

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 program Geodezija,
smer Geodezija

Kandidat:

Primož Mavsar

Izdelava 3R virtualnega modela kraške jame iz podatkov terestričnega laserskega skeniranja

Diplomska naloga št.: 669

Mentor:
doc. dr. Mojca Kosmatin Fras

Ljubljana, 1. 2. 2006

IZJAVA O AVTORSTVU

Podpisani **PRIMOŽ MAVSAR** izjavljam, da sem avtor diplomske naloge z naslovom:
**»IZDELAVA 3R VIRTUALNEGA MODELA KRAŠKE JAME IZ PODATKOV
TERESTRIČNEGA LASERSKEGA SKENIRANJA«.**

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

Ljubljana, 10.01.06

(podpis)

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK: 004:528.7(043.2)
Avtor: Primož Mavsar
Mentor: doc. dr. Mojca Kosmatin Fras
Somentor: uni. dipl. inž. Domen Smole
Naslov: Izdelava 3R virtualnega modela kraške jame iz podatkov
terestričnega laserskega skeniranja

Obseg in oprema: 55 str., 2 pregl., 31 sl., 2 en.

Ključne besede: lasersko skeniranje, model, internet, VRML

Izveček

V diplomskem delu je predstavljen princip izdelave 3R virtualnega modela kraške jame vse od zajema podatkov do končnega izdelka. Predstavljen je uporabljeni inštrument, terestrični laserski skener dolgega dosega Riegl LMS-Z420i in njegov pripadajoči računalniški program RiSCAN PRO, s katerim se krmili inštrument in nadzoruje skeniranje terena. Opisan je način skeniranja in združevanje posameznih skenov v en skupen projektni koordinatni sistem. Na kratko so podani tudi temelji 3R modeliranja in vizualizacije, osnovni princip delovanja interneta in njegove lastnosti ter opis VRML-ja. Celotna obdelava in modeliranje zajetih podatkov je potekalo v programu RiSCAN PRO. Izdelan je bil animiran model, ki služi predvsem za prezentacijo jame in njenih značilnosti. Poleg animiranega modela je bil izdelan tudi virtualni model v standardnem internetnem formatu za 3R vizualizacijo imenovanem VRML.

BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION

UDC: 004:528.7(043.2)

Author: Primož Mavsar

Supervisor: Assist. Prof. Mojca Kosmatin Fras, Ph.D.

Cosupervisor: Domen Smole, B.Sc.

Title: Generation of 3D virtual model of the karst cave from
terrestrial laser scanner data

Notes: 55 p., 2 tab., 31 fig., 2 eq.

Key words: laser scanning, model, internet, VRML

Abstract

The present work presents principle of generating the 3D virtual model of the karst cave from data acquisition to the final model. Long range terrestrial laser scanner Riegl LMS-Z420i and the enclosed software RiSCAN PRO, which is used for scanner controlling and data capture monitoring are presented as well. The method of scanning and transforming data from multiple scans into one well-defined project coordinate system is described. The work also presents fundamentals of 3D modelling and visualisation, basic principles of internet and it's features, and shortly describes VRML. The whole data processing and modelling was accomplished in RiSCAN PRO. The animated model suitable for presentations was generated. In addition to animated model, a generalised virtual model in VRML format was prepared.

ZAHVALA

Za pomoč pri nastajanju diplomske naloge se iskreno zahvaljujem mentorici doc. dr. Mojci Kosmatin Fras in somentorju uni. dipl. inž. Domnu Smoletu. Hvala tudi podjetju DFG Consulting d.o.o., ki mi je omogočilo uporabo podatkov za namen izdelave naloge.

Zahvalil bi se tudi svojim staršema, ki sta mi študij omogočila in me bodrila v najtežjih trenutkih.

KAZALO VSEBINE

1	UVOD	1
2	3R TERESTRIČNO LASERSKO SKENIRANJE	3
2.1	Sistem za 3R lasersko skeniranje Riegl LMS-Z420i	6
2.1.1	Splošno	6
2.1.2	Princip delovanja	8
2.1.3	Tehnični podatki	10
2.2	Programski paket RiSCAN PRO	11
2.3	Koordinatni sistemi in orientacija skenerja	14
2.4	Registracija podatkov	17
3	TEMELJI 3R MODELIRANJA IN VIZUALIZACIJE	20
3.1	Modeli in modeliranje	20
3.1.1	Objekti, atributi, medsebojni odnosi, operacije	21
3.2	3R vizualizacija in interakcija	23
3.2.1	Komponente scene	26
3.2.2	Točke in "prazni" poligoni	27
3.2.3	Polni poligoni	27
3.2.4	Interakcija in manipulacija	29
3.2.5	Stopnje detajla (Levels of Detail - LOD)	31
3.3	Svetovni splet (World Wide Web - WWW)	32
3.3.1	Tehnike dostopanja do podatkov na svetovnem spletu	33
3.3.2	VRML – spletni standard za 3R vizualizacijo	35
4	IZDELAVA MODELA KRAŠKE JAME	37
4.1	3R lasersko skeniranje – postopek snemanja	37
4.2	Rezultati snemanja	38
4.3	Odstranitev odvečnih točk in združitv vseh skenov v en objekt	39
4.4	Triangulacija modela – izdelava delov mreže	40

4.5	Animacija modela	46
4.6	Modifikacija modela in VRML zapisa za potrebe internetne objave	47
5	ZAKLJUČEK	53
VIRI		55

KAZALO SLIK

- Slika 1: Princip delovanja laserskega skenerja
- Slika 2: 3R laserski skener Riegel LMS-Z420i
- Slika 3: Koordinatni sistem skenerja in fotoaparata
- Slika 4: Zgradba sistema za terestrično skeniranje
- Slika 5: Programsko okno RiSCAN PRO
- Slika 6: Skenerjev lastni koordinatni sistem
- Slika 7: Transformacija koordinat med koordinatnimi sistemi
- Slika 8: Koordinatni sistemi laserskega skenerja
- Slika 9: Tarče za 3R lasersko skeniranje
- Slika 10: Refleksijska cilindrična tarča
- Slika 11: Model kot abstrakcija realnosti
- Slika 12: Abstrakcija sveta kot podatkovni model
- Slika 13: Klasična vizualizacijska shema
- Slika 14: Komponente scene
- Slika 15: Žični model z in brez uporabe algoritma za odstranjevanje nevidnih linij
- Slika 16: Klasičen odjemalec-strežnik spletni princip
- Slika 17: Merjene vezne točke
- Slika 18: Oblak točk na enem stojišču
- Slika 19: V vrsto nanizane točke in razpršen oblak točk
- Slika 20: Del območja za triangulacijo
- Slika 21: Spremenjen pogled na točke in selekcija območja primerne za triangulacijo v tem pogledu
- Slika 22: Triangulirana površina
- Slika 23: Ponovna triangulacija in odstranitev napake
- Slika 24: Napake – nepravilno triangulirana ograja, "šiv" ob stiku dveh delov modela
- Slika 25: Model brez uporabe algoritma glajenja
- Slika 26: Zglajen, končen model
- Slika 27: Model pred uporabo filtra *decimate* in rezultat uporabe filtra
- Slika 28: Razdelitev celotnega modela na dva dela
- Slika 29: VRML model brez in z uporabo *creaseAngle*

Slika 30: Model izdelan v RiSCAN PRO-ju

Slika 31: VRML model

KAZALO PREGLEDNIC

Preglednica 1: Vrste 3R terestričnih laserskih skenerjev

Preglednica 2: Posamezne faze redukcije izvozenih VRML datotek in njihove velikosti

1 UVOD

Potrebe in želje širših množic po čimbolj realnem prikazu sveta, ki jih obdaja, so pripeljale tudi do virtualnih trirazsežnih (3R) modelov sveta. To je seveda omogočila šele sodobna računalniška tehnologija z zmogljivimi procesorji in grafičnimi pospeševalniki ter visoko razvito grafično programsko opremo.

Področja uporabe teh modelov so zelo različna. Uporabni so predvsem v urbanističnem in krajinskem planiranju, medicini, strojništvu, elektro industriji, na področju reševanja, turizma in še bi lahko našteval. V primeru moje diplomske naloge gre predvsem za turistično uporabo modela. Obiskovalci jame se lahko že pred vstopom v jamo seznanijo z njenimi značilnostmi, posebnostmi in zanimivostmi, kasneje pa si jo ogledajo še v živo, pri čemer so lahko bolj pozorni na zanimive detajle, ki so jim vzbudili pozornost že pri virtualnem ogledu. Ker gre pri nalogi za študijski primer podjetja DFG Consulting d.o.o., ki mi je omogočilo uporabo njihovih zajetih podatkov, so ti podatki poslovna skrivnost podjetja in zaradi tega v nalogi ne omenjam imena konkretne kraške jame.

Izdelan primer naj bi nam podal predvsem nova znanja na področju 3R laserskega skeniranja, tako na področju zajema podatkov, kot tudi na področju obdelave in distribucije teh podatkov ter pokazal ali je taka opazovanja sploh smiselno izvajati in ali so pridobljeni izdelki tudi tržno zanimivi.

3R laserski skenerji za zajem prostorskih podatkov in programi za obdelavo tako pridobljenih 3R podatkov so že izpopolnjeni do te mere, da izdelava takega modela ne predstavlja prevelikega navora in ne zahteva posebnih znanj, če je model enostaven. V predstavljenem primeru pa gre za zelo kompleksen in obsežen model, z mnogo detajli, ki so za uporabnika tudi najbolj zanimivi. Zato je potrebno biti pri izdelavi modela predvsem potrpežljiv in natančen.

Zahtevno nalogo predstavlja tudi posredovanje in prezentacija izdelanega modela čim večjemu številu uporabnikov. To najlaže izvedemo preko interneta, ki ga uporablja vedno več ljudi. Kljub zelo zmogljivim računalnikom, tako na strani uporabnika, kot na strani strežnika,

ki te informacije posreduje uporabniku, pa preveliko število podatkov, oz. bolje rečeno prevelike datoteke formata VRML, še vedno predstavljajo problem. Te datoteke se morajo namreč v celoti shraniti na lokalni disk uporabnika, šele nato jih lahko le ta uporablja, oz. zažene in si ogleda njihovo vsebino. Prenos podatkov pa zahteva čas. Zaradi tega je potrebno model optimizirati do te mere, da se postopek lahko izpelje brez težav. To dosežemo z različnimi tehnikami izpuščanja točk v samem modelu in z dobro zgrajenim VRML modelom, kar vključuje predvsem njegovo optimizacijo zapisa in uporabo stopenj detajla.

Naloga je sestavljena iz petih poglavij, v katerih nazorno predstavljam potek izdelave modela in rešitve nekaterih problemov, ki se ob tem pojavljajo.

Po začetnem uvodu nadaljujem s principom zajema podatkov z laserskim skenerjem. Predstavljen je uporabljeni sistem Riegl LMS-Z420i, delovanje skenerja in njegovega pripadajočega programa Riscan pro.

V tretjem poglavju so predstavljeni temelji 3R modeliranja in vizualizacije. Razloženi so osnovni pojmi modeliranja, vizualizacije in delovanje interneta, vključno s kratko predstavitevijo VRML-ja, spletnega standarda za 3R vizualizacijo.

V četrtem poglavju predstavim potek izdelave konkretnega modela kraške jame, vse od poteka snemanja, obdelave podatkov, do končne predstavitvene animacije modela in optimizacijo zapisa izvozenih VRML datotek za internetno objavo.

Celotno nalogo zaključim s sklepom in predstavitevijo ideje, ki se ob nalogi poraja.

2 3R TERESTRIČNO LASERSKO SKENIRANJE

V novejšem času se na področju geodetskih opazovanj vse bolj uveljavlja metoda zajema podatkov z laserskimi skenerji, saj imajo le ti veliko prednosti pred ostalimi metodami celovitega zajema prostorskih podatkov (fotogrametrija, daljinsko zaznavanje). Te prednosti so (Jansa et al., 2005):

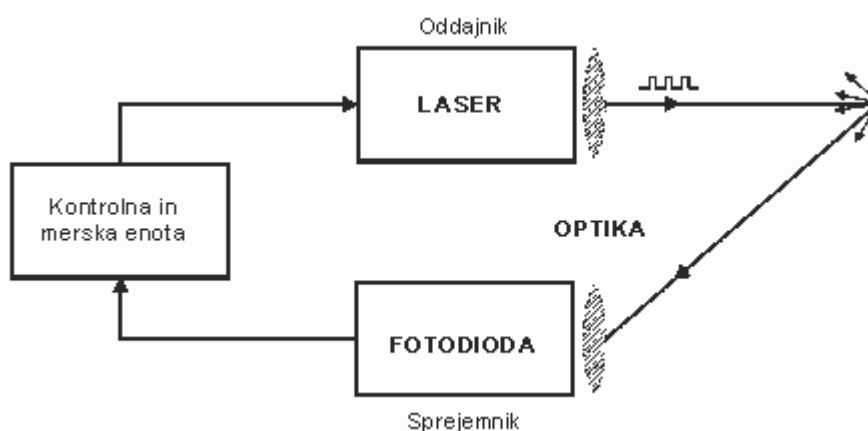
- Z laserskim skenerjem pridobimo gost oblak točk polarnih meritev z relativno visoko natančnostjo.
- Na enem samem stojišču lahko zajamemo podatke skoraj celotne vidne okolice okoli inštrumenta.
- Laserski žarek lahko do določene mere prehaja skozi vegetacijo in zaradi tega postane odbit žarek kompleksno oblikovan, časovno odvisen signal. Mnogo sodobnih inštrumentov je sposobno razbrati iz tega signala različne oblike objektov, ki se skrivajo za to vegetacijo, kot so veje ali grmovje.
- Lasersko skeniranje je aktivna tehnologija, kar pomeni, da za svoje delovanje ne potrebuje dnevne svetlobe ali kakršnekoli osvetlitve.
- Podatke pridobimo skoraj v realnem času, zato ni potrebe po dodatnem zapletenem in zamudnem delu, kot je to primer v fotogrametriji – npr. orientacija posnetkov.

Ima pa tudi nekaj slabosti, kot so:

- Omejena zmožnost zaznavanja lastnosti opazovanih površin (tekstura, barva).
- Problem zajemanja robov objektov (nenatančno merjenje razdalje).
- Manjka ustrezna topološka struktura podatkov, ki jo običajno potrebujemo za primerno uporabo izdelkov.

Te pomanjkljivosti uspešno odpravlja fotogrametrija, zato sta ti dve tehniki zajema prostorskih podatkov med seboj komplementarni. V praksi je težava odpravljena tako, da je na laserski skener pritrjen tudi visokoločljivi digitalni fotoaparati, s katerim obravnavane objekte fotografiramo. Če obe tehniki med sabo združimo, se rezultati na kritičnih mestih izboljšajo tudi do trikrat (Jansa et al., 2005).

Osnovni pogoj za izvedbo 3R laserskega skeniranja je ta, da skenirana površina odbija lasersko svetlobo. S 3R laserskimi skenerji skeniramo celotno površino objekta. To pomeni, da oddani laserski impulzi sistematično potujejo po celotnem obravnavanem objektu ter se od njega odbijajo nazaj k inštrumentu (slika 1), kjer se na osnovi časovnega intervala potovanja signala od oddajnika do sprejemnika ali meritve faznih razlik ali trikotniške mreže, izračuna razdalja med sprejemnikom in obravnavano točko. Metoda izračuna razdalje je odvisna od hitrosti oddajanja laserskih impulzov (preglednica 1).








Slika 1: Princip delovanja laserskega skenerja (Wagner, 2004)

Skeniranje je izvedeno z enim ali dvema gibljivima zrcaloma, ki omogočata zelo majhne spremembe v odklonskem kotu laserskega žarka, projiciranega na površino objekta (slika 4). Za končni izračun prostorskih koordinat posameznih točk potrebujemo podatek o odklonskem kotu zrcal in poševni razdalji med 3R laserskim skenerjem in obravnavano točko na objektu.

Glede na doseg skenerja in s tem posledično tudi na doseženo natančnost koordinat točk, v osnovi ločimo tri tipe 3R terestričnih skenerjev (Dorninger, 2005) (preglednica 1): skenerji dolgega dosega, skenerji srednjega dosega in skenerji kratkega dosega.

Preglednica 1: Vrste 3R terestričnih laserskih skenerjev

doseg		natančnost	metoda meritev		
			časovni interval potovanja signala (~ 10.000 točk/sek)	Fazne razlike (> 100.000 točk/sek)	Triangle mesh (~ 100.000 točk/sek)
Dolgi doseg	2–800 m	±10/ ±5 mm	<i>Riegl LMS- Z420i</i> - dolžina (d) - intenziteta (i) - RGB – tekstura (t) (11 MPix)		
	1–100 m	±4 / ±2 mm	<i>Leica HDS 3000</i> - d, i, t (1MPix)		
Srednji doseg	0.1–53 m	±7 / ±2 mm		<i>Leica HDS 4500</i> - d, i	
	0.1–80 m	±3/ ±0.5mm		<i>Faro LE 880 HE</i> - d, i, t (6 MPix)	
Kratki doseg	0.6-1.2 m	±0.1 mm			<i>Konica Minolta VIVID910</i> -d, t (11 MPix)

Natančnost določitve prostorske lokacije skenirane točke je odvisna od natančnosti določitve razdalje in natančnosti kotnih meritev.

S pomočjo preglednice 1 se lahko odločimo, kateri inštrument bomo uporabili, da bo izpolnil naše zahteve glede natančnosti in dosega. Seveda preglednica ne vsebuje vseh inštrumentov, ki jih trenutno ponuja trg, ampak samo tipične predstavnike posamezne vrste skenerjev. Kljub temu pa nam poda osnovno informacijo o tem, kaj lahko od posamezne vrste skenerjev pričakujemo. V primeru moje diplomske naloge nismo potrebovali najvišje natančnosti meritev ampak predvsem dolgi doseg. Zato je bil izbran inštrument Riegl LMS-Z420i, ki je v nadaljevanju tudi podrobneje opisan.

2.1 Sistem za 3R lasersko skeniranje Riegl LMS-Z420i

2.1.1 Splošno

Sistem za terestrično lasersko skeniranje Riegl LMS-Z420i sestavlja visoko zmogljiv, 3R laserski skener dolgega dosega, pripadajoči operacijski in procesni računalniški program RiSCAN PRO ter kalibriran visokoločljivi digitalni fotoaparati.



Slika 2: 3R laserski skener Riegl LMS-Z420i
(Riegl laser measurement systems – spletna stran, 15.09.2005)

Sistem zagotavlja podatke iz katerih lahko avtomatsko ali polatomatsko pridobimo izdelke, kot so teksturirane triangulirane površine ali ortofoto z globinsko informacijo.

Riegl LMS-Z420i je robusten in popolnoma prenosljiv senzor, oblikovan posebej za hitro pridobivanje visoko kakovostnih, tri razsežnih podatkov, tudi v najzahtevnejših pogojih. Zagotavlja dobro kombinacijo širokega vidnega polja, dolgega maksimalnega dosega in hitrega zajema podatkov.

Standardni prenosni računalnik in programski paket RiSCAN PRO omogočata uporabniku takojšnje pridobivanje 3R podatkov na terenu in zagotavljata različne načine registracije, naknadnega procesiranja in funkcije za izvoz teh podatkov.

S kombinacijo visoko zmogljivega laserskega senzorja in visokoločljivega digitalnega fotoaparata dosežemo kombinacijo prednosti laserskega skeniranja in digitalne fotogrametrije. Sam fotoaparat se pričvrsti na vrh inštrumenta in z njim lahko posnamemo celotno območje, ki ga pokriva tudi laserski senzor. Ker sta pozicija in orientacija fotoaparata glede na skenerjev koordinatni sistem dobro definirana, vsaka slika predstavlja kalibriran in registriran posnetek. Te posnetke lahko uporabimo za določitev barve vsakega verteksa skeniranih podatkov ali pa jih uporabimo kot visoko ločljive teksture na trianguliranih površinah izdelanih iz skenov.

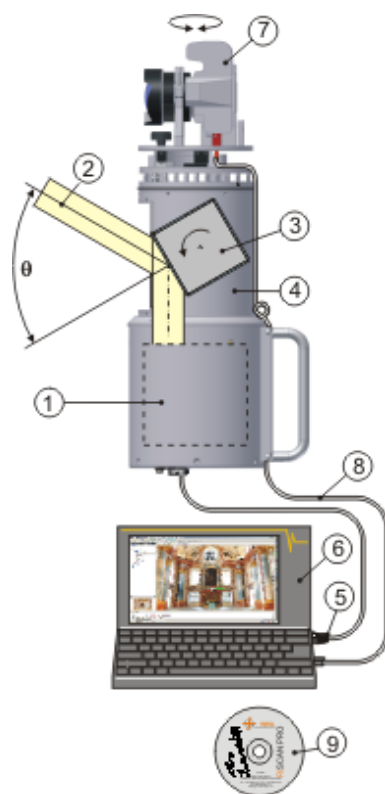


Slika 3: Koordinatni sistem skenerja in fotoaparata
(Riegl laser measurement systems – spletna stran, 15.09.2005)

V mojem primeru fotoaparata nismo uporabljali, zato tudi ni podrobneje predstavljen.

2.1.2 Princip delovanja

Na sliki 4 je prikazan sistem za terestrično skeniranje, kjer so posamezni elementi označeni s številko. **Elektronika za iskanje oddaljenosti (1)** (angl. **range finder electronics**) 3R skenerja je optimizirana tako, da skener zagotavlja visoko hitrost skeniranja. To pomeni hitro ponavljajoče se lasersko oddajanje (angl. fast laser repetition rate), hitro procesiranje signala in hiter podatkovni vmesnik (angl. high speed data interface).



Slika 4: Zgradba sistema za terestrično skeniranje
(Riegl laser measurement systems – spletna stran, 15.09.2005)

Vertikalni odklon **laserskega žarka (2)** pri linijskem skeniranju je realiziran s **poligonom (3)** s številnimi refleksijskimi površinami. Za visoke hitrosti skeniranja in / ali za vertikalni kot skeniranja od 0° do 80° , poligonalno ogledalo zvezno rotira z ustrezno hitrostjo. Pri počasnem skeniranju (angl. slow scanning rates) in / ali pri majhnih kotih skeniranja pa ogledalo linearno niha navzgor in navzdol. Horizontalno skeniranje je zagotovljeno z rotacijo celotne **optične glave (4)** za 360° .

Skenirani podatki: dolžina, kot in amplituda signala, se prenesejo v **računalnik (6)** preko **TCP/IP Ethernet vmesnika (5)**. Podatki **fotoaparata (7)** pa se prenašajo preko **USB/fireware vmesnika (8)**.

RiSCAN PRO software (9) omogoča operaterju izvajanje velikega števila operacij vključno s konfiguracijo senzorja, zajem podatkov, vizualizacijo podatkov, manipuliranje s podatki in njihovo arhiviranje. Deluje na operacijskih sistemih WINDOWS XP, 2000 SP2 ali NT SP4.

2.1.3 Tehnični podatki

Podatki o razdaljemeru

Zaščitni razred za oči po IEC60825-1:2001	razred 1
Meritveni doseg:	
Za naravne tarče, $\rho \geq 80\%$	do 1000 m
Za naravne tarče, $\rho \geq 10\%$	do 350 m
Minimalni doseg	2 m
Natačnost meritev	± 10 mm (ena meritev) ± 5 mm (povprečno)
Ločljivost	5 mm
Hitrost meritev	Do 12000 točk/s – nižja natančnost Do 8000 točk/s – višja natančnost
Valovna dolžina laserja	Blizu infra rdeče
Divergenca žarka	0.25 mrad

Podatki o skenerju

Vertikalno skeniranje

Območje skeniranja	0° do 80°
Mehanizem skeniranja	oscilirajoče/ rotirajoče ogledalo
Hitrost skeniranja	1 sken/s do 20 skenov/s pri 80°
Minimalni kotni hod	0.008°
Kotna natančnost	0.002°
Inklinacijski senzor	opcijsko
Interni sinhronizacijski časovni merilnik za sinhronizacijo z zunanjim GPS/INS sistemom	opcijsko

Horizontalno snemanje

Območje skeniranja	0° do 360°
Mehanizem skeniranja	rotirajoča optična glava
Hitrost skeniranja	0.01° /s do 15° /s
Minimalni kotni hod	0.01°

Kotna natančnost 0.0025°

Splošni tehnični podatki

Vmesnik:

- za konfiguracijo in izhod podatkov Ethernet TCP/IP, 10/100 MBit/sec
- za konfiguracijo RS232, 19.2 kBd
- za izhod podatkov ECP standard, paralelni

Napajanje 12 - 28 V DC

Poraba energije 72 W max. 80 W

Poraba toka:

- pri 12 V DC 6 A max. 6.6 A
- pri 24 V DC 3 A max. 3.3 A

Dimenzije 463 x 210 mm

Teža približno 14.5 kg

Temperaturno območje 0°C do + 40°C (operativno)

-10°C do +50°C (hranjenje)

Zaščitni razred IP64, odporen proti prahu in pljuskom vode

Še več o tem se nahaja na spletni strani http://www.riegl.com/terrestrial_scanners/lms-z420i_z420i_all.htm.

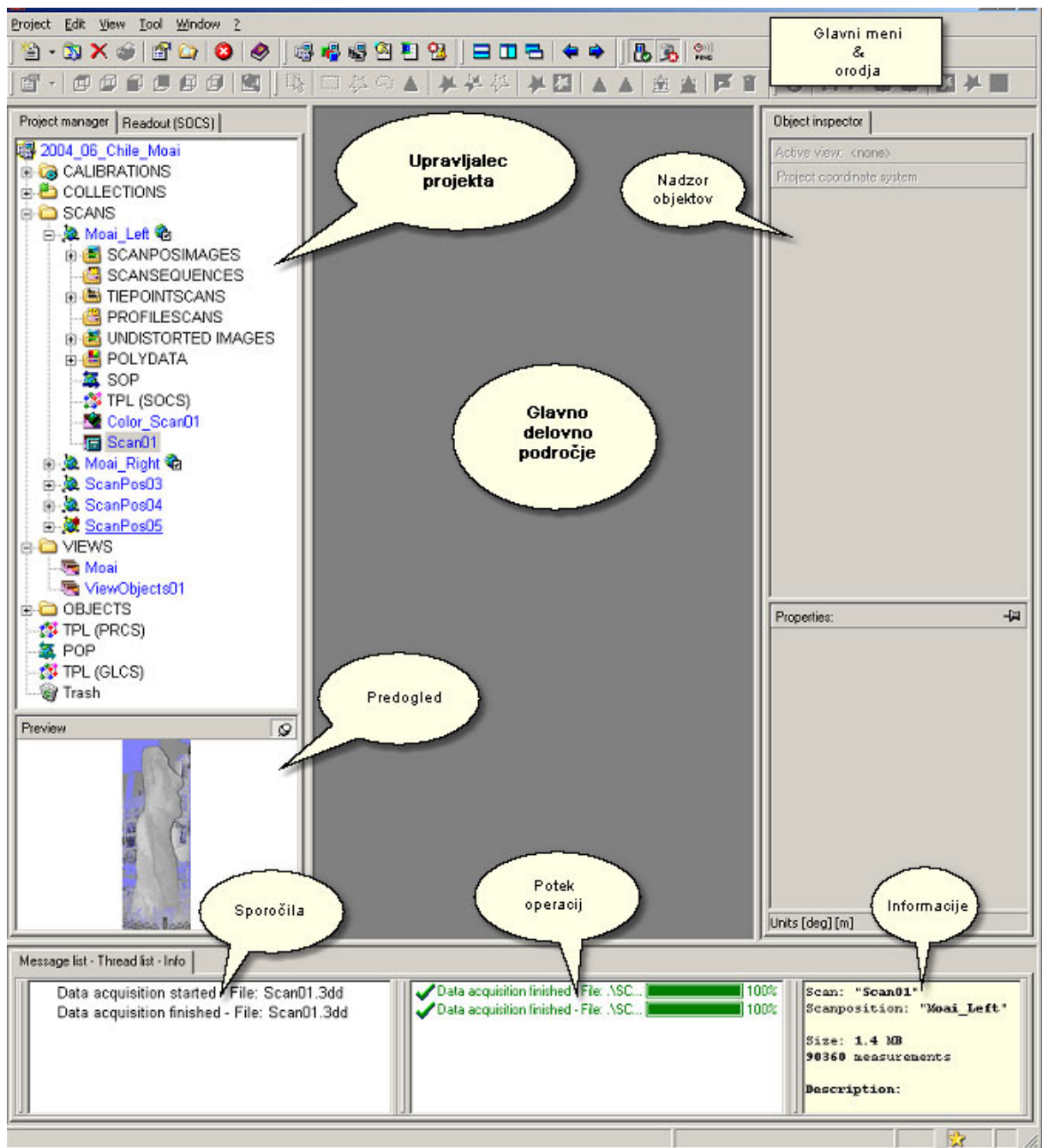
2.2 Programski paket RiSCAN PRO

Tako upravljanje skenerja kot tudi kasnejša obdelava surovih podatkov poteka v celoti, to je do končne animacije, v programskem paketu RiSCAN PRO, ki je sestavni del uporabljenega sistema Riegl LMS-Z420i. Osnovni paket vsebuje vse potrebne funkcije za izvedbo triangulacije zajetega oblaka točk, za določitev barvnih atributov vsaki posamezni laserski meritvi, za izdelavo združenih in nepopačenih visokoločljivih digitalnih slik za teksturo triangulirane površine modela, redukcijo točk v oblaku točk (ang. *decimate*), konstrukcijo objektov iz oblaka točk in še veliko več (glej <http://www.riegl.com>).

Aplikacija je projektno orientirana, kar pomeni, da so vsi podatki projekta shranjeni znotraj ene datotečne strukture. Ta vsebuje vse skenirane podatke, koordinate kontrolnih in veznih točk, kalibrirane fotografije, informacije o registraciji, dodatne opise in procesirane izhode ter vse potrebne transformacijske matrike za transformacijo podatkov iz večih skenov, to je celoten sklop podatkov zajetih iz enega stojišča, v en skupen, dobro definiran koordinatni sistem.

Taka struktura omogoča različnim uporabnikom direkten dostop do vseh koristnih podatkov znotraj projekta. Struktura projekta je shranjena na tekstovni bazi in dokumentirani projektne datoteke, kar omogoča uporabo XML jezika. Ime projektne datoteke je "project.rsp". Znotraj programa so zaradi lahke dostopnosti in jasnosti vsi podatki organizirani v drevesni strukturi.

Na sliki 5 je predstavljeno glavno programsko okno in njegovi deli.



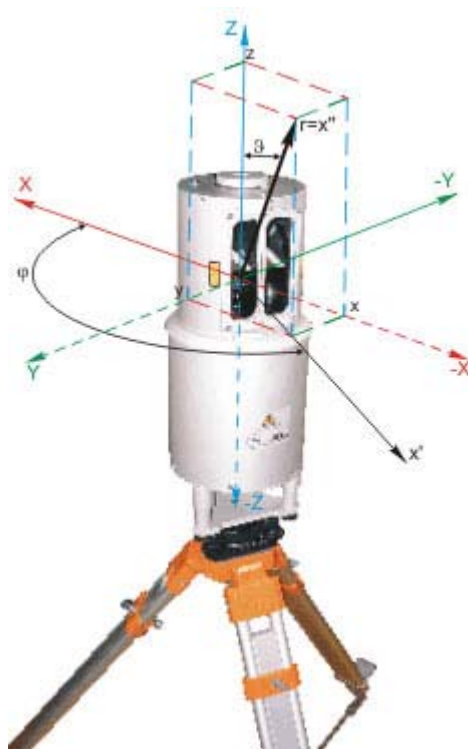
Slika 5: Programsko okno RiSCAN PRO

Program je zasnovan tako, da zmanjša porabljeni čas na terenu. Uporabniku zagotavlja vsa potrebna orodja za vizualni nadzor popolnosti zajetih podatkov v treh dimenzijah že na terenu. Avtomatska zaznava refleksijskih tarč mu namreč omogoča enostavno in takojšnjo združitev vseh skeniranih podatkov v en skupen, v naprej definiran koordinatni sistem.

2.3 Koordinatni sistemi in orientacija skenerja

Sistem uporablja več različnih koordinatnih sistemov, s pomočjo katerih lahko združujemo posamezne skene iz različnih stojišč v celoto in umeščamo posnete objekte v prostor. Tu so opisani najpomembnejši.

Skenerjev lastni koordinatni sistem (SLKS) je koordinatni sistem v katerem skener pridobi surove podatke. Podatki vsake laserske meritve vsebujejo geometrijske informacije (kartezične koordinate x, y, z ali polarne koordinate r, ϑ, φ) in dodatne opise (vsaj intenziteta, opcijsko tudi barve). Tako lahko izhod Rieglovega 3R laserskega senzorja opišemo kot (organiziran) oblak točk z dodatnimi opisi verteksov v lastnem skenerjevem koordinatnem sistemu.



Slika 6: Skenerjev lastni koordinatni sistem

(Riegl laser measurement systems – spletna stran, 15.09.2005)

Projektni koordinatni sistem (PRKS) je koordinatni sistem, ki je definiran s strani uporabnika in ga ponavadi predstavlja kar že obstoječi skenerjev koordinatni sistem prvega skena. Glede nanj se nato transformirajo vsi ostali SLKS v en skupen PRKS. Je pomožni koordinatni sistem. RiSCAN PRO zahteva, da so vsi geometrijski podatki znotraj tega

projektnege koordinatnega sistema predstavljeni z enotno natančnostjo (7 pomembnih števil). Na primer, če je zahtevana mm natančnost, maksimalna koordinata ne sme biti večja od 1 km.

Globalni koordinatni sistem (GLKS) je koordinatni sistem v katerega je umeščen projektni koordinatni sistem. Torej je to referenčni koordinatni sistem, npr. Gauss-Krügerjev koordinatni sistem. Ponavadi koordinate v globalnem koordinatnem sistemu vsebujejo zelo velika števila.

Koordinatni sistem fotoaparata (KSF) se nanaša na fotoaparatus, ki je opcijsko postavljen na vrh sistema za skeniranje in zagotavlja visokoločljive posnetke.

V skoraj vseh aplikacijah laserskega skeniranja zajem podatkov temelji na skeniranju objekta iz različnih lokacij z namenom, da se pridobi celoten sklop podatkov objektove površine brez lukenj oz. "skenerjevih senc". Različne lokacije skenerja se imenujejo pozicije skenerja. Preden začnemo s skeniranjem iz novega stojišča, moramo določiti začetno pozicijo skeniranja (po tovarniških nastavitvah ScanPos01 – sken iz prvega stojišča), šele nato lahko začnemo s skeniranjem. Na to pozicijo se bodo nato nanašali vsi zajeti podatki iz te lokacije.

Pozicija skena je določena z njegovim lastnim koordinatnim sistemom (SLKS), to je s pozicijo in orientacijo skenerja v projektne koordinatnem sistemu. Pozicijo in orientacijo lahko v splošnem določimo s šestimi parametri (3 za pozicijo in 3 za rotacijo) ali s transformacijsko matriko. RiSCAN PRO uporablja 4x4 transformacijske matrike imenovane SOP informacija (SOP – senzorjeva orientacija in pozicija).

$$M_{SOP} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Matriko sestavlja 9 elementov, ki predstavljajo rotacijo okoli osi (r_{11} do r_{33}) in 3 parametri translacije (t_1 do t_3). Uporaba homogenih koordinat dopušča enostavno izračunavanje rotacije in translacije z uporabo ene t.i. transformacijske matrike. Translacijski vektor je pozicija skenerja, vektorji po stolpcih (r_{1i} , r_{2i} , r_{3i}) pa so smeri lokalnih koordinatnih osi v PRKS. Točko P v prostoru zapisano s homogenimi koordinatami predstavljajo njene 3R koordinate x, y in z. Torej:

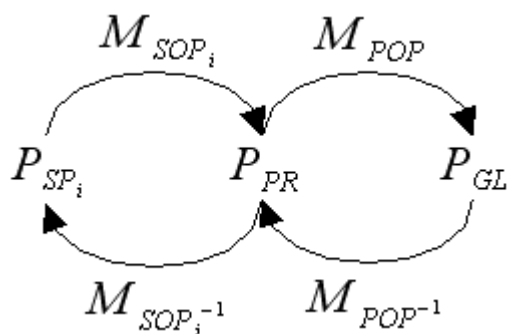
$$P_{\text{hom}} = (xyz1)^T \quad (2)$$

Opomba:

Sprememba skenerjeve orientacije na določeni lokaciji zahteva uporabo nove pozicije skeniranja (ang. scan position), čeprav sama skenerjeva lokacija ni bila spremenjena.

Vsaka pozicija skena vsebuje podatke pridobljene s te pozicije, shranjene v skenerjevem binarnem podatkovnem formatu s končnico **.3DD**. Nadalje, vsaka pozicija skena vsebuje svoje SOP informacije. Pretvorba podatkov iz SLKS v projektni koordinatni sistem se izvede enostavno z množenjem s SOP matriko pozicije skena.

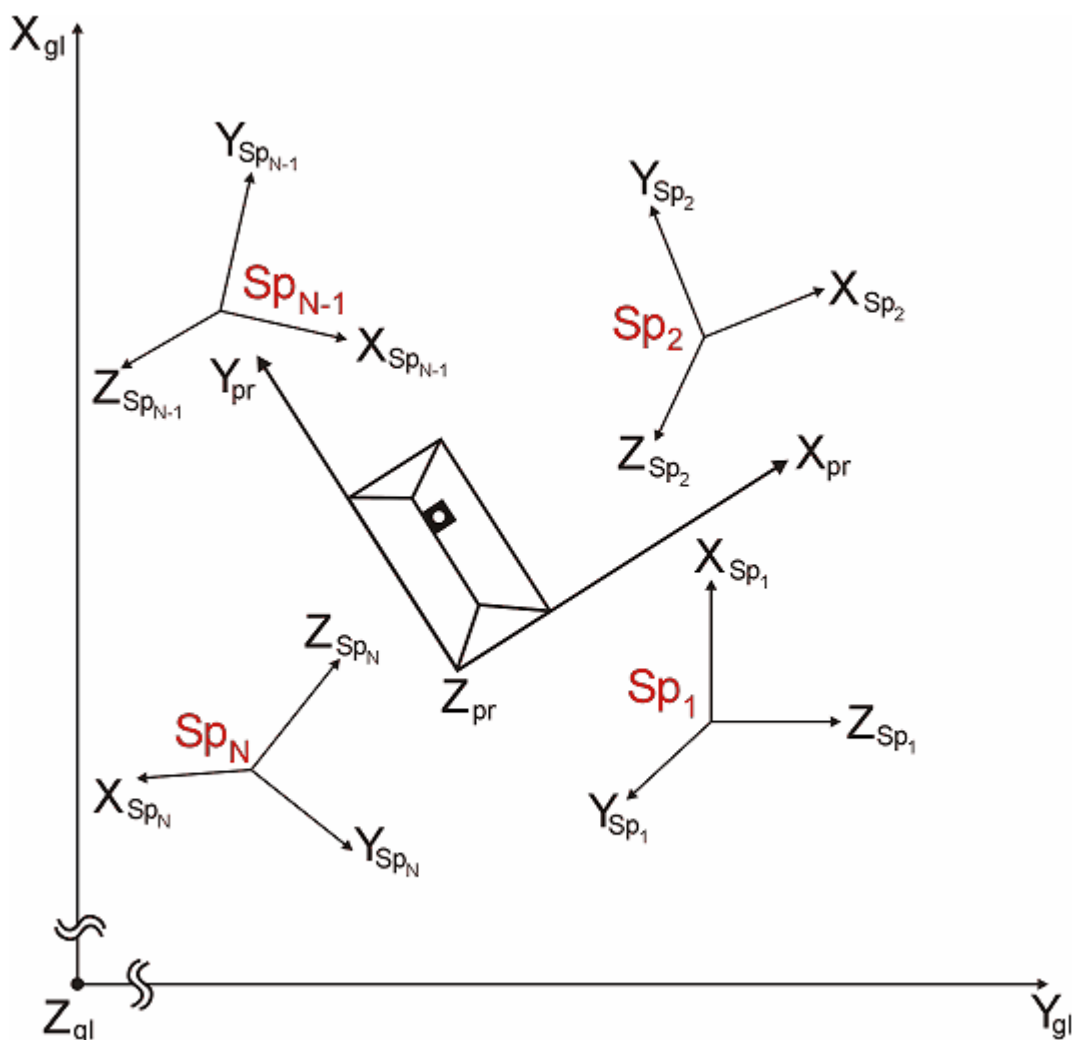
V primeru, da potrebujemo točko P transformirano iz določene pozicije skena v globalni koordinatni sistem, najprej njeno lokacijo množimo s M_{SOP} matriko, da pridobimo projektne koordinate, nato pa le te še enkrat pomnožimo z matriko M_{POP} , ki te koordinate dokončno transformira v globalni koordinatni sistem.



Slika 7: Transformacija koordinat med koordinatnimi sistemi (RiSCAN PRO - pomoč)

Spodnja skica kaže primer koordinatnih sistemov GLKS, PRKS in SLKS. Objekt predstavlja zgradba v tlorisnem pogledu. Projektni koordinatni sistem je definiran z Y_{pr} osjo, vzporedno s

steno, izhodišče pa sovpada z vogalom zgradbe. PRKS mora biti desnosučen sistem. Različne pozicije skenov so označene s S_{pi} , to je tam, kjer je bil postavljen skener. Vsaka pozicija skena ima svoj lasten koordinatni sistem (SLKS) s koordinatnimi osemi X_{spi} , Y_{spi} in Z_{spi} .



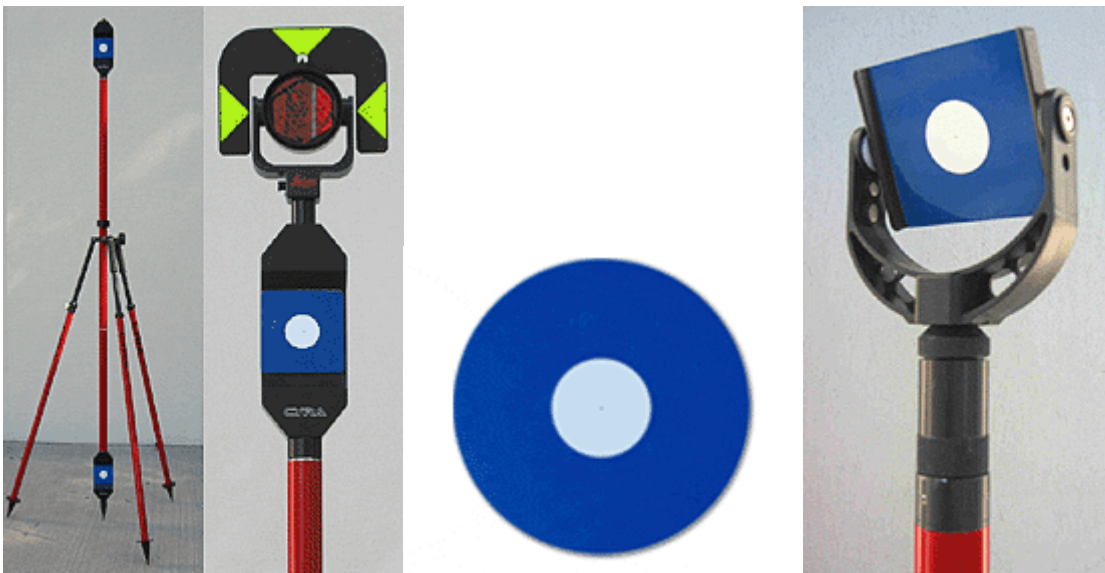
Slika 8: Koordinatni sistemi laserskega skenerja (RiSCAN PRO - pomoč)

2.4 Registracija podatkov

Termin registracija podatkov pomeni združitev in izravnavo podatkov iz različnih skenov tako, da dobimo ploskev opisano z enim samim oblakom točk. Torej pomeni to transformacijo vseh zajetih točk iz skenerjevih lastnih koordinatnih sistemov v en skupen projektni koordinatni sistem. Proces registracije skenov iz različnih pozicij matematično pomeni določitev matrike M_{SOP} .

Registracijo skenov lahko izvedemo na več načinov. Najpogosteje uporabljena metoda je z veznimi točkami. Obstaja pa še registracija s pomočjo inklinacijskih senzorjev inštrumenta, približna ročna registracija ter urez nazaj (ang. backsighting).

V mojem primeru je bil proces izveden z veznimi točkami, ki so zbrane in urejene v programu v t.i. zbirki veznih točk (ang. tiepoint lists). Vezne točke so ponavadi označene s posebnimi reflektorji, retro ali refleksijskimi tarčami, ki se dobro razlikujejo od ostalih podatkov na skenu. Njihova natančna pozicija se enostavno lahko določi z geodetskimi metodami. Vezno točko navadno predstavlja center tarče.



Slika 9: Tarče za 3R lasersko skeniranje (Leica geosystems – spletna stran, 16.10.2005)



Slika 10: Refleksijska cilindrična tarča
(Riegl laser measurement systems – spletna stran, 15.09.2005)

Vezne točke vključimo v zbirko veznih točk s posebnim programskim modulom, ki avtomatsko prepozna vse vidne tarče na vseh skenih. Ko imamo na voljo zadostno količino veznih točk, program izračuna transformacijsko matriko M_{SOP} , s katero se lahko nato vsi skeni pretvorijo, oz. združijo v en skupen projektni koordinatni sistem. Če imamo na voljo tudi koordinate veznih točk v globalnem koordinatnem sistemu, lahko objekt umestimo v prostor. Oblake združimo na podlagi najmanj treh identičnih veznih točk. Kadar koordinat veznih točk nismo merili v globalnem koordinatnem sistemu, izračunamo transformacijske parametre navadno glede na prvi sken v seriji, ki tako predstavlja naš projektni koordinatni sistem.

Navadno vse podatke obdelujemo v projektnem koordinatnem sistemu, torej obdelujemo vse zajete podatke naenkrat.

3 TEMELJI 3R MODELIRANJA IN VIZUALIZACIJE

V tem poglavju je podan splošen pregled koncepta podatkovnega modeliranja, računalniške grafike in spletnih tehnik. Poglavje je razdeljeno na tri dele. V prvem je opisan termin model, definirane so njegove komponente in posamezne faze, ki vodijo do nekega funkcionalnega sistema. Drugi del je posvečen tehnikam vizualizacije in ima namen razjasniti terminologijo, osnovne vizualizacijske mehanizme in komponente, ki sestavljajo realistično podobo. V zadnjem delu pa so predstavljene splošne metode dostopa, predstavitve in interakcije s podatki preko svetovnega spleta (ang. web-a), kamor sodi tudi VRML ter njegova vloga pri procesu vizualizacije.

3.1 Modeli in modeliranje

Termin model je eden izmed najbolj uporabljanih besed v mnogih vedah. Z njegovo pomočjo se postavljajo različne hipoteze, določajo predpostavke, izmenjujejo ideje in pridobivajo določena znanja. Modeli se uporabljajo za predstavitev določenega pojava na način, ki ga razumejo tudi drugi.

Glede na različne poglede sveta okoli nas obstaja več različnih vrst modelov. Ena izmed razdelitev modelov (Zlatanova, 2000) se nanaša glede na pojav, ki ga obravnavamo. Tako ločimo *realne* in *imaginarne* modele. V primeru realnih modelov moramo najprej pridobiti določene podatke o realnem objektu, ki ga obravnavamo. Opraviti moramo določene meritve, poizvedovnja, opazovanja in jih uporabiti pri izdelavi modela - *modeliranju*. V drugem primeru pa morajo biti lastnosti modela načrtovane (projektirane, izračunane, določene) s strani uporabnika. Jasno ločnico med obema vrstama modelov je težko postaviti, saj imamo v veliko primerih tako nezadostno količino merjenih podatkov, da je nujno "ročno" oblikovanje za predstavitev določenega objekta oz. pojava.

Druga kategorizacija modelov pa odraža njihovo formo. To so digitalni in analogni modeli (npr. klasične karte). Modele, ki so razumljivi računalniku, imenujemo *podatkovni modeli*. V splošnem so podatkovni modeli enostavnejši kot večina znanstvenih modelov.

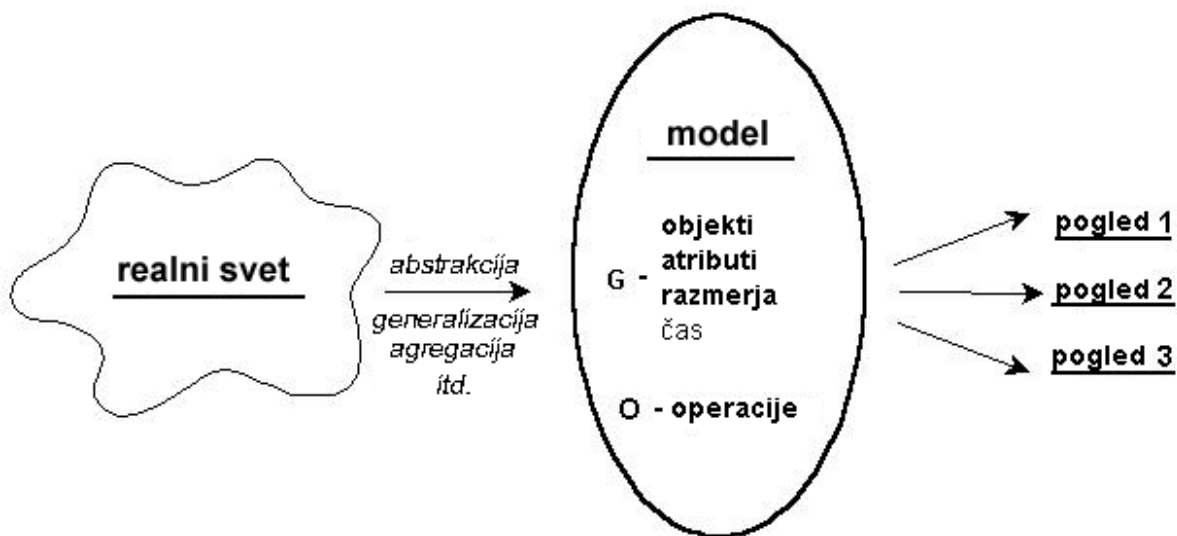
3.1.1 Objekti, atributi, medsebojni odnosi, operacije

Ena od striktnih definicij podatkovnega modela se glasi (Zlatanova, 2000): Podatkovni model, označimo ga z M, je orodje za prezentacijo sveta, ki je sestavljeno iz pravil generiranja G in operacij O. Torej lahko zapišemo v simbolih M(G,O) (slika 11). Pravila generiranja sestavljata dva sklopa specifikacij:

- strukturne specifikacije (GS),
- omejitvene specifikacije (GC).

Torej $G(G_s, G_c)$.

Strukturne specifikacije narekujejo tip in organizacijo podatkov, medtem ko omejitvene specifikacije določajo mogoče omejitve modela. Tako je na primer realna zgradba predstavljena v modelu kot sklop ravnih, pravokotnih površin (strukturna specifikacija). Te površine pa se morajo sekati samo vzdolž robov (omejitvena specifikacija). Pravila generiranja upravljajo s podatki, kar v splošnem predstavlja kategorije, njihove lastnosti in medsebojne odnose v nekem določenem časovnem trenutku.



slika 11: Model kot abstrakcija realnosti (Zlatanova, 2000)

Operacije opisujejo vse akcije, ki so lahko izvedene s podatki. Nekateri modeli omogočajo za svojo manipulacijo standardne poizvedovalne jezike, kot je npr. SQL. Tradicionalno ločimo pet osnovnih operacij:

- *nastavi vrednost* (ang. *set currency*) – vzpostavi logično pozicijo v prezentaciji modela,
- *povrni stanje* (ang. *retrieve*) – omogoči uporabniku dostopnost nekaterih podatkov,
- *vrini* (ang. *insert*) – dodaj nov podatek,
- *briši* (ang. *delete*) – odstrani podatek iz podatkovne baze,
- *posodobi* (ang. *update*) – spremeni obstoječi podatek.

Splošne podporne operacije so *selekcija*, *navigacija* in *specializacija*.

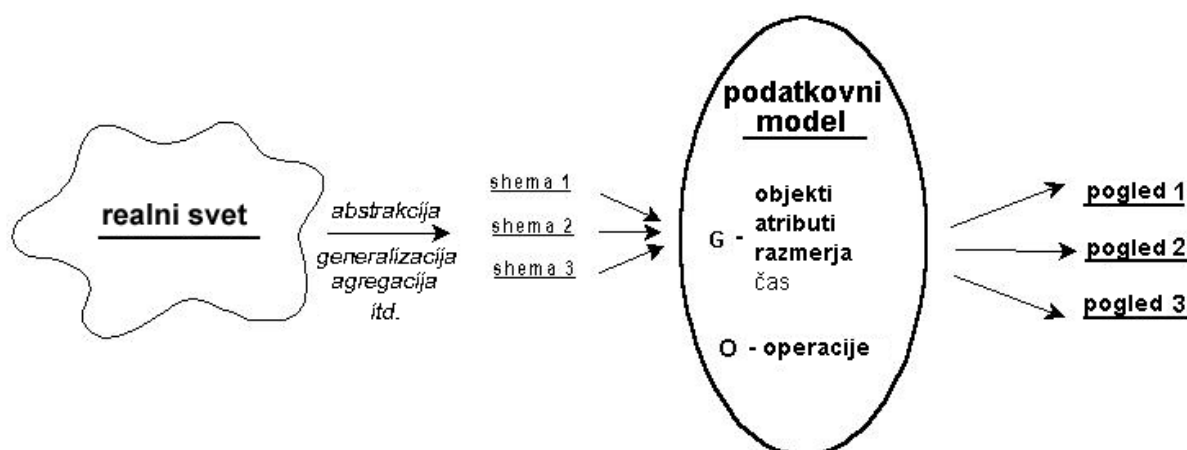
Selekcija (ang. *selection*) omogoča identifikacijo velikega števila podatkov na podlagi treh lastnosti: logične pozicije (prvi, zadnji, prioritetni), vrednosti ali medsebojnih odnosov. Na primer, uporabnik hoče videti prvih 10 (glede na identifikacijsko številko) stavb.

Navigacija (ang. *navigation*) je operacija, ki določi na podlagi selekcije logično pot, ki ji sledimo.

Specializacija (ang. *specialization*) je kompleksna operacija, ki omogoča ustvarjanje novega objekta (kategorije) na podlagi že obstoječih. Na primer uporabnik hoče ustvariti nek konglomerat imenovan "trgovski center", ki je sestavljen iz več zgradb (objektov). V nekaterih prostorskih modelih bi bila operacija izvedena z *Eulerjevimi operatorji* za dodajanje in odstranjevanje elementov iz prostorske baze podatkov na tak način, da ostane struktura mnogovrstna (raznolična). Tipični primer, ki se nanaša na relacijski podatkovni model, je kreiranje nove tabele na podlagi že obstoječih. Nato je z operacijami *unija*, *preseka*, *brisanje*, izvedena operacija *specializacija*. V resnici lahko uporabnik modela definira poljubno število operacij, kolikor jih pač potrebuje za določeno aplikacijo.

Razvitih je bilo veliko modelov na različnih področjih, glede na prezentacijo sveta, pravil generiranja in operacij. Nekateri modeli zajemajo zelo abstraktne pojme in zajemajo vse tipe podatkov, medtem ko so drugi bolj aplikativno usmerjeni in operirajo s specifičnimi vrstami podatkov (npr. točke, linije, površine v prostorskem podatkovnem modelu). Največja prednost splošnih podatkovnih modelov je v tem, da so neodvisni od aplikacij tako strukturno kot operacijsko. Torej lahko zgrajen sistem podatkov služi katerikoli aplikaciji. Aplikacijski

model je definiran s pravili splošnega modela v *shemi* (slika 12). Med vsemi splošnimi modeli so se samo trije uveljavili v širši uporabi. To so *relacijski*, *mrežni* in *hierarhični* podatkovni modeli. Ti modeli so produkt logičnega dizajna in se včasih opredelijo tudi kot logični modeli.



slika 12: Abstrakcija sveta kot podatkovni model (Zlatanova, 2000)

3.2 3R vizualizacija in interakcija

Vizualizacija je splošni pojem za opredelitev procesa ekstrakcije podatkov iz modela in njihovo predstavitev na ekranu. Ta naloga se nanaša na prezentacijo prostorskih podatkov in bo zato vizualizacija največkrat uporabljena v kontekstu s prikazom 3R grafike. Interakcija s 3R grafiko poteka tako, da se zazna uporabnikovo dejanje ter se na podlagi tega izračunajo novi parametri modela in se ustvari nova slika (pogled) na ekranu. Ker sta 3R vizualizacija in interakcija pri uporabniškem vmesniku kritična elementa, so tu na kratko opisani temeljni principi.

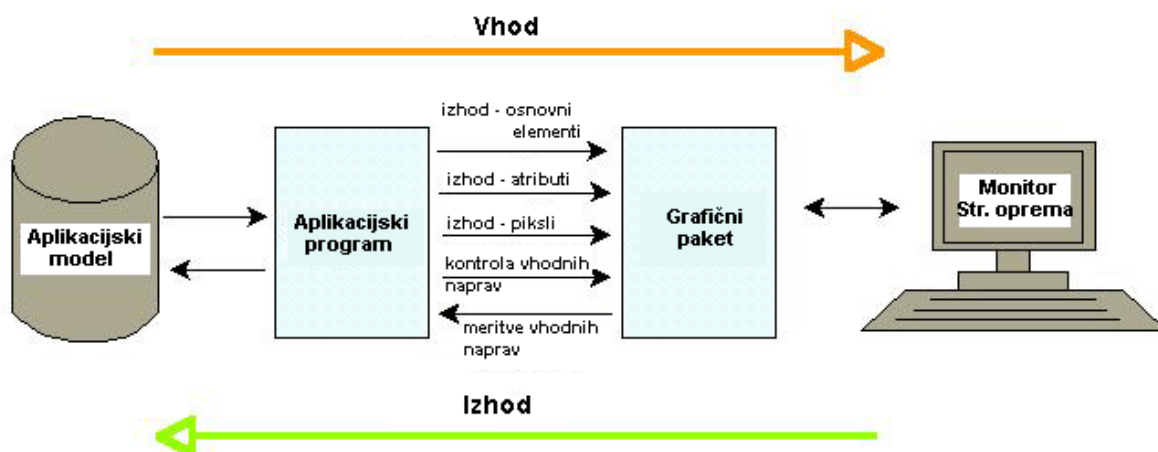
Proces vizualizacije in interakcije je izveden v dveh korakih, kar poznamo pod imenom "cevovod" (ang. *pipelines*), to je vhod (input) in izhod (output). Izhod zajema proces pošiljanja informacij na ekran, vhodni del pa vključuje uporabnikove akcije in odgovarjajoče predprocesiranje modela. Vizualizacijska shema je enaka za vse programe z grafičnim izhodom. To so t.i. CAD sistemi (npr. AutoCAD), Intergraph, Microstation, GIS programi (npr. ArcView), MapInfo, Web browserji (npr. Internet Explorer) itd.

Aplikacijski program je računalniški program, ki kontrolira in zagotavlja pravilen pretok podatkov v obe smeri. To je:

- dostopa do podatkov organiziranih v modelu, izvleče sklop podatkov, ki morajo biti prikazani na ekranu v nekem določenem trenutku in uporabi ustrezen grafičen paket (renderiranje) ter izvede potrebne algoritme za transformacijo 3R grafike v 2R sliko, ki se prikaže na ekranu ter
- kontrolira vhodne naprave.

Proces ustvarjanja slik (imidžov) iz modelov se pogosto imenuje kar renderiranje.

Grafični paket (ang. *graphics package*) je programski vmesnik do grafične strojne opreme (grafične kartice), ki izvaja renderiranje, vzdržuje okna in zaznava uporabnikove zahteve (akcije). Rezultat večletnega razvoja strojno neodvisnega programja je t.i paket za renderiranje (*rendering package*), ki se imenuje OpenGL (*Open Graphics Library*). To je programski jezik za 2R in 3R grafične vmesnike, razvit pri podjetju SCI, ki je postal industrijski standard na področju računalniške grafike. Gledano s programerskega vidika je OpenGL zbirka standardnih procedur (napisanih v C++), ki olajšajo upravljanje grafične strojne opreme. Ker je neodvisen od platforme, to je od operacijskega sistema, mora za upravljanje z okni in vhodnimi napravami (miška, tipkovnica) skrbeti sam aplikacijski program, ki uporablja OpenGL. To je urejeno z odgovarjajočimi knjižnicami. Kljub temu pa OpenGL izpolnjuje svoj namen, saj opiše objekte, jih uredi v 3R prostoru, izračuna barvne in osvetljevalne parametre in pretvori matematični opis objekta v piksle na ekranu. Podobni paketi renderiranja, ki pa so manj strojno neodvisni, so Direct 3D (Microsoft) in Java 3D (Sun Microsystems).



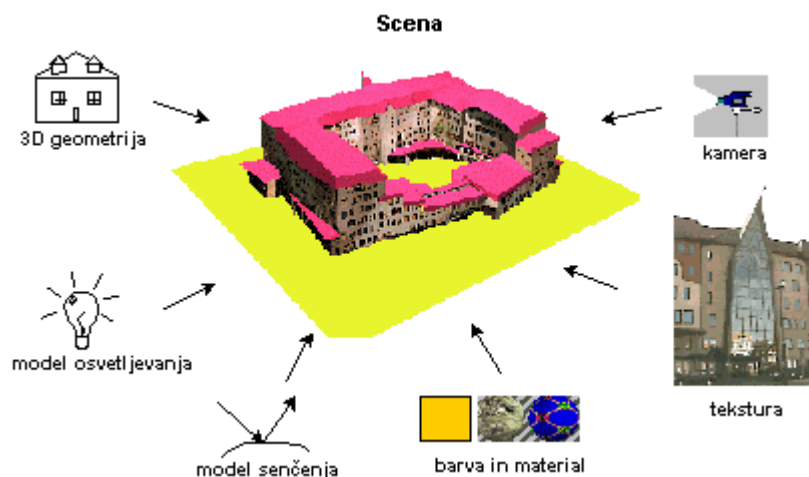
Slika 13: Klasična vizualizacijska shema (Zlatanova, 2000)

Model, ki se uporablja za vizualizacijo, se imenuje *aplikacijski model*. Če je aplikacijski program grafični urejevalec, potem se ta model uvrsti v kategorijo podatkovnih modelov računalniške grafike (RG). Tudi ta model ima seveda osnovne komponente modela opisanega zgoraj. Objekti, atributi in medsebojni odnosi v RG modelu zagotavljajo zadosten, konsistenten sklop podatkov za prikaz na ekranu v zelo kratkem času. Na primer, najpopularnejši modeli t.i. krilato-robi (ang. Winged-edge) podatkovni modeli, delujejo na osnovi 2R topologije. V glavnem, namen je zaznati najbližji naslednji primitiv (npr. ploskev) in ga pravilno prikazati. Atributi primitivov so povezani s parametri, ki so potrebni za izdelavo renderirane slike na ekranu. Na primer zgradba prikazana kot škatla, bo imela kot atribut barvo (radiometrično lastnost), ki je kontrast atributni informaciji (navadno semantična lastnost) zgradbe v 3R GIS. Druga specifična karakteristika je ta, da naj bi bili RG modeli organizirani v glavnem spominu, zaradi zahtev prikaza. To pa ne zadošča za prikazovanje velikega seta podatkov (npr. 3R urbanih področij) in zato zahteva mehanizem za dinamično ekstrakcijo podatkov iz baze podatkov, ki se nahaja na lokalnem ali oddaljenem disku.

Vizualizacija 3R grafike se značilno razlikuje od 2R grafike v naslednjem:

- kreiranju scen,
- orodjih za interakcijo in manipulacijo,
- količini podatkov.

V naslednjih vrsticah so opisani osnovni mehanizmi za pridobitev ustreznih scen in zagotovitve zadovoljivih orodij za interakcijo.



Slika 14: Komponente scene (Zlatanova, 2000)

3.2.1 Komponente scene

Beseda "scena" se uporablja za označitev renderiranih slik na ekranu. Da dosežemo lepo 3R sliko na ekranu, potrebujemo več ključnih komponent (slika 14):

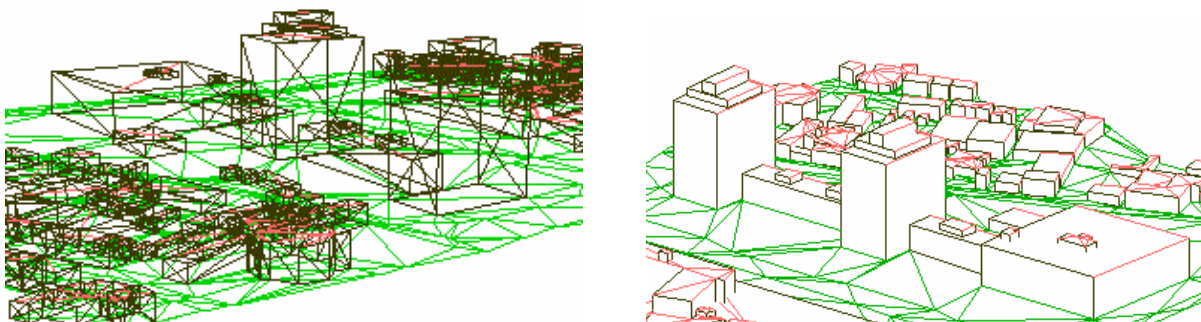
- geometrijo,
- osvetlitev in model senčenja,
- teksturo,
- pozicijo kamere (pogled).

Termin *geometrija* se tukaj nanaša na predstavitev geometrijskih lastnosti objekta v sceni. Vse te komponente skušajo zajeti oblike iz resničnega 3R sveta in jih predstaviti v skladu s človeško percepcijo. *3R geometrija*, to so *poligoni (površine)* in *linije*, sestavljene iz *verteksov* (točk z znanimi koordinatami), ki določajo pozicijo, obliko in velikost objektov. *Osvetlitev* in *modeli senčenja* nadzirajo svetlobo iz umetnega vira in odgovarjajočo refleksijo od površin (podlag). Cilj *barv* in *tekstur* je predstavitev lastnosti površin objektov. *Pozicija kamere* določa lokacijo opazovalca znotraj modela in glede na to orientira model. Razločnost scene je odvisna od uporabljenih komponent pri renderiranju slike ali njenega dela. S tega vidika lahko določimo tri osnovne tehnike:

- točke,
- "prazni" poligoni in
- "polni" poligoni.

3.2.2 Točke in "prazni" poligoni

Scena prikazana samo s točkami je znana pod imenom oblak točk. Tak prikaz modela nam podaja zelo slabo predstavo o resničnem objektu. Zato te točke povežemo v ustrezne poligone. Scena izdelana s t.i. praznimi poligoni se imenuje tudi žični model (ang. wire frame) (slika 15). Da predstavimo različne objekte, lahko uporabimo poligone različnih barv ali algoritme, ki skrijejo "nevidne" linije. Za še boljšo vizualno predstavitev pa obstaja še nekaj bolj sofisticiranih tehnik. Žični modeli uporabljajo manj zahtevne algoritme renderiranja slike, zato veliko modulov renderiranja zagotavlja navigacijo v realnem času z žičnimi modeli. Ti žični modeli pa imajo slabosti predvsem pri sami čitljivosti, razločnosti posameznih objektov, nizko stopnjo realnosti in težave pri orientaciji. Velikokrat se zgodi, da objekti tvorijo dvoumne situacije, saj ne moremo določiti kateri objekt je bližje, oz. dlje.



slika 15: Žični model z (desno) in brez (levo) uporabe algoritma za odstranjevanje nevidnih linij (Zlatanova, 2000)

3.2.3 Polni poligoni

Uporaba zapolnjenih poligonov, kar zajema tudi osvetljevalni model in odgovarjajoči model senčenja, izboljša realno podobo modela. Model osvetlitve je povezan z uporabljenim izvorom svetlobe in je sestavljen iz več parametrov, ki na renderirani sceni definirajo barvo vsakega piksla posebej. Model senčenja je bolj širok pojem, ki zajema tudi prej omenjen

model osvetlitve in definira uporabo modela osvetlitve in katere parametre le ta sprejme. Na primer nekateri modeli senčenja uporabijo model osvetlitve le na določenih delih celotne množice pikslov, za ostale pa pridobijo prave vrednosti z interpolacijo. Spet drugi modeli računajo vrednost za vsak piksel na ekranu. Če pa upoštevamo še interakcijo svetlobe z različnimi površinami, postaneta oba modela zelo kompleksna.

Model senčenja je odvisen od metode izračunavanja pikselske vrednosti po celotni površini.

Poznani so različni pristopi:

- konstantno senčenje, poimenovano *flat* ali *facet* – izračunana je ena vrednost in uporabljena za celotno površino;
- interpolirano senčenje: *Gourand-ov* model – izračun intenzivnosti v vsakem verteksu in nato z interpolacijo porazdeljeno po celotno površini. *Phong-ov* model – interpolacija normal na površino.

Kot prvi korak se vzame izračunana normala na robu te površine.

Zelo kompleksni modeli osvetlitve in senčenja nam lahko zagotovijo zelo realistične prikaze realnega sveta. Vendar pa ti postopki drastično zmanjšajo hitrost renderiranja slike. Npr. pri uporabi razpršene osvetljenosti (ang. radiosity) (zelo obsežna tehnika senčenja) se lahko ena scena izračunava tudi po več ur. V praksi takšni modeli seveda niso uporabni. Krajši čas renderiranja dosežemo z uporabo preprostih modelov senčenja in osvetljevanja kot so npr. *Gourand-ov* in *Phong-ov* model ali s teksturami. Vendar pa preveč preprost algoritem senčenja lahko ustvari ravno nasprotni učinek kot je v resnici. Lahko se npr. pojavijo neki moteči efekti, kot so silhuete poligona na stičnih robovih dveh ploskev ali pa nekonstantno senčenje površin med navigacijo po modelu. Vse naštetu kaže na to, da je uporaba tekstur še najprimernejša metoda za prikaz dokaj dobre realnosti v kratkem času renderiranja.

Uporaba tekstur je tehnika pri kateri navaden žični model "zavijemo" v sliko (skenirano realno sliko objekta ali pa umetno ustvarjeno, sintetično sliko). Princip je tak, da namesto izračunanih vrednosti pikslov iz modelov senčenja in osvetlitve, uporabimo odgovarjajoče piksele iz slike. Ker tu ne poteka nobeno preračunavanje vrednosti pikslov, poteka renderiranje veliko hitreje. V tem primeru se mora program ukvarjati z organizacijo slik v spominu, kot tudi z ekstrakcijo vsake nove slike iz baze, potrebne za določeno sceno.

Računalniški grafični standardi (npr. OpenGL, VRML, X3D, Java 3D) dajejo prednost enostavnim modelom senčenja in osvetlitve, s kombinacijo uporabe teksture. Ta kombinacija zelo pospeši renderiranje realnih scen.

3.2.4 Interakcija in manipulacija

Termin *interakcija* se nanaša na možnost premikanja in raziskovanja modela (zoom, prelet, raziskovanje, obračanje na enem mestu ang. pan, itd.) brez spreminjanja originalnih parametrov scene. *Manipulacija* pa je možnost dostopa do elementov scene (operacija select) in možnost spreminjanja celotne scene (operacije uredi, dodaj, briši, rotiraj, premakni, ...) ali nekaj njenih komponent, kot so geometrija, svetloba, pozicija kamere, slike za teksturo, koordinate teksture. Ta dva termina imata zelo pomembno vlogo pri procesu vizualizacije, predvsem pa pri cilju vizualizacije. Ker je cilj naloge uporaba tehnik navidezne resničnosti z interakcijo in manipulacijo, so spodaj razloženi še nekateri pojmi.

Čas, potreben da se sistem odzove na uporabnikovo zahtevo, je najpomembnejši faktor pri razločevanju med visoko in nizko stopnjo interaktivnosti in manipulacije. Časovni interval daje uporabniku občutek, da se nahaja "znotraj modela". Glede na ta časovni interval razlikujemo naslednje vrste prikaza:

- statični prikaz (ang. off-line plotting),
- interaktivni prikaz (ang. interactive plotting),
- animacija (ang. animation),
- navidezna resničnost (ang. virtual reality).

Statični prikaz je razred interakcije, kjer uporabnik nima možnosti upravljati s procesom risanja (ang. plotting). *Interaktivni prikaz* se nanaša na sistem orodij za interakcijo in manipulacijo, ki potrebujejo čas za preračunavanje in renderiranje nove scene. To je večji del CAD sistemov. *Animacija* omogoča uporabniku izračunavanje dinamike v naprej in prikazovanje v realnem času (3Dstudio, TrueSpace, Wavefront in drugi).

Navidezna resničnost je najvišja stopnja interakcije in manipulacije, pri kateri se operacije izvedejo v skoraj realnem času. To pomeni, da uporabnik ne zazna vmesnih kalkulacij med

dvema zaporednima scenama. Interakcija in manipulacija tu obsegata vsa sredstva (strojno in programsko opremo), ki nudijo uporabniku ustrezna orodja za kontakt z objekti in direktno manipulacijo z njimi. Sama percepcija uporabnika je zelo odvisna tudi od uporabljene strojne opreme. Medtem ko oprema sestavljena iz običajne miške in 2R ekrana omogoča samo navigacijo, naglavni prikazovalnik (ang. Head Mounted Display - HMD – naprava, ki omogoča postavitev ekrana pred vsako oko) pričara pravo 3R percepcijo. Z ozirom na razpoložljivo strojno opremo, lahko sisteme navidezne resničnosti - NR delimo v pet razredov:

- Začetni (ang. entry) NR sistemi sestavljajo osebni računalnik ali delovna postaja z 2R vhodno enoto, 2R ekranom in grafičnim pospeševalnikom (grafično kartico).
- Osnovni (ang. basic) NR sistemi sestavljajo specialne naprave za selekcijo, kot so PowerGlove, 3R ali 6R miška ali joystick in nekatere izboljšane prikazovalne naprave, npr. stereo grafični prikazovalnik.
- Napredni (ang. advanced) NR sistemi že vključujejo vse tipe pospeševalnikov renderiranja, to je tekstur, videa in zmogljivih procesorjev.
- Poglobljeni (ang. immersive) NR sistemi vključujejo kompleksne naprave za prikazovanje in zaznavanje (HMD, CAVE).
- Simulatorji letalskih kabin (ang. Cockpit Simulators) omogočajo fizično premikanje in ustvarjajo občutek realne prisotnosti v sceni in akcije.

Sodobna strojna in programska oprema omogoča delovanje začetnega NR sistema na vseh osebnih računalnikih. Realno časovna interakcija in manipulacija velikih modelov v načinu senčenja ali teksture so splošno obravnavane nizko stopenjske NR tehnike. Sistem, ki zagotavlja tako realno časovno interakcijo, kot tudi manipulacijo, se imenuje NR modelar (ang. *VRmodeller*). Sistemi, ki zagotavljajo orodja za interakcijo, vendar pa nudijo omejeno manipulacijo so poznani kot *NRbrskalniki* (ang. *VRbrowsers*) ali *pregledovalniki*. V nadaljevanju so še bolj detajlno opisane tehnike virtualne resničnosti podprte z VRML in NR brskalniki.

3.2.5 Stopnje detajla (Levels of Detail - LOD)

Koncept stopnje detajla je bil razvit zato, da se pospeši vizualizacija velikih scen. Ideja je ta, da postanejo objekti oz. njegovi elementi kompatibilni z velikostjo piksla na ekranu, ko so le ti zelo oddaljeni od opazovalca. To omogoča, da se originalna geometrična predstavitev zamenja z novo, nizko ločljivostno. Nizka ločljivost namreč zahteva manj spomina in časa za procesiranje, kar zelo pospeši sam vizualizacijski proces. Ta metoda se uporablja samo pri vizualizaciji, ko je objekt zelo oddaljen od uporabnika. Bližnji objekti so še vedno prikazani v polni ločljivosti. Še več, ko se približujemo oddaljenemu objektu, se le ta postopoma prikazuje v vedno večji ločljivosti. Te spremembe v ločljivosti morajo biti seveda za naše oči nezaznavne. Kontrolirane pa so navadno z enim parametrom, to je z razdaljo med točko gledanja in samim objektom.

Večina razvitih algoritmov predpostavlja, da je LOD v naprej izračunana različna prezentacija objekta. Na primer stavba ima lahko štiri različne geometrične opise:

- stene, streha, okna, vrata (to je LOD 0),
- stene, streha (LOD 1),
- okvir robov (LOD 2),
- ena točka (LOD3).

Vse štiri stopnje morajo biti izračunane v naprej in dostopne samemu sistemu. Sam postopek raziskav na tem področju je nagnjen k avtomatskemu generiranju LOD na osnovi prejšnjih scen, kar je do sedaj dajalo dobre rezultate samo za scene, ki so si med seboj podobne. Originalno razvit postopek za geometrično poenostavljanje se lahko uporabi tudi na imidžih uporabljenih za texture. Pri čemer gre za isti princip: slika uporabljena za bližnji objekt je prikazana v višji ločljivosti kot slika uporabljena na oddaljenem objektu.

Stopnja detajla je že uporabljena metoda na skoraj vseh sistemih vizualizacije, ampak upošteva samo geometrijo. Specifikacija VRML-ja pa že predstavlja fleksibilno organizacijo LOD geometrije in tekstur, ki temeljijo na ločenih, v naprej oblikovanih VRML dokumentih.

3.3 Svetovni splet (World Wide Web - WWW)

Na začetku moramo najprej pojasniti pojme kot so *internet*, *web* in *protokoli*. Internet najbolj označimo z besedami "svetovno računalniško omrežje" (Lenardič, 2002). Je največje računalniško omrežje, ki deluje kot mreža vseh mrež. V ožjem pomenu ga lahko označimo kot svetovno omrežje računalnikov velikih razsežnosti, ki med seboj komunicirajo z uporabo skupnih protokolov. Protokoli so skupki pravil ali dogovorov o načinu komuniciranja. So splošno sprejet format, ki omogoča prenos sporočil.

Internet omogoča več različnih storitev. To so (Lenardič, 2002): elektronska pošta, prenos datotek (ang. FTP – File Transfer Protocol), skupine novic (ang. USENET), večuporabniške razsežnosti (ang. Multi-User Dimensions ali MUD), knjižnični katalogi prek Telnet, pogovorna okna (IRC, Messenger), hrček (ang. Gopher) in seveda svetovni splet (ang. WWW ali 3W – World Wide Web), ki spada med najpogosteje uporabljane servise interneta.

Razvoj in začetek delovanja svetovnega spleta pomeni pravo revolucijo v izmenjavi in dostopu do podatkov med oddaljenimi računalniki. Od prvega pokaži in klikni (point-and-click) pregledovalnika (Mosaic, NCSA) leta 1993 pa do danes, se je število uporabnikov interneta močno povečalo. K temu so pripomogle naslednje lastnosti tega omrežja:

- Splet uporablja za izmenjavo podatkov protokole narejene za Internet. T.i. HyperText Transfer Protocol (HTTP) je za eno stopnjo višje kot Transmission Control protocol (TCP), ki predstavlja razširitev osnovnega Internet protokola (IP), oba splošno poznana kot TCP/IP protokol. Ker je TCP/IP protokol enak za vse računalnike, lahko že povezani računalniki v Internet z lokalno (Local Area Network – LAN) ali široko (Wide Area Network – WAN) mrežo, enostavno "posvojijo" protokol z namestitvijo določenega programja.
- Za iskanje in prikazovanje informacij uporabljamo t.i. pregledovalnik ali spletni pregledovalnik (ang. *Web browser*). To je program, ki deluje na uporabnikovi oz. odjemalčevi strani in podpira večino uporabljenih protokolov, razvitih tudi že pred HTTP-jem. To so FTP (File Transfer Protocol), TELNET (Text-oriented Remote Service), Gopher (univerzitetni informacijski protokol) in e-mail message protokol. Tak pregledovalnik zajema celotno

funkcionalnost, ki jo uporabnik potrebuje. Najbolj razširjeni pregledovalniki so Internet Explorer, Mozilla, Netscape, Opera.

- Splet uporablja enostavno in učinkovito shemo poimenovanja, t.i. URL (Uniform Resource Locator), za identifikacijo unikatne lokacije informacije in protokola za prikaz le te.

- Informacije na spletu so organizirane v standardnih enotah ali straneh. Stran je kakršenkoli dokument prikazan v pregledovalniku in ga uporabnik lahko bere. Še več, z uporabo hipertekstovnega mehanizma, lahko vsaka stran vsebuje poljubno število povezav (ang. linkov) na ostale strani na spletu, kar nedvomno pospeši in predvsem olajša brskanje za informacijami.

- Spletne strani so napisane v standardnem jeziku t.i. HTML-ju (Hyper Text Markup Language). Ta daje pregledovalniku navodila, kako določen dokument prikazati. Čeprav sam jezik določa veliko parametrov, kot so fonti in velikost teksta, je sama vizualizacija (npr. nove vrstice) naloga pregledovalnika na strani uporabnika. Tako je prikaz dokumenta praktično neodvisen od operacijskega sistema.

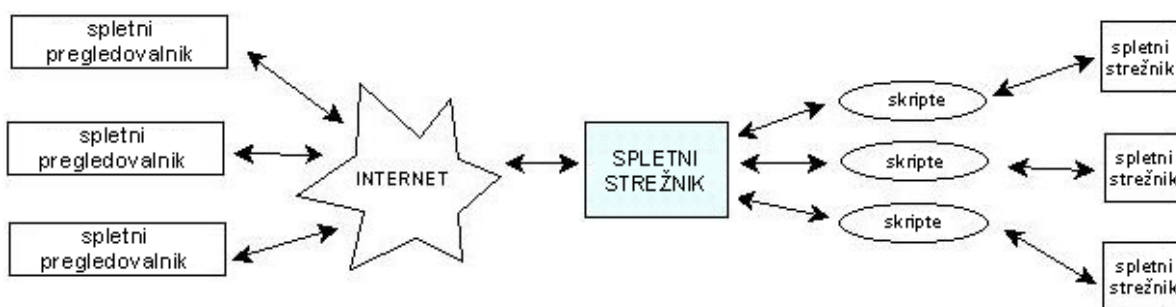
- Splet je lahko poljubno razširljiv z dodatnimi programi v dve smeri, in sicer na uporabnikovi in na strežnikovi strani. Na strani odjemalca, lahko pregledovalnik vsebuje različne dodatne programe imenovane *plug-ins*, *helper applications* in *applets*, z namenom prikaza specifičnih tipov dokumentov. Na strani strežnika pa se lahko razširi z različnimi *skriptami*, ki se na tem strežniku tudi izvedejo.

- Spletni pregledovalnik nudi uporabniški vmesnik, ki temelji na operaciji pokaži in klikni, ki ne zahteva shranjevanja ukazov. Tako je postal svetovni splet v zelo kratkem času popularen tako med navadnimi ljudmi različnih poklicev, kot tudi med različnimi velikimi podjetji.

3.3.1 Tehnike dostopanja do podatkov na svetovnem spletu

Osnovni mehanizem dostopanja do raznih dokumentov na spletu je zgrajen na principu *odjemalec-strežnik*. Ta princip deluje tako, da računalnik, na katerem delamo (osebni

računalnik ali delovna postaja), imenovan *odjemalec*, pošlje zahtevo za informacijo oz. proces drugemu računalniku (zopet osebnemu računalniku ali delovni postaji), ki zahtevano poseduje, imenovanemu *strežnik*, le ta pa zahtevano pošlje nazaj odjemalcu. Oba računalnika morata biti seveda medsebojno povezana v mrežo (LAN, WAN, Internet). Odjemalec mora imeti ustrezen program, ki mu omogoča povezavo s strežnikom ter pošiljanje zahtev in sprejemanje povratnih informacij. Strežnik pa mora imeti program, ki zaznava zahteve, jih izvrši in vrne zahtevane podatke ter zapre vzpostavljeno povezavo med računalnikoma.



Slika 16: Klasičen odjemalec-strežnik spletni princip (Zlatanova, 2000)

Podobno standardnemu konceptu odjemalec-strežnik, potrebuje splet, spletni strežnik (npr. *Apache, WebSite, IIS*) na strežnikovi strani in spletni pregledovalnik na odjemalčevi strani (slika 16). Spletni pregledovalnik prikazuje dokumente in zaznava uporabnikove akcije. Spletni strežnik pa lahko vzpostavlja povezave z drugimi strežniki ali aplikacijami z izvedbo t.i. *skript* in zagotavlja dostop do lokalnih datotek. Skripte (pogosto imenovane CGI skripte) so programi, napisani v skladu s standardom Common Gateway Interface (CGI) in so locirani v posebni mapi na serverju. CGI skripte, ki so lahko napisane v različnih programskih jezikih (Perl, C, Delphi, PHP) imajo tri dele: *glava* (ang. *header*), *telo* (ang. *body*) in *konec* (ang. *end*). Med vsemi različnimi opcijskimi parametri mora *glava* določiti tip dokumenta, ki pove ali lahko ta dokument prikaže spletni pregledovalnik ali se mora zagnati pomožni program, dodatek pregledovalnika ali poseben, ločen program. Na primer, če je dokument navaden tekst, ga pregledovalnik vizualizira kot HTML stran, če pa je dokument *model/vrml*, se aktivira VR pregledovalnik. *Telo* skripte lahko vsebuje razne izračune, povezave na druge strežnike, navodila za izdelavo dokumentov med delovanjem (ang. *on the fly*) ali različne kombinacije teh vsebin. *Konec* skripte daje strežniku znak, da zapre povezavo strežnik-odjemalec.

3.3.2 VRML – spletni standard za 3R vizualizacijo

VRML je kratica za *Virtual Reality Modeling Language* (modelirni jezik za navidezno resničnost). Njegovi začetki segajo v leto 1994, ko so bili izvedeni prvi poskusi uveljavljanja spletnega 3R grafičnega standarda. Prva verzija jezika VRML 1.0 je bila sposobna opisovanja statičnih 3R modelov in njihovo povezovanje s HTML in drugimi VRML dokumenti. Druga verzija, VRML 2.0 ali tudi VRML 97, pa je imela dodano še interakcijo, dinamiko, uporabo skript in multimedijски vmesnik. Decembra, leta 1997, je postal priznani odprti internetni standard ISO/IEC 14772.

S pomočjo VRML-ja lahko gradimo, urejamo in animiramo svoje 3R navidezne svetove in jih prikažemo na internetu. Za urejanje potrebujemo najmanj nek tekstovni urejevalnik, kot je npr. Beležnica, saj je koda zapisana v navadnem ASCII formatu. Zatorej pri VRML-ju ne gre za tipični programski jezik, v smislu kot ga poznajo programerji, saj za svoje delovanje ne potrebuje nekega vmesnega programa, imenovanega interpreter, ki daje računalniku inštrukcije za izvedbo. Koda je sestavljena iz opisa 3R scen in predvidenega obnašanja teh scen glede na akcije, ki jih bo sprožil uporabnik. Ponavadi je sestavljena iz več stotih oz. tisočih poligonov, ki predstavljajo osnovni gradnik v 3R scenah na računalnikih. Ker imamo v večini primerov opravka z zelo kompleksnimi navideznimi svetovi, je dobro, če jih urejamo z zato namenjenimi VRML urejevalniki, ki nam avtomatsko generirajo VRML kodo in ne s tekstovnimi urejevalniki, v katerih se zelo hitro izgubimo v kodi. Jaz sem uporabljal program *Internet Scene Assembler*, ki je produkt podjetja *Parallel Graphics*. Z njim sem z lahkoto urejal npr. različne poglede, osvetlitev, barvo, ozadje, animiranje in podobno, že pripravljenega modela. To je že optimiziranega do te mere, da zadostuje pogoju velikosti datoteke. Datoteke namreč ne smejo biti prevelike zaradi hitrosti prenosa le teh s strežnika na odjemalčev računalnik in tudi zaradi gladkejše navigacije, kar je razloženo v nadaljevanju.

Za pregledovanje, oz. prikazovanje ustvarjenih navideznih svetov potrebujemo seveda nek pregledovalnik, ki je lahko samostojni program, npr. *Openwrl* ali pa dodatek spletnemu pregledovalniku, npr. *Cortona VRML Client*. Ti pregledovalniki so večinoma brezplačni in jih lahko dobimo na spletu. Ko uporabnik pride na VRML orientirano stran, VRML pregledovalnik prejme navodila za prikaz od strežnika na drugi strani. Strežnik pošlje VRML

kodo, imenovano okolje, ki je naloženo na njegovem disku, uporabniku, ki lahko nato raziskuje ta navidezen svet na svojem računalniku. Vsak virtualni svet je svoje okolje, ki se zaradi lažje navigacije naloži iz strežnika na lokalni disk klientovega računalnika. Datoteke za prenos so velike od okoli 100Kb pa do več Mb, tako da pasovna širina za prenos podatkov igra pomembno vlogo pri izvajanju. Ko je virtualno okolje naloženo na lokalni računalnik, pa pride do izraza predvsem procesorska moč računalnika. Zmogljivejši kot je, hitrejši in uglajenejši je premikanje po prostoru. Zaradi povprečno ne najvišjih hitrostih prenosa podatkov pri današnjih domačih uporabnikih moramo skrbeti za to, da so VRML datoteke čim manjše. Razvijalci se pri tem tudi čimbolj trudijo, da bi ustvarili takšne VRML strani, ki bi zmanjšale komunikacijo med strežnikom in odjemalcem na minimum.

4 IZDELAVA MODELA KRAŠKE JAME

4.1 3R lasersko skeniranje – postopek snemanja

Snemanje je izvajalo podjetje DFG consulting d.o.o. iz Ljubljane, dne 13.07.2005. Ker je šlo pri snemanju bolj za testni primer, tudi meritve niso bile izvedene po nekem v naprej pripravljenem planu. Primer naj bi nam predvsem podal odgovore na vprašanja ali je takšna snemanja sploh smiselno izvajati in ali so pridobljeni izdelki tudi tržno zanimivi. Drug namen pa je bil pridobiti izkušnje in znanje na tem področju, ki bodo koristile predvsem pri načrtovanju kakovostnejše izvedbe drugih podobnih projektov, kar vključuje tako samo snemanje kot tudi naknadno obdelavo in predstavitev izdelkov.

Pri snemanju so bila uporabljena 4 stojišča, iz katerih je bilo pridobljeno 8 skenov. Zajem podatkov je bil izveden po vseh treh koordinatnih oseh. Vezne točke (skupno 10 točk) (slika 17) so bile fiksirane na tripodu in na steni, razporejene pa so bile enakomerno po prostoru. Oslonilne točke niso bile merjene, vseeno pa je možna umestitev modela v globalni koordinatni sistem z naknadno meritvijo oslonilnih točk.

Name	RefCount	RefType	Size	X	Y	Z	σX	σY	σZ
<input checked="" type="checkbox"/> PRCS_001	4	RIEGL Cylinder 10cm	0.00	-41.409	2.767	1.526	0.003	0.007	0.007
<input checked="" type="checkbox"/> PRCS_002	2	RIEGL Cylinder 10cm	0.00	-36.769	15.405	-0.597	0.006	0.004	0.011
<input checked="" type="checkbox"/> PRCS_003	3	RIEGL Cylinder 10cm	0.00	33.552	-7.659	-1.641	0.007	0.006	0.006
<input checked="" type="checkbox"/> PRCS_004	5	RIEGL Cylinder 10cm	0.00	57.663	18.159	-3.546	0.004	0.007	0.006
<input checked="" type="checkbox"/> PRCS_005	4	RIEGL Cylinder 10cm	0.00	36.173	15.613	-5.498	0.003	0.006	0.010
<input checked="" type="checkbox"/> PRCS_006	5	RIEGL Cylinder 10cm	0.00	20.888	10.580	-2.976	0.005	0.006	0.004
<input checked="" type="checkbox"/> PRCS_007	3	RIEGL Cylinder 10cm	0.00	-8.526	8.786	-1.247	0.003	0.001	0.002
<input checked="" type="checkbox"/> PRCS_008	4	RIEGL Cylinder 10cm	0.00	-13.938	-4.182	2.198	0.003	0.007	0.002
<input checked="" type="checkbox"/> PRCS_009	3	RIEGL Cylinder 10cm	0.00	9.505	-4.656	0.214	0.002	0.004	0.001
<input checked="" type="checkbox"/> PRCS_010	4	RIEGL Cylinder 10cm	0.00	40.250	5.515	-4.449	0.003	0.004	0.004

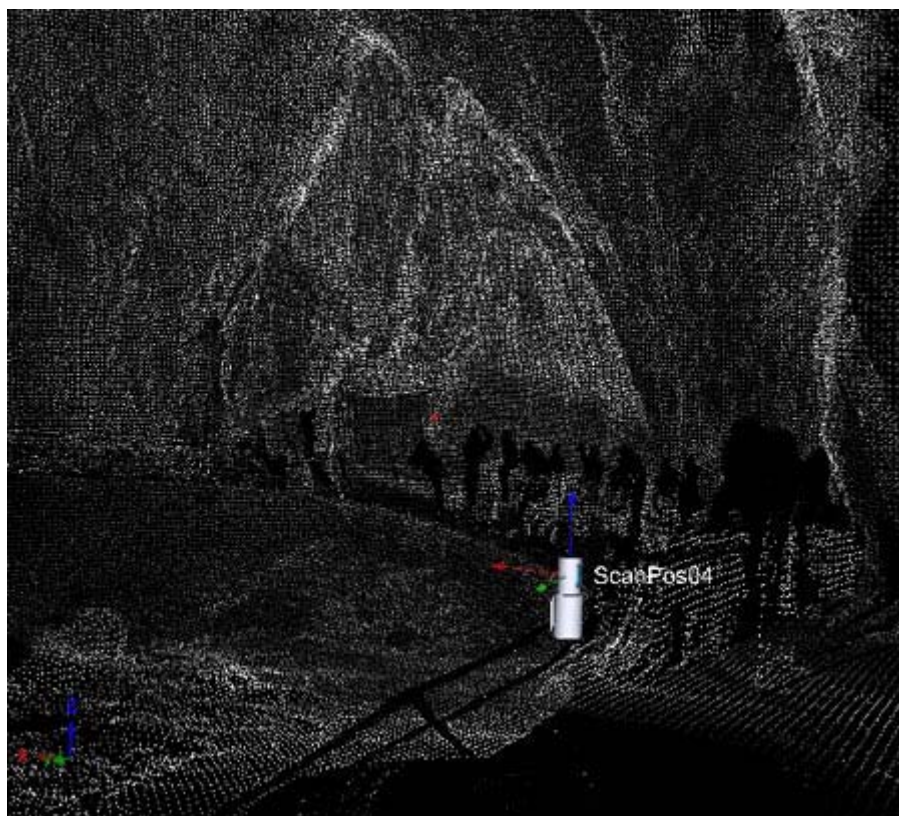
Slika 17: Merjene vezne točke

V jami ni bilo uporabljene posebne osvetlitve, saj je laserski skener za svoje delovanje tudi ne potrebuje. Potrebna je bila le tolikšna svetloba, da se je videlo, kam postaviti skener in tarče, za kar pa je zadostovala naravna svetloba iz zunanosti jame in umetna svetloba žarnic, ki so v jami že postavljene v turistične namene.

Celotno snemanje je bilo izvedeno v približno dveh urah, zajetih pa je bilo vsega skupaj 15.326.267 točk.

4.2 Rezultati snemanja

Na vsakem posameznem stojišču smo pridobili oblak velikega števila točk s prostorskimi koordinatami (x,y,z) v skenerjevem lastnem koordinatnem sistemu z izhodiščem v skenerjevem centru (slika 18).



Slika 18: Oblak točk na enem stojišču

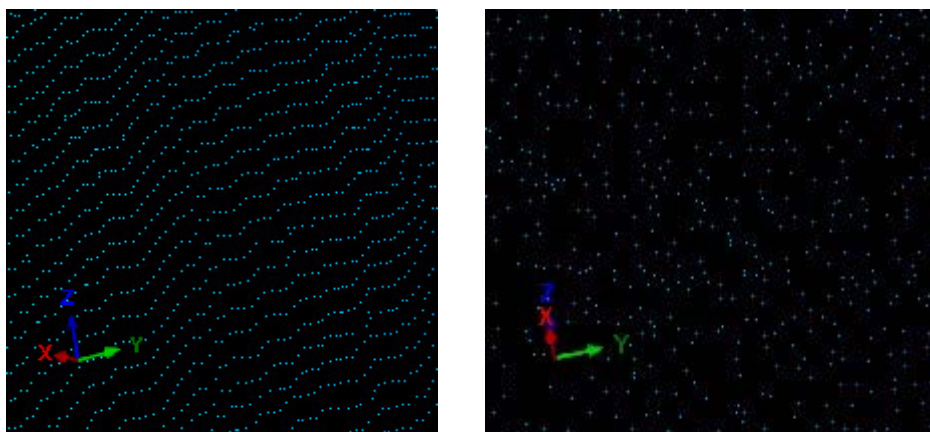
Povprečno vsebuje en sken okoli 2 milijona točk. Če združimo vse skene v en skupen koordinatni sistem, pomeni to močno obremenitev za računalnik in zato premikanje po modelu ni tekoče in tudi datoteke, ki se izvozijo iz programa, zasedejo preveč prostora. Večina teh točk je odvečnih, ker je isti detajl posnet večkrat z različnih pozicij, zato se točke podvajajo. Ker gre v tem primeru predvsem za vizualizacijo modela in ne za najatančnejše

meritve, tudi ne potrebujemo take gostote točk, kot jih je zajel inštrument. Zaradi tega sem pred triangulacijo modela te odvečne točke odstranil.

4.3 Odstranitev odvečnih točk in združitve vseh skenov v en objekt

Po zajemu podatkov celotnega obravnavanega območja pričnemo z obdelavo le teh. Naj povem, da je časovno razmerje med delom na terenu in kasnejšim pisarniškim delom nekje v razmerju med 1:3 in 1:5, če ga opravlja izkušen strokovnjak. Odvisno je od tega, kako zapleten je obravnavani objekt in kaj želimo pridobiti kot končni rezultat. Primer objekta v moji diplomski nalogi je eden izmed najbolj zapletenih, saj gre za objekt, ki ima mnogo detajla (kapniki). Ker imamo zajeto veliko količino točk in zelo kompleksno površino, moramo uporabiti različne metode reduciranja in triangulacije točk, da dobimo lepo gladek model, kar je opisano v nadaljevanju. Zaradi tega se tudi sam čas obdelave pridobljenih podatkov zelo poveča.

Najprej zmanjšamo število točk na vsakem skenu, da dobimo manjšo gostoto točk, kar sem omenil že prej. To storimo z uporabo filtra *octree*, ki odstrani točke po znani metodi oktalnega drevesa. Ta metoda temelji na kocki, ki se deli na osem enakih kock, ki so deljene naprej in tako do neke določene minimalne vrednosti velikosti kocke, navadno med 0.1 in 1.0 m, merjeno po stranici. Jaz sem uporabil vrednost 0.1 m. V te kocke so zajete vse točke obravnavanega območja. Filter pa deluje tako, da v vsaki najmanjši kocki lahko obstaja samo ena točka, ki je težišče vseh točk v tej kocki, ostale pa pobriše. Tako smo zmanjšali število točk npr. na prvem skenu iz 1 994 016 točk na 498 679 točk, kar predstavlja zmanjšanje za dobrih 75%. Za razliko od v vrstah nanizanih zajetih točk (surovi podatki), smo s tem tudi pridobili točke, enakomerno razpršene po površini modela (slika 19), kar nudi določene prednosti pri triangulaciji in hitrosti izvedbe drugih operacij.



Slika 19: V vrsto nanizane točke (levo) in razpršen oblak točk (desno)

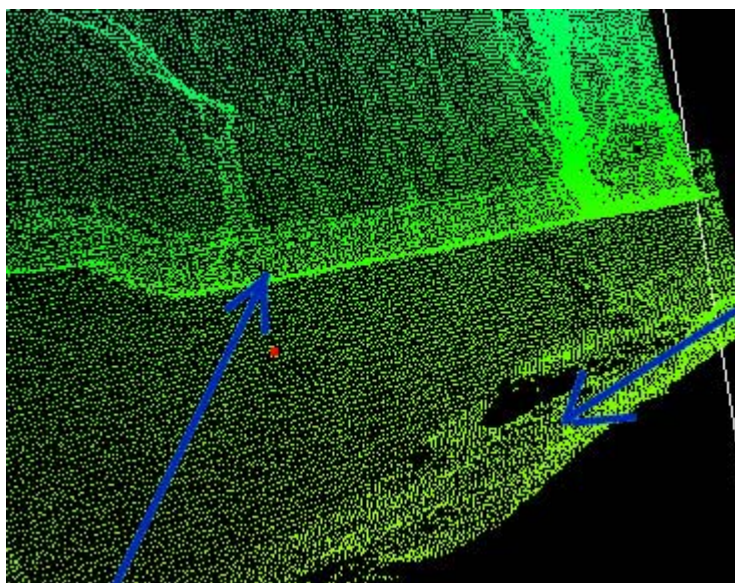
Naslednja stopnja je združitev vseh filtriranih skenov v en objekt oz. polydata. Tudi pri združevanju uporabimo filter octree, da izbrišemo morebitno podvajanje točk iz različnih skenov. Celoten objekt vsebuje tako 1 704 299 točk. Še vedno pa je prisotnih kar nekaj točk, ki nas ne zanimajo. To so npr. posneta drevesa, zunanost jame, ljudje in pa točke, ki so napačne (ne sodijo nikamor). Te točke moramo izbrisati ročno. Spet se zmanjša število vseh točk, in sicer vsebuje sedaj celoten model, pripravljen za postopek triangulacije, 1 325 155 točk, kar je seveda občutno manj (91% manj) od števila začetnih, surovih podatkov.

4.4 Triangulacija modela – izdelava delov mreže

Kot sem že omenil, predstavlja obravnavani objekt zelo zapleteno formo, kar zahteva poseben pristop k triangulaciji. Ker nisem nikjer zasledil podobnega primera, sem si postopek triangulacije tega obsežnega modela zamislil tako, da sem celoten objekt razdelil na več delov. Vsak posamezni del sem trianguliral, nato pa vse dele zopet združil v en objekt.

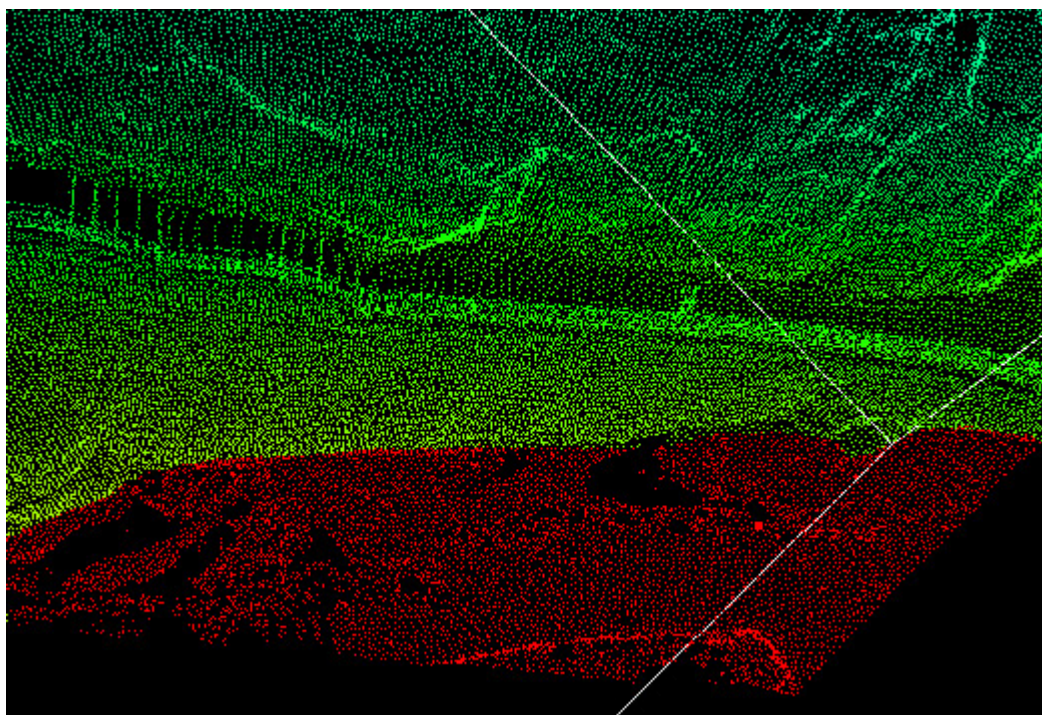
Ta način sem uporabil zato, ker program izvaja triangulacijo po Delaunay-jevi metodi, ki je algoritem za povezavo točk s koordinatami v dvorazsežnem prostoru, ki jih izračuna glede na položaj modela prikazan na ekranu ("ekranske" koordinate). Deluje pa tako (Vrenko, Petrovič, 2005), da med seboj optimalno poveže točke tako, da tvorijo, kolikor je le mogoče, enakostranične trikotnike. Zaradi tega je bilo treba celoten oblak točk razdeliti na več delov predvsem tam, kjer so karakteristične linije. To je bila predvsem ograja in pa spoj med tlemi

in steno. Program namreč ne omogoča zaznave karakterističnih linij, zato sem moral le-te definirati. Paziti sem moral tudi na sam pogled na točke (položaj modela prikazanega na ekranu), zaradi prej opisanega pogoja Delaunay-jeve triangulacije. Področje, ki sem ga hotel triangulirati, je moralo ležati približno pravokotno na ravnino gledanja. Le tako je bilo možno pridobiti lepo, gladko površino, brez nazobčanosti, ki se je pojavljala na mestih, kjer so bile točke preveč "ena nad drugo". Te težave nam pomagajo predstaviti slike 20, 21, 22, 23 in 24. Do napak je prihajalo tudi,, če so bile zajete tudi točke, ki sploh ne pripadajo tej površini.



Slika 20: Del območja za triangulacijo

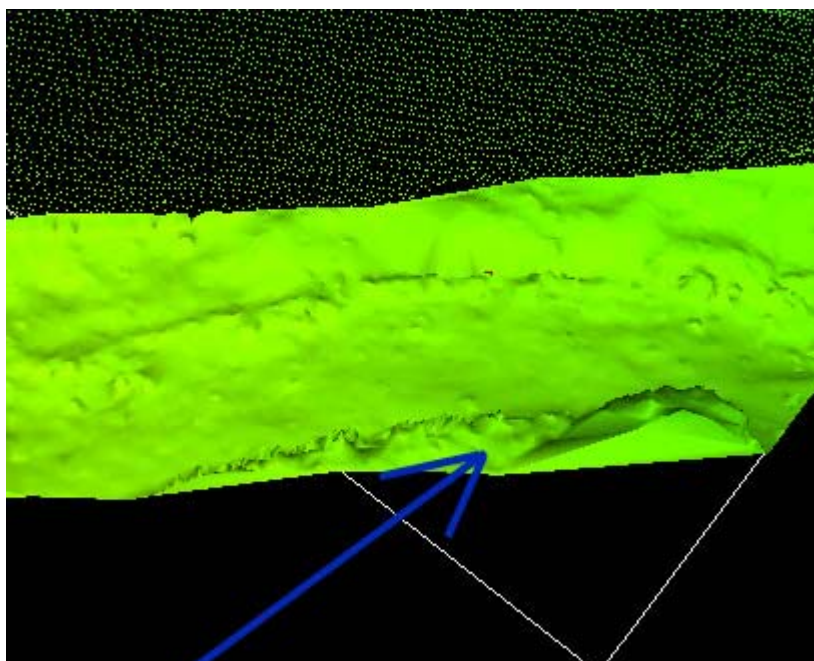
Na sliki 20 se lepo vidi, da se točke, kjer so stopnice (leva puščica), prekrivajo, zato bo tu prišlo do zmešnjave, če bomo triangulirali v takem pogledu. Isto se bo zgodilo tudi s tlemi, ki jih označuje desna puščica. Zato sem del, do kod seže ograja, razrezal in ga potem trianguliral skupaj z ograjo. Potko sem trianguliral s preostalim delom. Pri tleh sem postopal na ta način, da sem spremenil pogled tako, da sem videl točke pravokotno, jih označil in nato trianguliral.



Slika 21: Spremenjen pogled na točke in selekcija območja primerne za triangulacijo v tem pogledu

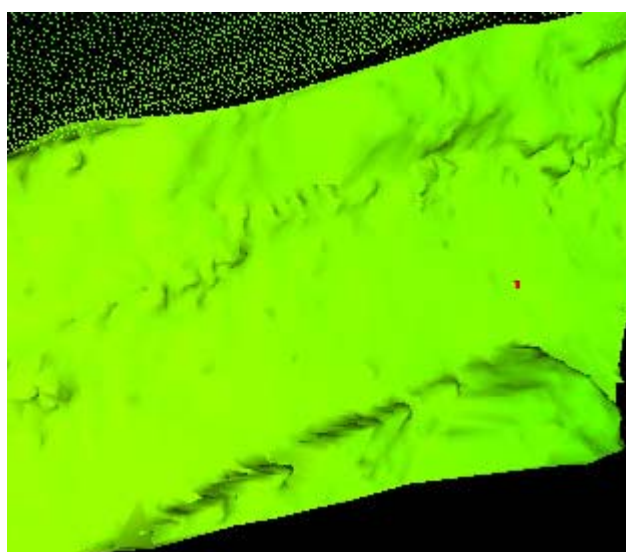
Pri triangulaciji imamo možnost spreminjanja parametrov. In sicer lahko nastavimo maksimalno dolžino stranic trikotnikov, kar pomeni, če je ena stranica trikotnika daljša od določene meje, se trikotnik odstrani. Drug parameter pa je maksimalni kot med normalo trikotnika in žarkom pogleda. Večinoma sem uporabljal nastavitve maksimalne dolžine stranic 5m in maksimalnega kota 90° .

Če sedaj trianguliramo izbrano področje, dobimo gladko površino, kot jo prikazuje slika 22.



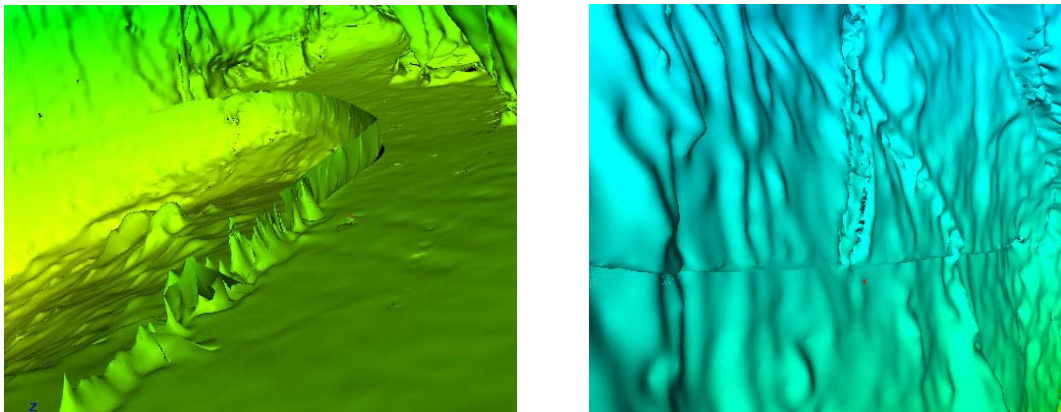
Slika 22: Triangulirana površina

Vendar pa ima ta površina napako, nazobčanost terena, ki se je pojavila zaradi prekrivanja točk in jo kaže puščica. V takem primeru sem spremenil pogled pravokotno na te točke, jih označil in zbrisal trikotnike. Nato sem ponovno označil te točke in jih na novo trianguliral. Rezultat prikazuje slika 23. Isti postopek sem uporabil tudi pri napakah, ki so se pojavile na mestih, kjer sem objekt razrezal in zopet združil, na t.i. šivih.



Slika 23: Ponovna triangulacija in odstranitev napake

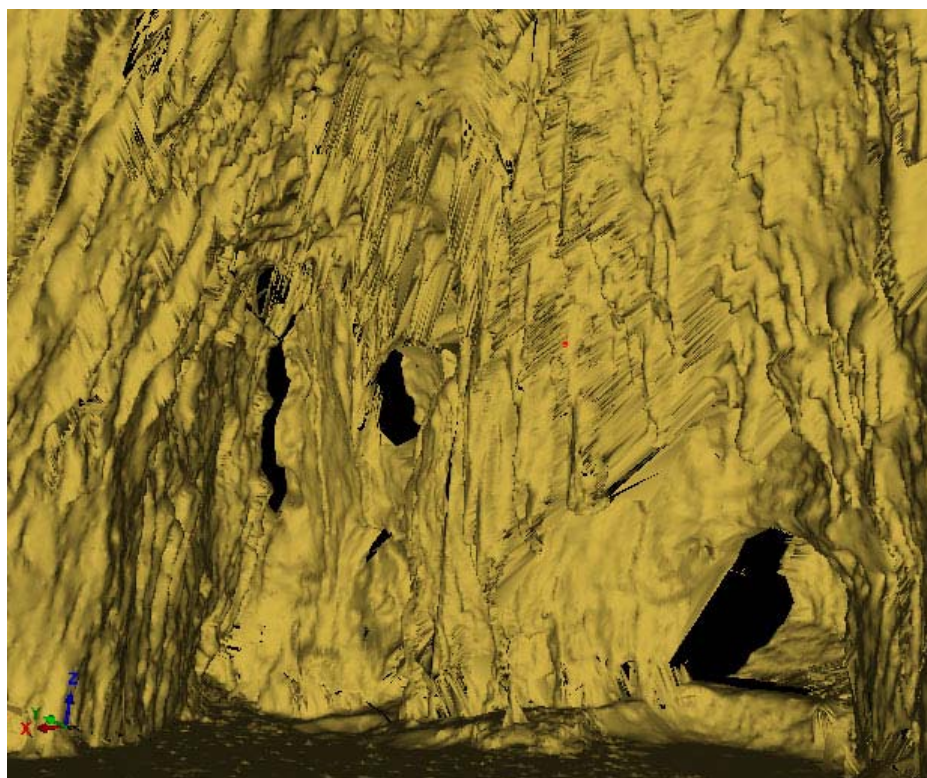
Na slikah 24 levo in 24 desno, vidimo še primere očitnih napak, ki nastanejo zaradi nepravilne postavitve pogleda in združevanja različnih delov modela.



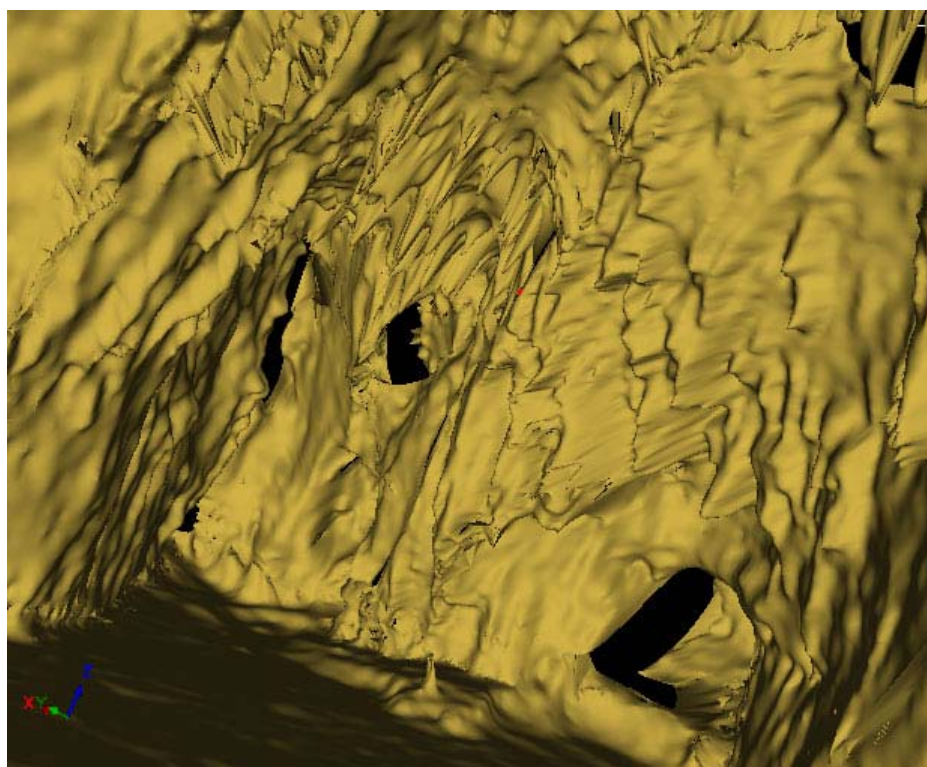
Slika 24: Napake – nepravilno triangulirana ograja (levo), "šiv" ob stiku dveh delov modela (desno)

Vse te napake odstranimo po istem postopku, kar zahteva ogromno porabljenega časa in potrpljenja. Kljub vsemu pa modela nisem uspel idealno triangulirati. To pomeni, da so še zmeraj prisotni nekateri deli, ki vsebujejo majhne luknjice, kar pa je posledica metode triangulacije.

Ko sem imel trianguliran celoten model, ga je bilo treba še zgladiti (ang. *smooth*). To pomeni, da je program zgladil ostre robove, kar povroči "mehkejšo" sliko modela. Razlika je vidna na sliki 25, kjer je prikazan model brez uporabe algoritma za glajenje in sliki 26, kjer je prikazan zglajen model.



Slika 25: Model brez uporabe algoritma glajenja



Slika 26: Zglajen, končen model

4.5 Animacija modela

Ko je model v celoti oblikovan tako, da ne odkrijemo več večjih napak, ga še animiramo. To je najbolj primeren način prikaza za razne prezentacije in podobno, kajti pri prostem gibanju po modelu hitro zaidemo preveč v stene modela ali iz njega ali pa se izgubimo. V RiSCAN PRO-ju je izdelava animacije dokaj preprosta. Temelji pa na vnaprej določeni poziciji navidezne kamere, pogleda in njene orientacije. Princip je tak, da definiramo dva pogleda, se pravi začetni pogled modela in pa končni pogled modela, vmes pa se po različnih metodah izračuna gibanje navidezne kamere. Giblje se lahko po premici, krivulji ali pa po krožnici. V vsakem primeru pa moramo programu določiti čas trajanja animacije od prve do druge točke in pa koliko vmesnih točk naj uporabi, da bo animacija tekla gladko. Ko smo to določili, lahko postavimo zopet nasledno pozicijo kamere, kjer se zopet izračuna pot med prejšnjim pogledom in novim ter tako naprej. Po določitvi vseh pozicij, si lahko ogledamo animacijo in opravimo morebitne spremembe v njej.

Animacijo shranimo v formatu *.avi*, ki je edini format, ki ga program RiSCAN PRO podpira. Pri tem pa je na voljo spet nekaj parametrov, glede na to, kakšno kakovost animacije želimo. Določimo lahko npr. število slik v sekundi (prednastavljeno je 25 slik/s), čas trajanja celotne animacije (pomnožimo čas trajanja celotne animacije z nekim številom), dimenzijo in ločljivost slike ter tudi katere "kodeke" bomo uporabili pri kompresiji video datoteke. Kodeki so algoritmi za kodiranje in dekodiranje video datotek. Če kodekov izrecno ne določimo, program izbere tiste, ki so nameščeni na računalniku na katerem delamo.

Program je porabil za izdelavo moje animacije, ki traja 1min 40s, približno 3 ure, kar pomeni, da je to kar zahteven proces. Pridobil pa sem datoteko velikosti 594 MB, ki sem jo nato še pretvoril v format *.mpeg*, s programom *Blaze Media Pro*. Ta format zapisa zasede bistveno manj prostora in sicer se velikost datoteke zmanjša na 30 MB. Tako dosežemo ob želeni kakovosti kar se da majhno datoteko, ki jo lahko nato posredujemo preko interneta. Seveda obstajajo tudi drugi formati video datotek (*.wmv*, *.ram*, *.mov*), ki omogočajo še manjše velikosti datotek, vendar pa ne izpolnjujejo pogoja kakovosti videa.

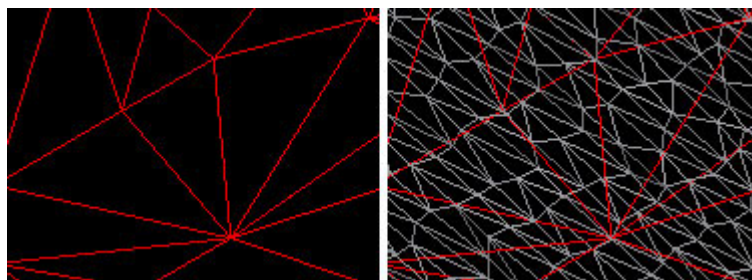
S programom *Blaze Media Pro* sem uredil in dodal video datoteki tudi zvok (format *.mp3*), ki nam nudi občutek dodatnega ugodja ob gledanju.

4.6 Modifikacija modela in VRML zapisa za potrebe internetne objave

V nalogi sem si zastavil cilj, da celoten objekt prikažem tudi prek interneta. Zaradi težav pri takem prikazu, opisanih v poglavju 3.3.2, moram ustvarjeni model modificirati. RiSCAN PRO ima možnost izvoza urejenega modela v obliki VRML datoteke (format *.wrl*). Vendar, če izvozimo model s tako stopnjo detajla, kot je narejena npr. animacija, nastane datoteka velikosti 208 MB, kar je občutno preveč. Ena VRML datoteka naj bi bila namreč velika maksimalno okoli 1.5 MB, da se še naloži na uporabnikov računalnik v nekem razumnem času. Torej moramo kar krepko poseči v model in nato tudi v sam zapis datoteke.

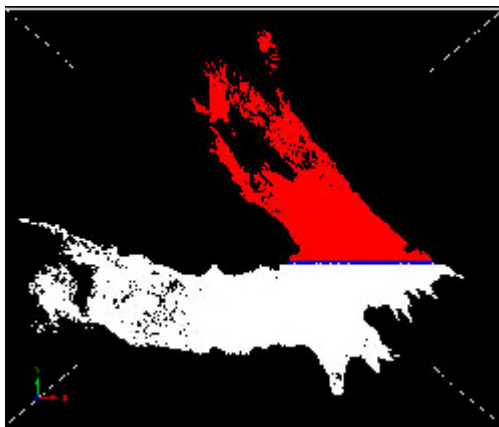
Po preučitvi delovanja VRML-ja in trikov na spletu, sem se problema lotil tako, da sem najprej zmanjšal stopnjo detajla samega modela v RiSCAN PRO-ju, ga razdelil na dva dela (dva kraka) in vsak del posebej izvozil kot VRML datoteko. Model se izvozi kot sklop imenovan *IndexFaceSet*. Na koncu sem še optimiziral sam zapis VRML kode teh dveh datotek in ju med seboj spet združil v tretji datoteki, imenovani *index.wrl*, od koder poteka tudi celotno nadzorovanje obnašanja modela (osvetlitev, pogledi, orientacija).

Prva stopnja zmanjševanja podatkov v modelu in s tem velikosti datoteke, je zmanjšanje detajla že v programu, s katerim sem ustvaril model. Program ima filter z imenom *decimate*, ki lahko zmanjša skupno število točk za več kot 90%. Deluje tako, da glede na predhodno določeno stopnjo redukcije, zmanjša število točk v modelu in s tem število trikotnikov, ki tvorijo površino modela (slika 27). Zaradi tega se močno zmanjša tudi stopnja detajla modela.



Slika 27: Model pred uporabo filtra *decimate* (desno) in rezultat uporabe filtra (levo)

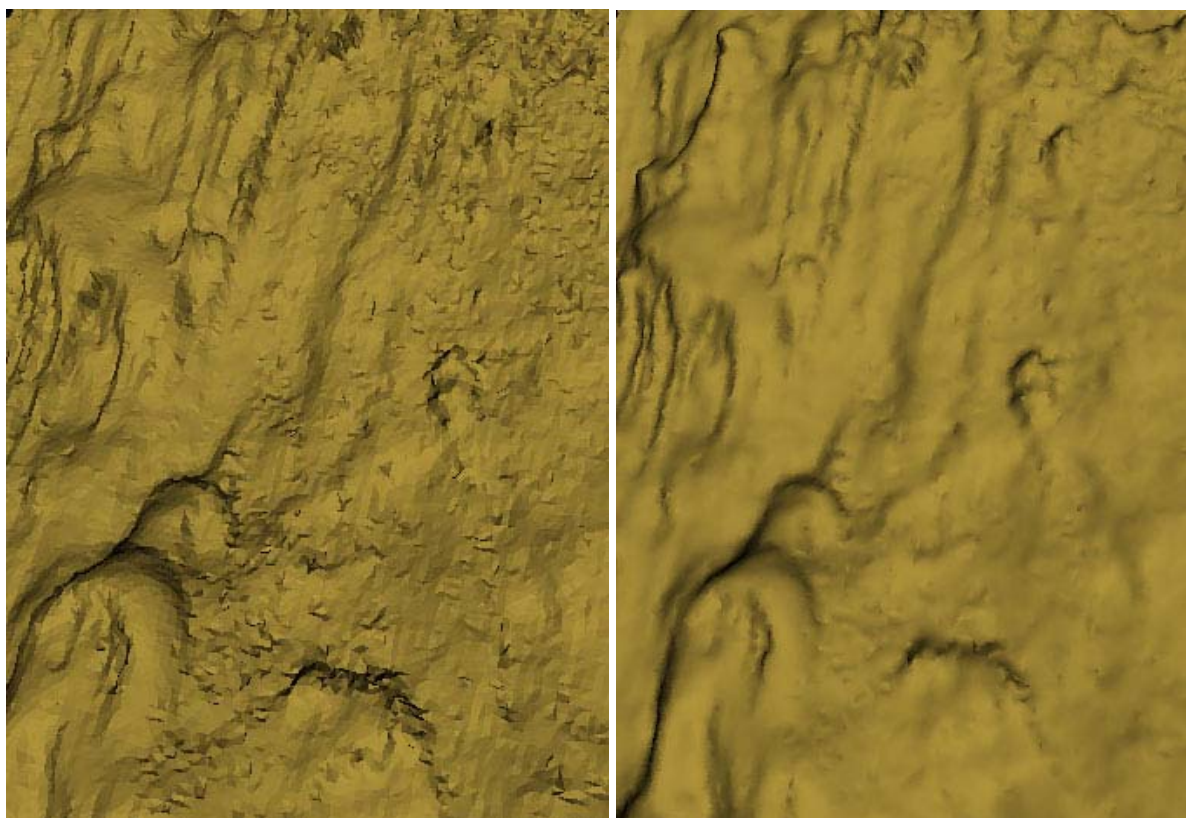
Tako modificiran model sem razdelil še na dva dela (slika 28) in ju izvozil v formatu *.wrl*. Koliko se zmanjša posamezna izvozena datoteka, vidimo v preglednici 2, ki prikazuje velikost *.wrl* datotek po posamezni fazi redukcije modela.



Slika 28: Razdelitev celotnega modela na dva dela (desni in levi krak)

Izvozeni datoteki posameznega kraka sem obdelal še s programom VIZup, podjetja VizUp Tehnology, ki se ukvarja z razvojem algoritmov za redukcijo poligonov in vizualizacijo CAD/CAE modelov. Program je narejen za izdelavo stopnje detajla (LOD) datotek *.wrl*. Deluje pa tako, da v VRML kodi zmanjša natančnost zapisa koordinat posameznih točk (zmanjša število decimalnih mest). Tu sem obema modeloma zmanjšal stopnjo detajla še za 20%. Taka stopnja namreč še ne odstopa preveč od stopnje detajla vhodnega modela.

Za tem je sledila ročna obdelava datotek. Veliko VRML kode je namreč nepotrebne, npr. indeksi barve posameznega verteksa (ang. *color node*) in presledki, zato sem jo izbrisal. Nekaj kode pa sem moral še dodati. Do sedaj ustvarjen VRML model je namreč zelo grob. To pomeni, da so vidni ostri robovi med posameznimi trikotniki, ki tvorijo površino modela (slika 29 levo). Da sem to odpravil, sem uporabil ukaz *creaseAngle*, ki zgladi te robove in tako postane model lepo gladek in prijetnejši na pogled (slika 29 desno).



Slika 29: VRML model brez (levo) in z uporabo (desno) *creaseAngle*

Zadnja faza redukcije velikosti VRML datotek pa je stiskanje (ang. compress) s programom Gzip. Te stisnjene datoteke, za razliko od navadnih stisnjenih datotek *zip* in *rar*, VRML brskalnik avtomatsko prepozna, zato jih lahko uporabljamo kot navadne VRML datoteke. Edino kar moramo storiti je, da preimenujemo končnico datoteke iz *.gz*, ki jo ustvari Gzip program, v navadno končnico VRML datoteke *.wrl*. Na ta način se velikost datotek zmanjša v povprečju za še dodatnih 70%.

Preglednica 2: Posamezne faze redukcije izvozenih VRML datotek in njihove velikosti

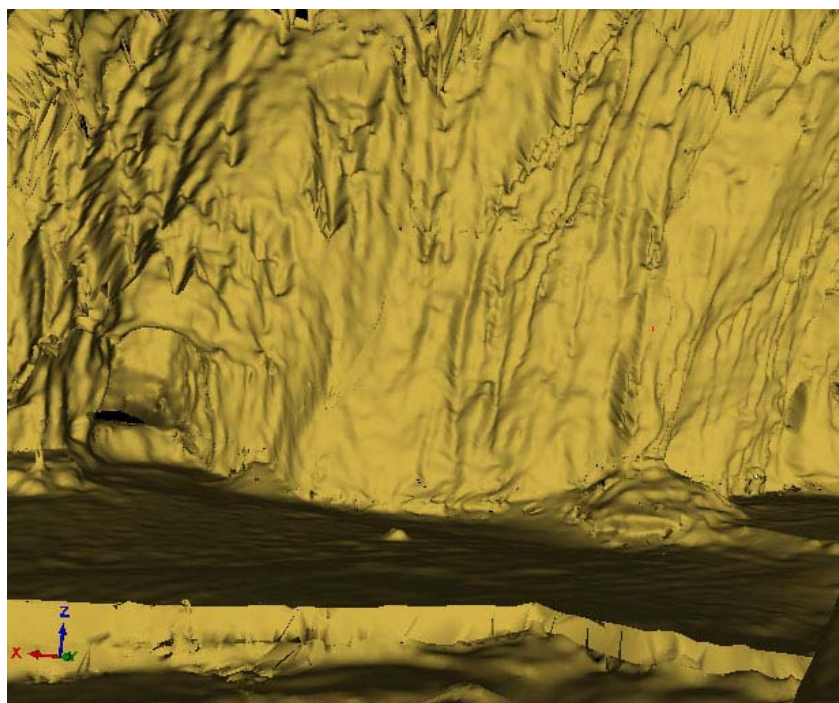
Faza redukcije	Velikost dat. [MB]	
	- Desni krak	- Levi krak
Izvoz iz RiScan-a - original model	141.489	67.431
Izvoz iz RiScan-a - reduciran model	13.241	6.612
Izbris <i>color node</i>	11.589	5.771
VIZup redukcija 20%	7.430	3.643
Odstranitev odvečnih presledkov in dodatek dopolnilne kode	5.055	2.465
Gzip	1.481	0.730

Po uspešno izvedeni redukciji obeh datotek sem ustvaril še tretjo datoteko (*index.wrl*), v kateri sem oba dela modela zopet združil. Ta princip razdelitve modela na dva dela povzroči, da se najprej naloži na računalnik prvi del modela, ki je tako že viden na ekranu, za njim pa še drugi del. Tako sem dosegel, da uporabnik veliko hitreje vidi model in se začne z njim "spoznavati", medtem ko se preostali del šele nalaga. Vsi se namreč predobro zavedamo, da je najbolj mučno čakanje na to, da se bo določen dokument prikazal na ekranu, potem, ko enkrat kliknemo na željeno ikono, ki pošlje zahtevo za prikaz tega dokumenta strežniku.

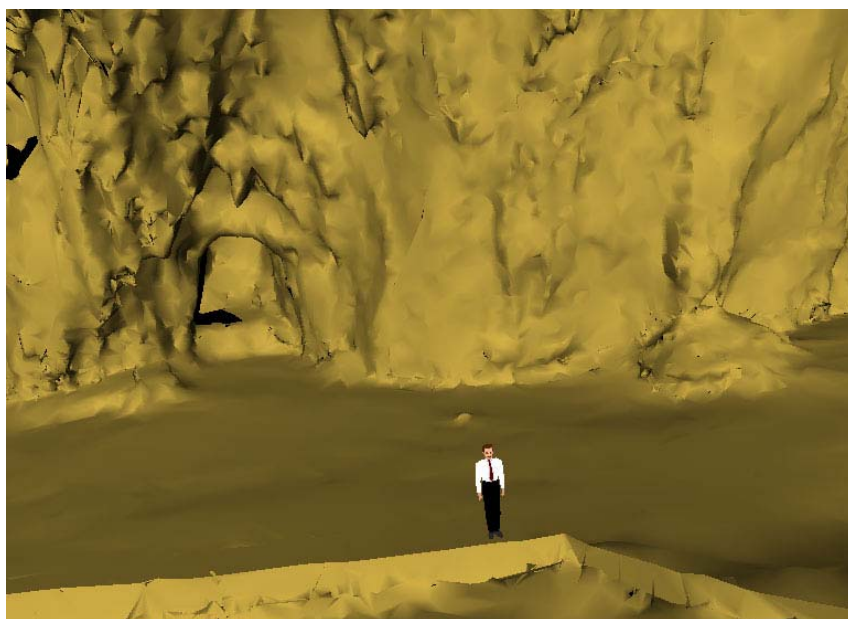
Ko se celoten model naloži na uporabnikov računalnik, lahko ta začne z raziskovanjem modela. Tu pa zopet nastopi težava in sicer, računalnik ne more obdelati tako velike količine podatkov v realnem času, ampak z manjšim zamikom. Zaradi tega se zgodi, da uporabnik prehitro zaide preblizu sten modela ali pa celo izven modela. Tako sem moral izdelani virtualni model še oplemenititi z v naprej pripravljenimi akcijami, ki nadzirajo obnašanje modela. To so predvsem v naprej pripravljeni pogledi modela in pa senzorji, ki zaznavajo bližino sten modela (*proximity sensor*). Celotno nadzorovanje obnašanja modela se izvaja iz datoteke *index.wrl*.

Sedaj je model pripravljen za objavo na spletu. Ogledate si ga lahko na spletni strani podjetja DFG Consulting d.o.o. (<http://www.dfgcon.si>).

Redukcija datotek je bila torej možna predvsem na račun stopnje detajla modela (slika 31). Zaradi tega tako kompleksni in obsežni modeli niso najbolj primerni za prikazovanje prek interneta na interaktiven način, če želimo prikazati njegove drobne detajle, kot so v mojem primeru kapniki. Tudi navigacija po takem modelu ne poteka tekoče. Vseeno pa dobi uporabnik dober občutek plastičnosti, kar je bil tudi moj namen.



Slika 30: Model izdelan v RiSCAN PRO-ju



Slika 31: VRML model

5 ZAKLJUČEK

Z izdelavo diplomske naloge sem pridobil znanje o izdelavi 3R virtualnega modela vse od zajema podatkov s 3R terestričnim laserskim skenerjem, preko obdelave zajetih podatkov in tehnik modeliranja, vse do končne izdelave animiranega prikaza izdelanega modela in nato distribucije preko interneta v VRML zapisu.

Skeniranje je izvedlo podjetje DFG Consulting d.o.o. iz Ljubljane, dne 13.07.2005. Zajetih je bilo 15.326.267 točk iz štirih stojišč. Uporabljeni sistem 3R terestričnega laserskega skeniranja je bil Riegl LMS-Z420i, ki je sestavljen iz laserskega skenerja in pripadajočega računalniškega programa Riscan Pro za krmiljenje skenerja in kasnejšo obdelavo podatkov vse do končnega izdelka. Opcijsko lahko pritrdimo na inštrument tudi visokoločljivi digitalni fotoaparati, s čimer združimo prednosti tako laserskega skeniranja kot fotogrametrije. V tem primeru fotoaparata nismo uporabljali.

Posebnost izdelanega modela je v tem, da je zelo obsežen, to pomeni, da vsebuje veliko število zajetih točk, vsebuje ogromno detajlov, kapnikov, ki so problematični za triangulacijo. Zaradi tega je bilo potrebno uporabiti različne metode zmanjševanja točk (*octree*), postopno trianguliranje, ki je zelo dolgotrajno, in na koncu še algoritem za glajenje izdelanega modela (*smooth*).

Za prezentacijo modela naročniku smo izdelali dinamično predstavitev (animacija), ki smo jo dodatno opremili z glasbo, kar je bilo izvedeno s programom *Blaze Media Pro*. Želja naročnika je bila tudi, da se omogoči interaktivno premikanje po modelu, kar sem izvedel s pomočjo internetnega standarda za 3R vizualizacijo, imenovanega VRML (Virtual Reality Modeling Language). Ker je bila datoteka, ki smo jo direktno izvozili iz programa Risca Pro v *.wrl* formatu (VRML format) prevelika za objavo na internetu (počasno nalaganje na uporabnikov osebni računalnik in nezmožnost brskalnika zagotoviti ustrezno hitre navigacije), sem jo moral zmanjšati. To zmanjšanje pa je bilo mogoče le na račun zmanjšanja stopnje detajla. Nekaj pa je k zmanjšanju prispeval tudi sam zapis VRML datoteke in pa uporaba programa za stiskanje datotek Gzip.

Testni primer modela jame je podal pričakovane rezultate tako v kakovosti izdelanega modela kot tudi v potrditvi smiselnosti izvajanja skeniranja v turistične namene. V prihodnosti je potrebno opremiti VRML model še z modulom za interaktivno merjenje v modelu, kar nam prinese resnično uporaben model za različne analize in načrtovanja, njegovo posredovanje širokemu krogu uporabnikov pa je v takem formatu praktično neomejeno, če ima le uporabnik dostop do svetovnega spleta. Sam program za obdelavo pridobljenih podatkov pa bi bilo potrebno dopolniti z algoritmi, ki bi nekatere zamudne postopke avtomatizirali. S tem mislim predvsem triangulacijo, kjer bi bilo mogoče uporabiti namesto 2R Delaunayeve triangulacije algoritem za 3R triangulacijo, na primer 3R Delaunayevo triangulacijo. Tako ne bi bilo potrebno več spreminjati pogledov na točke in jih postopno triangulirati.

VIRI

Uporabljeni viri

Dorninger, P. 2005. Terrestrial Laser Scanners for Documentation and Modeling of Historical Rooms. (Seminar - CD)

Jansa, J., Studnicka, N., Forkert, G., Haring, A., Kager, H. 2004. Terrestrial laserscanning and photogrammetry – acquisition techniques complementing one another. Pridobljeno s spletne strani:

http://www.ipf.tuwien.ac.at/publications/ISPRS_2004_Istanbul/Terr_Laserscanning_jj_ah_hk.pdf (22.11.2005)

Lenardič, J. 2002. Razvoj virtualnih svetov odpira nove možnosti poslovanja. Pridobljeno s spletne strani: http://www.cek.ef.uni-lj.si/u_diplome/lenardic304.pdf (29.08.2005)

Program RiSCAN PRO – pomoč

Vrenko, D., Petrovič, D. 2005. Izdelava trirazsežnega modela mesta Slovenske konjice. Pridobljeno s spletne strani:

http://www.geodetski-vestnik.com/49/2/gv49-2_231-240.pdf (30.08.2005)

Zlatanova, S. 2002. 3D GIS for Urban Development. Pridobljeno s spletne strani: <http://www.gdmc.nl/zlatanova/PhDthesis/html/content.html> (20.07.2005)

Wagner, W. 2005. The Laser Beam and Its Interaction with Different Types of Surfaces. (Seminar - CD)

Riegl laser measurement systems – spletna stran

http://www.riegl.com/terrestrial_scanners/lms-z420i_/420i_all.htm (15.09.2005)

Leica geosystems – spletna stran

http://hds.leica-geosystems.com/products/acc_tgt.html (16.10.2005)

Ostali viri

Ames, A. L., Nadeau, D. R., Moreland, J. L. 1997. VRML 2.0 Sourcebook, druga izdaja. New York, J. Wiley & Sons, Inc.: 654 str.

Mozetič, B. 2004. Terestrično 3D (trirazsežno) lasersko skeniranje. Pridobljeno s spletne strani: http://www.geodetski-vestnik.com/48/3/gv48-3_351-362.pdf (11.10.2005)

Pahor, D. 2002. Leksikon računalništva in informatike. Ljubljana: Pasadena. 786 str.

Santala, J., Joala, V. 2003. On the Calibration of a Ground-based Laser Scanner. Pridobljeno s spletne strani:

http://www.fig.net/pub/fig_2003/TS_12/TS12_4_Santala_Joala.pdf (11.10.2005)

Navidezna resničnost in VRML - spletna stran

http://projekti.svarog.org/navidezna_resnicnost/ (10.11.2005)

Web 3D cosortium - spletna stran

<http://www.web3d.org/index.html> (10.10.2005)

PRILOGA A: IZSEK KODE ZAPISA MODELA V VRML FOMATU

```
#VRML V2.0 utf8
#File exported by RiSCANPRO
#1 blocks
# This file was optimized with VIZup Free-Edition v 1.8.
# http://www.vizup.com/
```

```
Transform
```

```
{
  children
  [
    WorldInfo
    {
      info [ "Input Format: vtp" ]
    }
    Viewpoint
    {
      position 0.0 0.0 0.0
      orientation 0.0 0.0 1.0 0.0
    }
    DirectionalLight {
      ambientIntensity 0.7
      color 1 1 1
      direction 0 0 -1
      intensity 0.2
      on TRUE
    }
  ]
# Block 1
Shape
{
  appearance
```

Appearance

{

material

Material

{

diffuseColor 0.9 0.7 0.27

transparency 0

}

}

geometry

IndexedFaceSet

{

coord

Coordinate

{

point

[

26.1576 35.2845 -15.0175,

26.5332 35.0153 -14.9286,

27.1929 35.191 -14.4551,

28.5036 32.8645 -13.707,

27.1219 34.2359 -14.5913,