

Univerza  
v Ljubljani  
Fakulteta  
*za gradbeništvo  
in geodezijo*

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



Visokošolski program Geodezija,  
Smer za prostorsko informatiko

Kandidat:  
**Marko Zore**

## **Izdelava 3D-modela mesta za spletno uporabo**

**Diplomska naloga št.: 237**

**Mentor:**  
izr. prof. dr. Radoš Šumrada

Ljubljana, 28. 9. 2007

## **POPRAVKI**

## IZJAVA O AVTORSTVU

Podpisani **MARKO ZORE** izjavljam, da sem avtor diplomske naloge z  
naslovom: »**IZDELAVA 3D-MODELA MESTA ZA SPLETNO UPORABO**«.

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

Ljubljana, **XXXX**

---

(podpis)

## **BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK**

**UDK:** 004.738.5:004.92:711.4(043.2)

**Avtor:** Marko Zore

**Mentor:** izr. prof. dr. Radoš Šumrada

**Naslov:** Izdelava 3D-modela mesta za spletno uporabo

**Obseg in oprema:** 42 str., 6 sl.

**Ključne besede:** 3D-model, X3D-jezik, modeliranje reliefa in stavb

### **Izvleček:**

Naloga opisuje izdelavo tridimenzionalnega modela mesta z namenom uporabe na spletu. Model je izdelan v VRML-jeziku, oz. v sodobnejšem X3D-jeziku. Pri izdelavi modela se teži k cenovno ugodnim rešitvam, zato je postopek izdelave usmerjen v uporabo zastonjskih programov, ki so na voljo na spletu. Za prikaz reliefa se uporabi kombinacijo DMR in DOF posnetke, stavbe pa se izdeluje posamično v preprostem zastonjskem programu za modeliranje. Pri izdelavi se teži k iskanju kompromisa med vizualno kakovostjo modela in možnostjo zadovoljivega prenosa podatkov prek spleta.

## **BIBLIOGRAPHIC – DOCUMENTALISTIC INFORMATION**

**UDC:** 004.738.5:004.92:711.4(043.2)

**Author:** Marko Zore

**Supervisor:** Assoc. Prof. dr. Radoš Šumrada

**Title:** Making 3D-city model for web use

**Notes:** 42 p., 6 fig.

**Key words:** 3D-model, X3D-language, relief and buildings modelling

### **Abstract:**

This dissertation describes making a three-dimensional city model for web use. Model is made in VRML-language, respectively in X3D-language. At making, we tend to find low-price solutions, so we use freeware applications, available on the web. For relief visualization we use combination of DEM and DOF (digital orthophoto) snapshots. Buildings are separately made in simple freeware modelling application. What we also tend to reach at making the model is a compromise between visual quality of the model and satisfying data transmission over the web.

### **ZAHVALA**

Zahvaljujem se mentorju izr. prof. dr. Radošu Šumradi za korekten odnos in pomoč pri izdelavi diplomske naloge.

Nadalje se zahvaljujem staršem za finančno ter moralno podporo pri študiju, tako staršem kot dekletu Tjaši pa se zahvaljujem za potrpežljivost in zaupanje.

Zahvalil bi se tudi dobremu kolegu Dominiku Fajdigi, ki je bil vedno pripravljen svetovati in pomagati, tako pri študiju, kot tudi pri drugih stvareh.

Iskrena hvala vsem!

## KAZALO VSEBINE

<b>1</b>	<b>UVOD.....</b>	<b>1</b>
<b>2</b>	<b>UPORABLJENA INFORMACIJSKA TEHNOLOGIJA .....</b>	<b>2</b>
2.1	Internet .....	2
2.2	VRML .....	4
2.2.1	Splošen opis .....	4
2.2.2	Primer in pravila zapisa VRML .....	4
2.3	XML .....	7
2.4	X3D .....	8
2.4.1	Splošen opis .....	8
2.4.2	Primer X3D zapisa .....	8
2.4.3	Pravila XML/X3D zapisa .....	9
2.5	Osnove rastrskih podob .....	10
2.5.1	Rastrska podoba in barvni modeli .....	10
2.5.2	Lastnosti shranjevanja bitnih podob .....	11
2.5.3	Stiskanje bitnih podob .....	13
<b>3</b>	<b>UPORABLJENA GEOINFORMACIJSKA TEHNOLOGIJA .....</b>	<b>14</b>
3.1	Digitalni model reliefa .....	14
3.1.1	Oblike DMR .....	14
3.1.2	Zajem podatkov za izdelavo DMR .....	16
3.1.3	Uporabnost DMR .....	17
3.1.4	Glajenje in filtriranje digitalnega modela reliefa .....	20
3.2	Digitalni ortofoto .....	21
<b>4</b>	<b>UPORABLJENE APLIKACIJE .....</b>	<b>23</b>
4.1	Brezplačne aplikacije (freeware) .....	23
4.2	Zastonjski program za 3D modeliranje – Wings 3D .....	24
4.3	Zastonjski X3D urejevalnik – X3D EDIT .....	26
<b>5</b>	<b>POTEK IZDELAVE MODELA .....</b>	<b>27</b>
5.1	Izdelava stavb .....	27
5.1.1	Geometrično oblikovanje .....	27

<b>5.1.2</b>	<b>Izdelava tekstur .....</b>	<b>28</b>
<b>5.2</b>	<b>Prikaz površja v X3D obliki .....</b>	<b>29</b>
<b>5.3</b>	<b>Elementi animacije in drugi elementi v X3D .....</b>	<b>31</b>
<b>5.4</b>	<b>Združitev vseh elementov v končni model .....</b>	<b>32</b>
<b>5.5</b>	<b>Zmanjševanje prostorske obsežnosti podatkov .....</b>	<b>34</b>
<b>5.5.1</b>	<b>Krčenje bitne vrednosti podatkov.....</b>	<b>35</b>
<b>5.5.2</b>	<b>Variranje uporabe podatkov glede na vidnost .....</b>	<b>35</b>
<b>5.6</b>	<b>Postavitev modela na splet.....</b>	<b>36</b>
<b>6</b>	<b>OPIS POSTOJNE .....</b>	<b>37</b>
<b>7</b>	<b>ZAKLJUČEK .....</b>	<b>40</b>
<b>VIRI:</b>	<b>.....</b>	<b>41</b>



## KAZALO SLIK

Slika 1:	Primer prikaza VRML-zapisa z VRML-predvajalnikom .....	6
Slika 2:	Prikaz uporabe metode »UV mapping« v programu Wings 3D.....	29
Slika 3:	Prikaz DMR in DOF v X3D obliki z ustreznim prikazovalnikom.....	30
Slika 4:	Razlika med DKS in X3D-koordinatnim sistemom .....	31
Slika 5:	Primer prikaza stavbe v izdelanem 3D-modelu.....	33
Slika 6:	Izsek iz programa X3D - Edit.....	34

## **OKRAJŠAVE IN SIMBOLI**

CMYK	Subtraktivni barvni model
DMR	Digitalni model reliefa
DOF	Digitalni ortofoto posnetek
GIS	Geografski informacijski sistem
HTML	Markirni ali označevalni jezik (ang. Hypertext markup language)
ISO	Mednarodna organizacija za standardizacijo (ang. International organization for standardization)
IT	Informacijska tehnologija
RGB	Aditivni barvni model
URI	Enotni identifikator virov (ang. Uniform resource identifier)
URL	Enotni lokator virov (ang. Uniform resource locator)
URN	Enotno ime virov (ang. Uniform resource name)
X3D	Odprti standardni format za predstavitev in prenos 3D-scen in objektov
XML	Standardizirani jezik za zapis elektronskih dokumentov (ang. Extensible markup language)
VRML	Jezik za modeliranje v navidezni resničnosti (ang. Virtual reality modeling language)

# 1 UVOD

Internet je s svojim nastankom in razvojem popolnoma spremenil človeški vsakdan. Omogočil je komunikacijo s praktično vsakim obljudenim kotičkom na zemlji, omogočil je dostopnost in ustvarjanje ogromnih baz podatkov in informacij iz katerih se lahko učimo, informiramo in širimo svoja obzorja. V gospodarstvu je poenostavil poslovno komuniciranje, finančno poslovanje in podobno. Marsikateri del človeškega življenja je povezan z uporabo interneta in računalnika na sploh, lahko bi rekli, da je računalnik postal vsakdan povprečnega sodobnega človeka.

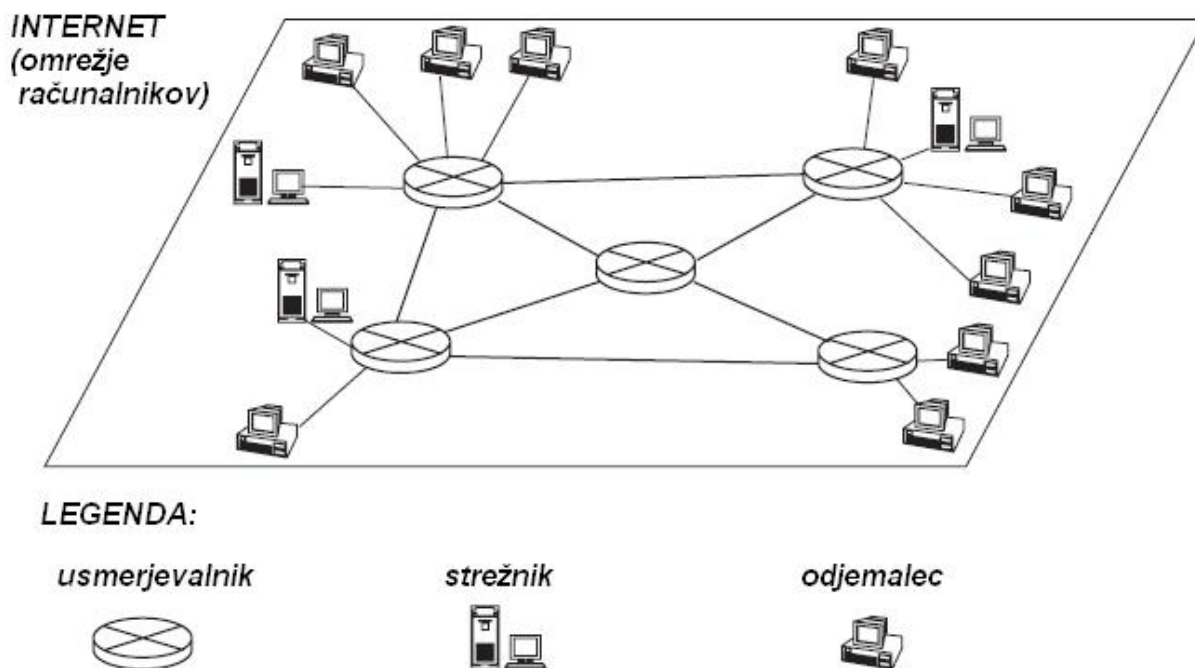
Ena izmed mnogih uporabnih funkcij informacijske tehnologije (IT) je obvladovanje prostorskih podatkov. S pomočjo IT izdelujemo prostorske prikaze vseh vrst in na njih upodablamo pravna, gospodarska, fizična, socialna in mnogo drugih dejstev, ki omogočajo njihovo obvladovanje in upravljanje. Aplikacije ali sisteme, ki omogočajo prikaz in poizvedovanje po prostorskih dejstvih, imenujemo geografski informacijski sistemi (GIS). Takih sistemov je predvsem zaradi hitrega napredka tehnologije vedno več, z povečevanjem zmogljivosti prenosa podatkov tudi na spletu. Poznamo vrsto aplikacij, ki tudi na spletu omogočajo načrtovanje potovanj, opozarjajo na stanje na cestah in prikazujejo stanje vremena v dejanskem času, omogočajo vodenje (tudi med vožnjo s povezavo GPS) potujočega od startne do ciljne lokacije. Obstajajo aplikacije, ki razkrivajo mnogo lokacij na svetu in predstavljajo njihove turistične znamenitosti, hotelske zmogljivosti, restavracije in druge zanimivosti, ki bi potencialne obiskovalce že iz domačega naslonjača prepričale, da bi obiskali prav to predstavljeno lokacijo. V kmetijstvu se uporabljajo aplikacije, ki omogočajo učinkovito upravljanje nad kmetijskimi in gozdnimi zemljišči, občine pa vzpostavljajo svoje informacijske sisteme za upravljanje svojega pristojnega območja in tako dalje.

Tovrstni prikazi so odvisni predvsem od strojnih omejitev računalnikov in omejitev prenosa informacij. V zadnjih letih se je razvijala predvsem tehnologija dvodimenzionalnega prikaza prostora, ki je danes že povsem uveljavljen način prikazovanja bolj ali manj kompleksnih GIS aplikacij za vse vrste uporabe. Nekaj korakov nazaj pa je prikazovanje prostora v treh dimenzijah, razlog je logičen, število podatkov se poveča kar za eno potenco, kar zaenkrat še ne omogoča istih uporabnih učinkov za prikaz enakega obsega površja, kot pri dvodimenzionalnih prikazih. Kljub temu pa se že nakazujejo smernice uporabe 3D-prikazov tudi za množično uporabo. Zanimiv primer je gotovo podjetje Google, ki s svojo aplikacijo Google Earth omogoča relativno zelo natančen pogled na praktično celoten planet, vse to pa omogoča ogromno število prostorskih podatkov, ki jim jih je uspelo poslati v praktično vsak dom, kjer imajo solidno povezavo z internetom in seveda željo po ogledu. Kljub temu pa, da so površje in nekatera glavna mesta prikazana v 3D obliki, še težko govorimo o pravi 3D-spletni aplikaciji. Veliko bolj konkretni so trenutno virtualni svetovi oz. 3D-spletne klepetalnice, ki dejansko združujejo množico ljudi v 3D prostoru, a tu govorimo o izmišljenih prostorih, ki ne predstavljajo prikaza dejanskega prostora. Vsem tem 3D »spletnim nakazovalcem prihodnosti« se pridružuje tudi ta diplomska naloga, ki poizkuša z X3D-standardom in zastonskimi spletnimi aplikacijami soustvarjati spletno prihodnost 3D-prikazov površja.

## 2 UPORABLJENA INFORMACIJSKA TEHNOLOGIJA

### 2.1 Internet

Internet (tudi medmrežje, svetovni splet) je v bistvu računalniško omrežje, ki povezuje množico manjših računalniških omrežij. Gre za najvišji možni nivo računalniškega omrežja. Če se nekdo želi priključiti na omrežje, prek katerega lahko sprejema informacije iz vseh možnih razpoložljivih virov, pravimo, da se želi priključiti na internet. Ker gre za omrežje, potujejo podatki iz posameznega priključenega računalnika do vozlišč, ki združujejo več računalnikov na določenem območju. Vozlišča sprejema vse podatke iz svojega območja in jih preusmerja tako znotraj območja kot tudi proti drugim vozliščem, ki pokrivajo druga območja. Vozlišča podatke preusmerjajo s pomočjo posebne naprave, ki ji rečemo »usmerjevalnik«. Usmerjevalniki pa podatke ne le usmerijo, vendar ji določijo tudi najkrajšo oziroma najhitrejšo pot, do željenega cilja.



*Shema računalniškega omrežja – Interneta  
(Hercog Drago: Svetovni splet, [www.fe.uni-lj.si/~hercog/html.pdf](http://www.fe.uni-lj.si/~hercog/html.pdf))*

Glede na to, ali podatke nudimo, ali pa po njih povprašujemo, ločimo računalnike na »odjemalce« in »strežnike«. V osnovi je lahko vsak računalnik lahko eno ali drugo, vendar računalnik pri katerem povprašujemo po podatkih imenujemo strežnik, računalnik s katerim dostopamo do strežnika pa nastopa v vlogi odjemalca, saj bo prejete podatke prikazal. (Turk, 1998)

Da pa lahko praktično ves svet nemoteno komunicira v dejanskem času in da vse gladko teče, pa je bilo potrebno dolgotrajno usklajevanje različnih metod, načinov in pripomočkov za komunikacijo. Poskrbeti je bilo treba za standardizacijo. Eden pomembnejših korakov k standardizaciji je bil storjen leta 1974 ko so definirali TCP/IP protokol (Transmission Control Protocol/Internet protocol). TCP/IP je običajen protokol za prenos podatkov med računalniki v internetu. Je standard, ki ga je moč preprosto uporabljati, zato se je hitro uveljavil in je v

uporabi še danes. TCP/IP sta pravzaprav dva standarda, ki pa se ju najpogosteje uporablja skupaj (Hoffman, 1996)

Da pa v omrežju ločimo posamezne računalnike med seboj, potrebujemo tudi ustrezno in učinkovito poimenovanje posameznega računalnika. Za programsko opremo, ki operira z imeni računalnikov, so najbolj učinkovita rešitev imena sestavljena iz številke. Tako so uvedli IP številke. Gre za štiri števila v razponu od 0 do 255, ločena s piko (npr. 195.3.46.3). Vsak računalnik v omrežju ima svojo IP številko, tako da se povsem loči od drugih. Tak način poimenovanja omogoča določiti več kot štiri milijarde različnih naslovov, a s porastom računalnikov že zmanjkuje številke. (Hoffman, 1996) Za človeka, ki uporablja splet, pa je ta številka nekoliko enolična za uporabo, IP številka pravzaprav nič ne govori o tem, katera številka predstavlja kateri računalnik.

Zato je bil uveden URI (ang. Uniform Resource Identifier) oz. »enotni identifikator virov«. Gre za enoten način za poimenovanje dokumentov in drugih virov na internetu ali v lokalnem omrežju v obliki črk in besed. Te pa človeku veliko več povedo o računalniku po katerem poizvedujejo. URI lahko uporabljamo v dveh oblikah: kot URL (ang. Uniform Resource Locator) oz. »enotni lokator virov«, ali kot URN (ang. Uniform Resource Name) oz. »enotno ime virov«. URL in URN imata podoben pomen, vendar delujeta na drugačen način.

URL (npr. <http://www.spletna-stran.si>) je sestavljen iz dveh ključnih delov: iz lokacije vira ([www.spletna-stran.si](http://www.spletna-stran.si)) in metode, s katero lahko do vira dostopamo (<http://>). Tak način označevanja vira je zelo praktičen, saj URL običajno vsebuje vse podatke, ki jih potrebujemo za odkritje na tak način označenega dokumenta. URL je relativno enostavno uporabljati, zato je v današnjem času pri spletni uporabi precej razširjen.

URL povezuje vir in lokacijo v nerazdružljivi povezani obliki. To pa včasih ni najboljše, saj vir in lokacija nista ista stvar. Problem nastane pri selitvi ali kopiranju nekega vira na drugo lokacijo, vir mora tedaj spremeniti svoj URL. Če ponujamo enolično datoteko na več lokacijah, bo imela ta datoteka toliko različnih URL, kot bo kopij teh datotek. To težavo lahko reši URN, ki datoteko ustrezno poimenuje, nič pa ne govori o lokaciji in metodi dostopa in je tako neodvisna od lokacije. Uporaba URN pa je prav zaradi neodvisnosti od lokacije za uporabo bolj zapletena in zahtevna, zato se v medmrežju ni uveljavila tako izrazito (Kozierok, 2003-2005)

Za prenos podatkov in komuniciranje prek interneta uporabljamo različne pristope. Lahko »brskamo« po spletu, oz. si ogledujemo in uporabljamo spletne strani, lahko pošiljamo elektronsko pošto, se neposredno pogovarjamo z drugimi uporabniki prek spletnih klepetalnic, si lahko izmenjujemo datoteke različnih vrst, itd. Vsak pristop uporablja nekoliko drugačen pristop poimenovanja računalnika s katerim stopamo v stik. Za spletno pošto uporabljamo »naslov elektronske pošte« (npr. [marko@elposta.si](mailto:marko@elposta.si)), v spletni klepetalnici preprosto kliknemo na »vzdevek« osebe, ki nas poveže z njegovim računalnikom, in podobno.

Za prikaz spletnih strani, ki nudijo vedno več informacij, na vedno več različnih načinov, pa se najpogosteje uporablja HTML (Hypertext Markup Language). HTML je markirni ali označevalni jezik, ki ga razumejo spletni brskalniki in nam prikaže kombinacijo teksta, slik, zvoka, in drugih elementov, kot si je zamislil avtor spletne strani.

Začetki interneta sodijo v šestdeseta leta prejšnjega stoletja. Takrat so bila računalniška omrežja precej bolj primitivna in občutljiva. Kot pri mnogo stvareh, je tudi na področju

interneta raziskave sprožila vojska. Obrambno ministrstvo ZDA ustanovi agencijo ARPA (Advanced Research Projects Agency), ki si je kot cilj zadala vzpostaviti omrežje, po katerem bi med posameznimi strateškimi točkami na ozemlju ZDA potekali podatki v digitalni obliki. Skozi raziskave so vzpostavili ARPANET mrežo, katero so skozi čas vedno bolj izpopolnjevali in razvijali. Ker so pri izdelavi ves čas sodelovale tudi posamezne univerze, je postajal projekt vse bolj zanimiv tudi za civilne uporabnike. Tako se že s koncem sedemdesetih let pojavi povezovanje računalnikov posameznih univerz. Število povezanih računalnikov je vse bolj raslo in v začetku devetdesetih let, so se že pojavila vprašanja, kako uporabo velikega omrežja poenostaviti za uporabo in približati sleherniku. Od takrat pa do danes je skokovit razvoj povzročil uporabo interneta v vsakdanji uporabi, vse od zabavnih vsebin, pa do najbolj resnih gospodarskih transakcij in povezovanj (Turk, 1998 in Hoffman, 1996).

Internet je kot del informacijske tehnologije ves čas podvržen hitremu razvoju in spremembami. Razvoj računalniške strojne opreme in telekomunikacijskih naprav in sredstev omogočajo prenesti v istem času vedno več podatkov, s tem pa je omogočen prikaz podatkov s prostorsko obsežnejšimi, a tudi vizualno in vsebinsko bolj bogatimi načini prikaza.

## **2.2 VRML**

### **2.2.1 Splošen opis**

VRML (Virtual Reality Modeling Language) pomeni v slovenskem jeziku "Jezik za modeliranje v navidezni resničnosti". S tem jezikom opisujemo navidezne svetove (Pesek, 2000). Jezik je podoben jeziku HTML, s to razliko, da opisuje HTML predvsem 2D-objekte, VRML pa je bil razvit za opisovanje v treh dimenzijah, katerih je človek najbolj navajen, saj je njegovo naravno okolje sestavljeno iz treh dimenzij. Zato govorimo o navidezni resničnosti, saj s tem jezikom poizkušamo objekte človeku prikazati v najbolj naravni obliki. Prednost jezika je v njegovi enostavnosti, saj okolje, ki ga želimo predstaviti, opišemo zgolj s tekstovno datoteko. Seveda za človeka, ki se prvič sreča z VRML stvar ni povsem enostavna. Poznati mora strukturo in pravila, ki so zapisana v VRML slovnici, da jih lahko potem učinkovito zapisuje.

Avtorja jezika Mark Pesce in Tony Parisi sta prvi osnutek jezika predstavila leta 1994. Najprej je nastal VRML 1.0, ki je omogočal zgolj modeliranje statičnih svetov, dodana podpora k interaktivnosti in dinamičnosti pa je leta 1997 pomenila nastanek VRML 2.0, ali VRML97. VRML se je razvijal v okviru neprofitne organizacije Web3D Consortium, takrat znane še kot VRML consortium. Ta organizacija skrbi za nadaljni razvoj jezikov VRML in novejšega X3D. Oba jezika sta postala tudi odprta standardna datotečna formata (uporabljata pripono .wrl in .x3d) v okviru mednarodne organizacije za standardizacijo – ISO.

Kot že rečeno, je VRML v osnovi tekstovni format, zato ga lahko zapisujemo v običajnih tekstovnih programih.

### **2.2.2 Primer in pravila zapisa VRML**

Vsak dokument lahko vsebuje naslednje gradnike (Pesek, 2000):

- komentarje ( ki jih označimo z #) ,

- objekte ( koščki celotnega navideznega sveta) ,
- attribute objektov ,
- vrednosti atributov ,
- script (java),
- ...

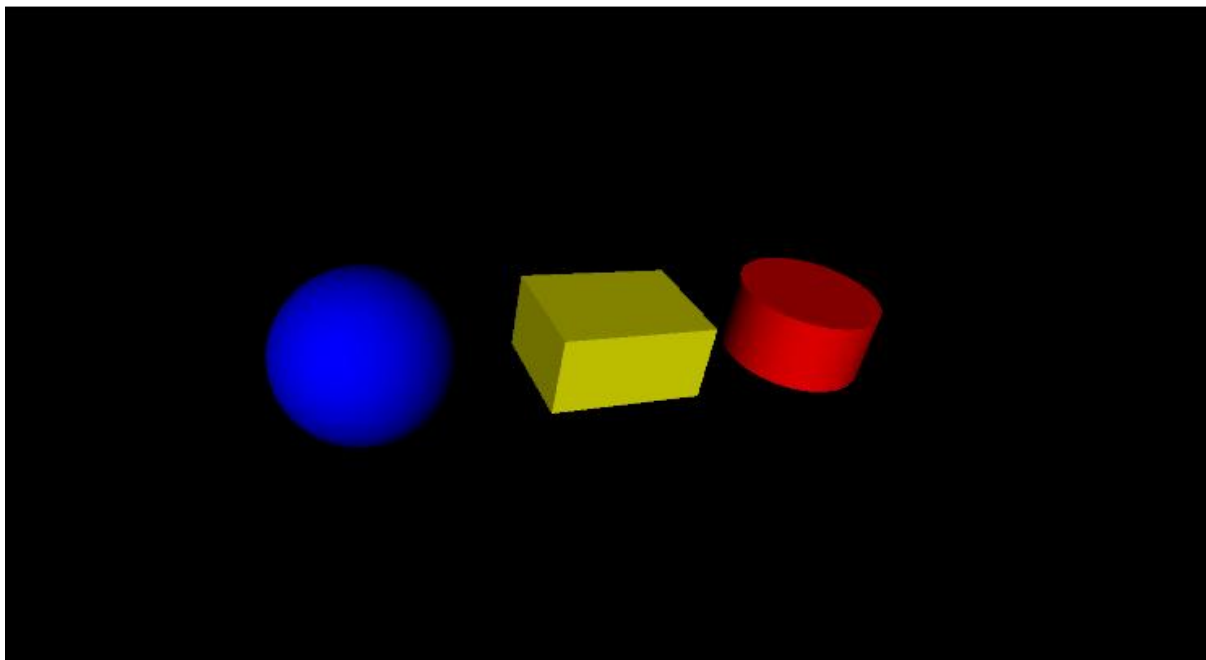
Spodaj je prikazan primer zapisa VRML-dokumenta v tekstovni obliki, s pomočjo katerega smo predstavili nekaj osnovnih geometrijskih teles v 3D-prostoru. VRML-dokument si lahko v taki obliki ogledamo, če ga odpremo z ustrezno tekstovno aplikacijo (npr. Beležnica v Microsoftovih Oknih.) Dokument vsebuje navodila o velikosti, obliki, legi ter barvi posameznega telesa, prolog dokumenta in komentarje:

```
#VRML V2.0 utf8
#izdelali bomo tri telesa
Transform {
    translation 3.0 0.0 0.0
    children[
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 1.0 0.0 0.0
                }
            }
            geometry Cylinder {
                height 1.0
                radius 1.0
            }
        }
    ]
}

Transform {
    translation -3.0 0.0 0.0
    children[
        Shape {
            appearance Appearance {
                material Material {
                    diffuseColor 0.0 0.0 1.0
                }
            }
            geometry Sphere {
                radius 1.0
            }
        }
    ]
}

Shape {
    appearance Appearance {
        material Material {
            diffuseColor 1.0 1.0 0.0
        }
    }
    geometry Box {
        size 2.0 1.0 1.5
    }
}
```

Če pa zgornji VRML-dokument »prebere« VRML-predvajalnik, ga ta spremeni v 3D-prikaz:



*Slika1: Na sliki vidimo rezultat zgoraj opisane VRML-kode kot jo prikaže VRML-predvajalnik. Po odprtju modela v predvajalniku smo spremenili kot pogleda s kontrolnim ukazi predvajalnika. Če bi želeli tak pogled v predvajalniku brez uporabe kontrolnih ukazov, bi morali smer pogleda definirati že v kodi.*

Oglejmo si zgornji primer kode podrobneje:

Na začetku dokumenta VRML je obvezna začetna vrstica ali prolog:

```
#VRML V2.0 utf8
```

Ta vrstica pove brskalniku, da je ta datoteka VRML-dokument različice 2.0 in da uporablja angleški nabor znakov UTF-8 (Pesek, 2000). UTF-8 je kodirni sistem za javno dostopen standard Unicode (ISO – 10646-UCS-2). Unicode vsakemu znaku določa ime in zaporedno številko. Število znakov je okoli 50000, ne določa pa grafične podobe znaka in kako podrobno znake kodiramo (Kovačič, 2005-2006).

Opazimo, da se posamezni elementi in atributi v kodi zamikajo v desno, tej obliki zapisa imenujemo drevesna struktura. V primeru zgoraj tri prej omenjena telesa najavimo z besedo »shape«. Telesa so podrejena atributom z imenom »Transform«, ki podrejena telesa iz središča prikaznega prostora premakne na določeno mesto z vrstico:

```
translation 3.0 0.0 0.0
```

»Translation« pomeni translacijo predmeta na mesto označeno z X,Y in Z koordinatami (3.0 0.0 0.0). Če translacije nebi izvedli, bi predmet stal v središču prikaznega prostora (0, 0, 0).

Telesom podrejeni atributi določajo njihov izgled. Z atributom »appearance« in atributu podrejenimi atributi določamo izgled ploskev elementa: barvo, material, prosojnost,... Z atributom geometry pa določimo vrsto telesa. VRML-jezik ima nekaj osnovnih oblik že



opredeljenih kroglo (sphere), kvader (box), cylinder (valj), in druge. Zahtevnejše oblike bodisi sestavimo iz že definiranih oblik, ali pa uporabimo kak drug definiran element, za predstavljanje zahtevnejših oblik.

## 2.3 XML

XML (Extensible Markup Language) je internetni standard razvit leta 1998. Namenjen je zapisu elektronskih dokumentov. Zapisan je v obliki besedila in je dovolj razumljiv tako računalnikom kot ljudem. Tudi XML vsebuje drevesno strukturo elementov, vsebuje pa tudi procesne inštrukcije (dodatna navodila programom, kako je potrebno obdelati dokument). XML je nastal predvsem z namenom poenotiti različne načine zapisov podatkov različnih programov. Vsak program ima svoja lastna pravila o zapisu svojih dokumentov, kar pomeni, da se je potrebno za programiranje datotek posameznih programov učiti pravila vsakič znova, kar pa je programerje stalo veliko časa, truda in denarja. XML tako s poenotenjem dokumentov pospešuje dobre učinke programiranja z dokumenti.

XML se je pojavil relativno pozno. Razlog se skriva v tem, da so računalniki hitreje obdelovali binarno predstavitev podatkov, kot pa predstavitev z znaki. Z razvojem strojne opreme pa so se povečali diski za shranjevanje podatkov in procesorji so postali dovolj hitri tudi za obdelavo podatkov v XML-obliki (Kovačič, 2005-2006).

XML je pravzaprav »osnovni« nivo jezika. Njegova, že v naslovu omenjena razširljivost (extensible) govori o tem, da podaja le glavne in osnovne lastnosti za poenotenje zapisa. Vsako specifično področje, ki se ukvarja z dokumenti, lahko razvije svojo slovnico, ki je podrejena slovnici XML a razširjena na svoje področje. Tako poznamo več izvedenk XML: XHTML, RSS, MathML, GraphML, SVG, MusicXML, in še in še. Poznamo tudi GML, ki je standard za prostorske podatke, nas pa seveda najbolj zanima izvedenka X3D za prikaz virtualnega 3D-okolja.

Če želimo definirati nov XML-jezik, napišemo slovnico za ta jezik. Slovnico, ki ureja XML-dokumente imenujemo DTD (ang. Document Type Definition) oz. »definicija tipa dokumenta«. DTD-slovnico določa standard XML. V njej določimo pravila za zapis podatkov določene vrste. Določa dovoljene elemente, katerim se lahko določi imena, attribute in vrsto vsebine. Slovnica določa tudi strukturo XML-elementov (Kovačič, 2005-2006). DTD slovnica je lahko samostojna datoteka, lahko pa je vgrajena v XML-dokument (v prologu, po začetni vrstici), ki mora ustrezati pravilom zapisanim v slovnici.

Leta 2001 se je pojavil nov standard XML-Schema, ki predstavlja alternativo DTD slovnici. XML-Schema je načeloma boljša in naprednejša od DTD slovnice, je pa tudi bolj obsežna in razmeroma bolj zapletena. Predstavlja temelj za razvoj novih orodij, ki temeljijo na standardu XML. Omogoča precizno določanje sintakse večine današnjih dokumentov. (Kovačič, 2005-2006) XML-Schema je lahko samostojen XML-dokument, lahko pa je kot DTD del XML-dokumenta, ki po njej povzema pravila.

## 2.4 X3D

### 2.4.1 Splošen opis

X3D je odprti standardni format za predstavitev in prenos 3D-scen in objektov. X3D v bistvu predstavlja posodobljen VRML-jezik prilagojen XML-standardu. Tudi X3D je sprejeti ISO-standard. Tudi ta standard je rezultat raziskav organizacije Web 3D Consortium, ki so si zadali nalogo izboljšati in posodobiti VRML-jezik. X3D je zasnovan modularno. Glavni modul predstavlja majhno jedro ("core profile"), ki vsebuje majhen nabor vozlišč, namenjenih geometriji in animaciji. Vsebuje mehanizem, ki omogoča preprosto dodajanje novih vozlišč s profili. Z uvedbo profilov je razvijalcem omogočeno, da izdelajo X3D-brskalnik, prirejen njihovim potrebam.

Pomembnejše nove lastnosti X3D:

- program je odprtokodni, torej ni potrebnih licenc,
- je uradno vključen v MPEG-4 multimedijski standard,
- zaradi podpore XML je prenos in obdelava podatkov lažja,
- kompaktilen z novo generacijo grafičnih datotek – SVG (Scalable Vector Graphics),
- 3D-objekte se lahko upravlja z programskimi jeziki C, C++ in java.

Organizacija Web 3d consortium je na svoji spletni strani ([www.web3d.org/about/faq/#general-2](http://www.web3d.org/about/faq/#general-2)) objavila 10 razlogov, v čem je X3D zapis boljši od VRML:

- X3D ohranja močno podobnost z VRML, X3D ni zavrgel nobene lastnosti VRML.
- X3D je kot izvedenka XML bolj povezljiva z drugimi vrstami datotek.
- X3D-dokument se v različnih predvajalnikih obnaša bolj predvidljivo kot VRML. (predvajalniki X3D-dokumente predvajajo bolj enotno kot VRML-dokumente, kjer so se občasno pojavljale razlike pri prikazu).
- X3D omogoča razvoj podrejenih verzij zapisa, ki so prilagojena različnim področjem na trgu (npr. CAD, medicina, vizualizacija,...).
- Dodani so novi elementi.
- Funkcionalnost se ves čas izboljšuje.
- X3D aplikacije so zanesljive in predvidljive.
- itd...

### 2.4.2 Primer X3D zapisa

Ker je X3D-standard izpeljan iz XML, oziroma je podrejen njegovim pravilom, ima definirano svojo slovnico v DTD ([www.web3d.org/specifications/x3d-3.0.dtd](http://www.web3d.org/specifications/x3d-3.0.dtd)), kot tudi v XML-Schema obliki ([www.web3d.org/x3d/content/x3d-3.1.xsd](http://www.web3d.org/x3d/content/x3d-3.1.xsd)). Ker je X3D-dokument zgrajen po pravilih ene ali druge slovnice (gre za isto slovnico predstavljeno v dveh oblikah), se v vsakem dokumentu običajno sklicujemo na eno izmed slovnice, s čimer zagotovimo preverjanje veljavnosti oziroma skladnost dokumenta z slovnico. X3D-predvajalnik pa bo dokument skladen z X3D-slovnico pravilno prikazal kljub temu, da slovnice v dokumentu posebej ne navajamo.

Prikažimo prejšnji primer treh geometrijskih teles v obliki VRML-zapisa še v X3D-zapisu (brez definirane slovnice v prologu):

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive" version="3.0">
  <head></head>
  <Scene>
    <Transform translation="3.0 0.0 0.0">
      <Shape>
        <Appearance>
          <Material diffuseColor="1.0 0.0 0.0"/>
        </Appearance>
        <Cylinder height="1.0"/>
      </Shape>
    </Transform>
    <Transform translation="-3.0 0.0 0.0">
      <Shape>
        <Appearance>
          <Material diffuseColor="0.0 0.0 1.0"/>
        </Appearance>
        <Sphere/>
      </Shape>
    </Transform>
    <Shape>
      <Appearance>
        <Material diffuseColor="1.0 1.0 0.0"/>
      </Appearance>
      <Box size="2.0 1.0 1.5"/>
    </Shape>
  </Scene>
</X3D>
```

Vidimo, da je drevesna struktura zapisa ohranjena. Drevesno strukturo uporabljata tako XML kot tudi VRML. Vidimo, da za razliko od VRML-zapisa v X3D-zapisu vsebino posameznih elementov ujamemo med začetnim in končnim imenom elementa, ki je naveden v dveh zašiljenih oklepajih »<<« in »>>«. Konec elementa označuje poševna črta, ki je lahko v začetku imena (npr.: <element> vsebina </element>), lahko pa začetni in končni oznaki združimo v eno (npr.: <element/>), vendar element ne bo mogel imeti vsebine. Omenjeni način je drugačen kot pri VRML-zapisu, kjer imen posebej ne označujemo, označujemo pa začetek in konec vsebine elementa z znakoma »{« za začetek in »}» za konec vsebine elementa.

Tudi začetni vrstici spominjata na VRML:

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile="Immersive" version="3.0">
```

V prvi vrstici izvemo, da gre za XML 1.0 zapis in kateri kodirni sistem uporablja, v drugi vrstici pa, da gre za X3D 3.0 obliko XML.

### 2.4.3 Pravila XML/X3D zapisa

Pri tvorbi XML/X3D-dokumentov je potrebno upoštevati naslednja osnovna pravila (Young, 2001):

- dovoljeno je gnezdenje. Element lahko vsebuje druge elemente, ki nadalje vsebujejo svoje elemente:

```
<element>
  <podrejeni-element>
    <podrejeni-element-podrejenega-elementa>
    </podrejeni-element-podrejenega-elementa>
  </podrejeni-element>
</element>
```

- Križanje elementov je prepovedano. Vsak podrejeni element, ki ga opisujemo v nadrejenem elementu, moramo končati pred koncem nadrejenega elementa:

```
<element1>
  <element2>
    </element1>
  </element2>
```

- Dokument mora imeti en sam element najvišjega nivoja (element dokumenta ali osnovni element). Ostali elementi morajo biti vgnezdjeni znotraj osnovnega elementa.
- Vsak element mora imeti svojo začetno in končno oznako. Za razliko od HTML, pri XML ne smemo izpuščati končne oznake elementa.
- Ime elementa v začetni oznaki mora biti povsem enako imenu v končni oznaki istega elementa (ime zapisano z malimi črkami ni enako istemu imenu zapisanemu z velikimi črkami)

## 2.5 Osnove rastrskih podob

### 2.5.1 Rastrska podoba in barvni modeli

Digitalna podoba ali fotografija je sestavljena iz mreže celic ali pikslov (ang. pixels), ki predstavljajo osnovni gradnik digitalne podobe. Vsaka celica nosi svojo barvno vrednost, ki v kombinaciji z drugimi celicami sestavlja kombinacijo barv tako, da lahko nosijo in predstavljajo neko podobo.

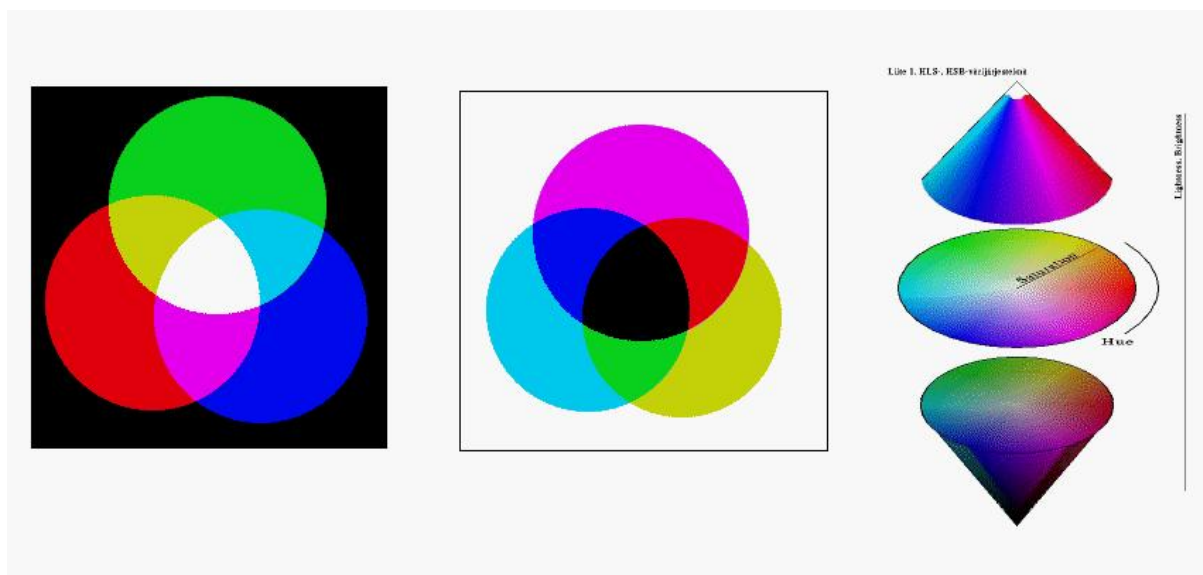
Število različnih barv, s katero lahko podobo upodobimo predstavlja barvna globina. Najmanjša enota računalnikovega spomina je bit, ki lahko nosi le dve vrednosti, 0 in 1. V grafiki predstavlja vrednost 0 belo, vrednost 1 pa črno barvo. Z kombinacijo dvobarvnih pikslov že lahko ustvarimo podobo, imenujemo jo enobitna podoba. Za uporabo več barv, potrebujemo več prostora, torej več bitov. Število barv, ki jih imamo na voljo, je enako številu 2 na določeno potenco, potenca pa je enaka prav številu bitov. Če barvo piksla opišemo s štirimi biti podatkov, bomo imeli na voljo  $2^4$  oz. 16 možnih kombinacij (0000, 0001, 0010,...). Štirje biti nam torej nudijo 16 možnih odtenkov, 8 bitna 256, 16 bitna 65536, 24 bitna pa že več kot 16 milijonov možnih barvnih odtenkov. (Corrigan, 1995)

Barve mešamo na aditivni in subtraktivni način. Pri aditivnem mešanju prehajamo od črne (ne mešamo nobene barve), do bele (vse barve zmešane skupaj). Vmesne barve pridobimo z različnimi deleži mešanja osnovnih barv. Pri aditivnem mešanju poznamo tri osnovne barve:

rdečo (angl. Red), zeleno (angl. Green) in modro (angl. Blue), ki tvorijo takoimenovani RGB barvni model. Tudi pri subtraktivnem mešanju poznamo tri osnovne barve: cianovo modro (angl. Blue), magenta vijolično (angl. Magenta) in rumeno (angl. Yellow), ki tvorijo barvni model CMY in so natančno nasprotje (komplementarne barve) trem osnovnim barvam v modelu RGB. Ker z mešanjem vseh treh osnovnih barv v polnem razmerju, ne bomo uspeli dobiti popolnoma črne barve, si pomagamo z dodajanjem četrte »pomožne« osnovne barve črne (angl. black), ki skupaj tvorijo barvni model CMYK. (Corrigan, 1995)

Oba modela sta plod različnega medija podobe. Aditivno mešanje barv je značilno za aktivne svetlobne izvore, kot je denimo računalniški zaslon. S pomočjo lastne svetlobe medija, osvetljujemo sicer temno podlago medija tako, da se prikaže željena barva. Pri subtraktivnem mešanju pa barve dobimo z odvzemanjem drugih barv, uporabljajo se za barvanje predmetov, ki niso aktivni svetlobni izvori. O subtraktivnem mešanju oz. CMYK metodi govorimo predvsem v zvezi z tiskanjem. Pri tiskanju na medij (običajno list bele barve) pada svetloba iz okolja, z barvami pa temnimo medij od bele prek poljubnih odtenkov do črne, ki je skrajni primer zatemnitve medija.

Veliko bolj naraven in usklajen način opisa barve pa ponuja model HSI (tudi HBI in HSL), katerega sestavljajo barvni ton (angl. Hue), nasičenost (angl. Saturation) in intenziteta (angl. Intensity). Barvni ton predstavljajo osnovne barve, kot so rdeča, zelena in modra. Nasičenost predstavlja čistost oziroma relativno jakost barve. Z zmanjšanjem nasičenosti dobimo bolj pastelne iz izprane barve. Svetlost lahko opišemo kot stopnjo črnega odtenka v barvi. Opisani model je tesno povezan s človekovim načinom dojemanja barve. Barvni ton se dejansko opira na valovno dolžino barve svetlobe, nasičenost povežemo z njeno jakostjo, svetlost pa s količino celotne svetlobe.



*Slika prikazuje RGB, CMYK in HSI barvni model ([www.wendler-im-netz.de/](http://www.wendler-im-netz.de/) in [www.mtholyoke.edu/acad/compsc/Honors/Lilla-Zollei/image40.gif](http://www.mtholyoke.edu/acad/compsc/Honors/Lilla-Zollei/image40.gif))*

## 2.5.2 Lastnosti shranjevanja bitnih podob

Poglejmo si še velikost bitnih podob v smislu zavzemanja prostora na računalniku. Za velikost so najpomembnejši trije parametri:

- velikost bitne podobe v točkah,
- barvna globina,
- format datoteke, v kateri podobo hranimo.

Povezava med velikostjo bitne podobe v točkah in njena velikost pri shranjevanju je približno premosorazmerna. Dvakrat večja podoba torej zavzame dvakrat več prostora na trdem disku. Tudi barvna globina daje logičen odsev pri zavzemanju prostora na disku, saj večbitna podoba zavzame več prostora. Velikost pri shranjevanju je v približnem sorazmerju s številom bitov podobe, ne pa tudi s številom barv. Če isto podobo zapišemo v eno in osem bitni obliki, bo osem bitna podoba zavzela približno osemkrat več prostora na disku. Tudi format datoteke, v katero zapišemo bitno podobo, lahko vpliva na količino podatkov pri shranjevanju. Vse je odvisno, ali format podpira določeno stiskanje podatkov ali ne (Corrigan, 1995).

Vsak format ima seveda svoje prednosti in slabosti, zato je njihova izbira odvisna od namena nadaljnje uporabe podobe oz. slikovne datoteke. Opišimo najpogostejše formate podob (Putrle, str. 14-15):

- JPEG (angl. Joint Photographers Expert Group) format ima dve pomembni značilnosti: kompletno barvno paleto in nizke zahteve glede prostora za shranjevanje. Uporablja stiskanje z izgubo podatkov, vendar pa stopnjo stiskanja in s tem tudi izgubo kvalitete običajno izberemo sami. Algoritem podobo uredi v kvadratke velikosti 8x8 slikovnih elementov in nato »zavrže« manj pomembne barvne podatke oz. barvne razlike znotraj posameznega kvadratika. Nivo izničenja barvnih razlik je odvisen od izbrane stopnje stiskanja. Velja, da je kvaliteta podobe še sprejemljiva pri stisnjenju do 10x. Torej s pomočjo JPEG (JPG) formata stisnjena 12 MB TIFF-datoteka ne sme biti manjša od 1,2 MB. Če želimo podobo, ki smo jo posneli v JPG-formatu, kasneje dodatno urejati s pomočjo računalnika, jo moramo najprej pretvoriti v TIFF-format. S tem ne bomo pridobili pri prejšnjem stiskanju datoteke izgubljenih podatkov (le ti so nepovratni), temveč se bomo izognili dodatnim izgubam kvalitete pri naslednjem shranjevanju podobe. Pri vsakem nadaljnjem shranjevanju sprememb, JPG namreč ponovno dodatno zavrže del barvnih razlik znotraj kvadratika 8x8, s tem kvaliteta podobe ponovno pade in meje med temi kvadratkami oz. razlike med kvadratkami lahko postanejo vidne.
- BMP-format je standardni format Microsoftovih Oken. Prepoznajo ga skoraj vsi programi za urejanje podob. Njegova pomanjkljivost je omejitev na 256 barv in ločljivost 96 dpi.
- GIF (Graphic Interchange Format) je uporabljan predvsem pri izmenjavanju računalniške grafike. Omejen je na 256 barv, z njim pa lahko učinkovito stiskamo podobe s homogenimi barvami (npr. vektorske podobe). Nima zveznih prehodov barv, kot jih ima JPEG. Podpira prosojnost ozadja, ki je včasih potrebna pri računalniški grafiki. Format podpira tudi animacijo podob.
- TIFF ali Tagged Image File Format je bil prvotno oblikovan za Macintoshove računalnike, danes pa ga množično uporabljajo tudi uporabniki drugih osebnih računalnikov. Ne uporablja stiskanja podatkov in je sposoben shraniti katerokoli barvo do velikosti 32 bitne barvne globine (skoraj 4,3 milijarde različnih barv). Format uporabljamo, če želimo imeti podobo visoke kakovosti ali če jo nameravamo kasneje dodatno urejevati. Njegova pomanjkljivost je, da datoteke shranjene v tem formatu

zavzamejo izredno veliko prostora za shranjevanje. S to lastnostjo je ta format manj primeren za uporabo v modelu.

### 2.5.3 Stiskanje bitnih podob

Ker shranjevanje podob v osnovi zavzema relativno veliko prostora, je pri grafični obdelavi pravzaprav nujno govoriti o stiskanju grafičnih podatkov. Poznamo naslednje najbolj običajne postopke stiskanja (Corrigan, 1995):

- Algoritem RLE (angl. Run Length Encoding) je eden najpreprostejših postopkov za stiskanje podatkov. Algoritem pri shranjevanju podobe ne beleži vsake celice v podobi posebej, temveč zapisuje število zaporednih celic, ki imajo enako vrednost. Postopek RLE pozna nekaj različic, ki se razlikujejo po tem, ali bo najprej zapisano število točk in nato njihova barvna vrednost ali obratno. Čeprav je postopek RLE razumljiv in enostaven, to še zdaleč ni najboljši način za stiskanje bitnih podob. Njegova šibkost se pokaže pri »pisanih« oziroma razdrobljenih podobah, ki vsebujejo veliko število barv, bolje pa se obnese pri podobah, ki vsebujejo veliko enobarvnih površin. Primeren je torej za preproste podobe sestavljene v računalniškem programu, kjer nastopa nekaj glavnih barv (npr. ikone, plakati, vizitke,...), manj primeren pa je za skenirane podobe, fotografije in tiste podobe, kjer upodabljamo osvetljeno okolje (sence, prehodi iz temnega v svetlo).
- Postopek LZW (po avtorjih Lempel, Ziv in Welch) je v primerjavi s postopkom RLE mnogo bolj natančen. Postopek ne zapisuje zaporedja posameznih pikslov, temveč s pomočjo izdelanih tabel s kodami prepozna določene večje ponavljajoče se vzorce v bitni podobi. S postopkom LZW podobo zmanjšamo približno od trikrat do štirikrat, pri podobah z izrazitimi vzorci pa celo do desetkrat. Tudi temu postopku delajo več težav skenirane podobe in fotografije, ki imajo običajno manj izrazite vzorce. Postopek LZW lahko poljubno izvajamo pri formatu TIFF, format GIF pa ga uporablja avtomatsko.
- JPEG postopek stiskanja smo že omenili. V nasprotju z LZW in RLE postopkoma je ta postopek primeren prav za skenirane podobe in fotografije, vendar pa pri tem postopku izgubimo del podatkov iz prvotne podobe, kar se pri LZW in RLE ne zgodi. JPEG je načeloma zelo učinkovit, saj podobo močno skrči, kljub temu pa so razlike na prvi pogled komaj opazne. Nekateri programi nudijo tudi stopnjo stiskanja podobe. Bistvo postopka JPEG sloni na dejstvu, da je človekovo oko zelo občutljivo na spremembe navidezne svetlosti, mnogo manj pa na spremembe barvnega tona. Postopek torej bolj upošteva spremembe v svetlosti, le v manjši pa spremembe v barvnih tonih med posameznimi točkami. Človekovo oko je za slednje mnogo manj občutljivo in skorajda ne zazna manjših sprememb v barvah sosednjih točk, zato se zdi stisnjena podoba zelo podobna prvotni.

## 3 UPORABLJENA GEOINFORMACIJSKA TEHNOLOGIJA

### 3.1 Digitalni model reliefa

Digitalni model reliefa (DMR) je zapis v obliki števil namenjen prikazovanju reliefa v obliki modela. Sestavljen je iz niza prostorskih točk, podanih v treh dimenzijah (x,y – za položaj in H – za višino) in v izbranem koordinatnem sistemu.

#### 3.1.1 Oblike DMR

Poznamo več geometričnih oblik DMR:

- Pravilna oblika DMR-mreže

Predstavnik pravilne geometrične oblike je kvadratna celična mreža. Gre za obliko DMR, ki mu geometrična pravilnost omogoča enostavno in razumljivo uporabo tako za računalniške aplikacije kot za ljudi, ter zavzema manj prostora pri shranjevanju. Ideja je predvsem v tem, da točke, ki v vnaprej določeni medsebojni razdalji določajo višino. Če imamo podano položajno koordinato (x,y) samo ene višinske točke, poznamo medsebojno oddaljenost med točkami ter poznamo projekcijo prikaza, potem za vsako podano višinsko točko natančno vemo kje leži.

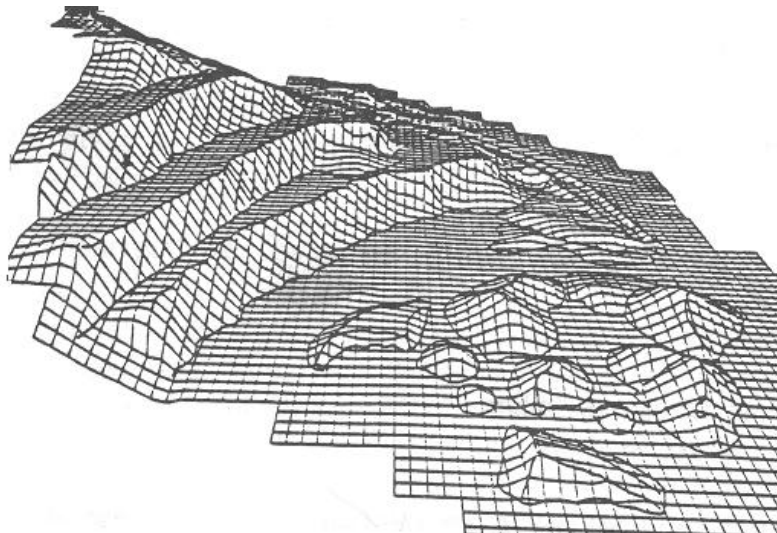
Primer dela datoteke, ki opisuje kvadratno celično mrežo (DMR 12,5):

```
ncols 181
nrows 241
xllcorner 439243.75
yllcorner 66993.75
cellsize 12.50
nodata_value -999.00
542.95 542.79 543.04 543.04 543.70 544.36 545.54 546.54 547.70 548.63 549.55
550.66 551.84 553.45 555.07 556.43 557.93 559.27 560.85 561.85 562.46 562.77
563.04 563.08 562.99 562.88 562.94 563.01 562.81 562.96 563.79 564.23 565.04
565.20 564.74 564.49 565.14 565.65 566.22 567.35 569.35 572.19 574.89 578.55
580.79 581.77 582.21 582.10 581.72 581.95 582.07 581.56 581.82 584.58 587.21
590.32 593.67 596.93 599.34 601.02 602.38 603.46 604.37 605.74 607.03 608.19
609.07 609.91 611.32 613.47 615.42 616.70 618.22 620.09 622.71 626.15 629.78
632.32 635.72 640.16 644.29 648.40 652.02 655.01 658.15 661.63 664.46 658.44
```

V zgornjem primeru lahko vidimo enostavnost zapisa, ki ga pravilna mreža omogoča. Definiramo samo položajne koordinate ene izmed točk (običajno vogala), določimo razdaljo med točkami in zapišemo še podatek o številu vrstic in stolpcev. Oddaljenost med točkami določa natančnost prikaza DMR. Manjša kot bo razdalja med točkami, večja bo natančnost, a tudi podatkov za obdelavo bo več. Tudi podatek o višini je lahko prostorsko potraten. V zgornjem primeru imamo podane višine do centimetra natančno. Če bi uporabili samo metersko točnost, bi prostornino datoteke zmanjšali kar za polovico (namesto 6 znakov bi potrebovali 3)! Zato je vedno potrebno iskati kompromis med natančnostjo modela in količino podatkov, ki lahko zelo obremenijo aplikacije za predstavitev terena.



Pravilna geometrična oblika DMR ima pa tudi svojo slabost. Tako ravninski kot hribovit svet bo po tej metodi predstavljen enako natančno, čeprav potrebujemo za hribovit svet bolj natančne podatke, za ravninski pa manj. Tako moramo pri tej metodi predstavitve reliefa vedno tehtati med natančnim prikazom hribov in velikim številom manj koristnih podatkov tudi za ravnine in obratno.

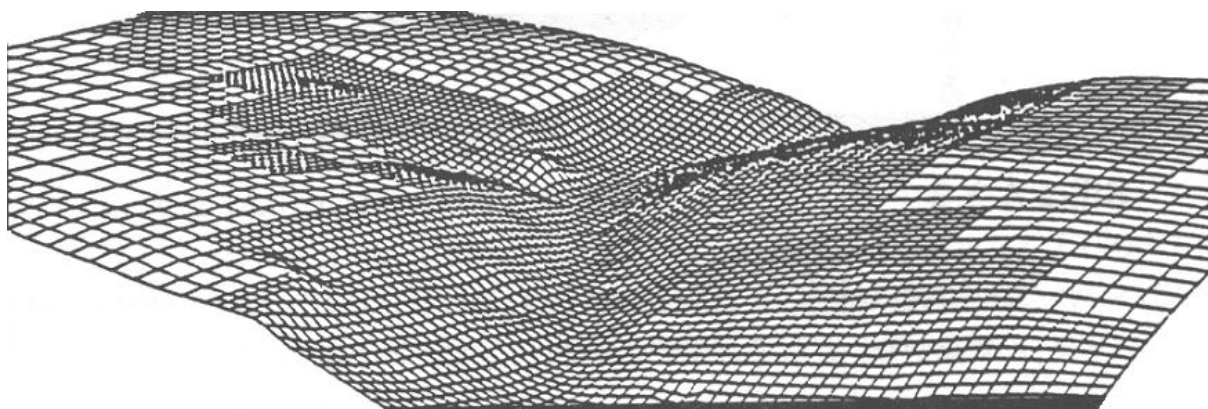


*Prikaz terena v obliki pravilne mreže*

*(Rihtarčič M., Fras Z., 1991: Digitalni model reliefa, 1. del: Teoretične osnove in uporaba DMR, Ljubljana, FAGG str. 47)*

- Polpravilna oblika DMR mreže

Predstavnik te oblike je progresivno vzorčena celična mreža. S to mrežo gostoto točk prilagajamo značilnostim terena. Mreža je še vedno sestavljena iz kvadratov, vendar razdalje med točkami niso več povsod enake. Na hribovitem terenu bomo v mreži večjih kvadratov, kvadrat razdelili na štiri manjše kvadrate, te štiri pa lahko še na manjše. Na ravninskem terenu večjih kvadratov ne bomo delili. S tem odpravimo slabost pravilne kvadratne mreže in natančnost prikaza prilagajamo terenu, vendar se na ta račun enostaven prikaz nekoliko oteži, saj je potrebno definirati, kateri kvadrat se deli in kateri ne.



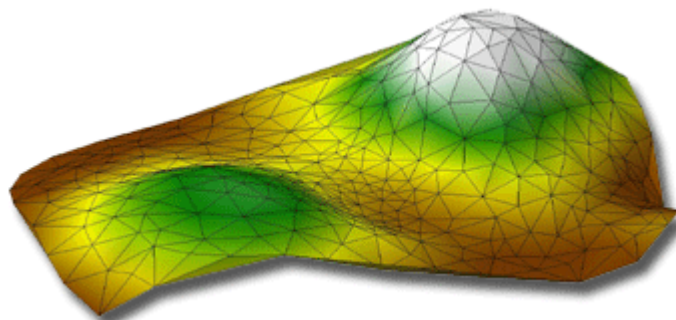
*Prikaz terena v obliki progresivne mreže. Na območju večjega naklona uporabimo manjše celice, na ravnini pa večje.*

*(Rihtarčič M., Fras Z., 1991: Digitalni model reliefa, 1. del: Teoretične osnove in uporaba DMR, Ljubljana, FAGG str.47)*

- Nepravilna oblika DMR mreže

Pri tej obliki govorimo predvsem o trikotniški ali TIN-mreži (angl. Triangular irregular network), ki se tvori in določa z Delaunayevo triangulacijo. Pri tej metodi teren prikažemo z trikotniškimi ploskvami, katerih vogali se združujejo v znanih višinskih točkah. (Kvamme in dr., 1997) Pogoji triangulacije je, da če trikotniku narišemo očrtani krog, ne pade vanj nobena druga točka iz mreže. Trikotniki morajo biti torej čimbolj enakostranični. Temelji te metode so v teoriji grafov, ki jih obravnava matematika. Postopek triangulacije je iterativen.

Glavna prednost TIN-mreže pred pravilno obliko predstavitve DMR je predvsem spremenljiva gostota točk glede na razgibanost terena. Kjer je teren razgiban, vključimo več točk v model, kot na manj razgibanem terenu. S tem prilagajamo kakovost modela glede na zahtevnost terena, kar omogoča ob isti količini podatkov na račun ravninskih predelov ustrezno predstaviti tudi razgibanega terena. Spremenljiva gostota točk pri TIN-mreži omogoča glede tudi uporabo podatkov, ki so pri pravilni obliki manj uporabni (npr. zabeležene spremembe naklona, poljubne višinske točke, itd...). Model TIN-mreže lahko v primerjavi z pravilno obliko sestavimo zelo hitro, saj potek interpolacije pri pravilnih oblikah zahteva precej več časa.



*Prikaz reliefa v obliki TIN mreže v kombinaciji s hipsometrično lestvico.  
([www.scisoftware.com/products/gms\\_details/tin.gif](http://www.scisoftware.com/products/gms_details/tin.gif))*

- Kombinirana oblika DMR mreže

Gre za uporabo kombinacije zgoraj naštetih oblik. Najpogosteje na pravilni mreži pravokotnikov potek grebena, jarka in kanalov prikažemo s dodanimi trikotniki.

### 3.1.2 Zajem podatkov za izdelavo DMR

Podatke za izdelavo DMR dobimo na več načinov:

- kartometrični zajem

Na karti s plastnicami višino posameznih točk preberemo z odčitanjem plastnic, oz. interpolacijo plastnic, če točka leži med plastnicama. Metoda je pogojno uporabna le za manjšo natančnost modelov. Na tak način so bili pri nas izdelani DMR100 in DMR500.

- stereofotogrametrija

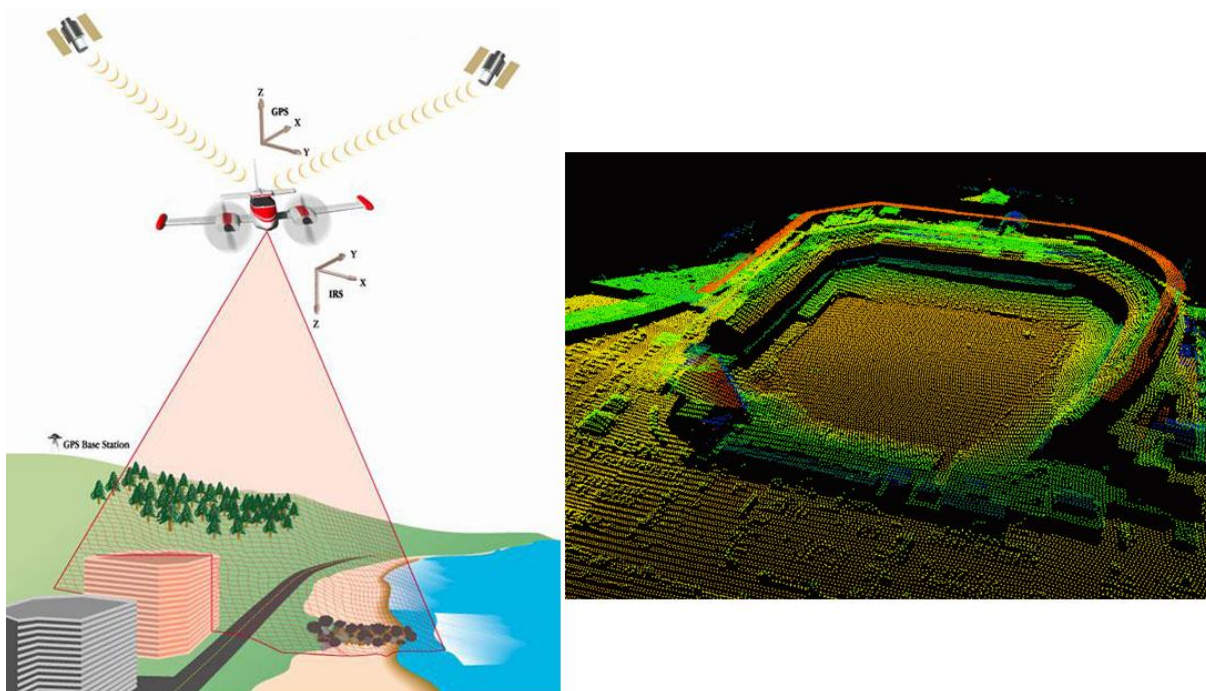
Iz dveh parov posnetkov se za posamezno točko zajame položajne koordinate in višino točke. Največkrat se uporabi progresivni način zajema. Na strmih predelih se gostota točk poveča.

Uporabi se lahko letalske ali satelitske posnetke. Pri nas so bil s pomočjo digitalnega ortofoto 1:5000 (letalski posnetki) zajeti podatki za izdelavo DMR20, Ministrstvo za obrambo pa uporablja tudi DMR zajet iz satelitskih posnetkov francoskega podjetja SPOT Image.

- lasersko skeniranje in radarska interferometrija

Pri laserskem skeniranju gre za tako imenovano LIDAR (LIght Detection and Ranging) tehnologijo skeniranja zemeljskega površja. Gre za relativno drago snemanje, ki zahteva zahtevno obdelavo z velikim številom podatkov.

Radarska interferometrija deluje na podobnem principu le z drugačno tehnologijo. Po tej metodi zajema je pri nas izdelan InSAR DMR25.



*Postopek LIDAR snemanja in rezultat snemanja  
([http://www.mbnms-simon.org/images/build/whats\\_lidar\\_01.jpg](http://www.mbnms-simon.org/images/build/whats_lidar_01.jpg) in  
[www.valtus.ca/images/gallery/lidarDEM.gif](http://www.valtus.ca/images/gallery/lidarDEM.gif))*

- transformacija med obstoječimi mrežami

Uporabimo lahko tudi že zajete podatke in izdelane mreže. Pri tej metodi veliko vlogo igrata ustrezna transformacija in interpolacija.

### 3.1.3 Uporabnost DMR

DMR je pravzaprav zelo uporaben izdelek, saj omogoča matematičen oz. avtomatski pristop pri analiziranju značilnosti terena.

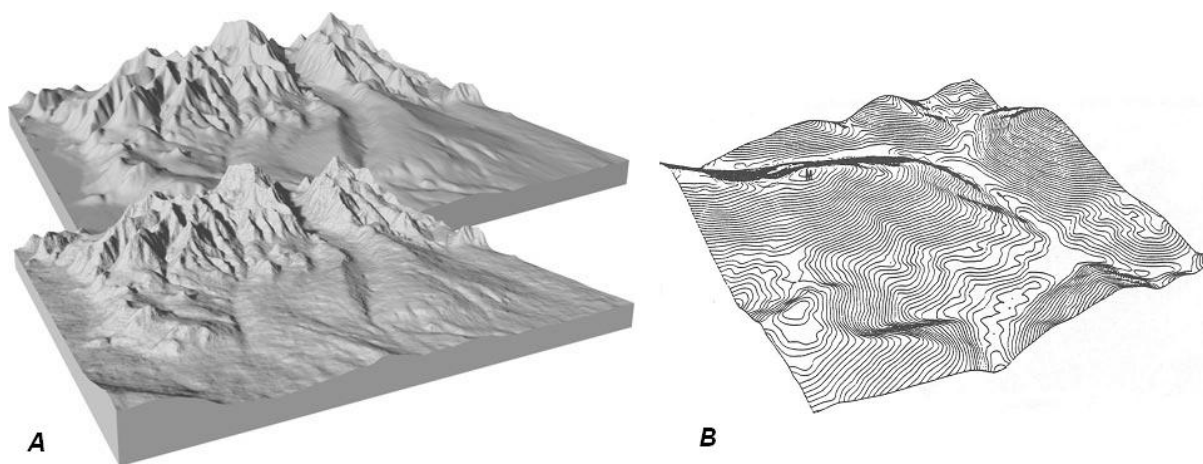
Uporabnost DMR sega na več področij:

- prostorski informacijski sistemi,
- kartografija v najširšem smislu,

- prostorsko planiranje in urbanizem,
- vojaška uporaba,
- preučevanje smeri in količine voda,
- preučevanje poplav in plazov,
- ugotavljanje pokritosti s telekomunikacijskimi signali,
- trasiranje prometnic,
- turistične predstavitve,
- itd...

DMR lahko uporabimo za prikaz tridimenzionalnega terena. Govorimo o projekciji DMR, ki je pravzaprav dvodimenzionalna podoba, ki daje prostorski vtis. 3D-prikaz je pregleden, jasen in dokaj realen prikaz površja. Za 3D-prikaz je najbolj uporabna pravilna DMR-mreža, ki daje gladke in vizualno realnejše rezultate, kot naprimer TIN-mreža, kjer je model opazno sestavljen iz trikotnikov. Algoritem prikazuje model na podlagi podanih parametrov, kot so: horizontalni in vertikalni kot gledanja, točko s katere »gledamo«, oz. točka iz katere izhajajo žarki centralne projekcije, merilo navpične povečave, itd... Ko spreminjamo pogled na model, algoritem na podlagi parametrov ves čas izračunava ustrezen prikaz modela. Algoritem pri prikazu upošteva prekrita območja, ki jih bližja vzpetina zakriva in jih zato ne prikaže.

Ker običajno model sestavlja le določen in omejen del površja, se model na robovih konča. Model, ki je prikazan v obliki blokdiagrama, lahko prikazujemo kot »letečo preprogo« (zgolj površje) ali pa z osnovnim kvadrom oz. »škatlo« (površje na robovih omejujejo stranice »škatle«, kot bi izdolbli površje v kvader).



V A primeru je prikazan model reliefa v obliki blokdiagrama v B primeru pa v obliki »leteče preproge« brez osnovnega kvadra.

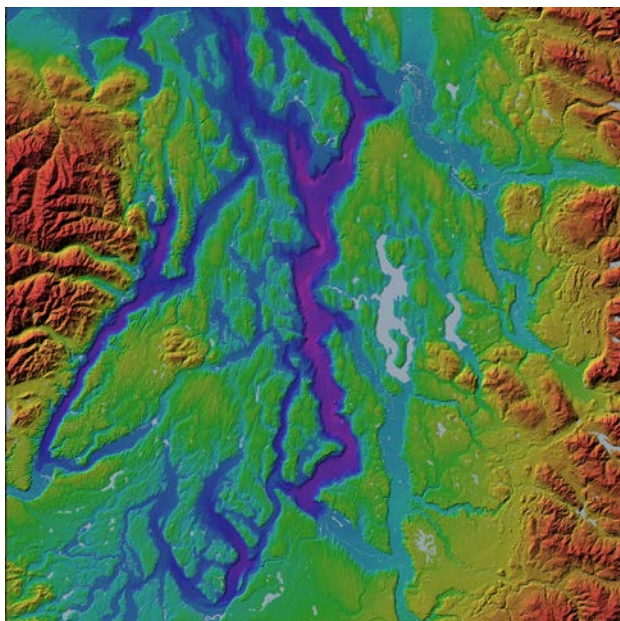
(<http://pubs.usgs.gov/of/2004/1451/graphics/garrity/fig4.jpg> in Rihtarčič M., Fras Z., 1991: *Digitalni model reliefa, 1. del: Teoretične osnove in uporaba DMR*, Ljubljana, FAGG, str. 47)

Ob istem merilu položajnih koordinat lahko povečamo faktor višinskih vrednosti in s tem povišamo vzpetine, s čimer dosežemo boljši vizualni vtis terena.

DMR je uporaben predvsem v prostorskih analizah. Izvajamo lahko naslednje funkcije (Kvamme in dr., 1997):

- hipsometrija

Vsaki posamezni celici DMR lahko določimo povprečno višino, kateri pripada določena barvna vrednost. Tako določene barve lahko uporabimo tako v dvodimenzionalnem kot tudi v tridimenzionalnem pogledu. Običajno izbiramo modro barvo za vodno gladino ali dno, zelene barve za nizke predele in prehod prek rumene v rdečo za vzpetine.



*Barvni (hipsometrični) prikaz površja.*  
(<http://duff.geology.washington.edu/data/prism/bathy/vivid.gif>)

- naklon

Celici DMR lahko določamo tudi naklon, smer naklona predstavlja smer padnice, torej smer največje strmine. Izdelamo lahko naklonsko karto, na kateri so narisani vektorji naklona, daljši je vektor večji naklon predstavlja. Tako karto lahko uporabimo za gradnjo, preučevanje odvodnjavanja, itd.

- senčenje

DMR se uporablja tudi za senčenje. Če govorimo o kvadratni celični mreži, predstavlja kvadrat med štirimi točkami v tridimenzionalnem pogledu ukrivljeno ploskev, katere prvi približek je hiperboloid. Za potrebe senčenja to krivuljo poenostavimo v ravnino, kateri izračunamo normalni vektor, ki je na ravnino pravokoten. Potrebujemo tudi vektor osvetlitve, ki ga določata azimut in elevacija. Naš končni cilj je določitev refleksije, ki pa predstavlja kot med normalnim vektorjem in vektorjem osvetlitve. Vrednost refleksije določata dve skrajni vrednosti 0 – črno in 1 – belo, vmesne vrednosti so torej med 0 in 1.

Podobno kot senčenje se proučuje tudi osvetljenost ali insolacija površja. Ugotovimo lahko, pod kakšnim kotom sonce pada na površje, pri tem pa moramo upoštevati tudi geografske koordinate oz. globalno lego ter datum in uro od katerih je osvetljenost odvisna. Upoštevati



moramo tudi sence, ki jih ustvarjajo hribi. Ugotavljamo lahko tudi, v katero smer neba je celica obrnjena, čemur pravimo eksporiranost. S tem ugotavljamo ugodne lege za vinograde, pozidavo...

- izdelava plastnic

Na podlagi DMR lahko izdelamo tudi plastnice, pri čemer uporabimo ustrezne algoritme. Plastnice lahko izdelamo tako iz geometrično pravilne oblike kot tudi iz TIN-oblike, vendar moramo pri slednji dobljene rezultate še zgladiti, saj plastnice zaradi trikotniške oblike vsebujejo ostre robove.

- izračun profilov in prečnih prereзов

Izračun profilov in prečnih prereзов je zelo uporaben v gradbeništvu, predvsem za določanje količine odkopov, s čimer dobimo jasne organizacijske in finančne podatke. Postopek je relativno enostaven. Obstoječe stanje na DMR odštejemo od načrtovanega stanja in dobimo razlike v prostornini zemeljske mase.

- izdelava kart vidnosti

Določitev območja, ki ga iz določene točke vidimo, je uporabno na številnih prej naštetih področjih (prostorsko planiranje in urbanizem, vojaška uporaba, ugotavljanje pokritosti s telekomunikacijskimi signali, turistične predstavitve,...) Algoritem, ki določa vidnost, poenostavljeno opišemo kot postopek, ki iz izbrane točke v vse smeri obravnavanega področja pošlje premice. Vsak del območja, katerega premica prvič prebode, ne da bi že prej prebodla kak drug del območja, se označi kot vidno. Območje, ki je prebodeno drugič, ali večkrat, se označi kot skrito. Gre za podoben algoritem, kot pri prikazovanju 3D-modela, saj je tudi tam potrebno prikazati le vidno območje.

- analize razvodij

Gre za funkcijo predvsem uporabno za analiziranje voda in plazov. Razvodje je črta, ki loči dve ali več porečji, oz. dve ali več področij, s katerega vode se umikajo v različno smer, oz. v različni(e) reki(e). Pri tej funkciji uporabimo algoritem, ki za vsako celico, najde najnižjo sosednjo celico, v katero bi se izlila voda, če bi jo celica »vsebovala«. Algoritem je relativno preprost, daje pa zanesljive rezultate.

### **3.1.4 Glajenje in filtriranje digitalnega modela reliefa**

Ker je za sam vizualno ustrezen prikaz DMR potrebna še obdelava samih podatkov, si podrobneje oglejmo glajenje in filtriranje DMR. Interpolacijski algoritmi, s pomočjo katerih DMR izdelamo, ne omogočajo vizualno ustreznega prikaza. Pojavijo se napake, vidne predvsem v perspektivnih pogledih, model pa je po interpolaciji grob in ima veliko umetnih majhnih vrhov, jam, čeri in dolinic. Grob izgled in napak se zaradi časovne omejenosti ne lotevamo z posameznim preučevanjem napak, ampak celoten model zgladimo. Zglajenemu modelu, ki je tudi vizualno privlačnejši, na ta način dovolj učinkovito in hitro odpravimo interpolacijske napake.

Glajenje izvajamo s filtri, ki so pomembna orodja pri GIS. Glajenje DMR se običajno lotimo s pomičnim oknom, ki pokrije devet celic. Obravnavano, tisto za katero rezultat računamo in

osem sosednjih, ki se obravnavane celice dotikajo. Z različnim uteževanjem sosednjih celic glede na obravnavano dosegamo različen učinek glajenja.

$$A \begin{bmatrix} 0,111 & 0,111 & 0,111 \\ 0,111 & 0,111 & 0,111 \\ 0,111 & 0,111 & 0,111 \end{bmatrix} B \begin{bmatrix} 0,05 & 0,10 & 0,05 \\ 0,10 & 0,40 & 0,10 \\ 0,05 & 0,10 & 0,05 \end{bmatrix} C \begin{bmatrix} 0,025 & 0,075 & 0,025 \\ 0,075 & 0,600 & 0,075 \\ 0,025 & 0,075 & 0,025 \end{bmatrix} D \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Gornje matrike (Kvamme in dr., 1997, str. 122) prikazujejo filtre za glajenje z različnimi učinki. Matrika A predstavlja močan učinek glajenja, B in C nekoliko manjši učinek, pri filtru D pa se stanje nebi spremenilo.

Glajenje pa ima tudi svoje slabe strani. Zaradi povprečenja vrednosti se lahko zgodi, da pomanjšamo vzpetine, in zapolnimo doline, kar še posebej velja za »močnejše« filtre. Zato se je za zahtevnejšo uporabo glajenja DMR vedno potrebno lotiti s premislekom, za zgolj vizualno kvaliteten prikaz, pa visoka natančnost prikazanih višin niti ni tako nujna (Kvamme in dr., 1997).

### 3.2 Digitalni ortofoto

Digitalni ortofoto (DOF) je fotogrametrični izdelek, ki ga dobimo s transformacijo fotografskega posnetka v ortogonalno projekcijo (Kosmatin Fras, 2004). V metričnem smislu je enak linijskemu načrtu ali karti. Pri nas se DOF uporablja z osnovnim slikovnim elementom (pikslo) 0,5m in izrisom v merilu 1:5000, imenujemo ga tudi DOF5.

DOF je kot metrična bitna podoba zelo uporaben na več področjih:

- v različnih geografskih informacijskih sistemih (GIS),
- pri prostorskem planiranju,
- pri planiranju geodetskih del,
- za zajem rabe zemljišč v kmetijstvu,
- za vzdrževanje kart in načrtov,
- itd...

DOF je produkt več različnih digitalnih postopkov, ki jih uvrščamo pod okrilje fotogrametrije (Kosmatin Fras, 2004):

- aerosnemanje

Aerosnemanje je pridobivanje fotografij površja s pomočjo letala. Ciklično aerosnemanje pomeni načrtovano večkratno snemanje istega območja v določenih časovnih intervalih. V Sloveniji se ciklično aerosnemanje izvaja po projektu CAS 2000, ki je podal nova podrobnejša pravila in postopke za snemanje. Po tem projektu se zanimivejše dele Slovenije posname vsake tri leta. Aerosnemanje je projekt Geodetske uprave, izvaja ga Geodetski zavod Slovenije (ki vzdružuje tudi arhiv posnetkov), financira pa se tudi iz proračuna. S tem projektom je Slovenija prva država v Evropi, ki površje iz zraka snema tako podrobno in pogosto.

Snemanje se izvaja v merilu 1 : 17500, saj en list pokriva en list temeljnega topografskega načrta (TTN) v merilu 1:5000. CAS 2000 je definiral enoto snemanja – blok, ki pokriva območje 15 x 22,5 kilometrov in vsebuje približno 80 posnetkov. Smer snemanja je V-Z ali S-J, izjeme so doline, ki se jih posname vzdolžno, posnetke pa se kasneje kombinira med seboj. Posnetki morajo biti kvalitetni, zato je potrebno vsako snemanje skrbno planirati. V planu se upoštevajo mejne sposobnosti letala, delovna območja instrumentna za izvedenost, vrsto končnega izdelka, relief zemljišča, zahtevano natančnost in razdelitev območja na liste. Posnetki se snemajo tako, da se v določeni meri mesebojno prekrivajo. Običajno je vzdolžni preklap posnetkov (v smeri leta) 60 %, prečni preklap (med pasovi) pa 30 %. Vzdolžni preklap posnetkov je odvisen od načina rabe posnetkov. Za stereoizvedenost je nujno, da se posnetki prekrivajo vsaj 60 %. Večji preklap omogoča večjo izbiro veznih točk za potrebe aerotriangulacije, vendar so stroški večji (večja poraba fotomateriala).

Posnetke se poleg izdelave DOF uporablja tudi za zajem topografskih podatkov (izdelana digitalna topografska baza - DTK5), zajem stavb, evidentiranje stanja prostora, interpretacijo, itd...

- skeniranje aeroposnetkov

Običajno se skenira na natančnosti 14 mikrometra, en črnobeli posnetek zavzema 360MB, barvni pa 1GB.

- orientacija posnetkov

Izvesti moramo dve vrsti orientacije, notranjo in zunanjo. Notranja orientacija se nanaša na uporabljen fotoaparatus. Pri orientaciji uporabimo konstanto fotoaparata in slikovne koordinate glavne točke, ki sta sestavni del kalibracijskega protokola fotoaparata.

Zunanja orientacija je ugotavljanje položaja fotografije v prostoru v trenutku nastanka posnetka. Elemente zunanje orientacije posnetkov izračunamo z izravnavo v projektu aerotriangulacije. Obstajajo tudi druge možnosti: s prostorskim urezom za en posnetek, ali z relativno in absolutno orientacijo stereopara za en stereopar.

- geometrične in radiometrične transformacije

Pri izdelavi ortofota govorimo o postopku diferencialnega redresiranja. Ker smo posnetke že izvedli notranjo in zunanjo orientacijo, lahko sedaj rekonstruiramo lego fotografije, kot je bila v trenutku nastanka posnetka. Skozi vsako točko na fotografiji lahko povlečemo premico, ki povezuje točko na posnetku, projekcijski center fotoaparata in točko na tleh. Vemo, da točka na tleh leži nekje na premici, vendar pa ne vemo, kje na premici leži. Lego točke na premici določamo na dva načina: lahko uporabimo stereopar, s čimer dobimo še drugo premico iste točke in njuno presečišče, ki sedaj določa lego na obeh premicah. Drugi način pa je uporaba DMR. Prej omenjena premica »preseka« DMR, s čimer dobimo vse tri položajne koordinate X, Y in Z. Poleg dobljenih koordinat, se zabeleži tudi barvna vrednost mrežne celice posnetka, ki se na DOF preslika na ustrezen položaj, s čimer zagotovimo metričnost DOF-posnetka.

Obstoječi DMR uporabimo, če je dovolj natančen. Če dovolj natančnega DMR nimamo na voljo, ga lahko izdelamo kar v okviru izdelave ortofota z avtomatskimi postopki slikovnega ujemanja. To je relativno hitra metoda, a tudi manj zanesljiva. Bolj zanesljive metode



izdelave zahtevajo tudi več časa. Metoda slikovnega ujemanja se slabše odnese predvsem na območju gozda, saj tam podoba ne zajame tal, temveč vrh krošenj dreves, višin dreves pa ne poznamo. Zato je DOF narejen s pomočjo DMR te vrste manj natančen na področju gozda. DOF je tudi splošno odvisen od uporabe DMR, bolj natančen DMR kot uporabimo, bolj natančen bo tudi DOF.

Zgoraj naštetim postopkom sledijo še združevanje in mozaičenje posnetkov, oprema lista karte, zapis, arhiviranje in končna uporaba posnetkov.

Digitalni ortofoto, ki ga pri nas poznamo pod nazivom DOF5, je sistemski izdelek Geodetske uprave RS, ki se v Sloveniji izdeluje že skoraj 10 let. DOF5 je izdelan na osnovi aeroposnetkov merila od 1:15000 do 1:25000 in je izdelek, katerega izdelavo zagotavlja država. Konec leta 2002 je bilo naše ozemlje v celoti pokrito s tem izdelkom, tako da se sedaj že odvija prvi cikel obnove. Vzdrževanje ortofota bo predvidoma potekalo v povprečju na 6 let (3 leta na intenzivnih območjih, 9 let na izrazito neintenzivnih območjih). Po naročilu je seveda možna tudi izvedba ortofota v drugih merilih; predvsem v večjih mestih naročajo izdelavo ortofota v merilu 1 : 1000. Geometrična ločljivost skeniranja je 14 mikrometrov, kar je enako 1814 DPI. Poleg osnovne skenirane podobe se običajno zapišejo še posamezni nivoji slikovne piramide (prevzorčene podobe z različno geometrično ločljivostjo, ki omogočajo hitrejšo avtomatsko obdelavo; običajno v štirih nivojih), kar računalniški zapis osnovne podobe poveča za približno tretjino.

Ob izdelavi DOF5 se je z metodo avtomatskega slikovnega ujemanja, ki je opisana zgoraj, zajemal tudi višinski model, ki smo ga s kratico poimenovali DMR25. DMR se je sprva zajemal le na 40 metrov in se nato interpoliral na 25 metrov. Natančnost novjših podatkov DMR25 je bila na odprtem terenu ocenjena na 0,5 do 2m, v gozdu pa zaradi težav z interpretacijo in vidnostjo tal na 3 do 5 m, lahko tudi 10 m ali več. Bistveno slabšo natančnost dosega DMR 25 na tistih listih DOF 5, ki so bili narejeni s 40-metrsko celico in naknadno interpolirani v 25-metrsko mrežo. DOF5 torej zaradi načina izdelave ne dosega natančnosti državne topografske karte 1:5000 (DTK5). Zadnjih nekaj let se izvaja tudi sistemska kontrola izdelanih listov DOF 5, ki pa žal ne vključuje kontrole pozicijske natančnosti (Kosmatin Fras, 2004).

## **4 UPORABLJENE APLIKACIJE**

### **4.1 Brezplačne aplikacije (freeware)**

Pred samim pričetkom izdelave 3D-modela, sem moral poiskati aplikacije, s katerimi bi najbolj hitro in učinkovito izdelal model. Iskal sem enostaven, hitro naučljiv program, ki omogoča 3D-modeliranje, uporabo tekstur (podob prilepljenih na model) in izvoz izdelka (gre za funkcijo programa, ki izdelan model zapiše v določenem standardnem zapisu) v VRML-obliko. Iskati sem začel pri najbolj znanih 3D-modelirnih programih, katerih uveljavljenost med uporabniki so garancija za zanesljiv in funkcionalen program, kot so na primer: Maya, 3D Studio Max, Cinema4D, itd...

Dobra stran teh aplikacij je zanesljivost in funkcionalnost, širok krog uporabnikov, kar omogoča veliko gradiva za učenje in svetovanje in možnost prenosa datotek v druge

aplikacije (3D Studio Max omogoča močno povezljivost s programom AutoCAD). Take aplikacije pa imajo tudi svoje slabe strani. Ker so namenjene za močno podporo arhitekturi in modeliranju za druge namene, imajo vgrajeno široko paleto zmogljivih in učinkovitih algoritmov, ki pa jih pri izdelovanju modelov za VRML ne potrebujemo. Program je zato prostorsko preobsežen glede na uporabno vrednost te naloge, zmogljivost in učinkovitost pa zahteva pri izdelavi tudi ogromno programerskega dela in posledično denarja, zato je cena takih programov precej visoka. Ker pri izdelavi VRML-modela nebi potrebovali vseh vgrajenih algoritmov takih programov, bi bila izdelava modela s temi programi finančno vprašljiva.

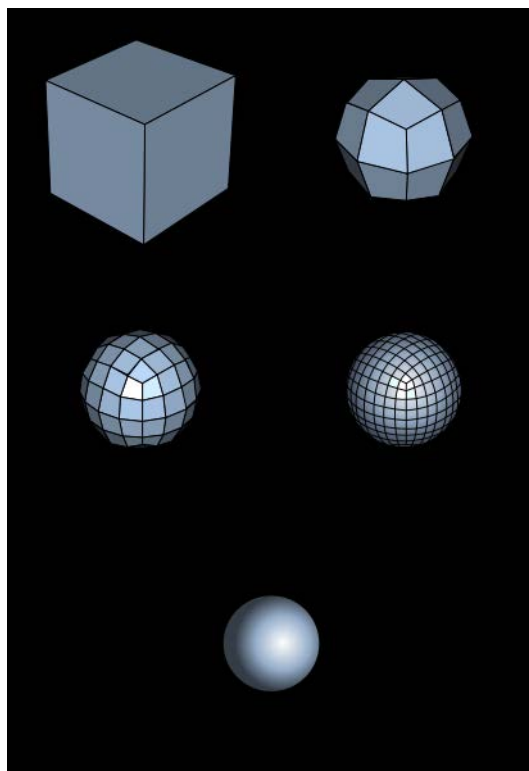
Prav zato sem se z mislimi obrnil na ponudbo brezplačnih aplikacij, ki so na voljo v medmrežju. Brezplačne aplikacije (ang. freeware oz. free software) so programi, ki jih avtorji ponujajo uporabnikom, ne da bi v zameno za uporabo programa zahtevali plačilo. Take programe različni ljudje izdelujejo predvsem iz lastnega hobija in veselja, iz nasprotovanja visokim cenam aplikacij, nekateri programi so morda ekonomsko nezanimivi, itd...

Seveda imajo programi brez plačila tudi slabe strani. Plačilo programa za proizvajalca pomeni neko obvezo, da bo program tudi brezhibno deloval. Zato je kupcu programa v določenem časovnem obdobju zagotovljena vsa dodatna podpora v obliki poporavkov in dodatkov. Brezplačnost programa pa tega ne zagotavlja, podpora je odvisna od dobre volje avtorja. Brezplačni programi si praviloma manj uporabniško dodelani in manj zanesljivi, še posebno če gre za enega samega avtorja, saj izgradnja učinkovitega kompleksnega programa zahteva več programerjev. Ta problem je bil rešen z konceptom odprtokodnih programov (ang. »open source software«), kjer ima uporabnik zastonjskega programa dostop do programske kode izdelanega programa in ga poljubno spreminja, dopolnjuje in odpravlja napake. Tak program sčasoma uporabniki dodelajo, tako da postane bolj zanesljiv in učinkovit.

## **4.2 Zastonjski program za 3D modeliranje – Wings 3D**

Wings 3D je zastonjski in odprtokodni program za modeliranje in deluje na treh najbolj znanih operacijskih sistemih (Windows, Linux in Mac). Avtorjev je več, vseh 12 je omenjenih na spletni strani programa ([www.wings3d.com](http://www.wings3d.com)). Program je narejen po zgledu podobnih programov »Nendo« in »Mirai« izdelanih v Izware organizaciji.

Vsi trije omenjeni programi modelirajo posamezne objekte po metodi »subdivision«, kar pomeni, da se gladki objekti prikazujejo z razčlenjevanjem osnovnih oglatih oblik. Tako lahko kroglo po Catmull-Clark – ovi metodi prikažemo z razčlenjevanjem vogalov kock do poljubne natančnosti.



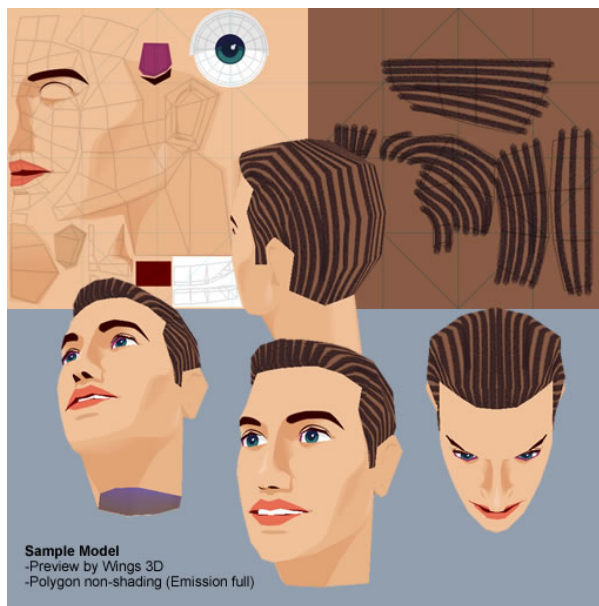
*Krogla izdelana po Catmull-Clarkovi metodi.*

*([http://en.wikipedia.org/wiki/Image:Catmull-Clark\\_subdivision\\_of\\_a\\_cube.svg](http://en.wikipedia.org/wiki/Image:Catmull-Clark_subdivision_of_a_cube.svg))*

Podobno lahko stožec prikazujemo z razčlenjevanjem vogalov osnovne ploskve piramide, itd...

Ob prvem delu s programom ugotovimo, da so osnovni ukazi precej enostavni, kar po osebni izkušnji nebi moral reči za še en modelirni zastojni program Blender. Programa se lahko, kljub precej možnostim za modeliranje, hitro naučimo in privadimo, vsaj kar se osnovnih ukazov tiče. Na polje, ki ga predstavljajo 3 glavne osi (x,y,z), lahko nanizamo osnovna telesa (kocka, valj, piramida,...), ki so osnova za nadaljnje oblikovanje zahtevnejših oblik. Osnovno preoblikovanje lahko izvajamo s premiki, zasuki in spremembo merila po izbrani eni ali več oseh. Omenjene operacije lahko izvajamo nad celotnim telesom, lahko pa izbiramo tudi zgolj posamezne ploskve, robne premice ali robne točke. Prednost takega načina dela je predvsem enostavnost in hitra naučljivost, po prvih urah dela se pa kot slabost izkaže veliko število zaporednih ukazov, ki so potrebni za posamezne operacije nad telesi. To slabost sčasoma odpravimo z iznajdljivostjo, spretnostjo in prostorsko predstavo, ki pa se krepijo – tako kot pri drugih podobnih programih – z izkušnjami.

Program uporablja nekoliko nerodno obliko nanašanja tekstur na objekte imenovano »UV mapping«, ki lahko povzroči kar nekaj sivih las, vsaj kar se te naloge tiče. Poljubno teksturo (v našem primeru fotografije stavb) v program uvozimo in jo dodelimo posameznemu telesu. Vsako telo ima lahko le eno teksturo, na kateri združimo texture za vse posamezne ploskve telesa.



*V primeru na sliki obraz »oblečemo« iz posameznih elementov, ki so nanizani na teksturi.  
(<http://www.muranon.com/axel/character/mapping.html>)*

Posamezno ploskev lahko ustrezno premikamo, sukamo in spreminjamo velikost nad teksturo v ozadju tako, da ustrezno zajame željen del texture.

Program poleg drugih vrst datotek izvaša končni izdelek tudi v VRML-obliki, kar je bil tudi eden glavnih pogojev pri izboru.

### 4.3 Zastonjski X3D urejevalnik – X3D EDIT

X3D – Edit je zastonjski urejevalnik X3D-datotek, ki omogoča enostavno delo z X3D-gradniki in njihovimi atributi. Avtor programa je Don Brutzman, računalniški strokovnjak iz Združenih držav, zaposlen na Podiplomski pomorski šoli v Kaliforniji.

X3D ali VRML-urejevalniki so pravzaprav logična poenostavitev dela s kodo ene ali druge vrste zapisa. Čeprav sta obe vrsti zapisa dokaj logični, pregledni in hitro naučljivi, pa lahko z orodjem, kot je urejevalnik, pohitrimo in poenostavimo urejanje kode. Glavna lastnost urejevalnika je, da izboljša preglednost kode. Urejevalnik ohranja drevesno strukturo kode, vendar dodaja možnost skritja podrejenih elementov in podrobnosti, ki nas v tistem trenutku ne zanimajo. To je prvi ukrep izboljšanja preglednosti, saj nam za iskanje določenega elementa ni treba iskati po enoličnem zapisu kode dolgem več strani. Drugi ukrep je uvedba ikon. Vsak tip elementa ima svojo ikono, katero si možgani zapomnijo in jo pri iskanju elementa zagledamo precej hitreje, kot zgolj ime elementa, ki je skrit v množici elementov in atributov, vseh opremljenih z eno in isto vrsto in barvo pisave.

Tretja prednost urejevalnika je napeljevanje na iskano rešitev. Če urejamo kodo brez urejevalnika, moramo za učinkovito delo poznati vse možne elemente, njihove attribute in kakšen je sploh rezultat uporabljenih elementov in atributov. X3D – edit ima vse možne elemente, ki jih lahko uporabimo, nanizane v svojem okvirju, po katerem lahko iščemo. Če odkrijemo kak nov atribut ali element, ki ga še ne poznamo, ali pa nam njegovo delovanje ni povsem jasno, nam urejevalnik že sam omogoča kratek opis atributa ali elementa. Za nalogo

pomembna funkcija programa, pa je tudi možnost uvoza VRML-datoteke in pretvorba v X3D-zapis.

## **5 POTEK IZDELAVE MODELA**

### **5.1 Izdelava stavb**

#### **5.1.1 Geometrično oblikovanje**

Stavbe sestavimo iz dveh bistvenih elementov in sicer iz geometrije stavbe oz. geometričnih elementov, ki prikazujejo geometrijo stavbe in tekstur stavbe. Pred samo izdelavo posamezne stavbe je potrebno zajeti njene prave geometrične in vizualne lastnosti. Ker izdelujemo model v prvi vrsti z namenom vizualne uporabnosti in ne z namenom pridobivanja realnih metričnih podatkov iz modela, natančnosti podatkov posvečamo manj pozornosti. To je pomembno tudi iz finančnega vidika, saj imajo natančni merski podatki zaradi natančnega zajema določeno vrednost in ceno, ampak v model namenjen vizualni predstavitvi glede na njihovo ceno ne prinesejo bistvene dodatne vrednosti. Zato se poslužimo manj natančne metode in sicer uporabimo DOF-posnetek iz katerega zajamemo dolžino in širino strehe. Višino stavbe in druge mere podrobnejših detajlov stavbe pridobimo iz fotografij, ki smo jih posneli na terenu z namenom pridobitve geometrije in oblike stavb in izdelavo tekstur stavb. Pridobljene mere strehe iz DOF-posnetka sorazmerno primerjamo z ostalimi merami, ki so vidne na fotografiji in na tak način pridobimo mere stavbe. Ker fotografije na terenu pogosto niso povsem pravokotne glede na lego stavbe, so zaradi lastnosti perspektivne projekcije tako pridobljene meritve nekoliko popačene in netočne, vendar se razmerje posamezne izmerjene vrednosti do drugih vrednosti na stavbi ali do vrednosti na drugih stavbah ne poruši tako močno, da bi oblika in mera stavb opazno kvarila vizualno kvaliteto končnega modela. Mere in obliko stavbe, ki smo jih iz vrednotili iz DOF-posnetka in na terenu posnetkih fotografij zabeležimo v ročno izdelano skico, ki nam pomaga pri oblikovanju modela stavbe.

Oblikovanje stavbe izvedemo v programu Wings 3D. Obliko stavbe poizkušamo čimbolj poenostaviti v najosnovnejša geometrijska telesa in se poizkušamo izogibati zapletenim oblikam, saj te v primerjavi z osnovnimi telesi zavzemajo več prostora na disku, prikazovanje tekstur na takih oblikah je težje, bolj pa obremenijo tudi algoritme X3D-prikazovalnika. Tukaj gre za večno kartografsko dilemo, saj moramo doseči kompromis, pri katerem bo stavba dovolj avtentično prikazana, vendar ne na račun zapletenih geometričnih oblik. Zato se je potrebno vsakič posebej odločati, kateri elementi dajejo stavbi razpoznavni znak in so vizualno bistvenega pomena in katere elemente lahko zanemarimo in jih ne prikažemo. Po izkušnjah pri izdelavi modela so najlažje prikazljive novejši gospodarske stavbe in prodajni centri, katere obliko lahko sestavimo le iz nekaj kvadrov in lahko dosežemo visoko stopnjo avtentičnosti kljub majhni geometrični zahtevnosti. Najtežje pa se prikazuje starejše stavbe, ki so zaradi nesimetričnosti, mnogih dograjevanj v preteklosti in zapletenih streh lahko precejšen problem, saj porabimo za avtentični prikaz stavbe več časa in uporabimo več zapletenih geometrijskih oblik, ki bolj obremenjujejo končni prikaz modela.

Pri tem velja omeniti pomembno lastnost prikazovanja 3D-modelov, pri kateri ne igra bistvene vloge velikost objekta ampak njegova geometrična zahtevnost. Tako lahko več deset metrov velik trgovski objekt enostavnih oblik manj obremenjuje prikaz modela, kot na primer ena sama geometrijsko zapletena streha majhne stanovanjske hišice. Po izkušnjah lahko

rečem, da model med drugim najbolj obremenjujejo prav zapletene strehe. Pri tem se nam pravzaprav pojavi vprašanje, kako natančno naj pravzaprav strehe prikažemo? Odgovor ni lahek predvsem zaradi naslednjega paradoksa: v končnem modelu bomo stavbe pretežno gledali z vrha navzdol, pri premikanju skozi mesto v realnem okolju, pa opazujemo stavbe predvsem od spodaj navzgor ali naravnost. Pogled z vrha si redkeje privoščimo zgolj na razglednih točkah mesta. Odgovor nam bo verjetno dal model sam; če bodo strehe prezahtevno in preveč podrobno prikazane, bo model omejeno uporaben ali pa sploh ne.

S programom Wings 3D se enostavne geometrijske oblike, ki so tudi najbolj pogoste, oblikuje relativno hitro in enostavno in ne zahteva večjega znanja in spretnosti. Nekoliko se zaplete pri oblikovanju zahtevnejših oblik, kjer so zaželenje daljše izkušnje s programom, poznavanje naprednejših funkcij programa in veliko spretnosti.

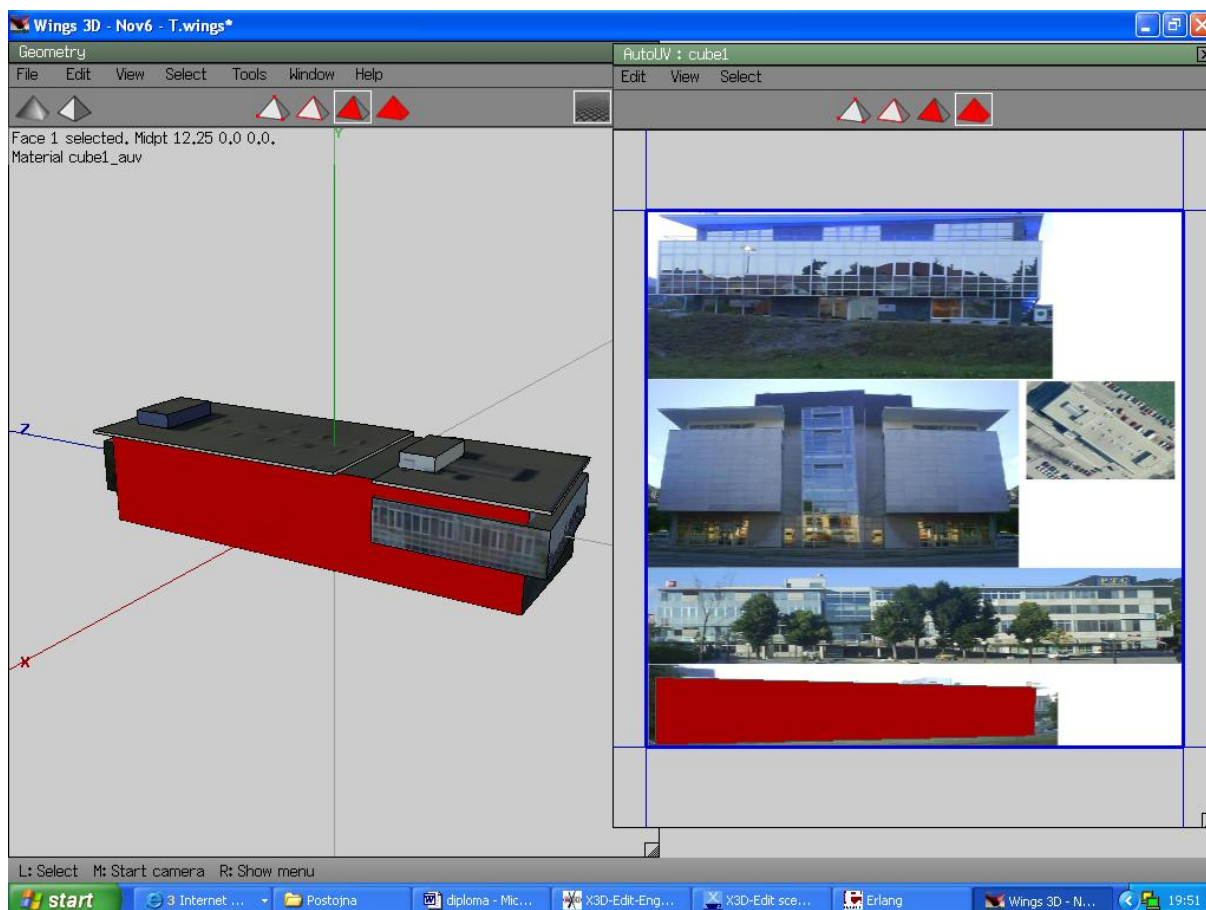
### **5.1.2 Izdelava tekstur**

Ko je stavba geometrično oblikovana jo lahko »oblečemo« v texture. Texture pridobimo iz fotografij, ki smo jih posneli na terenu za vsako stavbo posebej. Fotografiranje na terenu je učinkoviteje, ko že izdelamo nekaj stavb s teksturami, saj takrat vemo na kaj se je potrebno osredotočiti in kaj lahko izpustimo. Z izkušnjami zmanjšamo število posnetih fotografij stavbe in čas zajema. Fotografiramo iz dveh namenov in sicer za izvednotenje geometričnih oblik stavbe ter za pridobitev tekstur stavbe. Pomembno je, da sliko zajemamo čim bolj pravokotno na steno stavbe. To nam prvič zagotavlja bolj natančno izvednotenje geometrije iz fotografije, saj se tako zmanjša vpliv perspektivne projekcije in dobimo verodostojne rezultate, pravokotnost pa zagotavlja tudi realen in nepopačen izgled texture na stavbi. Ker pa zaradi ovir in nedostopnosti (ozke ulice, vegetacija, ograje,...) ne moremo vedno stavbe posneti pravokotno na steno, uporabimo bodisi popačene texture ali pa steno upodobimo zgolj v barvi fasade, brez oken vrat in drugih podrobnosti. Na teksturah se pojavljajo tudi predmeti in vegetacija, ki jih seveda na stenah stavb ni, so pa postavljeni med žarke perspektivne projekcije, ki poteka od fotoaparata, do posnete stavbe (avtomobili, telekomunikacijski tabli, ljudje, drevesa, javna razsvetljava, prometne označbe,...). Ta situacija je neizogibna, saj bi grafično urejanje in odstranjevanje motečih predmetov vzelo preveč časa za premalo učinka. Pri pogledu na končni model se izkaže, da je najpomembnejši splošni vtis texture, ki ga moteči predmeti načeloma ne pokvarijo, če se na teksturi pojavljajo v dovolj majhnem deležu. Pojavijo se tudi situacije, ko drevesa popolnoma zakrivajo pogled na stavbo, takrat lahko prikažemo zakrito teksturo, lahko zgolj nakažemo barvo fasade, ali pa texture preprosto ne uporabimo, kar je v primeru omejene količine podatkov še najboljša rešitev. Olajševalna okoliščina težko dostopnih in zakritih tekstur je ta, da jih tudi ljudje v realnem okolju težko opazijo in jih niso vajeni. Ljudje so najbolj vajeni tistih objektov, katere se najboljše opazi iz več strani, take pa se tudi najlažje posname.

Za snemanje fotografij lahko uporabimo povprečen fotoaparat. Če imamo na voljo kvalitetnejši fotoaparat, nam sicer v določenih primerih olajša delo (oddaljenost od objekta in kvaliteta barv), vendar pa je načeloma važen splošni vtis texture in ne podrobnosti.

Zajete podobe ne zahtevajo veliko obdelave. Za pripravo tekstur povsem zadostuje že aplikacija Slikar, ki je del Microsoftovega operacijskega sistema - Oken. Za morebitno natančnejšo obdelavo je na internetu na voljo kar nekaj zmogljivih zastojnih programov za obdelavo podob (npr. Gimp). Podobam največkrat odrežemo odvečno vsebino, jih zmanjšamo ali jih združujemo. Združevanje podob zahteva program Wings 3D (metoda UV Mapping), saj je za vsak element (kocka, stožec,...) stavbe potrebno definirati skupno teksturo. Če

naprimer kocka s štirimi ploskvami predstavlja štiri stene stavbe, moramo v teksturo združiti posnetke vseh štirih strani stavbe. Za več elementov pa lahko uporabimo isto teksturo, česar se poizkušamo držati, saj več tekstur pomeni več zavzetega prostora. Več podob torej združimo v eno samo, v programu pa za vsako ploskev posebej navedemo kater del podobe naj uporabi za teksturo. Vsaka ploskev ima torej definirane svoje slikovne koordinate, ki prikazuje njeno lego na podobi. Ploskev (in ne podobo, kot bi morda pričakovali) lahko glede na podobo poljubno vrtimo in premikamo.



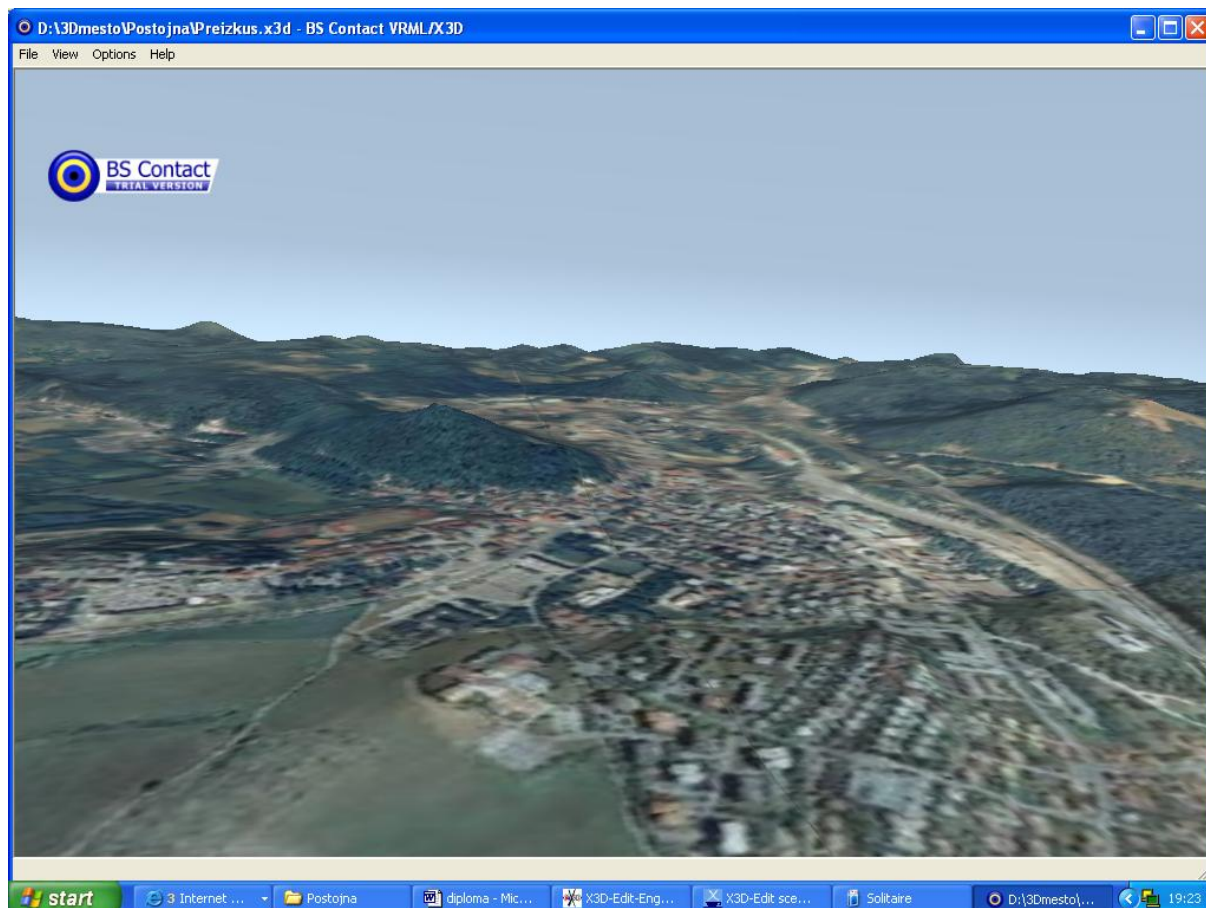
Slika 2: Metoda »UV mapping« programa Wings 3D

## 5.2 Prikaz površja v X3D obliki

Teren v modelu prikažemo z geodetskimi podatki, ki so na voljo. To sta v našem primeru digitalni model reliefa z 12,5 metrsko razdaljo med točkami (DMR12,5) in digitalni ortofoto v merilu 1:5000 (DOF5). Ker se površina DMR v eni datoteki površinsko pokriva z DOF, je prikaz terena v X3D-obliki zelo enostaven. X3D-jezik vsebuje element »elevation grid« oz. višinsko mrežo, torej gre za element namenjen prav za prikazovanje površja. V atributih elementa navedemo število vrstic in stolpcev mreže, razdaljo med stolpci in vrsticami v metrih, in višine vseh točk v mreži. V atributu »creaseAngle« pa navedemo stopnjo zglajenosti prikazanega modela (0 = nezglajeno, 3,14 = povsem zglajeno). Model sedaj lahko prikažemo, vendar je brez prikazane vsebine. Zato elementu »elevation grid« v okviru nadrejenega elementa »shape«, dodamo še element »appearance«, ki definira zunanji videz.



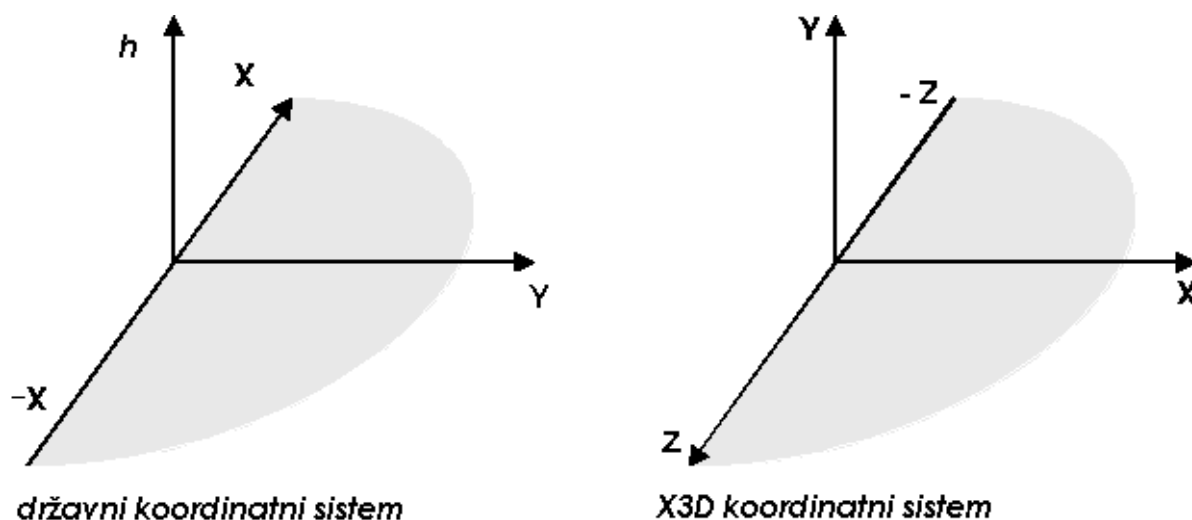
Elementu »appearance« dodamo podrejeni element »ImageTexture«. Ta element predstavlja povezavo do podobe, ki jo želimo prikazati na modelu reliefa, v našem primeru je to povezava do ustreznega DOF-posnetka. Vsaka tekstura pa se po tako definirani začetni vrednosti v X3D izriše narobe in sicer »z glavo navzdol«. Zato moramo poleg elementa »ImageTexture« dodati še element »TextureTransform«, ki teksturo prikaže pravilno obrnjeno. V našem primeru spremenimo atribut »scale« iz vrednosti »1,1« na vrednost »1,-1«. Sedaj lahko prikažemo model reliefa z ustrezno vsebino v X3D prikazovalniku.



*Slika 3: Prikaz DMR in DOF v X3D obliki z ustreznim prikazovalnikom.*

Tako definiran model pa se pojavi v izhodišču X3D-koordinatnega sistema, zato so koordinate na površju definirane od 0, pa do meja DMR in DOF posnetka (2250 in 3000 metrov). Želimo si, da bi X3D-koordinate ustrezale koordinatam državnega koordinatnega sistema, saj lahko le tako umeščamo posamezne stavbe na pravilno mesto v modelu. Uporabimo element »Transform«, v katerega navedemo koordinate državnega koordinatnega sistema (vrednost, ki ustreza vogalu prikazanega DOF in DMR). Vendar pa se tu pojavi težava. X3D-koordinatni sistem se ne sklada z državnim koordinatnim sistemom. Da so zamenjana imena osi (X je y, Z je x, Y je h) še ne predstavlja večjega problema. Problem predstavlja dejstvo, da X3D predvajalnik vsebino v državnih koordinatah preslika prek X osi, saj pozitivna os Z, ki predstavlja državno koordinato x, poteka v smeri negativne državne koordinate x.





Slika 4: Razlika med DKS in X3D-koordinatnem sistemom

Prikazana vsebina je tako prezrcaljena v smeri y (vzhod je na zahodu in obratno). Možnost rešitve predstavlja označevanje Z osi (oziroma x DKS) s predznakom minus, kar verjetno prinaša še najmanj dodatnega dela in napak. Ko upoštevamo vse zgoraj navedeno, imamo končno pripravljen pravilno umeščen model reliefa za dodajanje dodatne vsebine.

### 5.3 Elementi animacije in drugi elementi v X3D

Prva verzija VRML (VRML 1.0) je omogočala prikaz elementov v 3D-prostoru, ni pa omogočala premikanja elementov v modelu med opazovanjem, oziroma animacije. To se je povsem spremenilo z drugo verzijo VRML 2.0 oz. VRML 97, ko se je lahko statičen svet spremenil v gibljivega. X3D-zapis je v celoti povzel lastnosti in sposobnosti zadnje verzije VRML in dodal nove.

Osnova animacije je spreminjanje stanja oziroma atributov elementov med samim opazovanjem modela. Spreminjamo lahko tako lego, usmerjenost in velikost elementov, kot tudi barve ali teksture. Za obvladovanje animacije pa nujno potrebujemo enega osnovnih pogojev animacije – obvladovanje časa, zato »zna« X3D-zapis med prikazom meriti čas. Za obvladovanje časa skrbi več uporabnih elementov. Prvi izmed njih je »TimeSenzor«, oziroma senzor časa. Običajno deluje skupaj z elementom »Interpolator«. Prvi določa časovno dolžino dogodka (premika, zasuka, predvajanja,...) in morebitno ponavljanje dogodka, drugi pa znotraj te določene časovne dolžine določa dinamiko sprememb. Interpolatorjev je več vrst, ki so prilagojeni atributom posameznih elementov.

X3D-zapis omogoča tudi uporabo predvajanja multimedijskih datotek, kot sta zvok in filmske datoteke. Zvok in filmske datoteke lahko omejimo bodisi prostorsko, pri čemer jima določimo jakost in radij širjenja zvoka, bodisi časovno, pri čemer jima določimo začetek in konec predvajanja ter morebitno ponavljanje.

Uporabna sta tudi elementa »Viewpoint« in »Anchor«. Prvi element določa posamezno lokacijo in smer pogleda na element. Večina predvajalnikov omogoča sprehajanje po teh vnaprej določenih točkah pogleda. Te elemente uporabimo, če želimo uporabniku ponuditi kak uporaben, pomemben ali lep pogled na vsebino z določene perspektive. Ta element se

pogosto družiti z elementom »Anchor«, ki omogoča, da se na vnaprej določeno perspektivo pomaknemo s klikom na določen predmet ali skupino predmetov.

Animacije pa nebi bilo brez ključnega elementa v X3D-zapisu, ki pošilja podatke od enega elementa do drugega. To je element »ROUTE«, ki je neke vrste usmerjevalec sporočil. Ta element uporabimo pri vseh prej naštetih in drugih elementih, ki si za dinamični prikaz izmenjujejo vrednosti različnih vrst (številске, logične, časovne,...). Element »ROUTE« sestavljajo štiri vrednosti, dve za ime elementa in atributa od koder vrednost prihaja in dve za ime elementa in atributa, kamor je vrednost namenjena. Ime posameznega elementa moramo definirati v njegovem atributu »DEF«, imena pa se ne smejo podvajati. Imena atributov so že določena v okviru X3D-zapisa.

Omenimo še dva elementa, ki vizualno izboljšujeta model. To sta ozadje oziroma »Background« in megla oziroma »Fog«. Ozadje je podoba, ki jo vidimo, če nam pogleda ne zastira nobeden od geometričnih elementov, ki smo jih prikazali. Če ozadja ne definiramo, je obarvano črno. Ozadje prilagajamo namenu modela. V našem primeru, ko izdelujemo model na zemeljskem površju, ozadje prikažemo v barvah neba, s čimer ohranjamo realnost modela. Tudi element, ki prikazuje meglo, je v našem primeru lahko zelo uporaben in avtentičen. Meglo pri modelu zemeljskega površja lahko uporabimo za prikrivanje robov zemeljskega površja, s čimer ohranjamo realnost, saj je megla naraven pojav in smo ga ob pogledu na pokrajino vajeni.

## 5.4 Združitev vseh elementov v končni model

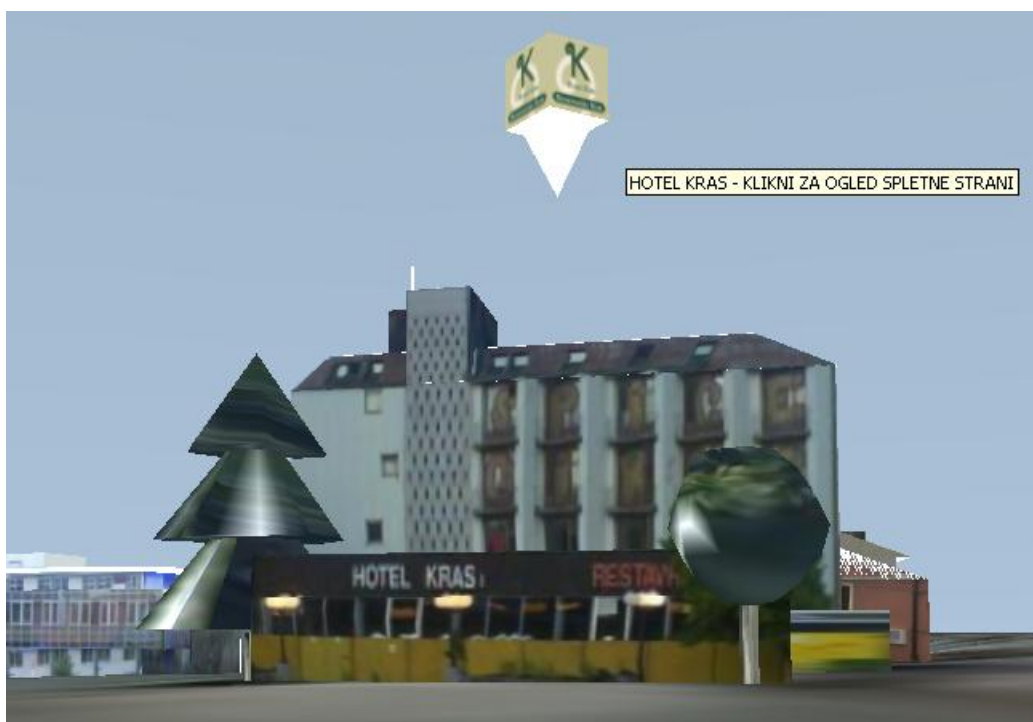
Izdelane modele posameznih stavb v programu Wings 3D lahko izvozimo v obliko VRML, katero potem lahko uvozimo v program X3D – Edit. Ker imajo vse tri vrste datotek (Wings 3D, VRML in X3D) identičen koordinatni sistem, tu ni težav. V X3D-obliko uvožena stavba, je sedaj množica »Transform« elementov (»Transform« elementi določajo medsebojno lego osnovnih geometričnih elementov, ki skupaj tvorijo enotno obliko - stavbo), ki so nadrejeni posameznim geometričnim in teksturnim elementom. Te elemente združimo v še en nadrejeni »Transform« element, ki stavbo umesti na njeno ustrezno mesto na prikazanem modelu reliefa. Stavbo transformiramo s pomočjo koordinat DKS, ki jih pridobimo z DOF, višino pa nastavimo s poizkušanjem. Stavbo, če je potrebno, tudi ustrezno zasukamo, vse to z atributi elementa »Transform«.

Če je skupna velikost modela, ki ga model zavzema na disku oz. na serverju še sprejemljiva, lahko za boljši vizualni učinek izdelamo tudi druge elemente, kot so na primer drevesa, luči, semaforji, prometni znaki in podobno. Z izdelavo dreves izboljšamo realnost modela, ob enem pa pokrivamo stene stavb, ki jih tudi v realnem okolju ni videti. Smotno je izdelati drevesa na način, ki najmanj obremenjujejo model pri uporabi, saj se s tem izognemo tudi prikazu potratnih tekstur na stavbah, ki jih lahko z drevesi prekrijemo.

Z elementi animacije lahko statični model še poživimo. Z animacijskimi elementi »TouchSensor«, »TimeSensor« in »PositionInterpolator«, katerih podatke povezujejo elementi »ROUTE«, lahko ustvarimo relativno enostaven in zanimiv način pridobivanja podatkov o stavbah. Element »TouchSensor« umestimo v skupino, ki združuje vse geometrične oblike posamezne stavbe. Sedaj lahko pri uporabi modela kliknemo na stavbo z miško in element »TouchSensor« zabeleži čas klika. Ta se pošlje do elementa »TimeSensor«, ki se sproži kot stoparica in meri vnaprej definiran čas. »TimeSensor« pošilja podatke o

vnaprej definiranem časovnem obdobju elementu »PositionInterpolator«. V tem elementu vnaprej definiran čas, ki ga pošilja »TimeSenzor« razdelimo na poljubno število časovnih enot. Te časovne enote so lahko medseboj enako dolge, lahko pa poljubno različne, s čimer se ustvarja različno hitrost premika elementa. V našem primeru razdelimo interval desetih sekund na dvajset enako dolgih časovnih enot. Vsaki časovni enoti, moramo definirati poljubno 3D-koordinato v X3D-koordinatnem sistemu (lahko bi rekli, da definiramo »urnik« ali »vozni red« premikajočega se elementa). Ko »TimeSenzor« sproži čas, »PositionInterpolator« pošilja izbran geometrični element od ene koordinate do druge, po definiranem »urniku«. (»PositionInterpolator« pošilja podatke elementu »Transform«, ki je nadrejen poljubnim geometričnim elementom, ki jih želimo premikati).

V našem primeru smo z zgoraj opisanimi postopki ustvarili gibljiv označevalnik stavbe. Ta je sestavljen tako, da nakazuje puščico z oznako podjetja, ki ima v stavbi sedež. Če na stavbo ne kliknemo, ta znak nad stavbo v nevidnem območju čaka na klik. Ko pa kliknemo na stavbo, se s pomočjo elementa »PositionInterpolator« znak spusti nad stavbo v vidno območje in tam niha deset sekund. Nihanje smo definirali v dvajset časovnih enotah po pol sekunde, ki skupaj sestavljajo deset sekund. Vsakemu enosekundemu intervalu smo definirali premikanje za meter dol in nazaj gor, kar ustvarja nihanje in s tem vizualno privlačnost. Sedaj lahko znak, ki predstavlja stavbo, »pripnemo« še na element »Anchor«, ki povzroči, da se s klikom na nihajoči znak odpre spletni brskalnik z spletno stranjo podjetja, ki ga znak predstavlja.

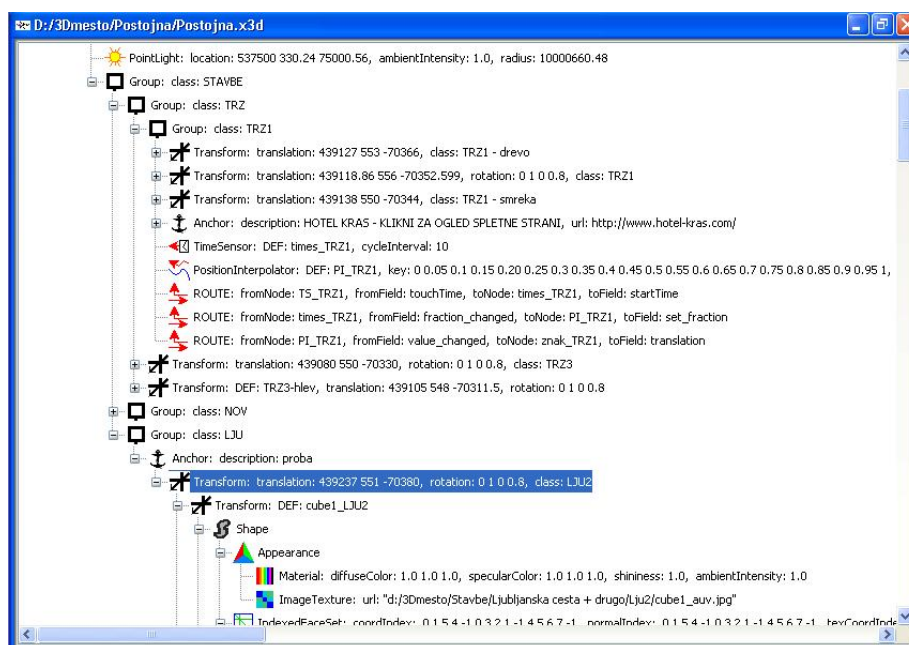


*Slika 5: Izsek iz 3D-modela, na katerem je vidna stavba in njen znak, ki se pokaže s klikom na stavbo. Klik na znak stavbe nas poveže s spletno stranjo podjetja v stavbi.*

Na podoben način lahko ustvarjamo podobne elemente, ki omogočajo zanimivo ogledovanje modela. Z elementi animacije lahko statični model poživimo z elementi, ki se gibljejo tudi v naravi. Lahko na primer izdelamo letalo, ki preleti mesto, vlak, ki se zapelje po tirih, morda nevihtni oblak, ki se »pripelje« nad mesto, iz katerega švigajo strele in s pomočjo predvajanja zvoka, lahko celo zagrmí. Z zvokom lahko simuliramo petje ptic v parku, zvonjenje iz zvonika

in tako dalje... Pri uporabi elementov nas omejuje zgolj iznajdljivost in domišljija, še najbolj pa tehnične omejitve, saj si želimo manj prostoren in za prikazovanje manj zahteven model.

Ker se v X3D-zapisu pri združitvi vseh opisanih elementov v model pojavi velika nepregledna množica X3D-elementov, moramo za učinkovito delo vzpostaviti pregledno strukturo teh elementov. Posamezne elemente zato združimo v istovrstne skupine, naprimer teren posebej, stavbe posebej, itd. Stavbe še nadalje razčlenjujemo glede na ulico, v kateri stojijo. Vsaki stavbi določimo ID-oznako, ki se pri drugi stavbi ne sme ponoviti. Sam sem stavbe poimenoval po naslovu: ID-oznaka je tako sestavljena iz prvih treh črk naslova in hišne številke, za stavbo na Notranjski ulici 5 je naprimer ID-oznaka »NOT5«. Za animacijske in druge elemente, ki jih dodajamo stavbam, tvorimo ID na naslednji način: vsak element »TouchSensor« poimenujemo po kratlici z »TS« dodamo mu pa še oznako stavbe, pri kateri je uporabljen (npr. TS\_NOT5). S tem zagotovimo neprekrivanje oznak, preglednost in učinkovitost.



Slika 6: Izsek iz programa X3D – EDIT. Množica X3D elementov, ki sestavljajo model, je za lažje delo drevesno členjena in označena z ikonami

## 5.5 Zmanjševanje prostorske obsežnosti podatkov

Čas v informacijski tehnologiji je opazen dejavnik, saj je razvoj tehnologije relativno hiter. Če danes kupimo neko opremo ali tehnologijo, je jutri lahko že zastarela. Za uporabnika tehnologij je to lahko bodisi slabo, lahko pa tudi dobro. Če vemo, da se hitrost prenosa podatkov in zmogljivost računalnikov povprečnega uporabnika interneta in s tem potencialnega uporabnika modela te naloge povečuje, je čas na strani tu predstavljenega modela. Razumeti je treba, da je realen model večjega mesta opisan na ta način za prenos podatkov, programsko in strojno opremo še vedno velik zalogaj. Ker se okoliščine izboljšujejo sčasoma, smo torej obsojeni na sprejemanje kompromisov med vizualno realnostjo in uporabnostjo modela.

Zmanjševanja prostorske obsežnosti podatkov v smislu zavzemanja prostora podatkov na disku se lotevamo na dva načina: z zmanjšanjem bitnih vrednosti, ki prenašajo informacijo o

uporabljenem podatku in z uporabo boljših podatkov bolj izpostavljenih predelov in slabših podatkov na relativno skritih predelih modela.

### **5.5.1 Krčenje bitne vrednosti podatkov**

Ko je model iz vseh opisanih sestavin končno sestavljen, lahko ugotovimo, da bi praktično pri vsakem uporabljenem elementu in metodi prihranili nekaj bitne prostornine. Pri oblikovanju stavb smo že omenili zahtevo po uporabi čim bolj osnovnih geometričnih elementov, ki jih naj bo čim manj. Prikazovanje geometričnih lastnosti stavb in drugih elementov prostornino in prikaz modela obremenjuje v relativno majhnem obsegu. Veliko bolj obremenjujoče so teksture samih stavb in drugih elementov, torej digitalne podobe. Če imamo geometrično razdrobljeno stavbo in na vsakem posameznem delu določeno teksturo, taka stavba močno obremenjuje algoritme predvajalnika. Poleg geometrični obliki se bomo morali posvetiti tudi krčenju uporabljenih tekstur. Uporabiti je treba čimbolj avtentično podobo stavbe ob čim manjši prostornini podobe, torej je za izdelavo modela potrebno dobro poznavanje lastnosti rastrskih podob, posameznih formatov in metod krčenja podob.

Tudi uporabljene geodetske podlage imajo dosti možnosti za krčenje. DOF krčimo tako kot druge teksture, medtem ko je DMR poglavje zase. Pri DMR imamo dva parametra, ki ju v procesu krčenja lahko variramo. Eden izmed parametrov je razdalja med posameznimi točkami DMR. DMR5 veliko bolj natančno prikazuje prikazano površje, omogoča boljše prilaganje k stavbam in bolj realen prikaz, a zavzema veliko prostora; DMR25 ravno obratno, DMR12,5 je logično nekje vmes. Drugi parameter pa je točnost višinskih podatkov. V DMR12,5 so višine podane s točnostjo centimetra, kar pri uporabi pomeni 6 znakov (npr. 534,23). Če podatek zaokrožimo na meter, imamo le še 3 znake za prikaz, s čimer zmanjšamo količino podatkov, a s tem nekoliko pokvarimo natančnost prikaza. Obeh parametrov se tako kot povsod lotevamo z iskanjem kompromisa med sposobnostjo za prikaz in realnostjo prikaza.

### **5.5.2 Variranje uporabe podatkov glede na vidnost**

Drugi ukrep za zmanjševanje prostorske obsežnosti podatkov je variranje uporabe podatkov glede na vidnost. Vsako mesto ima bolj in manj zanimive predele. Vsem potencialnim uporabnikom modela (npr. meščanom, turistom,...) bo najbolj zanimiv center mesta, saj je tam običajno v majhnem prostoru združenih več pomembnih objektov in ustanov, tam pa se običajno srečujejo poti iz različnih predelov mesta. Bolj zanimive so tudi lokacije s turističnimi znamenitostmi, trgovskimi centri, stanovanjski predeli z večjo koncentracijo ljudi. Logično je, da se bomo bolj zanimive predele prikazali bolj kvalitetno in natančno kot manj zanimive. Pri tem pa je treba upoštevati še eno pomembno vizualno lastnost: manj natančno prikazan objekt bo v daljavi izgledal enako kvaliteten, kot bolj natančno prikazan objekt v bližini. Na podlagi teh lastnosti lahko torej učinkovito variramo uporabo podatkov in zmanjšujemo kvaliteto prikaza proti obrobju mesta, s čimer ohranjamo vizualno kvaliteto in hkrati prihranimo na prostorski obsežnosti in manj obremenjujemo algoritme predvajalnika. Natančnost prikaza variramo tudi na posameznih objektih, kot je to že opisano v izdelavi tekstur.

## 5.6 Postavitev modela na splet

VRML in X3D sta že od nastanka predvsem spletno orientirana zapisa, saj so mnogi predvajalniki v bistvu vključki (angl. plug-ins) spletnih brskalnikov, zadnje čase pa vse bolj tudi samostojni programi. Postavitev modela na splet je zato relativno enostavna. V programu Microsoft Frontpage, aplikaciji za izdelavo spletnih strani, model, predstavljen v X3D-datoteki preprosto dodamo v HTML-kodo z enim samim ukazom (Insert Web Component, Advanced Controls, Plug-In), pri čemer dodamo sporočilo za obiskovalce spletne strani, ki še nimajo nameščenega vključka, s spletno povezavo, kjer je ta vključek na voljo. Definiramo še velikost okna, in njegovo poravnavo znotraj objavljene spletne strani. Samostojno stoječi, zgolj na X3D-zapisu in X3D-redvajalniku sloneči model je pripravljen za spletno objavo in uporabo.

Model pa bi lahko naredili še bolj uporaben, če bi ga povezali z elementi spletne strani in bi ga od tam lahko tudi upravljali. Kot nek primer možne uporabe lahko navedemo informacijsko spletno stran za prebivalce in morebitne obiskovalce mesta, podprto z spletnim 3D-prikazom. Spletna stran bi lahko vsebovala bazo vseh obiskovalcem zanimivih ustanov in sedežev komercialnih podjetij z osnovnimi opisi, po kateri bi lahko uporabniki poizvedovali, rezultat poizvedovanja pa bi bil tudi 3D-prikaz lokacije. Lahko bi dodali še prikaz načrtovanih dogodkov v mestu, komercialno naravnane predstavitve ponudbe, in podobno. Tehnično dovršena, vizualno privlačna in informacijsko bogata in uporabna spletna stran, bi lahko nudila zadostno število obiskovalcev, da bi s sponzorskim denarjem lahko pokrivali stroške izdelave in vzdrževanja tovrstne spletne strani.

Določene pomisleke o uporabnosti spletne strani v povezavi z X3D-modelom vzbuja problem prenosa podatkov s strežnika. Novi obiskovalec spletne strani, bi si moral s spleta najprej pridobiti ustrezen predvajalnik, ki bi podprl prikaz modela v X3D, potem pa bi moral pridobiti še vse podatke, ki sestavljajo 3D-prikaz. Internetni uporabniki smo z povečevanjem zmogljivosti prenosa postali razvajeni in smo vajeni hitrega nalaganja spletnih strani brez daljšega čakanja. Vizualno kvaliteten model pa bi zaradi svojega obsega zahteval daljše čakanje, ki bi bilo v povezavi z prej opisano spletno vsebino za marsikoga moteče in posledično pogojno uporabno.

Rešitev za ta problem bi bili različni koncepti kombinacije prenosa podatkov, ki jih nekatere spletne aplikacije že uporabljajo. Omogočajo na primer takojšnjo uporabo aplikacije, ko najprej vidimo le osnovne obrise, ostala vsebina pa se nalaga še med tem, ko aplikacijo že uporabljamo. Nekatere aplikacije osnovne podatke na gostujoči računalnik prenesejo samo enkrat, ob naslednjem obisku pa se prenašajo samo nastale spremembe v podatkih od zadnjega ogleda in podobno. Osnovna X3D-datoteka takih možnosti ne dopušča. Ob morebitni vsaki najmanjši spremembi podatkov, moramo zamenjati celotno datoteko, ki jo mora potem uporabnik na svoj računalnik znova prenesti v polnem obsegu. X3D-datoteka se lahko običajno uporabi šele takrat, ko se prenese celotna vsebina modela, tudi tista, ki jo morda uporabnik sploh ne bo pogledal. Za zgoraj opisano uporabo modela bi torej morali v sam spletni prikaz vložiti še nekaj več časa in truda.

## 6 OPIS POSTOJNE

Mesto Postojna s čez 8000 prebivalci in sedežem občine leži v jugozahodnem delu Slovenije in predstavlja tudi regionalno središče Notranjske in tudi možni sedež nove Notranjsko–kraške regije. Umeščena je na rob Pivške kotline, tik pod Postojnskimi vrati (609m), ki predstavljajo najnižji prehod med Zahodno Evropo in Vzhodno Evropo nižje od Alp. V bližini se reka Nanoščica izliva v Pivko, ki ponikne v Postojnski jami in dolga leta tvori to kraško znamenitost. Neposredno ob mestu se s svojimi 677 metri strmo vzpne Sovič, ki opazno posega v obliko mesta, ki se okrog vzpetine razvija že od svojega nastanka. Tudi nekoliko odmaknjena Pečna Reber (763m), ena prvih vzpetin, ki na vzhodu začenjajo gozdove in kraške planote Javornikov, tudi nudi lep pogled na mesto ter Pivško kotlino proti Razdrtem in Nanosu.



*Grb občine Postojna, ki slovi kot eden najlepših Slovenskih grbov.  
([www.postojna.si](http://www.postojna.si))*

Postojno se v zgodovinskih virih prvič omenja leta 1226, domnevno pa že 1149. Leta 1251 je prvič omenjen grad Postojna (castrum Arnesprech), leta 1262 pa vas Postojna (villa de Arnesprech). Središče posesti je bil prav grad, ki je pogosto menjal lastnike. Leta 1689 je grad na Soviču pogorel in leta 1700 so ob vznožju zgradili novo graščino, ki stoji še danes v samem središču mesta, danes je v njem Inštitut za raziskovanje krasa ZRC SAZU. Največ je k nastanku Postojne pripomogel pravzaprav promet, ki se je ob koncu 13. stoletja v teh krajih močno povečal, začetki prometa čez te kraje pa segajo do rimskih časov in še dlje. Pot iz Ljubljane do Postojne in od tod v eno smer proti Reki in v drugo proti Trstu, se je imenovala »cesarska cesta«. Leta 1748 je postala Postojna sedež Notranjske kresije, sredi 19. stoletja pa je kresijski urad zamenjalo okrajno glavarstvo za postojnski okraj, ki je obsegalo 38 notranjskih in vipavskih občin. Sprva je spadala v župnijo Slavina, leta 1793 pa je postala samostojna župnija. Leta 1857 je mimo Postojne zapeljala tudi južna železnica, ki je po eni strani pospešila razvoj, je pa pomenila tudi hud udarec za lokalno prebivalstvo, ki se je takrat preživljalo predvsem s prevozništvom. Leta 1909 je Postojna postala mesto. Po prvi svetovni vojni je bila Postojna priključena Italiji, meja je tekla vzhodno, čez Hrušico, Planino in naprej čez Javornike. Bližina meje je pritegnila tudi vojaško navzočnost, v neposredni bližini so gradili Alpski zid, v Postojni pa je bilo zgrajenih več vojaških in drugih objektov, ki svojemu, ali kakemu drugemu namenu, služijo še danes. Po 2. svetovni vojni se je mesto s stanovanjskim naseljem Kremenca razširilo proti vzhodu in dobilo značilno obliko podkve, katero je leta 1972 povdarila še prvo zgrajena avtocesta v takratni Jugoslaviji, ki danes poteka praktično skozi mesto tik ob železnici in s tem pravzaprav potrjuje pomembno prometno lego Postojne. Nekdanjo obrtno dejavnost, pozvezano z žagami in mlini ob reki Pivki, s prevozništvom in predelavo lesa in bližnjih gozdov, je prerasla v lesno industrijo, razvila se je tudi strojna in živilska industrija. Proti zahodu ob Zalogu, se razvija novi industrijski del Postojne.



V Postojni imajo sedež številne pomembne ustanove, kot so Inštitut za raziskovanje krasa, Višja strokovna šola, Srednješolski center, Gozdarska srednja šola, Notranjski muzej, Upravna enota, Okrajno sodišče, Policijska uprava, 24. vojaškoteritorialno poveljstvo in območna enota Zavoda za gozdove Slovenije, ki upravlja večino gozdov na Javorniki, Hrušici, Nanosu in Snežniku.



*Postojnska jama*  
([www.hotel.krek.si](http://www.hotel.krek.si))

Začetno prometno pobudo k razvoju Postojne, je v dvajsetih letih 19. stoletja prevzela Postojnska jama in turizem je postal eden najpomembnejših dejavnikov razvoja mesta. Vhodni del so obiskovali že v 13. stoletju, leta 1818 pa je domačin Luka Čeč odkril notranje dele jame, kar je omogočilo turistični razvoj. Leta 1883 je bilo ustanovljeno Turistično in olupševalno društvo, leta 1874 pa so zgradili prvi hotel. Postojnska jama ima kot glavna in najstarejša turistična destinacija na Postojnskem osrednjo vlogo v turizmu Postojne in širše okolice. K obisku vsekakor pripomore tudi ugodna prometna lega, saj leži na pol poti do morja mnogim dopustnikom. Skozi čas so poleg Postojnske jame ponudili gostom tudi ogled Predjamskega gradu, ki ga danes dopolnjuje tudi viteški turnir, pa tudi ostale jame, ki so sicer namenjene manj množičnemu turizmu. Ob vходу v Pivko jamo so uredili tudi kamp, ki ga gostje cenijo predvsem zaradi narave ter miru, pa tudi možnosti športnih aktivnosti. V obdobju po osamosvojitvi se je v lokalnem turizmu začutil trend nazadovanja. Značilni slab priokus enodnevnega turizma, oziroma največ ene nočitve obiskovalcev na območju, je nazadovanje le stopnjevalo. V turistično infrastrukturo se ni vlagalo dovolj in ponudba hotelskih sob je postajala nekonkurenčna. Zamrl je hotel Kras v samem središču mesta, ki čaka na podrtje stare stavbe in izgradnjo nove. Vendarle pa se je v zadnjih letih začutilo ponovno zavedanje, da predstavlja turizem za Postojno, že kljub mnogo pretečenim desetletjem, še vedno velik in ne dovolj izkoriščen potencial. Pred kratkim se je po dolгих letih propadanja odprl tudi hotel in hostel Šport, ki poleg tradicionalnih gostov za eno noč, poizkuša zadržati turiste v kraju z pestro športno – rekreativno ponudbo, s poudarkom na kolesarjenju. Tudi novo odprto trgovsko in storitveno središče Epicenter, ki kljub le delni turistični usmerjenosti, prinaša nove trende v ponudbo Postojne, ki pritegne veliko obiskovalcev tudi izven občine. Omeniti je treba tudi poletni festival Zmaj ma' mlade, Furmanski praznik, in druge prireditve, ki bogatijo turistično dogajanje v Postojni.

Gospodarski razvoj se je, kot že omenjeno, pričel z prometom skozi kraje. Ljudje so se ukvarjali z prevozništvom (furmanstvom) in spremljevalnimi dejavnostmi. Prvo »gospodarsko



krizo« je v mesto pripeljala železnica, saj so mnogi poklici propadli in ljudje so se morali preusmeriti v druge dejavnosti. Veliko je bilo tudi gozdarstva, na reki Pivki so delovali mlini in žage. Nek nov gospodarski zagon se je mestu zgodil po 1. svetovni vojni, zaradi umeščenosti neposredno ob jugoslovansko – italijanski meji in zaradi prihoda vojske v mesto. Po drugi vojni so v mestu nastala nekatera večja podjetja, ki delujejo še danes, v kovinski industriji je to Liv, v lesno-gozdarski pa Javor in Gozdno gospodarstvo, nastali pa sta tudi močno trgovsko podjetje Nanos in avtoprevozniško podjetje Transavto, ki pa danes ne obstajata več. Osamosvojitve in zamenjava družbenega sistema je za postojnsko gospodarstvo pomenila manjši šok, saj so se zašibila vsa večja podjetja, pomenila pa je konec prav Nanosa in Transavta. Kovinska in lesno-gozdarska industrija sta se uspeli obdržati, v zadnjih letih pa je zraslo tudi nekaj novih močnih proizvodnih in storitvenih podjetij, ki skrbijo za delovna mesta in gospodarski doprinos.



*Pogled na Postojno iz SZ strani. Dobro je viden Sovič, ki se zajeda v mesto in ga deli, v vznožju je hotelski kompleks pri vhodu v Postojnsko jamo, v daljavi pa kraške planote Javornikov.*

*(<http://www.ntz-nta.si/>)*

Z prvo svetovno vojno je Postojna prvič dobila tudi večji vojaški pomen, ki se je ohranil do danes. V času soške fronte je bilo mesto pomembna zaledna točka, v mestu je bilo celo poveljstvo celotne soške armade z Svetozarjem Borojevičem von Bojno na čelu. Po vojni, ko je območje pripadlo Italiji in je meja potekala neposredno ob mestu, se je povečala navzočnost italijanske vojske. Zaradi obrambe meje, je Italija zgradila več vojašnic, gradila pa je tudi sistem utrd – Alpski zid. Po drugi svetovni vojni je vojaške objekte zasedla jugoslovanska vojska, ki je v okolici zgradila še več objektov, skladišč uporabljali so tudi širše območje na Počku. Po osamosvojitvi se je navzočnost vojske sicer zmanjšalo, a še vedno je v postojnski vojašnici barona Andreja Čehovina (podatek iz 2006) Šola za podčastnike SV, 24. vojaškoteritorialno poveljstvo Postojna, enota vojašnice Postojna, 460. artilerijski bataljon SV, 2. četa 17. bataljona vojaške policije, logistična baza, skupina za pridobivanje kadra in poveljstvo osrednjega vadišča Poček, ki velja za največje vojaško vadbišče v Sloveniji in predstavlja večje breme za okoliško prebivalstvo in okolje.

## 7 ZAKLJUČEK

X3D-zapis se je izkazal kot primeren in uporaben zapis tudi za predstavitve površja in objektov na površju. Še posebej uporaben je prikaz površja s pomočjo podatkov, ki so v Sloveniji na voljo. DOF5 in DMR12,5, katerih datoteke obsegajo isto območje (2250m x 3000m), lahko enostavno vključimo v model v dobri minuti, zapis pa omogoča tudi poljubno stopnjo glajenja predstavljenega DMR. Ker je X3D relativno nov zapis, še nima dovolj podpore s strani aplikacij za 3D-modeliranje. To pomankljivost rešujejo številni prevajalci iz VRML v X3D-zapis, saj je VRML pri modelirnih aplikacijah veliko bolje zastopan. Izdelanemu modelu da posebno težo sposobnost animacije X3D-zapisa, ki naredi model gibljiv ter tako bolj živ in realen.

Da pa ni vse samo rožnato, ugotovimo pri gradnji bolj natančnih in slikovno podprtih modelov stavb. Obdelava tekstur pri uporabi modela in premikanju skozi mesto je s strani obdelave grafičnih podatkov in še vedno zelo zahtevna naloga, ki ji povprečna grafična kartica ni kos. Skozi model z visoko stopnjo realnosti se je težko premikati in usmerjati z običajnimi ukazi predvajalnikov. Tudi zato je nujno govoriti o iskanju kompromisa med kvaliteto in uporabnostjo modela, za boljše manevriranje skozi model bomo morali zmanjšati teksturno natančnost in s tem vizualno realnost modela.

O zgoraj omenjenem kompromisu moramo govoriti tudi pri morebitni spletni uporabi modela. Prikaz v X3D-datoteki namreč zahteva pridobitev primerne aplikacije za prikaz (X3D-predvajalnika) in celotne X3D-datoteke z uporabljenimi teksturami, kar ob bolj obsežnem modelu zahteva več čakanja. Tako objavljen model mora uporabnik že ob najmanjših posodobitvah znova pridobiti v celotnem obsegu. Morebitne prilagoditve so možne z X3D-elementom »script«, ki omogoča podporo jave in nekaterih drugih programskih jezikov, s katerimi bi lahko model bolj prilagodili spletni strani in omogočili hitrejšo uporabo in uporabo usklajeno z elementi spletne strani.

Skozi nalogo smo preizkusili tudi koncept »zastonjskih aplikacij«, s katerimi smo nadomestili praviloma drage komercialne 3D-modelirne aplikacije. Ugotovimo lahko, da se za manj zahtevno 3D-modeliranje izplača uporabiti zastonjske aplikacije. Te so v nekaterih primerih že prav na zavirljivi ravni in omogočajo učinkovito uporabo tudi zahtevnejših algoritmov. Seveda ima prihranek tudi svoje slabosti, a od vsakega posameznega primera je odvisno ali včasih razlika v ceni odtehta manjšo zanesljivost zastonjskih programov. V primeru te naloge, draga programska oprema vsekakor ne bi prišla v poštev.

Na strani tega modela in vseh 3D-spletnih aplikacij je čas. Tehnologija se nadalje razvija in kvaliteta prikaza in uporabnost se bo posledično povečevala. Nadalje lahko pričakujemo tudi razvoj in še večjo uveljavitev X3D-zapisa, najmanj take, kot jo je (bil) deležen starejši brat VRML.

## **VIRI:**

Atlas Slovenije. 2005. Ljubljana, Mladinska knjiga Založba: 491 f.

Corrigan John. 1995. Računalniška grafika. Nova Gorica, Flamingo trade: 304 f.

Forum odprta koda: <http://forum.odprtakoda.org/> (avgust 2007).

Geodetska uprava Republike Slovenije: [www.geodetska-uprava.si/gu/aplik/cepp/GURS\\_izpisiso.jsp?ID={1614DDAF-5216-11D2-BC1C-00A0C9067C11}](http://www.geodetska-uprava.si/gu/aplik/cepp/GURS_izpisiso.jsp?ID={1614DDAF-5216-11D2-BC1C-00A0C9067C11}) in [www.geodetska-uprava.si/gu/podatki/Topograf/Aero/Aero.asp](http://www.geodetska-uprava.si/gu/podatki/Topograf/Aero/Aero.asp) (avgust 2007).

Hoffman Paul. 1996. Vse o internetu & World Wide Webu. Ljubljana, Pasadena: 203 f.

Kosmatin Fras Mojca. 2004. Vpliv kakovosti vhodnih podatkov na kakovost ortofota. Geodetski Vestnik, 48, 2.  
[www.geodetski-vestnik.com/48/2/gv48-2\\_168-178.pdf](http://www.geodetski-vestnik.com/48/2/gv48-2_168-178.pdf) (avgust 2007).

Kovačič Iztok. 2005-2006. Uvod v podatkovni standard XML, študijsko gradivo pri predmetu Avtomatska obdelava podatkov.

Kozierok M. Charles. 2003-2005. The TCP/IP Guide.  
[www.tcpipguide.com/free/t\\_TCPIPApplicationLayerAddressingUniformResourceIdent.htm](http://www.tcpipguide.com/free/t_TCPIPApplicationLayerAddressingUniformResourceIdent.htm) (2.9.2007)

Kvamme Kenneth, Oštir-Sedej Krištof, Stančič Zoran, Šumrada Radoš. 1997. Geografski informacijski sistemi. Ljubljana, Znanstvenoraziskovalni center Slovenske akademije znanosti in umetnosti: 476 f.

Mihevc A., Mihevc B., Frelih M., Polajnar Frelih N., Polak S., Simić M. Zbirka priročnikov »Slovenija total«, Notranjska A-Ž, Pomurska založba: 268 f.

Pesek Igor. 2000. VRML. Maribor.  
[www.pfmb.uni-mb.si/didgradiva/nastopi/didrac2/00/3/index.html](http://www.pfmb.uni-mb.si/didgradiva/nastopi/didrac2/00/3/index.html) (3.9.2007)

Podobnikar Tomaž. 2006. Digitalni model reliefa iz različnih podatkov, Življenje in tehnika, april 2006.  
<http://iaps.zrc-sazu.si/Files/File/Publikacije/digitalni%20model.pdf> (avgust 2007).

Putrle Dejan. Delovanje digitalnega fotoaparata.  
[www.o-ckpiran.kp.edus.si/predmetnaP/biologija/dejan/digifoto.pdf](http://www.o-ckpiran.kp.edus.si/predmetnaP/biologija/dejan/digifoto.pdf) (avgust 2007).

Rihtarčič Mateja, Fras Zmago. 1991. Digitalni model reliefa, 1. del: Teoretične osnove in uporaba DMR. Ljubljana, FAGG: 135 f.

Turk Žiga. 1998. Uvod v Internet, Storitve na internetu, prosojnice predavanj.  
[www.fgg.uni-lj.si/~sdrobne/Pouk/SR/SR.htm](http://www.fgg.uni-lj.si/~sdrobne/Pouk/SR/SR.htm) (avgust 2007).

Vintar Rastislav. Shareware, freeware in public domain.  
[www2.arnes.si/~rzjtopl/slo/sharewar.htm](http://www2.arnes.si/~rzjtopl/slo/sharewar.htm) (avgust 2007).

Web 3D consortium. [www.web3d.org](http://www.web3d.org) (avgust 2007).

Wings 3D. [www.wings3d.com](http://www.wings3d.com) in [http://en.wikipedia.org/wiki/Wings\\_3D](http://en.wikipedia.org/wiki/Wings_3D) (avgust 2007).

X3D - EXTENSIBLE 3D. [http://www.pfmb.uni-mb.si/didgradiva/2005/uc\\_vrml/x3d.htm](http://www.pfmb.uni-mb.si/didgradiva/2005/uc_vrml/x3d.htm)  
(avgust 2007).

Young J. Michael. 2001. XML Korak po korak. Beograd, CET Computer Equipment and Trade: 382 f.



Hotel Kras

Restavracija Kras







Hotel Kras

Restavracija Kras





















