijsko zemljišče, ki se začasno ne uporablja zaradi gradnje infrastrukture ali je neobdelano zaradi socialnih ali drugih razlogov. Kmetijsko zemljišče, na katerem je ograda za konje, prašiče ali druge živali in ki ni poraslo s travinjem.
GOZD	2000	Gozd (2500m <sup>2</sup> )	Zemljišče, ki je v skladu s predpisi o gozdovih opredeljeno kot gozd.
OSTALA NEKMETIJSKA ZEMLJIŠČA	3000	Pozidano in sorodno zemljišče (25 m <sup>2</sup> )	Površina, na kateri so zgradbe, ceste, ki vodijo do naselij ali hiš, parkirni prostori, rudniki, kamnolomi in druga infrastruktura, ki služi za opravljanje človeških dejavnosti.
	4100	Barje (5000 m <sup>2</sup> )	Nizko ali visoko barje, ki se ne uporablja za kmetijsko rabo. Vegetacija je navadno višja kot na barjanskih travnikih in se ne kosi.
	4210	Trstičje (5000 m <sup>2</sup> )	Močvirno zemljišče, na katerem raste trstika. Na tem zemljišču ni kmetijske pridelave.
	4220	Ostalo zamočvirjeno zemljišče (5000 m <sup>2</sup> )	Nizko ležeča zemljišča, pogosto poplavljena in ves čas bolj ali manj namočena, ki se ne uporabljajo v kmetijske namene.
	5000	Suho, odprto zemljišče s posebnim rastlinskim pokrovom (5000 m <sup>2</sup> )	Ne-gozdno zemljišče, pokrito z nizko vegetacijo (pod 2 m), ki je nerodovitno ali nedostopno. Pokritost z vegetacijo ni večja od 75 %.
	6000	Odprto zemljišče brez ali z nepomembnim rastlinskim pokrovom (5000 m <sup>2</sup> )	Nezazidano zemljišče z malo ali brez vegetacije, zaradi česar takšne površine ne moremo vključiti v kakšen drug razred. Sem sodijo vsa zemljišča, prekrita z golimi skalami, peščene plaže in sipine, prodnate površine ob oziroma v vodotokih, melišča in ostale odprte površine.
	7000	Voda (25 m <sup>2</sup> )	Površina, pokrita s površinskimi vodami, kot so jezera, reke, potoki in jarki, v katerih se nahaja voda.