Tekstuurien määritteleminen pinnoille VRML 1.0 -syntaksia käyttäen

Maa-57.285
Fotogrammetrian ja kuvatulkinnan maastoharjoitukset
Korvaava suoritus
Erik Lindberg
38821E

1. Johdanto 4

2. Yleistä 5

2.1 Koordinaatisto 5

2.2 Esimerkkitekstuuri 6

2.3 VRML-koodi 6

3. Suorakulmainen kolmio 8

3.1 Pinnan normaali 9

3.2 Tekstuurikoordinaatit 10

4. Yleinen ratkaisu tekstuurin määrittelemiselle kolmion pinnalle 12

4.1 Ulkoinen tekstuuritiedosto 12

4.2 Tekstuurin määrittelyn normit 13

5. Johtopäätökset 16

Kuvat

Kuva 1: Käytetty tekstuuri. Kuva on keltamusta ja kooltaan 50x50 pikseliä. Vasemman reunan musta viiva on vahvuudeltaan 5 pikseliä, yläviiva 2 pikseliä ja diagonaali 1 pikselin verran. ...6

Kuva 2: Edellisen VRML-koodin tuottama kuva selaimella. Kuvaa tarkastellaan täsmälleen pinnan normaalivektorin suunnasta. Sininen alue on “määrittelemätöntä avaruutta”.....8

Kuva 3: Edellistä kolmiota on pyöritelty selaimella. Varsinkin läheltä tarkasteltaessa tekstuuri on erittäin karkea, mikä johtuu tekstuurikuvan matalasta resoluutiosta. 8

Kuva 4: Tekstuuri näkyy silloin kun pintaa tarkastellaa pinnan normaalivektorin (N) ja pinnan määrittelemästä puoliavaruudesta käsin. Toiselta puolelta pinta on näkymätön, ellei tekstuuria erikseen määritellä. 9

Kuva 5: Tekstuurikoordinaatit vaihtelevat välillä 0–1. Bittikartan vasen alareuna on origo.....10

Kuva 6: Esimerkkitekstuuri, kun ulkoinen tekstuuritiedosto on GIF-formaatissa. Käyttämätön, valkoinen alue pakkautuu hyvin LZW-algoritmilla. 13

Kuva 7: Esimerkkitekstuuri, kun ulkoinen tekstuuritiedosto on JPEG-formaatissa. Kuva jatkuu jonkin verran käytettävän tekstuurin ulkopuolelle, jottei tekstuurin reuna muutu epäselväksi.13

Kuva 8: Kuvalle saadaan oikea kiertosuunta, kun ensimmäiseksi pisteeksi valitaan min y ja toiseksi pisteeksi max x. Kiertosuunta on tällöin vastapäivään. Kuvassa on myös kolmiopinnan sivujen mitat (3D-avaruudessa). 14

Kuva 9: Tekstuurikoordinaatit kuvan 8 määrittelemälle kolmiolle, jos tekstuuritiedosto on neliön muotoinen. 15

Kuva 10: Optimaalisesti valitut tekstuurikoordinaatit. 15

Kuva 11: Optimaalisesti valittujen tekstuurikoordinaattien käyttämä kuvatiedosto. Kuvaa on skaalattu. Optimoinnin ansiosta kuvadatasta käytetään maksimaaliset 50%.....15

1. Johdanto

Tämä harjoitustyö käsittelee tekstuurien määrittelyä pinnoille VRML 1.0-syntaksin mukaisesti. VRML (Virtual Reality Modelling Language) on kolmiulotteisten mallien kuvauskieli, jota käytetään Internetissä. Standardi on avoin.

Kieltä on kehitelty vuodesta 1994 lähtien. Tässä harjoitustyössä käsitellään version 1.0 mukaista syntaksia. Tähän päädyttiin, koska versio 1.0 sisältää tekstuurin määrittelyyn liittyvän syntaksin sekä erityisesti siitä syystä, että tälle standardille löytyy varmimmin tuki kaupallisista ja ei-kaupallisista ohjelmista.

Harjoitustyön aihe liittyy käytännön ongelman ratkaisemiseen. Maanpintaa kuvaava valokuva halutaan tuoda kolmiulotteiseen esitykseen. Valokuva on tässä taso, joka halutaan kiinnittää koordinaateilla oikeaan asentoon määritellyssä XYZ-koordinaatistossa. Koska tehtävä ratkaistaan VRML-kielellä, ratkaisu on käyttöjärjestelmäriippumaton: tavallisimpiin käyttöjärjestelmiin on saatavissa VRML-selain.

2. Yleistä

VRML-kieli on rakenteellinen kieli, joka määrittelee kolmiulotteisia tiloja. Tiloja muodostetaan kappaleprimitiiveistä: pallo (*sphere*), kuutio (*cube*), sylinteri (*cylinder*), kartio (*cone*). Näitä kappaleita täydentävät vapaastimääriteltävät tasot (*IndexedFaceSet*) ja murtoviivat (*IndexedLineSet*) (VRML Specification, 1994). Koska tässä työssä haetaan ratkaisua tekstuurin määrittelylle pinnalle, päädytään seuravaan:

- Tekstuuri määritellään vapaastimääritettävällä tasolle.
- Vapaastimääriteltävä taso muodostuu kolmioista.
- Kolmiotasoilla määritellään kaikki pinnat (niin kaarevat pinnat kuin myös tasot), joihin liitetään tekstuuri.

Kaikki pinnat paitsi tasot määritellään siis kolmioina. Käytännössä tämä tapahtuu muodostamalla kolmiulotteinen TIN (*triangular irregular network*). Kolmioverkon kolmioiden määrä on siten suhteessa lopputuloksen laatuun: mitä enemmän kolmioita pinta-alayksikköä kohden sitä tarkempi lopputulos. Tasopinta voitaisiin määritellä n-kulmiona, jossa kaikkien kulmapisteiden koordinaatit sijaitsevat samalla tasolla. Käytännössä tason jakaminen kolmioverkkoon saattaisi kuitenkin olla selkeämpää varsinkin, jos ohjelmoitava kolmiulotteinen tila sisältäisi sekä kaarevia että tasopintoja.

Tässä harjoitustyössä selvitetään, kuinka tekstuuri määritellään pintaprimitiiville. Ainoaksi primitiiviksi valitaan kolmiopinta. Tekstuuri määritellään 50x50 pikselin kokoisella bittikartalla. Tässä työssä bittikartan koko pidetään vakiona.

Tämän työn esimerkkikuvat on tuotettu *Microsoft Internet Explorer*in *VRML AddIn*-ohjelmalla. Microsoft on lisensioinut WorldView-ohjelmistoon perustuvan VRML-selaimen *InterVista*lta. Valinta tehtiin, koska Netscapen VRML-selaimella esiintyi ongelmia tekstuurien renderoinnissa. Lisäksi Microsoftin tuote toisti värit paremmin.

2.1 Koordinaatisto

VRML-kieli tukee kolmiulotteista suorakulmaista koordinaatistoa. Oletusorientoinnissa koordinaattiakselit osoittavat seuraaviin suuntiin (VRML Specification, 1994):

- X-akseli osoittaa oikealle
- Y-akseli osoittaa ylös
- Z-akseli osoittaa sisään kuvaruutuun.

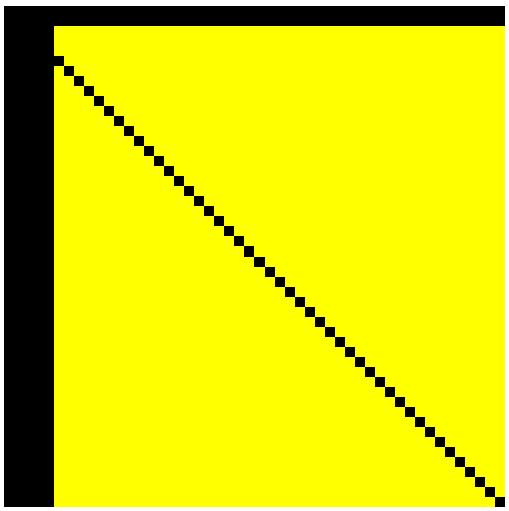
2.2 Esimerkkitekstuuri

VRML-standardi tukee niin ulkoisina tiedostoina kuin myös numerokoodein esitettyjä kuvia (VRML Specification, 1994). Kun kuva liitetään johonkin pintaan, siitä tulee tekstuuri. Ulkoiset kuvatiedostot voivat olla joko GIF- tai JPEG-standardin mukaisia kuvia. Numerokoodein voidaan kuvat sisällyttää VRML-koodiin. Tämä on hyvä keino, jos tekstuuri on hyvin yksinkertainen (esim. shakkiruutu, 4x4 pikseliä). Jos tekstuuri on esimerkiksi valokuvan osa, on ainoa järkevä keino ulkoisen tiedoston käyttäminen.

Ulkoiset kuvatiedostot käyttävät useimmiten jotain kuvantiivistysalgoritmia. Esimerkkikuvan tiivistysalgoritmi on Lempel Ziv Welch -algoritmiin perustuva kuvantiivistysmenetelmä. Se soveltuu hyvin binäärikuviin tai kuviin joissa on alle 256 väriä. Mitä suurempia yhtenäisen värisävyn pinnat ovat sitä paremmin kuva tiivistyy. JPEG-kuvat on tiivistetty diskreettiin kosinimuunnokseen perustuen: se soveltuu paremmin luonnollisiin kuviin, missä siirtymiset kahden värin välillä tapahtuu vähitellen. Se on tehokas pakkausmenetelmä valkouville. (Lindberg, 1997.)

Esimerkkitekstuuri on 50x50 pikselin kokoinen GIF-kuva. Sen tiedostokoko on 252 tavua. Kunkin pikselin värille on varattu 4 bittiä, tiivistämättömänä kuva veisi 1250 tavua tilaa. Pakkaamisella on saatu aikaan merkittävä tilan säästö. Käytännössä tämä tarkoittaa sitä, että kuva latautuu nopeammin.

Tiedon pakkaaminen on erityisen tärkeää Internetissä, missä verkon hitaus tekee suurien tietomäärien siirtämisen tuskallisen hitaaksi (Lindberg, 1997).



Kuva 1: Käytetty tekstuuri. Kuva on keltamusta ja kooltaan 50x50 pikseliä. Vasemman reunan musta viiva on vahvuudeltaan 5 pikseliä, yläviiva 2 pikseliä ja diagonaali 1 pikselin verran.

2.3 VRML-koodi

Tässä harjoitustyössä ei perehdytä VRML-kielen yleiseen syntaksiin. Ainoastaan tehtävälle keskeisimpiä kohtia tarkastellaan. VRML-koodi esitetään tässä harjoitustyössä `erilaisella`

`fontilla`. VRML-koodi ei sisällä rivinumeroita, mutta tässä harjoitustyössä koodin rivit on numeroitu selkeyden vuoksi.

3. Suorakulmainen kolmio

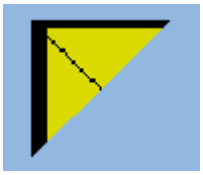
Ensimmäinen tekstuurikokeilu tehtiin suorakulmaiselle kolmiolle, jonka kateetit olivat yhtä pitkiä. Tarkoituksena on kiinnittää puolet esimerkkikuvan tekstuurista kolmion pinnalle.

Tämä on mahdollisimman yksinkertainen tapaus, josta lähdetään liikkeelle.

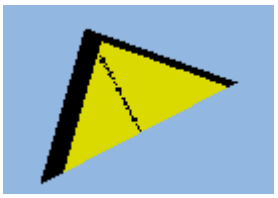
Seuraava VRML-koodi tarvittiin kolmiulotteisen mallin luomiseen:

```
1:   #VRML V1.0 ascii
2:
3:   Separator {
4:     Texture2 {
5:       filename koe.gif
6:     }
7:     TextureCoordinate2 {
8:       point [0 1,0 0,1 1]
9:     }
10:    Coordinate3 {
11:      point [-5 5 0,-5 -5 0, 5 5 0]
12:    }
13:    IndexedFaceSet {coordIndex [1, 2, 0, -1]}
14:   }
```

Listaus 1: Tekstuurin kiinnittäminen suorakulmaiseen kolmioon. Tekstuurista käytetään ainoastaan puolet.



Kuva 2: Edellisen VRML-koodin tuottama kuva selaimella. Kuvaa tarkastellaan täsmälleen pinnan normaalivektorin suunnasta. Sininen alue on “määrittelemätöntä avaruutta”.



Kuva 3: Edellistä kolmiota on pyöritelty selaimella. Varsinkin läheltä tarkasteltaessa tekstuuri on erittäin karkea, mikä johtuu tekstuurikuvan matalasta resoluutiosta.

Ensimmäinen rivi kertoo selaimelle, että tieto on VRML 1.0-kieltä. Ensimmäisen rivin jälkeen on jätettävä tyhjä rivi.

Rivien 4–6 *Texture2*-komento määrittelee tekstuuriksi koe.gif-tiedoston sisältämän kuvan.

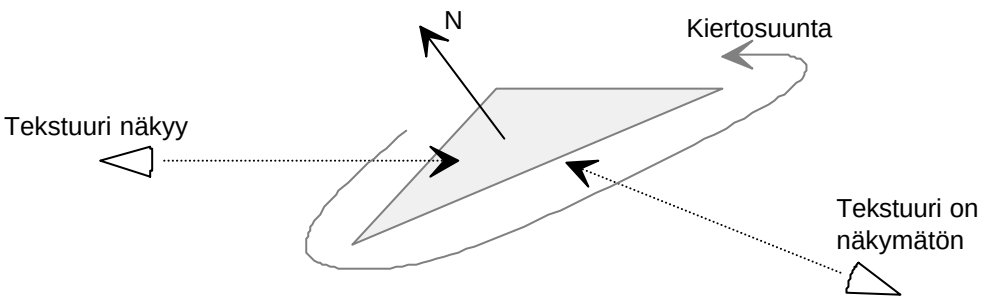
Rivien 7–9 *TextureCoordinate2* määrittelee, kuinka tekstuurin kuvakoordinaatit muunnetaan seuraavan pinnan sijaintikoordinaateiksi.

Rivien 10–12 *Coordinate3* määrittelee tässä esimerkissä kolme XYZ-koordinaattia, jotka määrittelevät kolmion nurkkapisteet. Tämä komento määrittellee ainoastaan koordinaatteja, jotka tallennetaan muistiin. Vasta seuraava komento määrittelee, kuinka pisteet yhdistetään.

Rivin 13 *IndexedFaceSet* määrittelee pinnan, joka käyttää määriteltyä tekstuuria ja määriteltyjä 3D-koordinaatteja. Pinnan XYZ-koordinaatit annetaan määrittelemällä sen reunaviivojen pisteet järjestyksessä. Ne kuvataan järjestysluvulla (ensimmäinen XYZ-koordinaatti saa järjestysluvun 0), joka viittaa *Coordinate3*-komennolla annettuun XYZ-pisteeseen.

3.1 Pinnan normaali

Ensimmäisessä esimerkissä kolmion sivut kierretään vastapäivään. Silloin, oikean käden säännön mukaan, pinnan normaali osoittaa katsojaa kohti. Tässä tapauksessa pinnan normaali osoittaa kuvaruudusta ulospäin, eli se on Z-akselin suuntainen.



Kuva 4: Tekstuuri näkyy silloin kun pintaa tarkastellaa pinnan normaalivektorin (N) ja pinnan määrittelemästä puoliavaruudesta käsin. Toiselta puolelta pinta on näkymätön, ellei tekstuuria erikseen määritellä.

Pinnan normaali on tärkeä tekijä, koska tekstuuri näkyy ainoastaan siihen suuntaan, mihin pinnan normaalivektori osoittaa. Toinen puoli jää näkymättömäksi. Jos sama tekstuuri halutaan kiinnittää molemmille puolille, rivi 13 olisi seuraavanlainen:

```
13:   IndexedFaceSet {coordIndex [1, 2, 0, -1, 1, 0, 2, -1]}
```

Silloin kolmion kärkipisteet kierretään ensin vastapäivään, jonka jälkeen samat pisteet käydään läpi myötäpäivään. Koska pisteet on määritelty samassa komennossa, ne käyttävät samaa tekstuuria. Koordinaatit ovat myös automaattisesti samat, koska kärkipisteiden järjestysnumerot viittaavat samaan *Coordinate3*:n määrittelemään koordinaattitaulukkoon.

Jos kolmion molemmille puolille liitetään sama tekstuuri, ainoastaan kiertosuuntaa vaihtamalla, tekstuurit ovat toistensa peilikuvia. Käytännössä tämä aiheuttaa sen että, esimerkiksi tekstiä

sisältävä tekstuuri on luettavissa ainoastaan katseltaessa kohdetta oikealta puolelta. Väärältä puolelta teksti on peilitekstiä.

Jatkossa tekstuuri määritellään ainoastaan toiselle puolelle. Jotta ikäviltä yllätyksiltä vältyttäisiin, kolmion kärkipisteet kierretään aina vastapäivään. Jos suurempaa kolmioverkkoa määriteltäessä ei pidättäydytä tässä sopimuksessa, lopputulos on “tilkkutäkki”, jossa osa kolmioista on näkymättömiä.

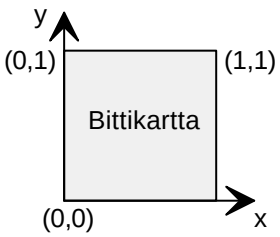
3.2 Tekstuurikoordinaatit

Kuten jo aikaisemmin todettiin, *TextureCoordinate2* määrittelee, kuinka tekstuurin kuvakoordinaatit muunnetaan seuraavan pinnan sijaintikoordinaateiksi.

Koska tekstuuri määritellään kolmiolle, tekstuurikoordinaatteja on määriteltävä kolme. Tekstuurikoordinaatit ovat xy-koordinaattipareja, jotka viittaavat tekstuurina käytettävän kuvan pikselikoordinaatteihin, siten että:

$x \in [0,1] \subset \mathrm{R},$ missä x on bittikartan sarakkeen suhde bittikartan leveyteen

$y \in [0,1] \subset \mathrm{R},$ missä y on bittikartan rivin suhde bittikartan korkeuteen



Kuva 5: Tekstuurikoordinaatit vaihtelevat välillä 0–1. Bittikartan vasen alareuna on origo.

Jos tekstuurina käytettävän bittikartan koko olisi esimerkiksi 640x320 pikseliä, niin tekstuurikoordinaattipari (0.1, 0.3) vittaisi pikselikoordinaattiin (64,96). Tässä harjoitustyössä käytetty selain ei interpoloinut pikseleiden väliin osuneille tekstuurikoordinaateille likiarvoja vaan valitsi lähimmän pikselin arvon. Tämä piirre aiheuttaa rakeisuutta kuvilla, jos tekstuurina käytetty bittikartta on pieni.

Listauksen 1 riveistä 7–12 voidaan todeta, että valitut tekstuurikoordinaatit ja kolmion nurkkapisteiden 3D-koordinaatit muodostavat yhtenevät kolmiot.

```
7:  TextureCoordinate2 {
8:      point [0 1,0 0,1 1]
9:    }
10:   Coordinate3 {
11:     point [-5 5 0,-5 -5 0, 5 5 0]
12:   }
```

Käytännössä tämä tarkoittaa sitä, että tekstuuri saadaan levitettyä kolmion pinnalle käyttämällä ainoastaan skaalausta. Tekstuurista leikatun kolmion kateettien pituus on 1, kolmiulotteisin koordinatein määritellyn kolmion kateettien pituus on 10. Koska kateetit ovat samanpituiset, kuvaan ei tule vääristymää kumpaankaan tekstuurikoordinaatin suunnassa. Tämä on johtava ajatus siinä vaiheessa, kun siirrytään tekstuurin määrittellyyn mielivaltaisesti valitulle kolmiolle.

4. Yleinen ratkaisu tekstuurin määrittelemiselle kolmion pinnalle

Rakennettaessa VRML-maailmoja Internet-käyttöön suurin puute on hitaus. VRML-maailmat kuormittavat helposti verkkoa, ja joustava selaaminen tulee hankalaksi. Kun VRML-mallin rakentaminen painottuu tekstuureihin, kuvadatan käsittely nousee keskeiseksi asiaksi.

Seuraavassa luvussa käsitellään mallin rakentamisvaiheessa huomioonotettavia seikkoja. Luvussa 4.2 käsitellään yleisten normien määrittelyä. Normit voidaan määritellä projektikohtaisesti, mutta niiden järjestelmällinen käyttö ja noudattaminen rakentaa hyvän pohjan siirryttäessä monimutkaisempiin malleihin.

4.1 Ulkoinen tekstuuritiedosto

Tekstuuri määritellään ulkoisella tiedostolla. Ulkoinen tiedosto on aina muodoltaan mxn pikselin matriisi. Käytännössä tämä aiheuttaa sen, että jokaisessa tekstuurissa puolet pikseleistä jää käyttämättä. Koska tämä aiheuttaa turhaa tiedonsiirtoa, käyttämättä jäävä osa tiedostosta pitää saada tiivistettyä mahdollisimman tehokkaasti. GIF-kuvien käyttämä LZW-kompressio tiivistää tehokkaasti suuret yhtenäiset väripinnat, joten se on sopiva käyttötarkoitukseen.

Suurissa tekstuuritiedostoissa JPEG-kuvien käyttämä kompressio on käyttökelpoisempi kuin LZW. Se soveltuu paremmin valokuvien ja luonnollisten kohteiden tiedontiivistämiseen. Jos kuvassa on jyrkkiä reunoja JPEG-kuvien koko kasvaa helposti suureksi. Tämä tulee ottaa huomioon määriteltäessä tekstuuria: tekstuurin raja ei saa olla jyrkkä.

Seuraavat kuvat havainnolistavat, millainen tekstuuritiedoston sisältö on em. Kuvaformaateille erikseen. Punainen viiva on käytettävän tekstuurin raja.

Kuva 6: Esimerkkitekstuuri, kun ulkoinen tekstuuritiedosto on GIF-formaatissa. Käyttämätön, valkoinen alue pakkautuu hyvin LZW-algoritmilla.

Kuva 7: Esimerkkitekstuuri, kun ulkoinen tekstuuritiedosto on JPEG-formaatissa. Kuva jatkuu jonkin verran käytettävän tekstuurin ulkopuolelle, jottei tekstuurin reuna muutu epäselväksi.

4.2 Tekstuurin määrittelyn normit

Tässä luvussa määritellään normit tekstuurin määrittelemiseksi kolmionmuotoiselle tasolle. Normeja voidaan käyttää joko yhdessä VRML-projektissa tai niitä voidaan pitää perustana kaikissa uusissa projekteissa. Tämän harjoitustyön määrittelemät normit ovat vain murto-osa siitä, mitä pitää ottaa huomioon rakennettaessa VRML-maailmaa.

Normit ovat tässä työssä eräänlaisia sääntöjä, joita säännöllisesti noudattamalla vältytään loogisilta ristiriidoilta. Loogisia ristiriitoja tekstuurin osalta ovat tekstuurin vääristymät, joita on kolmea eri tyyppiä (sekä niiden yhdistelmiä):

1. Tekstuuri on väärällä puolella pintaa (= näkymätön katselijan suuntaan)

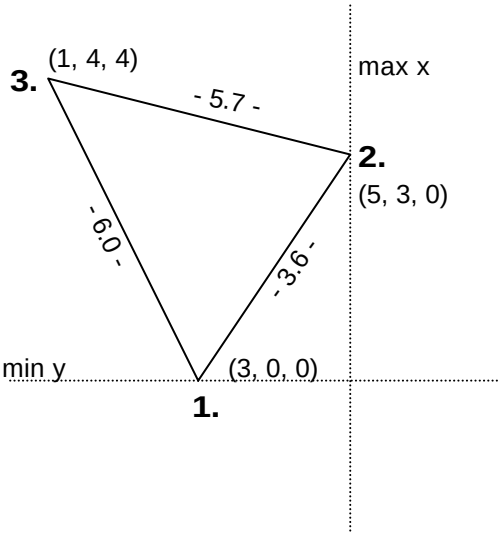
2. Tekstuuri on kiertynyt (=tekstuurin kulmapisteet ovat eri kuin kolmiopinnan vastinpisteet)

3. Tekstuuri on vääristynyt (=tekstuurikoordinaattien muodostama kolmio ei ole yhtenevä kolmiopinnan kanssa)

Ensimmäinen ja toinen ristiriita liittyvät samaan ongelmaan: tekstuurikoordinaatit ja kolmiopinnan koordinaatit eivät vastaa toisiaan. Jotta tämä virhe saataisiin eliminoitua, koordinaattipisteet tulee valita aina seuraavan periaatteen mukaan.

Periaate 1: Kolmiopinnan kiertosuunta on vastapäivään

kolmiopinnan 1. piste on se piste, jonka y-koordinaatti on pienin. Jäljellejäävistä koordinaateista 2. piste on se, jonka x-koordinaatti on suurempi. Tällä menetelmällä pisteiden kierto saadaan oikeaan suuntaan eli vastapäivään.

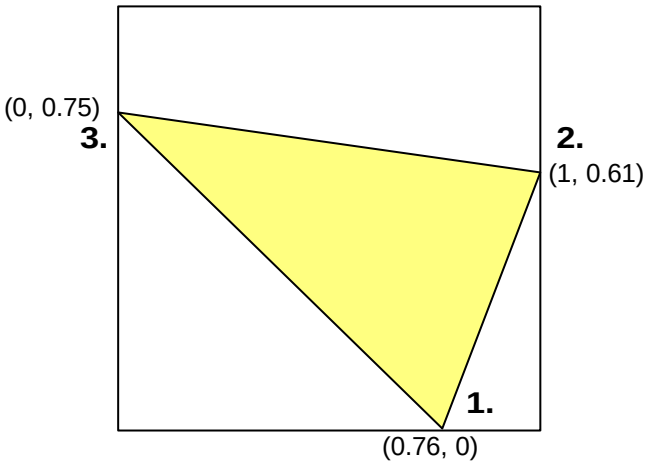


Kuva 8: Kuvalle saadaan oikea kiertosuunta, kun ensimmäiseksi pisteeksi valitaan min y ja toiseksi pisteeksi max x. Kiertosuunta on tällöin vastapäivään. Kuvassa on myös kolmiopinnan sivujen mitat (3D-avaruudessa).

Periaate 2: tekstuurikoordinaattien kiertosuunta on sama kuin kolmiopinnan.

Tekstuurikoordinaatit on valittavat samassa järjestyksessä. Tämä takaa sen, että kierto-suunta säilyy samana (tekstuurin kiinnittämisen väärälle puolelle). Lisäksi tekstuurikolmion pitää olla yhtenevä (eliminoi tekstuurin vääristymät).

Jos tekstuurin sisältämä kuvatiedosto olisi neliönmuotoinen (tekstuurikoordinaatisto olisi yhtenevä pikselien muodostaman koordinaatiston kanssa), määritellyt tekstuurikoordinaatit olisivat seuraavan kuvan mukaiset.

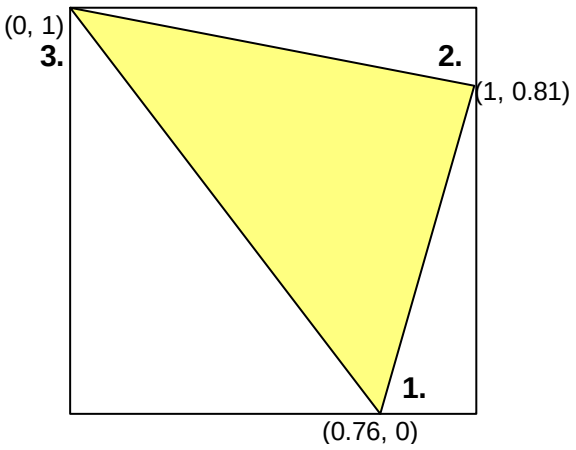


Kuva 9: Tekstuurikoordinaatit kuvan 8 määrittelemälle kolmiolle, jos tekstuuritiedosto on neliön muotoinen.

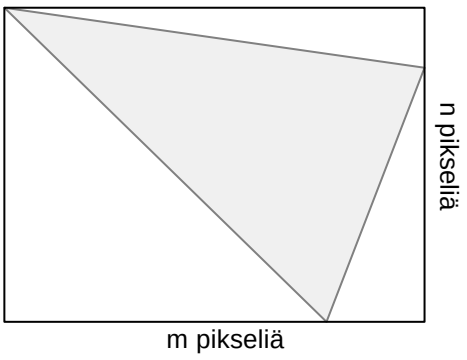
Kuvasta nähdään, että kuvatiedostosta käyttämättä jäävä alue on suuri. Tämä aiheuttaa turhaa tiedonsiirtoa. Tämän välttämiseksi kuvatiedosto ei voi olla neliönmuotoinen.

Periaate 3: Tekstuuritiedoston kuva skaalataan siten, että se sisältää mahdollisimman paljon informaatiota

Jos tekstuuritiedoston resoluutio on m x n pikseliä, niin skaalatun kuvan resoluutioksi valitaan m x k · n, missä k = $((y_{max} - y_{min})/(x_{max} - x_{min}))$. Uudet tekstuurikoordinaatit ovat tämän kuvalle tapahtuvan skaalauksen jälkeen x → x ja y → y/k. Jos k>1, niin tekstuurikoordinaatit muunnetaan seuraavasti: x → x · k ja y → y. Tekstuurikoordinaatit skaalataan siten niin, että ne ulottuvat välille [0..1].



Kuva 10: Optimaalisesti valitut tekstuurikoordinaatit.



Kuva 11: Optimaalisesti valittujen tekstuurikoordinaattien käyttämä kuvatiedosto. Kuvaa on skaalattu. Optimoinnin ansiosta kuvadatasta käytetään maksimaaliset 50%.

5. Johtopäätökset

VRML-kieli on standardoitu. Sen sijaan menetelmät, joilla VRML-maailmoja rakennetaan, eivät ole. Kuinka kolmiulotteiset rakennetaan ja millä periaatteilla, jää ohjelmoijan harteille.

Jotta VRML-maailmojen teko olisi mahdollisimman joustavaa, oppimisen tulee tapahtua kerralla. Tähän auttaa kurinalaisuus: rutiinit suoritetaan joka kerta samoin periaattein. Tämä harjoitustyö pukee sanoiksi sen, mitä pitää miettiä tehtäessä ensimmäisiä VRML-kokeiluja tekstuureille.

Koska harjoitustyö liittyy ainoastaan tekstuurien määrittelemiseen pinnoille, tämä ei ole mikään yleispätevä teos. Tämä harjoitustyö luo normiston vain yhdelle pienelle osalle VRML-ohjelmointia.

Lähteet

Lindberg, 1997.
Lindberg, Erik. WWW-kartat ja kartografia. Erikoistyö. Espoo, 1997.

VRML Specification, 1994.
The Virtual reality Modelling Language. Version 1.0 Specification. 3rd Draft - November 2, 1994.