

Integrirani sistemi premera

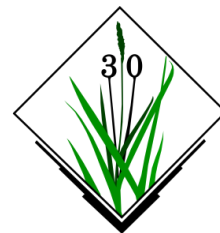
GRASS GIS – izrada DMT

Antonić Nenad o347
nenadsbc@gmail.com

Mentor: Prof. Bulatović Vladimir
11 / 5 / 2014

Rezime

Rad pokušava da ilustrije osnovne mogućnosti GRASS GIS softverskog paketa primarno vezane za kreiranje digitalnog modela terena na bazi rasterske i vektorske strukture ulaznih podataka. Sekundarno, rad ima za cilj da pojasni sve bitne termine vezane za pomenuti softverski paket, način njegovog funkcionisanja, strukturu njegovih podataka, značaj softvera otvorenog koda i osnovnu sintaksu. Rad prikazuje niz dostupnih mogućnosti za kreiranje DMT, objašnjava njihovu sintaksu, definiše osnovne parametre i primerima ilustrije rezultate bazirane na konkretnim podacima dostupnim na zvaničnom sajtu softvera.



Ključne reči: *GRASS, GIS, Open, Source, DTM, Free*

1. Uvod

Softverski paket koji je prikazan u radu je usko povezan i praktično nerazdvojiv od Geografskih informacionih sistema pa je iz tog razloga navedeno i nekoliko najčešće korišćenih definicija GIS-a. Kasnije kroz rad biće pojašnjen i termin Digitalnog modela terena, njegov značaj i struktura.

Geographical Resources Analysis Support System (GRASS) je jedan od najvećih besplatnih GIS (*Geographic Information System*) projekata realizovanih pod GNU GPL (*General Public License*). On kombinuje moćan „engine“ za rastere, vektore i prostorni procesing u jedan integrirani softverski paket koji uključuje alate za prostornu analizu, modelovanje, obradu slike/snimka i vizualizaciju [1].

GRASS GIS, ili samo GRASS (*Geographical Resources Analysis Support System*), je besplatan GIS (*Geographic Information System*) softverski paket otvorenog izvornog koda (*Open Source*) koji se koristi za upravljanje i analizu geoprostornih podataka, obradu slike, proizvodnju karata, prostorno modelovanje i vizualizaciju. GRASS GIS se danas koristi u akademske i komercijalne svrhe sirom sveta, koriste ga mnoge državne agencije kao i kompanije za zaštitu životne sredine [2].

Postoje brojne definicije GIS-a. Jednu od najčešće korišćenih definicija dala je firma ESRI: *Geographic Information System* (GIS) integriše hardver, softver i podatke za snimanje, upravljanje, analiziranje i prikaz svih oblika geografskih informacija od važnosti. GIS omogućava da vidimo, tumačimo, razumemo i vizualizujemo podatke sa prostornim geografskim karakteristikama, da utvrdimo obrasce, šeme, zavisnost i trendove posmatrajući mape, globuse, grafikone [3].

U najopštijem smislu, Geografski informacioni sistem predstavlja organizovan skup računarskog hardvera, softvera, podataka, osoblja i mreža radi efikasnog prikupljanja, skladištenja, ažuriranja, rukovanja, analize, modelovanja, prenosa, i prikaza svih oblika prostornih informacija.

2. *Open Source* koncept, GRASS kao *Open Source*

Po rečima Neteler i Mitasova, autora knjige *Open Source GIS: A GRASS GIS Approach*: Ideja o *Open Source* konceptu, konceptu softvera otvorenog koda, stara je verovatno koliko i razvoj samog softvera. Naime, kod stvaran na univerzitetima i državnim laboratorijama, često je prvo bio dostupan u javnom domenu. Richard M. Stallman je prvi definisao ovaj koncept u formi četiri stepena slobode:

0. *Freedom*: Sloboda korišćena softvera u bilo koju svrhu.
1. *Freedom*: Sloboda učenja o tome kako program funkcioniše, mogućnost adaptacije programa svojim potrebama.
2. *Freedom*: Sloboda distribucije softvera.
3. *Freedom*: Sloboda poboljšanja programa, publikovanje poboljšane verzije u javnost u cilju opšte koristi.

Softver baziran na ovim principima se može nazvati *Free Software*. Richard M. Stallman 1984 godine počinje da radi na GNU projektu, a 1985 godine osniva "*Free Software Foundation*" kako bi podržao gore navedeni koncept. Licenca GNU projekta, GNU *General Public License* (GPL), ne samo da garantuje pomenuta četiri stepena slobode nego ih štiti. Upravo ovim projektom GPL je postala najčešće korišćena licenca za *Free Software* danas [4].

Osnovna ideja je bazirana na pretpostavci da će softver evoluirati ako se dopusti da programeri (korisnici) mogu da čitaju, razmenjuju i modifikuju njegov kod. To će ga poboljšati, greške će biti otklonjene a mogućnosti softvera će biti unapređene. S obzirom na ekspertizu korisnika (programera) promene i poboljšanja samog softvera mogle bi se odvijati znatno brže nego inače u poređenju sa konvencionalnim razvojem. Kompletan, potpun pristup softverskom kodu je jako važan za GIS jer određeni podalgoritmi i funkcije mogu biti jako komplikovane i mogu u velikoj meri uticati na krajnje rezultate, prostornu analizu ili modelovanje. Za potpuno razumevanje funkcionisanja jednog sistema ili nekog njegovog dela, važno je razumeti funkcionisanje svih podsistema i delova koji ga čine. Često će se desiti da običan korisnik nije u situaciji da ispravi kod, već da samo uoči grešku u funkcionisanju. Ponekad je i to značajno jer će ukazivanjem na grešku podstaći nekog stručnijeg da izvrši kompletnu analizu, testiranje i popravku koda. Jako je bitna i šarolikost u struci i ekspertizi ljudi koji koriste jedan *Open Source* softver, upravo to će značiti brže rešavanje problema, unapređenje i stabilnost jednog velikog projekta. Takođe, *Open Source* licenca i slobodan pristup softveru preko interneta omogućava saradnju na napuštenim projektima, njihov dalji razvoj i "održavanje u životu" ako za tim ima potrebe [4].

Kao što je gore već spomenuto, 1999 godine GRASS je usvojio GNU GPL. Sa ovim, GRASS je prihvatio psihologiju *Open Source* softvera, i upravo to je podstaklo njegovu široku primenu i prihvatanje od strane korisnika. Ovakav pristup razvijanju softvera privlači programere da i sami učestvuju u njegovom daljem razvoju.

3. Razvojni put projekta GRASS

GRASS je najveći besplatan softver iz grupe GIS projekata i po obimu svog izvornog koda nalazi se na listi među prvih deset svih *Open Source* projekata. Pored toga, 80-ih godina prošlog veka, GRASS je među prvim *Open Source* GIS softverima dostigao proizvodni status i prvi je podržao oba modela podataka, i rasterski i vektorski [4].

GRASS je razvijan u periodu između 1982 – 1995 godine od strane američke vojske, tačnije *U.S. Army Corps of Engineers* (Construction Engineering Research Laboratory - CERL) u *Champaign*, država *Illinois*. Prvenstvena zamisao je bila razviti pouzdan softverski alat za potrebe upravljanja zemljištem u vojne svrhe. Tokom druge polovine osamdesetih godina, CERL je publikovao GRASS sa kompletnim izvornim kodom na internet. Ekspanzija razvoja i korišćenja interneta u to vreme, svakako je doprinela da se GRASS kao GIS softver utvrdi na svetskom nivou [4].

Kasnije, 1995 godine CERL odustaje od daljeg razvoja GRASS-a i napušta projekat. Dve godine potom pojavljuje se nova verzija, GRASS 4.2, publikovana od strane *Baylor University*. Verzija GRASS 4.2.1 je realizovana i publikovana 1998 godine. Kordinirana je od strane dva profesora, Marcus Neteler i Helena Mitasova, a razvijena na *Institute of Physical Geography and Landscape Ecology, University of Hannover*. Ovom verzijom ispravljene su gotovo sve primećene softverske greške i razvijeno je i dodato više od 50 novih modula. Razvoj naredne verzije, GRASS 5.0, počinje 1999 godine. Oktobra iste godine realizovana je GRASS 5.0beta verzija pod GNU GPL (*General Public License*) licencom. Ova licenca štiti autore softvera od bilo kakve zloupotrebe i pritom pruža uvid u kompletan izvorni kod. Krajnji korisnik može analizirati postojeće metode, razumeti njihovu funkcionalnost, modifikovati program prema svojim potrebama, korigovati i poboljšavati postojeće module. Od 2001 godine, GRASS istraživački tim dobija nov štab u Italiji, tačnije ITC-irst (*Centro per la Ricerca Scientifica e Tecnologica*), *Trento, Italy*. GRASS 5.0.0 verzija je zvanično realizovana 2002 godine sa značajnim poboljšanjima u odnosu na verziju 4.x. Uključivala je podršku za *float* raster podatke i 3D interaktivnu vizualizaciju. U isto vreme radilo se na GRASS 5.7 verziji i implementaciji 3D višelejerskih vektorskih podataka, 3D TINa... Povećane aktivnosti na razvijanju GRASS softvera dovele su do toga da se septembra 2002 godine organizuje prva međunarodna *Open Source / Free Software GIS – GRASS* konferencija u Italiji.

Stabilna verzija, GRASS 6.4.1, rezultat je više od 9000 softverskih unapređenja i popravaka u odnosu na prethodnu stabilnu verziju GRASS 6.2.3. Poslednja publikovana verzija, GRASS 6.4, uključuje novo modernije grafičko okruženje (GUI) sa integriranim „*Location Wizard*“ kojim se može definisati baza GRASS podataka kao i vršiti odabir projekcionih parametara. Pored toga, ova verzija još integriše i vektor digitajzer, alat za *SQL* upite, 3D mod pogleda, alate za georeferenciranje.

4. GIS koncept, geoprostorni modeli podataka

Prostorni podaci su „srce“ svake GIS procedure i aplikacije. Prostorni podatak je georeferenciran svojom lokacijom na površini Zemlje. To podrazumeva tačno zabeleženu lokaciju, određen koordinatni sistem, jedinice mere i projekciju karte. Kada se prikazuje, prostorni podatak uvek ima odgovarajuću razmeru kao i na papirnoj karti [5].

4.1. Rasterski model podataka

Rasterski model podataka predstavlja digitalni zapis podatka baziran na regularnoj matričnoj strukturoj formi. Naime, njegova struktura je predstavljena matricom koja je definisana određenim brojem kolona i redova. Presek jednog reda i jedne kolone definišu osnovni i najmanji element matrice. Ako govorimo o 2D rasterskoj strukturi onda je to je piksel ili ćelija. U slučaju da se radi o 3D modelu podataka, najmanji volumenski element je voksel. Upravo odavde se može zaključiti da je svaki piksel unutar jedne matrične strukture jednoznačno određen brojem reda i kolone u čijem preseku se nalazi. Pored toga, svaki piksel „uz sebe“ ima pridruženu vrednost atributa koja može predstavljati intenzitet reflektovanog signala, boje, visinu. Vrednosti ovih atributa mogu se odnositi na sam piksel ili samo na njegova temena. Kada se odnose na ceo piksel onda takav rasterski podatak predstavlja sliku (satelitski snimak, skenirana mapa). U slučaju kada se vrednosti atributa odnose na temena onda takav raster predstavlja kontinualnu površinu (visina, temperatura). Osnovne karakteristike jedne rastrske strukture podataka su: rezolucija, dimenzija, broj boja i format zapisa. Rezolucija predstavlja veličinu piksela izraženu u dužinskim jedinicama ili brojem piksela na određenom prostoru. Dimenzija je definisana brojem kolona i redova. Broj boja predstavlja domen vrednosti pojedinačnog piksela (1 bit, 8 bit, 24 bit...).

4.2. Vektorski model podataka

Vektorski podaci su poseban tip podataka čiju strukturu čine osnovne geometrijske primitive: tačka, linija i poligon. S obzirom da se položaj primitiva definiše koordinatama, odnosno vektorom položaja, ovi podaci su dobili naziv vektorski podaci.

Osnovni element vektorskog sadržaja je tačka. Položaj tačke definisan je njenim koordinatama. S obzirom da nema dimenziju, tačka predstavlja nula-dimenzionalnu geometrijsku primitivu i njom su prikazani entiteti koji se zbog malih dimenzija ne mogu prikazati u razmeri pomoću linija i poligona. Linija je organizovan skup povezanih tačaka, predstavlja jednodimenzionalnu geometrijsku primitivu. Koristi se za prikaz linijskih objekata koji se zbog mali dimenzija ne mogu prikazati poligonom u razmeri. Poligon predstavlja organizovan skup linija kojima se definiše neka oblast, a od kojih se prva i poslednja tačka poklapaju. Poligon je dvodimenzionalna geometrijska primitiva.

Vektorski model podataka baziran je na „luk - čvor“ reprezentaciji, sastoji se od krivih koje se ne presecaju (lukovi). Jedan luk definisan je nizom tačaka, kranje tačke luka

su čvorovi a tačke duž luka su verteksi. Dve uzastopne tačke na jednom luku ili krivoj predstavljaju jedan segment.

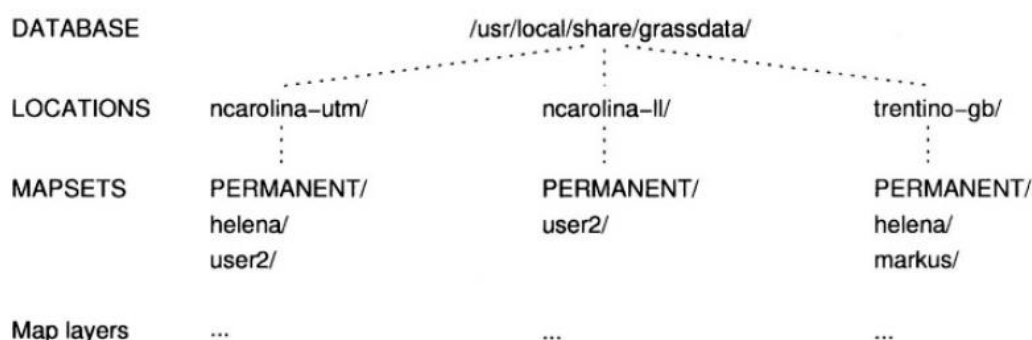
4.3. Atributivni podaci

Atributskim podacima se izražavaju negeometrijske karakteristike entiteta. Najveći deo atributskih podataka u GIS-u dat je u alfa-numeričkom obliku (slova I cifre). U najvećem broju slučajeva podaci su prikazani kroz tabele pa se nazivaju I tabelarni podaci. Oni predstavljaju kompletan skup izabranih podataka spremnih za upotrebu (ready-to-use). Atributske tabele sadrže opisne informacije o karakteristikama objekata u realnom svetu, koriste se na različite načine I najraznovrsnije svrhe. U postojećim GIS rešenjima tabele se automatski pridružuju prostornim podacima.

5. Struktura GRASS baze podataka

Predefinisana i netipična organizacija podataka unutar GRASS baze podataka je nešto što je svojstveno isključivo ovom softveru. Naime, GRASS egzistira na jedinstveno definisanoj strukturi podataka koja skladišti prostorne i druge informacije kroz niz direktorijuma i poddirektorijuma u okviru jedinstvene baze podataka. Svaki direktorijum sa svojim poddirektorijumima i folderima, pored karakterističnog naziva nosi i niz drugih značajnih informacija. Iz ovoga se može zaključiti da je najpre potrebno dobro upoznati ovu strukturu i način njenog funkcionisanja kako bi bilo koji softverski zadatak bio uspešno obavljen.

GRASS baza podataka predstavlja direktorijum unutar memorijskog kapaciteta računara, napravljen i imenovan od strane korisnika. Najčešći naziv ovog direktorijuma je *GISDATABASE* ili *DATABASE*. Kreiranje ovog direktorijuma je prvi korak pre početka bilo kakvog rada u GRASS softveru. Struktura podataka unutar ovako definisane baze podataka počiva na principu *LOCATIONS / MAPSET*, što prikazuje slika 5-1 ispod.



Slika 5-1: Organizacija GRASS GIS baze podataka

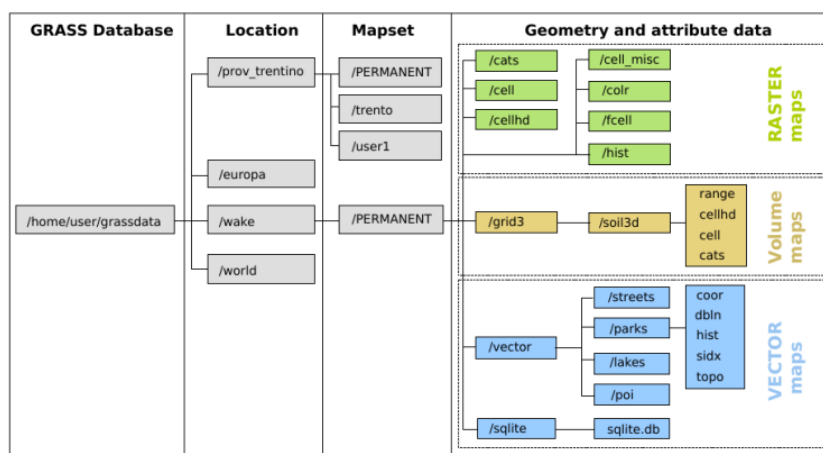
LOCATION se u najopštijem slučaju može posmatrati kao biblioteka, poddirektorijum unutar GRASS baze podataka, koji skladišti podatke unutar istog

geografskog regiona. Svaki *LOCATIONS* definisan je koordinatnim sistemom, projekcijom i geografskim granicama. Unutar baze podataka može se nalaziti niz *LOCATIONS*, s tim da se svaki odnosi na tačno definisan geografski region. Unutar svakog *LOCATIONS* foldera nalaze se jedan ili više *MAPSET*. Tehnički, *MAPSET* je samo poddirektorijum *LOCATIONS*. Najčešće, *MAPSET* predstavlja tematsku bazu podataka ili skup podataka za jedan podregion u okviru regiona koji definiše njegov *LOCATIONS*. Isto tako, podaci u okviru jednog *MAPSET*s mogu se odnositi i na ukupan, ceo region. Postoji više dobrih razloga za podelu *LOCATIONS* na *MAPSET*. Jedan od njih je i obezbeđivanje simultanog pristupa za više korisnika jednom istom *LOCATIONS*. Naime, svaki korisnik pristupa svom *MAPSET* sa svojim podacima i ima mogućnost da ih pregleda, menja, ažurira. Sve ostale *MAPSET* može samo pregledati (*read only*). U situaciji kada više korisnika radi na jednom istom projektu ovo je i više nego korisno.

Unutar svakog *LOCATIONS* nalazi se i jedan poseban *MAPSET* pod imenom *PERMANENT*. Ovaj *MAPSET* se automatski kreira od strane GRASS čim korisnik kreira novu *LOCATIONS*. U njemu se nalaze svi bazni, osnovni podaci, za jedan *LOCATIONS*. Samo jedan korisnik, kreator, ima mogućnost dodavanja, ažuriranja, brisanja podataka u okviru *PERMANENT MAPSET*. Podaci odatle mogu biti podeljeni u ostale *MAPSET*, gde im mogu pristupi i ostali korisnici. U okviru *PERMANENT* foldera nalazi se i fajl *DEFAULT_WIND* koji sadrži koordinate podrazumevanig (*Default*) regiona. Ovaj *MAPSET* je posebno koristan jer obezbeđuje zaštitu svih osnovnih podataka u situacijama kada više korisnika radi na istom projektu.

5.1. Fajl struktura

Prethodno poglavlje definiše koncept i osnovnu strukturu jedne GRASS baze podataka. Sada je potrebno detaljnije objasniti značenje pojedinih foldera i fajlova koji se odnose na rasterske / vektorske podatke, unutar samog *MAPSET*. Slika 5.1-1 daje strukturni pregled fajlova baze podataka.



Slika 5.1-1: Strukturni pregled fajlova unutar baze podataka

GRASS rasterski lejeri definisani su nizom fajlova koji su raspoređeni u nekoliko foldera unutar jednog *MAPSET*. Folderi su organizovani na sledeći način:

cellhd/ Predstavlja zaglavlje (*header*) mape. U njemu se nalaze osnovne informacije kao što su kod projekcije, ivične koordinate raster mape, broj redova i kolona, rezolucija, informacije o kompresiji mape.

cell/, fcell/, ili grid3/ Sadrži matricu vrednosti u kompresovanom formatu koji zavisi od rasterskog tipa podataka (*integer, floating* ili 3D grid).

hist/ Fajl koji sadrži metapodatke kao što su izvor podataka, komande koje su korišćene pri kreiranju raster mape i drugo.

cats/ Opcioni fajl, može sadržati tekst ili numeričke oznake vezane za određenu kategoriju raster mape.

colr/ Dostupne kolorne table za mapu.

cell_misc/ Opcioni fajl, najčešće sadrži vremenske oznake.

GRASS vektorske mape su smeštene u nekoliko odvojenih foldera u okviru jednog direktorijuma. Folderi su organizovani na sledeći način:

head Predstavlja zaglavlje vektorske mape u ASCII formatu sa osnovnim informacijama o kreiranju mape (vreme, datum, ime).

coor Binarni fajl koji sadrži koordinate grafičkih elemenata vektorske mape (primitiva).

topo Binarni fajl, definiše prostorni odnos između geometrijskih elemenata mape.

Hist Fajl u ASCII formatu koji sadrži sve komande korišćene pri kreiranju vektorske mape, kao i ime, vreme, datum kreiranja.

5.2. Geografski region, region

Pojam geografskog regiona je nešto što je svakako svojstveno ovom softveru i jako je važno upoznati se sa njegovim značenjem kako bi kasniji rad u softveru imao smisla i bio lakši. Termin će se i u samom radu često pominjati.

Naime, u GRASS-u region asocira na diskretnu geografsku oblast. Ona je definisana svojim granicama u okviru oređenog koordinatnog sistema i projekcije. Takođe, unutar svakog regiona definisana je njegova rezoluciji, odnosno dimenzije njegovih najmanjih delova (celija). Rezolucija se definiše po pravcu sever-jug i istok-zapad i mora biti ista po oba pravca. Potrebno je praviti razliku između *default*, *current* i *saved* regiona. Upravljanje regionom unutar *MAPSET* dostupno je preko modula *g.region* [1].

6. Funkcionalnost i osnovna sintaksa GRASS softvera

Poslednja stabilna verzija GRASS softvera obezbeđuje 425 modula za upravljanje i analizu podataka. Moduli su grupisani po funkcijskim klasama (*display, raster, vector...*); referenca na svaku klasu je njen prefiks sto je najčešće početno slovo (*d.*, r.*, v.*...*). U tabeli 6-1 data je lista osnovnih funkcijskih klasa sa prefiksom kojim se pozivaju iz komandne linije i opisom same funkcije [6].

<i>Prefix</i>	<i>Function class</i>	<i>Type of command</i>
d.*	display	graphical output
s.*	sites	site data processing
r.*	raster	raster data processing
i.*	imagery	image processing
v.*	vector	vector data processing
g.*	general	general file operations
m.*	misc	miscellaneous commands
p.*	paint	map creation in PPM format
ps.*	postscript	map creation in Postscript format

Tabela 1: Spisak osnovnih funkcijskih klasa sa prefiksom i opisom

Ovakva kategorizacija prilično olakšava korisniku navigaciju kroz mnoštvo alata i funkcija. Dakle, prefiks daje referencu za određenu klasu, iza klase obavezno dolazi tačka a iza nje jedna ili dve ključne reči, razdvojene tačkom, koje definišu šta se tačno izvršava unutar navedene funkcije. Upravo ovih par pravila definiše osnovnu sintaksu softvera:

modul [-flag1[flag2...]] parametar1=mapa1[,mapa2,...]\

[parametar2=broj...] [--o] [--q] [--v]

- **Modul;** označava ime komande tj. funkcije.
- **Flag;** opciono, predstavlja specijalne karakteristike, poziva se određenim slovom.
- **Parameter;** predstavlja ime za ulazni ili izlazni fajl, konstantu, ime metode, simbol.

Važno je znati da između ključnih reči koje su razdvojene tačkom ne sme biti razmaka kao ni pre i posle znaka jednakosti. Sa druge strane, između parametara i *flags* se nalazi razmak dok je više parametara pri nabranjanju razdvojeno zarezom. Takođe, prvi parametar koji se navodi ne zahteva ispis ključne reči *parameter*. Sa nekoliko sledećih primera biće ilustrovano korišćenje gore navedenih pravila. Recimo, *v.in.shape* predstavlja vektorsku komandu za učitavanje ESRI SHAPE fajla, *r.buffer* računa bafer zonu raster linije i raster zone, *g.list rast* će ispisati listu raspoloživih rastera unutar *MAPSET* itd.

7. Digitalni model terena, osnovni pojmovi

„DMT (Digital Terrein Model - DTM) je statistička reprezentacija kontinualne površi terena pomoću velikog broja odabranih tačaka sa poznatim X, Y i Z koordinatama u određenom koordinatnom sistemu“ – prva definicija DMT (Miller i La Flamme 1958).

Po rečima Dr. Borisov Mirka autora članka Digitalni modeli terena za teritoriju SCG formiranih na osnovu vojnih topografskih karata: Odavno postoji potreba i težnja da se pri kartiranju zemljišta na konvencionalnim podlogama prikaže i treća dimenzija. Sa više ili manje uspeha to je činjeno različitim tehnikama, senčenjem, šrafiranjem, višebojnom hipsometrijskom skalom boja, izohipsama, konstrukcijom perspektivnih pogleda, ispisivanjem kota karakterističnih tačaka itd. U izradi topografskih karata skoro isključivo je korišćen metod konstrukcije izohipsi i ispisivanja kota, a visinska predstava se odnosila na ortometrijske visine, tj. na visine iznad površi geoida.

Danas, mnoge analize bazirane na digitalnoj predstavi zemljišta kao i projektovanja različitog tipa zahtevaju drugačiji način organizacije i prikaza podataka vezanih za treću dimenziju. Neki od najčešćih zahteva su: konstrukcija podužnih, poprečnih i drugih profila, računanje kubatura zemljanih masa, prikaz linija terena istog nagiba, izrada perspektivnih prikaza terena itd. Uvođenjem računara u upotrebu moguće je, drugačijim prikazom i organizacijom podataka, uspešno odgovoriti složenim zahtevima. Danase se u praksi najčešće koriste dva termina za organizaciju podataka o prostoru kada je u pitanju visinska predstava:

- Digitalni model visina (*Digital Elevation Model - DEM*), koji se odnosi na sistem visina u vidu predefinisiranog rasporeda tačaka koji prekriva površ terena i
- Digitalni model terena (*Digital Terrein Model - DTM*), koji pored visina tačaka, uzima u obzir i reljefne odlike terena, odnosno posebne tačke, linije i površi.

Digitalni model terena kao statistička predstava kontinualne površi zemljišta, prema načinu čuvanja podataka o visinama može imati dva oblika: pravilna rešetka – GRID i sistem trouglova – TIN. I jedna i druga forma organizacije podataka predstavljaju izveden prikaz DMT. Ovaj prikaz predstavlja generalizaciju podataka izvornog prikaza DMT pa je iz tog razloga prvo potrebno definisati izvorni DMT [7].

Izvorni DMT opisuje se tačkama, linijama i površima, a njihovim međusobnim rasporedom i uklapanjem daju se relativna nadvišenja između određenih lokacija na zemljištu. Ukupan broj tačaka, linija ili visinskih površi zavisi od samog oblika i konfiguracije terena. Dakle, manji broj tačaka za ravničarske terene i terene jednolikog pada i veći broj tačaka za terene složenijeg reljefa. Trebalo bi težiti ravnomernom rasporedu tačaka duž čitave oblasti terena. Posebno je potrebno voditi računa da se za kotiranje biraju i najviši i najniži delovi terena (vrhovi, prevoji, sedla), sastavi reka, raskrsnice puteva, mostovi, tačke vododelnica itd. Visinska predstava zemljišta u digitalnom obliku na nivou izvornog DMT obuhvata:

- masovne tačke (nekarakteristične tačke poznate visine),
- karakteristične tačke (vrhovi, sedla, prevoji),
- kontrolne tačke (trigonometri, reperi),
- prelomne (prekidne) linije duž kojih se teren lomi u vertikalnom smislu i
- karakteristične površi (vodena ogledala, stenjaci, kamenolomi i konture interpolacionih područja) [7].

Izveden DMT predstavlja transformaciju svih informacija sadržanih u izvornom skupu podataka, odnosno u njihovoj strukturi koja je raspoloživa za opis visinske predstave Zemljine površi, bez degradacije kvaliteta. Raspoloživa su dva već pomenuta načina:

- razbijanje područja interpolacije na pravilnu mrežu (GRID) i
- razbijanje područja interpolacije na mrežu nepreklopnih trouglova (TIN).

Organizovanje podataka na bazi pravilne mreže tačaka predstavlja najjednostavniji način modelovanja površi terena. Mreža se obično sastoji od kvadrata čija temena predstavljaju tačke sa poznatim visinama. Pored ovog rešenja postoji i mogućnost modelovanja površi mrežom pravougaonika, pravilnih trouglova ili šestougaoicima. Glavna prednost ovakvog pristupa jeste jednostavna struktura podataka koja omogućava brze prostorne operacije nad elementima u modelu GRID. Takođe, postoje i znatne uštede u memoriji jer za tačke u modelu GRID nije potrebno čuvati sve podatke. Naime, u bazi DMT nalaze se osnovni podaci o elementima: koordinate početne tačke, dimenzije GRID modela, broj redova i kolona, rotacija mreže i drugi relevantni podaci. Pored ovih podataka potrebno je uvođene dodatne topologije za čuvanje karakterističnih tačaka i linija terena koja padaju van temena GRID strukture.

Kod TIN modela površ se diskretizuje prostornim nizom nepreklopajućih trouglova. Temena trouglova, koja su u opštem slučaju nepravilnog oblika, jesu tačke sa poznatim visinama, odnosno rezultat neposrednih merenja. Uz koordinate temena trouglova, u TIN strukturu se jednostavno mogu ugraditi i druge neophodne informacije kao što su karakteristične linije terena i površi sa posebnim značenjem. Iz tog razloga se ova struktura smatra fleksibilnijom. Stalni predmet rasprave je poređenje ova dva modela:

- tačke u GRID imaju jednostavnu strukturu,
- vreme pristupa u GRID do određenih elemenata značajno je manje nego u TIN,
- količina podataka u GRID je znatno manja,
- model visina u TIN bolje aproksimira reljef..

Uopšteno, DTM definisan je kao digitalna (numerička) reprezentacija terena i može sadržati sledeće četiri grupe (topografskih i netopografskih informacija):

- Zemljišne forme kao što su visine, nagibi, razdelne linije, preseci i druge složene geomorfološke karakteristike koje se koriste pri opisu reljefa.

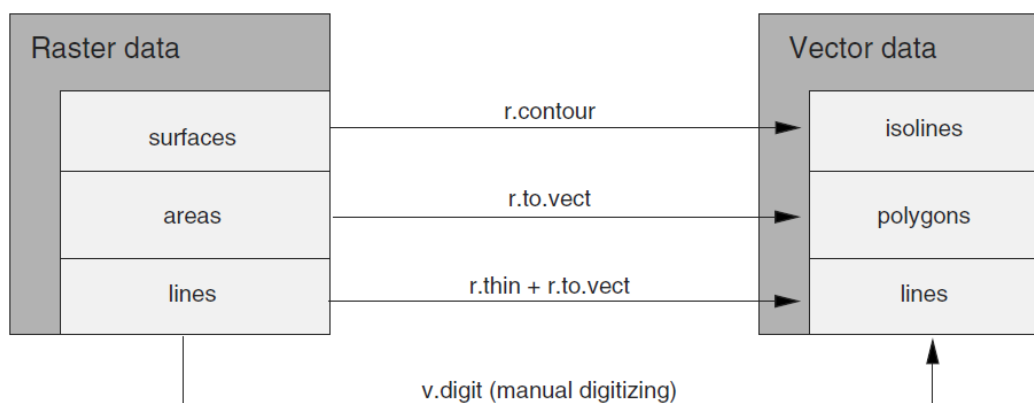
- Karakteristike terena, hidrološke karakteristike (reke, jezera, obalne linije), transportne mreže (putevi, pruge, staze), naselja, granice, međne linije.
- Prirodni resursi i životna sredina, tlo, vegetacija, klima.
- Socio-ekonomski podaci kao što su distribucija stanovništva u određenoj oblasti, industrija i poljoprivreda, kapitalni prihodi [8].

8. Rad sa raster podacima i kreiranje DMT

U poglavljima gore data su osnovna objašnjenja vezana za rastersku strukturu podataka i njegove karakteristike. Ovde će diskusija biti usmerena na rad sa GRASS rasterskim podacima u cilju kreiranja DMT kao i na njihovu konverziju u vektorske podatke. Biće prikazano korišćene osnovnim rasterskih GRASS modula vezanih za kreiranje DMT kao i objašnjenja njihovih osnovnih parametara i mogućnosti.

8.1. Transformacija raster podataka (vektorizacija)

Pristup i način konverzije raster podataka pre svaga zavisi od tipa rasterskog podatka kao i od korišćene aplikacije (Slika 8.1-1). Kao što je i ranije rečeno, raster strukturom mogu se prikazati linijske karakteristike, tačke ili homogene oblasti. Bez obzira o kakvom se tipu geometrije radi, raster podaci mogu biti konvertovani u vektorske korišćenjem modula *r.to.vect*. Modul skenira raster mapu koja predstavlja ulazne podatke, ekstrahuje geometrijske karakteristike iz tačaka, linija ili oblasti i konvertuje podatke u GRASS vektor format. Kompletna sintaksa modula sa dostupnim parametrima i njihovim objašnjenjem data je ispod [9].



Slika 8.1-1: Prikaz tipa podataka i njihove konverzije

```
r.to.vect [-svzbt] input=ime output=ime type=tekst [column=name] [--overwrite] \  
[--help] [--verbose] [--quiet]
```

Flags:

-s Glatke, zaobljene ivice oblasti

- v** Korišćenje rasterskih vrednosti kao kategorija
- z** Ispisivanje raster vrednosti kao Z koordinate
- b** Bez izgradnje vektor topologije
- t** Preporučuje se kod konverzije velikog broja tačaka, bez kreiranja atributivne tabele
- overwrite** Dozvola da rezultujući fajl bude prikazan preko postojećeg
- help** Rezime korišćenja modula, pojašnjenje
- verbose** Verbalan rezultujući modul
- quiet** Rezultujući *Quiet* modul

Parametri:

input=ime [obavezno]

Ime ulaznog raster lejera/mape

output=ime [obavezno]

Ime rezultujućeg vektor lejera/mape

type=tekst [obavezno]

Ulazni tip podatka

Opciono: tačka, linija, oblast

column=ime

Ime kolone u atributivnoj tabeli koja će skladištiti rezultujuće vrednosti

Ime mora biti SQL kompatibilno

Podrazumevano: *value*

8.2. Modul *r.contour*

Modul *r.contour* generiše vektorsku mapu. Ulazni podaci su konturne linije rasterskog formata [10].

r.contour input=ime output=ime [step=float] [levels=float[,float,...]] [minlevel=float] \ [maxlevel=float] [cut=integer] [--overwrite] [--help] [--verbose] [--quiet]

Parametri:

input=ime [obavezno]

Ime ulazne raster mape

output=ime [obavezno]

Ime rezultujuće vektor mape

step=float

Korak između konturnih linija (ekvidistanca)

levels=float[,float,...]

Lista konturnih linija data njihovim visinama

minlevel=float

Minimalna vrednost izohipse

maxlevel=*float*

Maksimalna vrednost izohipse

cut=*integer*

Minimalni broj tačaka koji generišu jednu konturnu liniju (0 -> nema ograničenja)

Podrazumevano: 0

Konturne linije mogu se definisati parametrom *levels* tako što se jednostavno navode visine koje generišu presečnu horizontalnu površ i odvajaju se zarezom ili definisanjem parametra *step*, *minlevel* i *maxlevel*. Ako se ne specificiraju parametri *minlevel* i *maxlevel* modul će automatski koristiti minimalnu i maksimalnu vrednost raster ćelije. Modul može koristiti samo jedan od ova dva načina, ako su definisani i *step* i *levels* parametri modul će ignorisati *step* parametar. Vrednost parametra *step* predstavlja ekvidistanciju. U procesu određivanja ovog parametra potrebno je dobro poznavati teren kako bi se njegova vrednost optimalno odredila. Opcioni *cut* parametar dozvoljava korisniku da specificira minimalni broj raster ćelija na osnovu kojih će biti konstruisana izohipsa [10].

Primer ispod ilustruje proces generisanja izohipsi u vektorskom formatu od ulaznih raster podatka. Izohipse su iscrtane sa ekvidistacijom od 10 m za delove iznad 70 i ispod 120 m. Slika 8.2-1 prikazuje preklopljene ulazne i rezultujuće podatke.

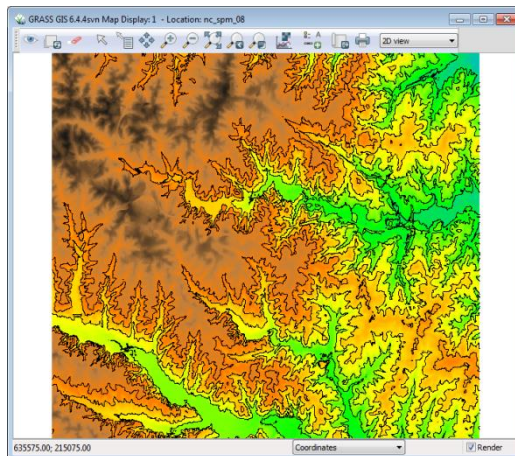
```
g.region rast=elevation -p
```

```
d.rast elevation
```

```
r.info elevation -r
```

```
r.contour elevation out=elev_contour_10m step=10 minlevel=70 maxlevel=120
```

```
d.vect elev_contour_10m
```



Slika 8.2-1: Preklopljene izohipse i ulazni raster podatci

8.3. Modul *r.resamp.interp*

Modul *r.resamp.interp* sadrži nekoliko resampling algoritama. Specificiranim resampling algoritmom vrši interpolaciju koristeći određen broj susednih piksela iz ulaznih raster podataka [11].

r.resamp.interp **input=ime** **output=ime** [**method=string**] [--**overwrite**] [--**help**] [--**verbose**]\ [--**quiet**]

Parametri:

input=ime[**obavezno**]

Ime ulazne raster mape

output=ime [**obavezno**]

Ime rezultujuće raster mape

method=string

Sampling interpolacioni metod

Dostupno: *nearest, bilinear, bicubic, lanczos*

Podrazumevano: *bilinear*

nearest: Nearest-neighbor interpolation

bilinear: Bilinear interpolation

bicubic: Bicubic interpolation

Sve ćelije koje predstavljaju susedne ćelije moraju imati ne *null* vrednost kako bi rezultat u izlaznom rasteru bio ne *null*. Modul sadrži četiri algoritma koja se specificiraju parametrom *method*. Svaki od dostupnih algoritama koristi ponderisane vrednosti određenog broja susednih piksela iz ulaznih podataka kako bi odredio vrednost svakog piksela u izlaznom skupu podataka. Dostupni algoritmi:

1. *Nearest neighbor* (1 ćelija)
2. *Bilinear* (4 ćelije)
3. *Bicubic* (16 ćelija)
4. *Lanczos* (25 ćelija)

Važno je napomenuti da je ovaj modul prvenstveno namenjen reinterpolaciji (promena rezolucije) podataka koji reprezentuju kontinualnu površ dok se slabije koristi za rasute podatke koji predstavljaju nekakvu geometriju. U tu svrhu može se koristiti modul *v.surf.**. Resampling moduli (*r.resample, r.resample.stats, r.resamp.interp, r.resamp.rst*) resampluju mapu tako da se ona pri zumiranju određenih delova poklopi sa početno definisanim regionom. Pri tom naravno, cilj je očuvanje kvaliteta što zavisi od korišćenog algoritma. Potrebno je znati da kod *bilinear, bicubic* i *lanczos* interpolacije ćelija za koju se računa vrednost ne sme biti okružena ćelijama čija je vrednost *null*. Do ovoga bi moglo doći ako bi se desilo da se ćelije iz ulaznog skupa podataka nalaze ven definisanog regiona.

Primer ispod prikazuje generisanje DMT koristeći dve različite metode (*bilinear* i *bicubic*), kao i uporednu analizu rezultata prikazanu slikama 8.3-1 i 8.3-2.

g.region rast=elevation -p

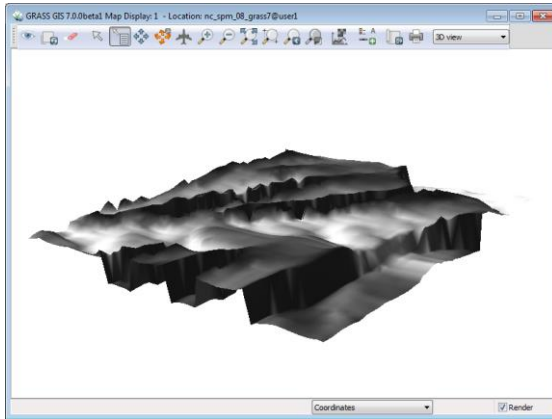
r.resamp.interp elev_ned_30m out=elev_ned10m_bil meth=bilinear

r.resamp.interp elev_ned_30m out=elev_ned10m_bic meth=bicubic

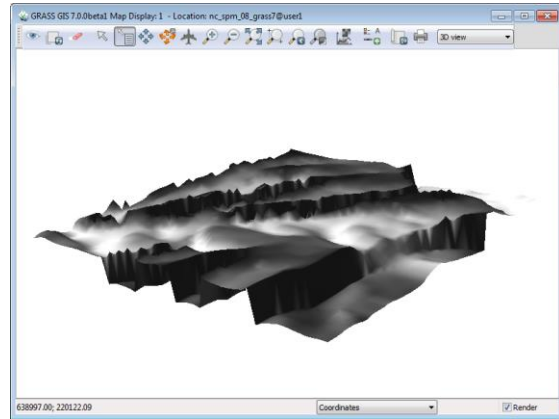
r.slope.aspect elev_ned10m_bil aspect=aspect_ned10m_bil

r.slope.aspect elev_ned10m_bic aspect=aspect_ned10m_bic

d.rast aspect_ned10m_bil
d.rast aspect_ned10m_bic



Slika 8.3-1: Bilinear



Slika 8.3-2: Bicubic

8.4. Modul *r.resamp.rst*

Reinterpolacija sa mogućnošću topografske analize primenjeno na ulaznu raster mapu, proizvodi takođe raster mapa (može biti specificirana nova rezolucija). Modul koristi regulacioni *spline* metod sa parametrima *tension* i *smoothing* [12].

```
r.resamp.rst [-td] input=ime ew_res=float ns_res=float [elev=string] [slope=string] \  
[aspect=string] [pcurv=string] [tcurv=string] [mcurv=string] [smooth=string] \  
[maskmap=string] [overlap=integer] [zmult=float] [tension=float] [theta=float] \  
[scalex=float] [--overwrite] [--help] [--verbose] [--quiet]
```

Flags:

- t Koristi slobodan *tension* parametar
- d Prikaz parcijalnih izvoda umesto topografskih parametar

Parametri:

input=ime[obavezno]
Ime ulazne raster mape
ew_res=float [obavezno]
Definisanje rezolucije po pravcu istok-zapad
ns_res=float [obavezno]
Definisanje rezolucije po pravcu sever-jug
elev=string
Rezultujući fajl sa z-vrednostima, elevaciona mapa
slope=string
Rezultujuća *slope* mapa (nagib)
aspect=string
Rezultujuća *aspect* mapa (pogled)

pcurv= *string*
Rezultujuća *profile curvature* mapa (profil zakrivljenosti)

tcurv= *string*
Rezultujuća *tangential curvature* mapa

mcurv= *string*
Rezultujuća mean curvature mapa

smooth= *string*
Ime raster mape koja sadrži *smoothing*

maskmap= *string*
Ime raster mape koja će se koristiti za masku

overlap= *integer*
Broj redova/kolona preklapanja za segment
Podrazumevano: 3

zmult= *float*
Faktor skaliranja za z-vrednosti
Podrazumevano: 1.0

tension= *float*
Vrednost *spline tension* parametra
Podrazumevano: 40.

theta= *float*
Anisotropy ugao

scalex= *float*
Anisotropy faktor skaliranja

Ovaj modul prvenstveno je namenjen za interpolaciju podataka pri promeni rezolucije koji reprezentuju kontinualnu površ. Ređe se primenjuje na raster mape koje obiluju „razbacanim“ pikselima. Reinterpolacija (resampling) se može vršiti na istu, manju ili veću rezoluciji zavisno od postavljenih vrednosti parametara *ew_res* i *ns_res*. Svaka mapa koja predstavlja kreirani proizvod, kreira se koristeći parametre trenutnog regiona koji se ne mora poklapati sa regionom ulaznih podataka.

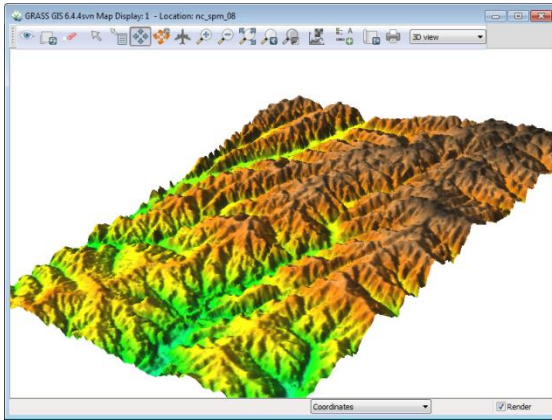
Simultano sa procesom interpolacije moguće je i računanje topografskih parametara koristeći z-vrednosti ulaznog seta podataka koje predstavljaju visinu/dubinu određenog temena piksela. Parametri koji se mogu računati su najčešće nagibi, pogledi, profili zakrivljenosti (u pravcu najvećeg pada), tangencijali profili (pravac od tangente ka konturnoj liniji). Kako bi se izvršila navedena računanja potrebno je specificirati parametre *slope*, *aspect*, *pcurv*, *tcurv*, *mcurv* respektivno.

Postavljanjem *smoothing* parametra na nulu rezultujuća površ će prolaziti tačno kroz tačke koje je generišu. Korisnik takođe može definisati raster mapu, *maskmap*, kojom će obezbediti primenu algoritma na tačno određene vrednosti. Interpolacioni algoritam neće uzimati u obzir ćelije sa vrednostima *null* ili nula pri kreiranju *mask*. Parametar *zmult* dozvoljava korisniku da transformiše vrednosti z skale što može biti jako korisno ako je potrebno konvertovati vrednosti visina iz jednih u druge jedinice mere. Podrazumevana vrednost ovog parametra je 1, dakle nema konverzije.

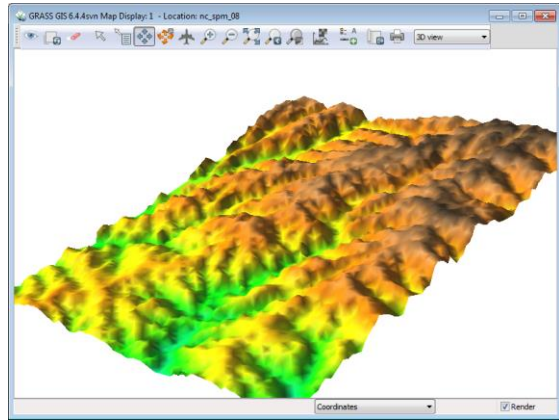
Tension parametar definiše karakter rezultujuće površi. Više vrednosti ovog parametra redukuju promašaje/greške koje mogu nastati na površi sa značajnom promenom gradijenta. Primer ispod pokazuje način generisanja DMT modulom *r.resamp.rst* i uticaj promene *tension* parametra (Slika 8.5-1 i Slika 8.5-2).

```
g.region rast=elev_ned_30m -p
r.resamp.rst elev_ned_30m elev=elev_ned10m_ten40 ew_res=10 ns_res=10
r.resamp.rst elev_ned_30m elev=elev_ned10m_ten80 ew_res=10 ns_res=10
```

```
d.rast elev_ned10m_ten40
d.rast elev_ned10m_ten80
```



Slika 8.5-1: Parametar *tension* 40



Slika 8.5-2: Parametar *tension* 80

8.5. Modul *r.surf.idw* (*Inverse Distance Weight - IDW*)

Modul predstavlja prostorni interpolacioni alat namenjen obradi rasterske strukture podataka [13].

```
r.surf.idw [-e] input=ime output=ime [npoints=integer] [--overwrite] [--help] \
[--verbose] [--quiet]
```

Flags:

-e Prikaz interpolovanih grešaka

Parametri:

input=*ime* [obavezno]

Ime ulazne raster mape

output=*ime* [obavezno]

Ime rezultujuće raster mape

npoints=*integer*

Broj tačaka pri interpolacionom koraku

Podrazumevano: 12

Modul *r.surf.idw* „popunjava“, računa, vrednosti ćelija rastera interpolacijom vrednosti zadatih ulaznim setom podataka. Koristi tehniku numeričke aproksimacije baziranu na ponderisanim kvadratnim rastojanjima koja se odnose na određen broj susednih piksela. Broj susednih piksela od kojih se vrši aproksimacija potrebno je specificirati parametrom *npoints*, u suprotnom podrazumevan broj tačaka je 12 [13].

Za razliku od modula *r.surf.idw2* koji procesira sve ulazne podatke pri svakom koraku interpolacije, *r.surf.idw* teži da minimizuje broj ulaznih podataka koji će biti uključeni u proračun. Brzina izvršenja interpolacije je u funkciji pretrage pogodnih tačaka i ne povećava se značajno sa brojem ulaznih podataka. U tom smisli najgori scenario pri korišćenju modula *r.surf.idw* se dešava kada se ne poklapaju regioni procesiranih i rezultujućih podataka. Stepenn glatkosti dobijene površi interpolacijom povećava se sa brojem susednih tačaka koje se procesiraju, tj. glatkost je indirektno definisana parametrom *npoints* [13].

8.6. Modul *r.surf.contour*

Modul generiše površ u obliku raster mape pri čemu su ulazni podaci rasterizovane konturne linije. Iz gore navedenog se može zaključiti da je korak koji prethodi ovom modulu rasterizacija vektorskih izohipsi, korišćenjem modula *v.to.rast* i *r.thin* [14]. Sintaksa ima sledeći oblik:

```
r.surf.contour input=ime output=ime [--overwrite] [--help] [--verbose] [--quiet]
```

Parametri:

input=*ime* [obavezno]

Ime ulazne raster mape koja sadrži konturne linije

output=*ime* [obavezno]

Ime rezultujuće raster mape

Postupak određivanja visina ovim modulom sličan je manuelnom postupku. Naime, potrebno je prvo uočiti susedne izohipse na mapi unutar kojih se nalazi željena tačka čiju visinu je potrebno odrediti, zatim se metodom linearne interpolacije dolazi do visine tačke. Na sličan način radi i ovaj modul. Dakle, *r.surf.contour* algoritam baziran je na principu linearne interpolacije između susednih izohipsi. Ovaj modul će dati dobre rezultate ako postoje kote najviših vrhova i najnižih depresija mape i ako se ispoštuju sledeće okolnosti:

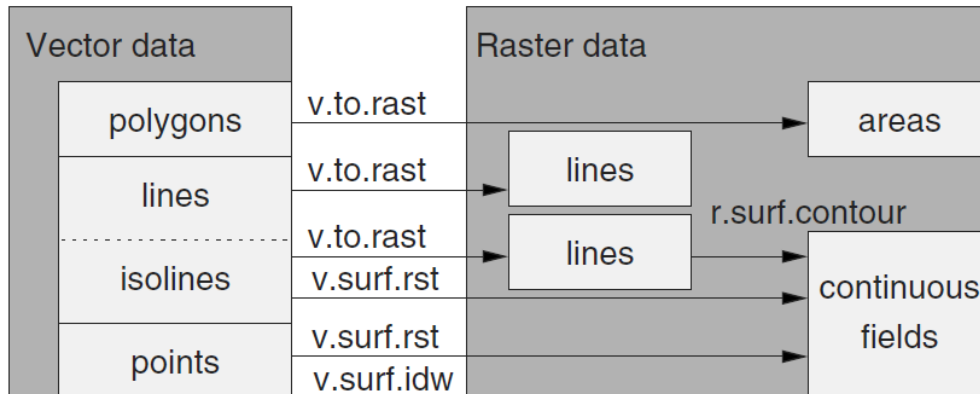
1. Konturne linije se pružaju do granica ragiona u kom se radi
2. Modul generiše mapu koja ima istu rezoluciju kao i ulazna raster mapa
3. Nema nepovezanih (otvorenih, isprekidanih) konturnih linija
4. Jedini podaci koji definišu visinu su izohipse (nema podataka između)

Potencijalne anomalije mogu se uočiti kreiranjem „*aspect*“ mape korišćenjem modula *r.slope.aspect*, gde će ulazni podatak biti raster mapa dobijena modulom *r.surf.contour*. Brzina izvršavanja modula *r.surf.contour* zavisi od „razmaknutosti“ konturnih linija (mereno brojem piksela) [14].

9. Rad sa vektorskim podacima i kreiranje DMT

9.1. Transformacija vektorskih podataka (rasterizacija)

Modul *v.to.rast* vrši konverziju GRASS vektor map lejera u GRASS raster map lejere (Slika 9.1-1). Rad ovog modula odnosiće se samo na one podatke koji se nalaze unutar granica vektorskih oblasti u radnom geografskom regionu. Opciono, moguće je konvertovati i vektorske attribute tako da predstavljau raster kategorije. Pre pokretanja ovog modula potrebno je proveriti granice regiona kao i vrednost rezolucije [15].



Slika 9.1-1: Prikaz tipa podataka i konverzije

```
v.to.rast [-d] input=ime[layer=string] [type=string[,string,...]] [cats=range] \
[where=sql_query] output=ime use=string [attrcolumn=ime] [rgbcolumn=ime] \
[labelcolumn=ime] [value=float] [rows=integer] [--overwrite] [--help] [--verbose] \
[--quiet]
```

Flags:

-d Kreiraj zgusnute linije (podrazumevano: razredene linije)

Parametri:

input=ime [obavezno]

Ime ulazne vektor mape ili izvora podataka za direktni OGR pristup

layer=string

Ime lejera ili broj

Vektorske karakteristike mogu imati kategorijske vrednsoti u više lejera. Ovaj broj određuje koji lejer se koristi.

Podrazumevano: 1

type=string[,string,...]

Ilazni tip podataka

Dostupno: *point, line, area*

Podrazumevano: *point,line,area*

cats=*range*

Vrednosti kategorija

Na primer: 1,3,7-9,13

where=*sql_query*

WHERE kondicional *SQL* upita bez ključne reči *where*

output=*name* [**obavezno**]

Ime rezultujuće raster mape

use=*string* [**obavezno**]

Izvor odakle se crpe raster vrednosti

Dostupno: *attr, cat, val, z, dir*

attr: čitanje/korišćenje vrednosti iz atributivne tabele

cat: korišćenje kategorijske vrednosti

val: korišćenje vrednosti određene *value* opcijom

z: korišćenje *z* vrednosti (tačke i konturne linije)

dir: output as flow direction (linije)

attrcolumn=*ime*

Ime kolone za *attr* parametar

rgbcolumn=*ime*

Ime kolone za definisan set boja (dostupno RRR:GGG:BBB)

labelcolumn=*ime*

Ime kolone koja će se koristiti za raster kategorije

value=*float*

Raster vrednosti (za *use=val*)

Podrazumevano: *1*

rows=*integer*

Broj redova koji će se čuvati u memoriji

Podrazumevano: *4096*

9.2. Modul *v.surf.bspline*

Modul ima integrirana dva metoda interpolacije, *bilinear spline* i *bicubic spline* sa *Tykhonov* regularizacionim parametrom. Ulazni podaci su 2D ili 3D vektorski *point* lejeri. Modul može interpolovati *z*-vrednosti, 3D tačke ili vrednosti određene atributske kolone iz 2D ili 3D vektor mape koje je specificirao korisnik. Rezultat jednog od metoda interpolacije može biti i raster i vektor mapa [16].

```
v.surf.bspline [-ce] input=ime [layer=string] [sparse_input=ime] [output=ime] \  
[raster_output=ime] [mask=ime] [sie=float] [sin=float] [method=string] \  
[lambda_i=float] [column=ime] [solver=ime] [maxit=integer] [error=float] \  
[memory=integer] [--overwrite] [--help] [--verbose] [--quiet]
```

Flags:

- c Pronalaženje optimalne vrednosti *Tykhonov* parametra
- e Procena gustine tačaka i međusobne udaljenosti

Parametri:

input= *ime* [**obavezno**]

Ime ulazne vektor *point* mape, ili izvora podataka za direktan OGR pristup

layer= *string*

Ime lejera ili broj. Vektorske karakteristike mogu imati kategorijske vrednosti u više lejera. Ovaj broj određuje koji lejer se koristi.

Podrazumevano: *1*

sparse_input= *ime*

Ime ulazne vektor mape sa proređenim tačkama ili izvora podataka za direktan OGR pristup

output= *ime*

Ime rezultujuće vektor mape

raster_output= *ime*

Ime rezultujuće raster mape

mask= *ime*

Ime raster mape koja se koristi za masku (primenjuje se isključivo na *raster_output*)

sie= *float*

Dužina *spline* koraka po pravcu istok-zapad

Podrazumevano: *4*

sin= *float*

Dužina *spline* koraka po pravcu sever-jug

Podrazumevano: *4*

method= *string*

Spline interpolacioni algoritam

Dostupno: *bilinear, bicubic*

Podrazumavano: *bilinear*

bilinear: *Bilinear* interpolacija

bicubic: *Bicubic* interpolacija

lambda_i= *float*

Tykhonov regularizacioni parametar (efekat *smoothing*)

Podrazumavano: *0.01*

column= *ime*

Ime atributske kolone iz koje će se uzimati vrednosti za aproksimaciju

Ako parametar nije definisan a ulazni podatak je 3D mapa, koristiće se z-vrednosti

solver= *ime*

Ime algoritma koji će rešiti linearni sistem jednačina

Dostupno: *cholesky, cg*

Podrazumevano: *cholesky*

maxit= *integer*

Maksimalni broj iteracija pri rešavanju sistema jednačina

Podrazumevano: *10000*

error= *float*

Kriterijum za zaustavljanje iterativnog postupka

Podrazumevano: *0.000001*

memory= *integer*

Maksimalna količina memorije koja će se koristiti za čuvanje raster mape (u MB)

Podrazumevano: *300*

Sa teoretske strane proces interpolacije se bez obzira na metod izvršava u dva koraka: prvi predstavlja računanje linearnih koeficijenata *spline* funkcije na osnovu vrednosti posmatranih tačaka; drugi korak je generisanje interpolovane površi (ili interpolovanih vektorskih tačaka). Dužina svakog *spline* koraka definisana je vrednostima dva parametra, *sie* i *sin* (korak ne mora biti isti po oba pravca). Optimalna dužina koraka ne bi smela biti manja od dužine između dve posmatrane tačke [16].

U slučaju pravilno raspoređenih tačaka po čitavom regionu, dužina *spline* koraka može uzeti vrednost maksimalne udaljenosti dve posmatrane tačke. Međutim, često su tačke nepravilno distribuirane i u tim slučajevi se zahteva statistička regularizacija ili procena. U svakom slučaju, *v.surf.bspline* će pokušati da minimizuje *bilinear spline* gradijent i *bicubic spline* krivu u onim delovima gde nedostaje ulaznih podataka (tačaka). Generalno, može se izvesti zaključak da *spline* korak ne bi trebao biti manji od srednjeg rastojanja posmatranih tačaka, čak, dva puta veće rastojanje predstavljalo bi dobro specificiran parametar. Mala vrednost *spline* koraka može u velikoj meri povećati vreme procesiranja [16].

Dobro je i znati da je i broj *spline* izračunavanja po svakom pravcu i u svakom trenutku ograničen, bez obzira na njegovu dužinu. Naime, region će se u situacijama kada postoji previše *spline* izračunavanja automatski podeliti na subregione gde će svaki imati maksimalno 150x150 *spline* koraka.

Tykhonov regularizacioni parametar (λ_i) definiše glatkost interpolacije. Ako mu se dodeli mala vrednost interpolovana površ će prolaziti bliže ulaznim podacima, dok će u drugom slučaju površ biti više glatka.

Kao što je gore i rečeno, ulazni podaci mogu biti 2D ili 3D vektor mape. Ako su u pitanju 3D mape i parametar *column* se ne specificira, onda će z-vrednosti biti korišćene za interpolaciju. Parametar *column* je obavezno specificirati kada su ulazni podaci 2D mape.

9.3. Modul *v.surf.idw* (*Inverse Distance Weight*)

Modul stvara raster matičnu strukturu sa interpolovanim vrednostima generisanim od neregularno raspoređenih tačaka vektorske mape, koristeći tehniku numeričke aproksimacije. Interpolovane vrednosti dodeljene pikselima su izračunate na osnovu vrednosti susednih tačaka ulaznih podataka i dužine između njih i posmatranog piskela. U poređenu sa drugim metodama, numerička aproksimacija dozvoljava predstavljanje kompleksih površina (naročito površina sa anomalijskim karakteristikama), isključuje prostoni uticaj bilo kakvih grešaka i generiše površ koristeći vrednosti ulaznih tačaka [17].

v.surf.idw [-n] **input**=ime [layer=string] [column=ime] **output**=ime [npoints=count] \ [power=float] [--overwrite] [--help] [--verbose] [--quiet]

Integrirani sistemi premera

Parametri:

input=ime [obavezno]

Ime ulazne vektor mape ili izvora podataka za direktan OGR pristup

layer=string

Ime lejera ili broj. Vektorske karakteristike mogu imati kategorijske vrednosti u više lejera. Ovaj broj određuje koji lejer se koristi.

Podrazumevano: 1

column=name

Ime atributske kolone sa vrednostima za interpolaciju

Ako parametar nije definisan a u ulazni podatak je 2D vektor mapa, biće korišćenje kategorijske vrednosti. Ako je ulazni podatak 3D mapa biće korišćene z-vrednosti.

output=ime [obavezno]

Ime rezultujuće raster mape

npoints=count

Broj tačaka koje će se koristiti za interpolaciju

Podrazumevano: 12

power=float

Veće vrednosti ovog parametra daju veću težinu tačkama na manjim udaljenostima od rezultujuće tačke pri interpolaciji

Podrazumevano: 2.0

Vrednosti koje ovaj modul interpoluje su vrednosti koje se specificiraju kroz *column* parametar. Ako se ovaj parametar ne definiše (nije obavezan), modul koristi *categories* kao vrednosti za interpolaciju ili z-vrednosti ako se radi o 3D mapi.

Količina memorije koju ovaj modul koristi zavisi od broja vektorskih tačaka u zadatom regionu. U slučajevima kada su tačke gusto raspoređene, može se desiti da programu „zafali“ memorije za izvršavanje interpolacije.

Važno: Vektorske karakteristike koje nisu kategorisane neće biti uzete u obzir prilikom interpolacije.

U slučajevima kada korisnik definiše mapu za masku interpolacije će se odnositi samo na tačke koje padaju unutar *mask* područja. Bez obzira na to, pri interpolaciji biće korišćene vrednosti svih tačaka koje se nalaze unutar regiona, tačke van regiona neće biti korišćene.

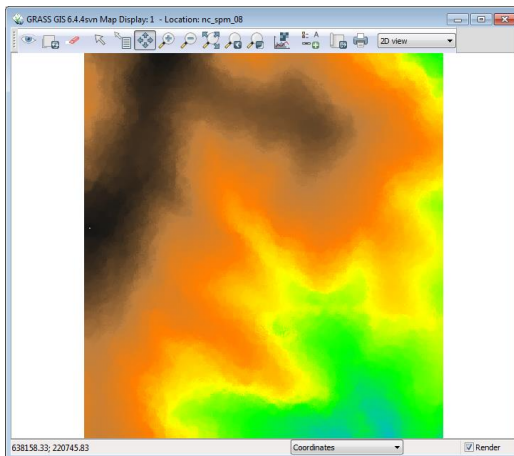
Ako unutar jedne ćelije rezultujuće raster mape pada više tačaka od broja specificiranog parametra *npoints*, tada će vrednost te ćelije biti određena kao srednja vrednost svih tačaka (sem ako je aktivna *-n flag*, tada će biti interpolovana samo tačka koja je najbliža centru).

Power parametar definiše težine za tačke koje učestuju u interpolacionom koraku. Naime veće vrednosti ovog parametra davaće veću uticaj (težinu) vrednostima tačaka koje su bliže posmatranom rezultujućem pikselu, dok će tačke koje su više udaljene imati manji uticaj na sam rezultat krajnje interpolovane vrednosti. Za vrednosti ovog parametra

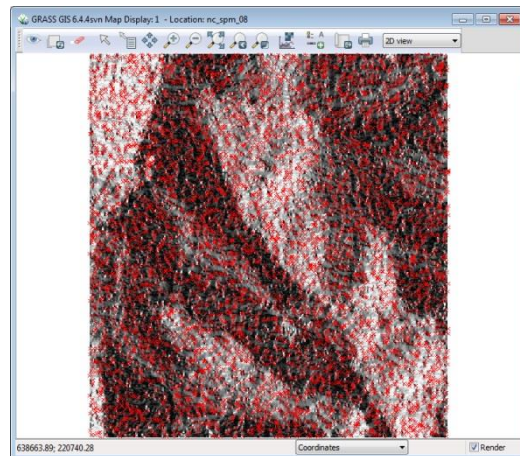
između 0-1 interpolaciona funkcija će napraviti oštre vrhove iznad datih tačaka, dok će za veće vrednosti od 1 vrhovi biti više glatki.

Parametar *npoint* definiše broj tačaka koje će se interpolovati pri svakom koraku, podrazumevan broj je 12 tačaka. Za vrednost parametra *npoint*=1, rezultat modula će biti *Voronoi* diagram. Primer ispod ilustruje korišćenje ovog modula i prikaz rezultata u 2D režimu pogleda (Slika 9.3-1) kao i „*aspect*“ mapu sa preklapljenim ulaznim vektorskim podacima (Slika 9.3-2).

```
g.region rural_1m res=2 -p
v.info elev_lid792_randpts -c
v.surf.idw elev_lid792_randpts out=elev_idw_2m col=value npoint=16 power=3
r.colors elev_idw_2m col=elevation
r.slope.aspect elev_idw_2m aspect=asp_idw_2m
d.rast asp_idw_2m
d.rast elev_idw_2m
d.rast elev_lid792_randpts
```



Slika 9.3-1: 2D prikaz



Slika 9.3-2: Aspect mapa i ulazni podaci

9.4. Modul *v.surf.rst* (Regularized Spline with Tension)

Prostorna interpolacija vektor *point* mape *spline* algoritmom. Prostorna aproksimacija i topografska analiza na bazi ulaznih podataka (tačke ili izohipse) u vektorskom formatu u cilju generisanja *floating* vrednosti raster mape [18]. Sintaksa i paraemtri:

```
v.surf.rst [-ctd] input=ime [layer=string] [zcolumn=ime] [where=sql_query] \
[elevation=ime] [slope=ime] [aspect=ime] [pcurv=ime] [tcurv=ime] [mcurv=name] \
[devi=ime] [cvdev=ime] [treeseg=ime] [overwin=ime] [mask=ime] [tension=float] \
[smooth=float] [scolumn=string] [segmax=integer] [npmin=integer] [dmin=float] \
[dmax=float] [zmult=float] [theta=float] [scalex=float] [--overwrite] [--help] \
[--verbose] [--quiet]
```

Flags:

- c *Cross-validation* procedura bez raster aproksimacije
- t Korišćenje zavisnog *tension* parametra od skale
- d Prikaz parcijalnih izvoda umesto topografskih parametara

Parametri:

input=ime [obavezno]

Ime ulazne vektor mape ili izvora podataka za direktan OGR pristup

layer=string

Ime lejera ili broj. Vektorske karakteristike mogu imati kategorijske vrednosti u više lejera. Ovaj broj određuje koji lejer se koristi.

Podrazumevano: 1

zcolumn=ime

Ime kolone atributske tabele iz koje će se koristiti vrednosti za aproksimaciju

where=sql_query

WHERE kondicioni *SQL* upit bez *where* ključne reči

elevation=name

Ime rezultujuće *elevation* raster mape

slope=ime

Ime rezultujuće *slope* mape (nagib)

aspect=ime

Ime rezultujuće *aspect* mape (pogled)

pcurv=ime

Ime rezultujuće raster mape koja prikazuje profil zakrivljenosti

tcurv=ime

Ime rezultujuće raster mape koja prikazuje tangencijalnu zakrivljenost

mcurv=ime

Ime rezultujuće raster mape koja prikazuje srednju zakrivljenost

devi=ime

Ime rezultujuće vektor *point* mape odstupanja

cvdev=ime

Ime rezultujuće vektor *point* mape sa greškama dobijenim *cross-validation* metoda

treeseq=ime

Ime rezultujuće vektor mape sa prikazom *quadtree* segmentacije

overwin=ime

Ime rezultujuće vektor mape koja prikazuje preklapajući prozor

mask=ime

Ime raster mape koja se koristi za masku

tension=float

Tension parametar

Podrazumevano: 40.

smooth=float

Smoothing parametar

scolumn=string

Ime atributske kolone sa *smoothing* parametrom

Integrirani sistemi premera

segmax=*integer*

Maksimalan broj tačaka u jednom segmentu

Podrazumevano: 40

npmin=*integer*

Minimalni broj tačaka za aproksimaciju u segmentu (>segmax)

Podrazumevano: 300

dmin=*float*

Minimalna udaljenost između tačaka (uklanjanje identičnih tačaka)

dmax=*float*

Maksimalna udaljenost između tačaka na izohipsi (ubacivanje novih tačaka)

zmult=*float*

Faktor skaliranja za vrednosti koje se koriste za apriksimaciju

Podrazumevano: 1.0

theta=*float*

Anisotropy ugao

scalex=*float*

Anisotropy faktor skaliranja

Modul *v.surf.rst* izvršava prostornu aproksimaciju baziranu na z-vrednostima (ulazni podatak je vektor 3D mapa i *zcolumn* parametar nije zadat), *categories* (ulazni podatak je vektor 2D mapa i *zcolumn* parametar nije zadat) ili atributima (specificiran parametar *zcolumn*) tačaka ili izohipsi vektorskih mapa koje predstavljaju ulazne podatke. Izlazni podaci su raster mape koje reprezentuju interpolovanu površinu [18].

Kao dodatna opcija, simultano sa aproksimacijom, mogu se računati i topografski parametri kao što su nagibi, pogledi, profili zakrivljenosti, tangencijalne zakrivljenosti definisanjem parametara *slope*, *aspect*, *pcurv*, *tcurv* respektivno.

Ulazni podaci bivaju proveravani od strane modula u smislu da se tačke koje se poklapaju ili nalaze na manjem rastojanju od specificirane vrednosti parametra *dmin*, uklanjaju. Takođe, ako se kao ulazni podaci koriste izohipse koje su previše „razređene“, modul će sračunati poziciju novih tačaka između svake dve susedne tačke na liniji, pod uslovom da je njihovo rastojanje veće od vrednosti parametra *dmax*. Parametar *zmult* predstavlja vrednost kojom se po potrebi skaliraju podaci koji se koriste za aproksimaciju (npr. konverzija dužnih jedinica za vrednosti visina koje se koriste kao podaci za generisanje površi). Podrazumevana vrednost je 1, dakle nema skaliranja [18].

O parametru *tension* i *smooth* bilo je reči i pre. Parametar *tension* određuje karakter rezultujuće površine, „od tanke ploče do membrane“. Smoothing parametar kontroliše odstupanje između ulaznih podataka i rezultujuće mape. Ako je parametar *smooth* postavljen na nulu rezultujuća površ će biti generisana kroz ili jako blizu ulaznih podataka.

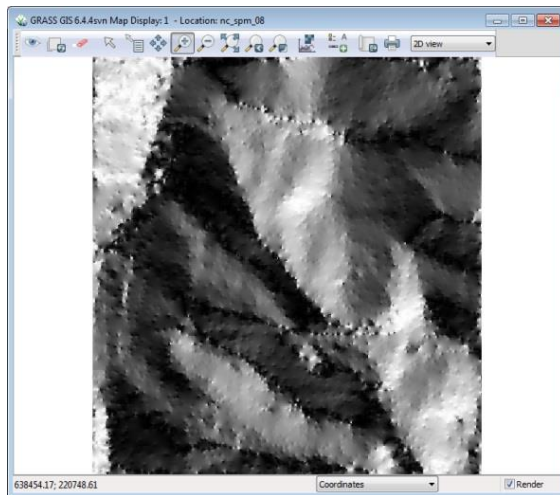
Ako je broj ulaznih podataka veći od vrednosti parametra *segmax*, procesiranje podataka će se vršiti po segmentima. Ceo region biće izdelfen na niz pravougaonika koji će sadržati manje tačaka od vrednosti parametra *segmax*, i aproksimacija će se vršiti po tako formiranim regionima. Kako bi se obezbedili dobri i glatki prelazi između segmenata,

Integrirani sistemi premera

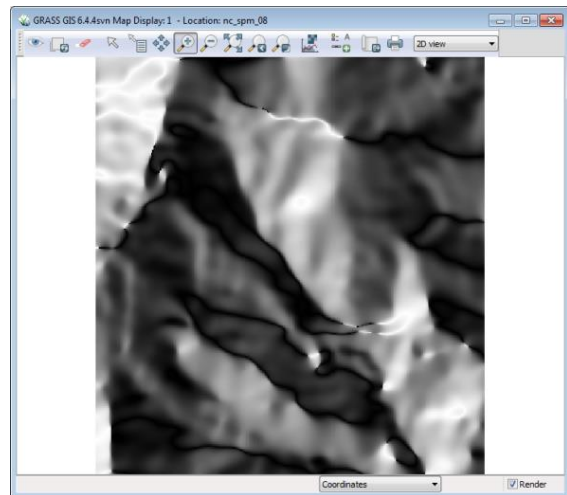
aproksimaciona funkcija će za računanja u okviru jednog segmenta koristiti ne samo tačke iz tog segmenta već i vrednosti tačaka iz susednih segmenata. Broj tačaka koje će se koristiti za aproksimaciju definisan je parametrom *npmin* i ta vrednost mora biti veće od vrednosti parametra *segmax*. Pre pokretanja ovog algoritma potrebno je specificirati rezoluciju i podesiti region za aproksimaciju koristeći modul *g.region* [18].

Primer ilustruje generisanje DMT modulom *v.surf.rst* i poređenje rezultata u zavisnosti od *tension* parametara (slike 9.4-1 i 9.4-2) i u zavisnosti od *smooth* parametra (slike 9.4-3 i 9.4-4).

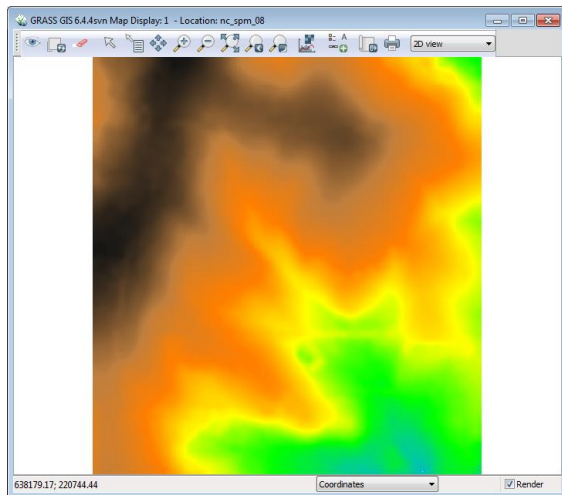
```
g.region rural_1m res=2 -p
v.surf.rst elev_lid792_randpts elev=elev_rstt10_2m asp=asp_rstt10_2m zcol=value \
ten=10 seg=30 npmin=140
v.surf.rst elev_lid792_randpts elev=elev_rstt160_2m asp=asp_rstt160_2m zcol=value \
ten=160 seg=30 npmin=140
v.surf.rst elev_lid792_randpts elev=elev_rstsm0_2m asp=asp_rstsm0_2m zcol=value \
smo=0 seg=30 npmin=140
v.surf.rst elev_lid792_randpts elev=elev_rstsm10_2m asp=asp_rstsm10_2m zcol=value \
smo=10 seg=30 npmin=140
r.colors asp_rstt10_2m col=aspect
r.colors asp_rstt160_2m col=aspect
d.rast asp_rstt10_2m
d.rast asp_rstt160_2m
d.rast asp_rstsm0_2m
d.rast asp_rstsm10_2m
```



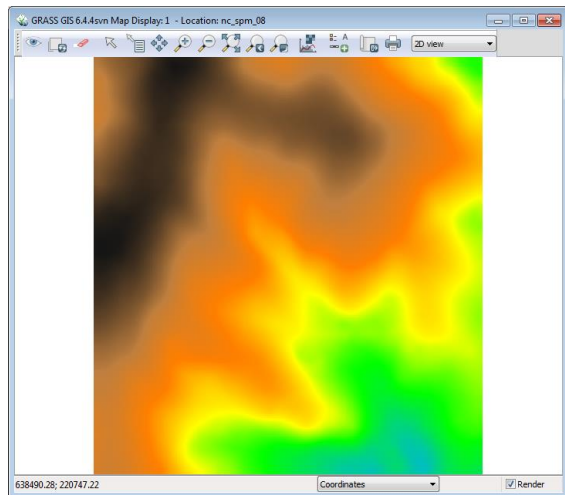
Slika 9.4-1: Parametar *tension=160*



Slika 9.4-2: Parametar *tension=10*



Slika 9.4-3: Parametar smooth=0



Slika 9.4-4: Parametar smooth=10

10. Zaključak

Digitalni model terena, kako je i napomenuto, predstavlja znatno složenu i kompleksnu strukturu prostornih podataka integriranih u digitalnom obliku. Za razliku od recimo digitalnog elevacionog modela, kroz DMT moguće je kvalitetnije sagledati prostor, integrirati više lejersku strukturu različitih podataka, kvalitetno istaći karakteristične i druge bitne linije terena, vršiti topografska izračunavanja itd. „Sirov“ kvalitet jednog DMT zavisice pre svega od kvaliteta i načina akvizije prostornih informacija. Sa druge strane, potrebno je raspolagati kvalitetnim softverom i drugom opremom koja će omogućiti kvalitetno procesiranje, poznavati metodologije, algoritme i njihov uticaja na određen tip podataka i naravno posedovati radno iskustvo. Iz svega navedenog može se izvesti zaključak da je kreiranje DMT i obezbeđivanje „dovoljno“ kvalitetnih podataka krajnje ozbiljan zadatak.

Ovaj rad je pre svega imao za cilj da ukaže na značaj softvera otvorenog koda, da pokuša da razjasni način funkcionisanja GRASS softvera, da pojasni organizaciju baze podataka i značenje fajlova koje ona integriše. Rad se dotakao vektorskih i rasterskih podataka, pojasnio način konverzije između tih modela i pokušao da na jednom mestu sumira rasterske i vektorske module bitne za kreiranje DMT. Način korišćenja modula nije detaljno razrađen niti su detaljno objašnjene funkcije i algoritmi koji su integrirani unutar modula. Cilj je jednostavno bio napraviti najosnovniju kategorizaciju, dati objašnjenje najčešće korišćenih parametara i kreirati DMT na bazi dostupnih podataka.

Literatura:

- [1] Neteler M, Mitasova H. 2008. *Open Source GIS: A GRASS GIS Approach*. 3.izd. New York: Springer Science+Business Media. 406 str.
- [2] <http://grass.osgeo.org/documentation/general-overview/> [citirano 13/5/2014]
- [3] http://www.esri.com/what-is-gis/overview#overview_panel [citirano 13/5/2014]
- [4] Neteler M, Mitasova H. 2004. *Open Source GIS: A GRASS GIS Approach*. 2.izd. Boston: Springer Science+Business Media. 401 str.
- [5] Jovanović V, Đurđev B, Srdić Z, Stankov U. 2012. *Geografski informacioni sistemi*. 1.izd. Beograd: Univerzitet Singidunum. Novi Sad: Univerzitet u Novom Sadu. 209 str.
- [6] Neteler M, Bowman Hamish M, Landa M, Metz M. 2010. *Environmental Modelling & Software*.
- [7] Borisov M. 2005. *Digitalni modeli terena za teritoriju SCG formiranih na osnovu topografskih vojnih karata*. Geodetska služba. Br. 102: 8 – 13 str.
- [8] Li Z, Zhu Q, Gold C. 2005. *DIGITAL TERRAIN MODELING Principles and Methodology*. Florida: CRC Press. 318 str.
- [9] <http://grass.osgeo.org/grass64/manuals/r.to.vect.html> [citirano 21/5/2014]
- [10] <http://grass.osgeo.org/grass64/manuals/r.contour.html> [citirano 21/5/2014]
- [11] <http://grass.osgeo.org/grass64/manuals/r.resamp.interp.html> [citirano 21/5/2014]
- [12] <http://grass.osgeo.org/grass64/manuals/r.resamp.rst.html> [citirano 21/5/2014]
- [13] <http://grass.osgeo.org/grass64/manuals/r.surf.idw.html> [citirano 21/5/2014]
- [14] <http://grass.osgeo.org/grass64/manuals/r.surf.contour.html> [citirano 21/5/2014]
- [15] <http://grass.osgeo.org/grass64/manuals/v.to.rast.html> [citirano 23/5/2014]
- [16] <http://grass.osgeo.org/grass64/manuals/v.surf.bspline.html> [citirano 23/5/2014]
- [17] <http://grass.osgeo.org/grass64/manuals/v.surf.idw.html> [citirano 23/5/2014]
- [18] <http://grass.osgeo.org/grass64/manuals/v.surf.rst.html> [citirano 23/5/2014]