SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4642

MONTAŽA TEKSTURE NA TRODIMENZIJSKE OBJEKTE

Dean Babić

Zagreb, lipanj 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 17. ožujka 2016.

ZAVRŠNI ZADATAK br. 4642

Pristupnik: **Dean Babić (0036470256)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Montaža teksture na trodimenzijske objekte**

Opis zadatka:

Proučiti postupke preslikavanja teksture na trodimenzijske objekte. Za zadanu sliku teksturu proučiti postupke određivanja uv koordinata u postupku preslikavanja teksture tako da slika teksture prati značajke objekta. Načiniti ocjenu i usporedbu ostvarenih rezultata.
Izraditi odgovarajući programski proizvod. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 18. ožujka 2016.
Rok za predaju rada: 17. lipnja 2016.

Mentor:

Prof. dr. sc. Željka Mihajlović

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za završni rad modula:

Prof. dr. sc. Siniša Srbljić

Sadržaj

1. Uvod

Od samih početaka računalne grafike bilo je potrebno stvoriti realistične i detaljne objekte te pozadine. Kako bi to ostvarili, potrebno je izraditi mrežice (engl. Mesh) i teksture koje se montiraju na objekte definirane pomoću mrežica. Što su mrežice gušće, a teksture veće rezolucije, time su objekti kompleksniji i realističniji. Naglim razvojem računalnih igara, simulatora te virtualne realnosti potrebno je sve brže i točnije stvarati teksture. Isto tako rastu i zahtjevi za poboljšanjem videoigara kako od strane industrije igara, tako i od samih korisnika.

Kako bi doskočili tom problemu osmišljen je novi pristup izradi tekstura, inače težak i spor postupak, koji bi dao rudimentarno rješenje problema uz eventualnu korekciju dobivenog rješenja. Osmišljen postupak kombinira već prethodno definiranu mrežicu zajedno sa slikama tog predmeta, čime se vrlo brzo može dobiti preslikavanje piksela sa slike na objekt definiran mrežicom. No postupak nije ograničen samo na preslikavanje predodređenih slika na predodređeni objekt, moguće su sve kombinacije objekata i tekstura.

Ovaj rad opisat će matematičke osnove u računalnoj grafici, kako se stvaraju 3D teksture te način povezivanja tih dvaju elemenata u novu funkcionalnu cjelinu zvanu montiranje teksture, koja nudi precizno i brzo stvaranje tekstura pomoću slika predmeta te njegove mrežice. Također biti će prikazan i objašnjen rad aplikacije „Obložitelj“, koji implementira željene funkcionalnosti prethodno navedene.

2. Vrste i tehnike 3D teksturiranja

Teksturiranje 3D modela je proces kojim se oživljava model, daje mu se boja i svojstvo. Teksturiranje modela se može raditi na više načina, prema potrebi i mogućnostima. Za dobro teksturiranje poželjno je koristiti dodatne programe za obradu fotografija, kako bi model dobio karakteristike koje su potrebne u skladu s namjenom modela. Teksturne mape omogućavaju da se 3D modeli na sceni učine stvarnijim, posebnim i zanimljivim. Veliku ulogu u postavljanju i prilagođavanju teksture na model imaju i UV točke bez kojih bi bilo teško zamisliti proces teksturiranja. Za većinu karakteristika modela odgovorni su postavljeni materijali na modelu. Boja, prozirnost i refleksija su neki primjeri koji mogu biti promjenjivi s različitim materijalima, a koji služe kao podloga teksturnoj mapi.

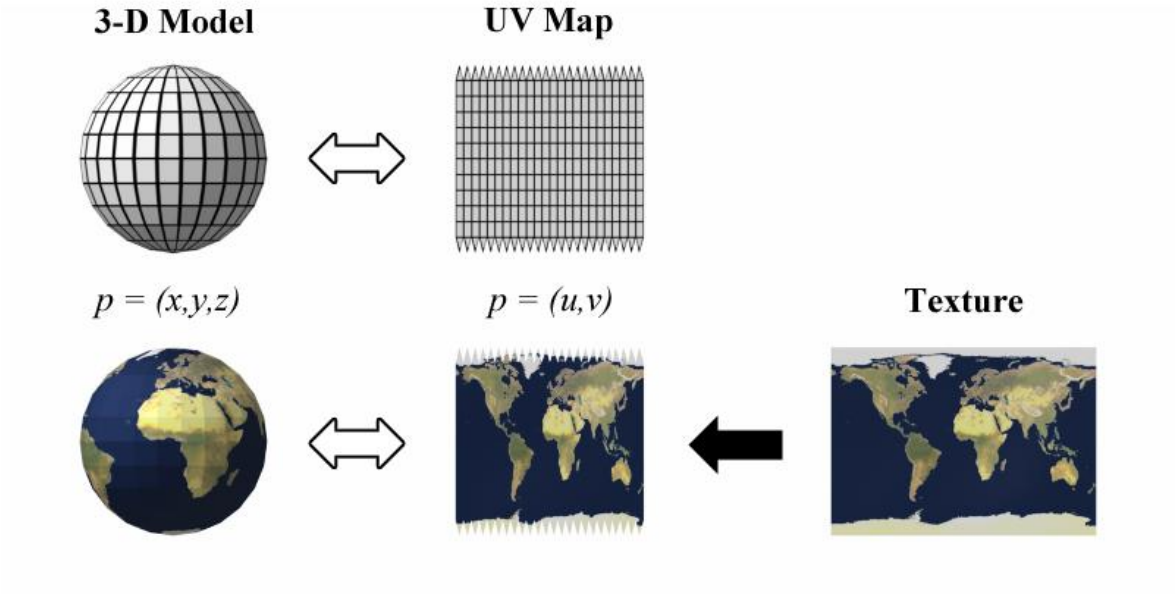
2.1. Materijali

Na samom početku teksturiranja odabire se tip materijala koji se postavlja na model. Nekoliko različitih vrsta materijala koji se nude određuju hoće li model biti mat ili sjajan, proziran ili reflektirajući. Osnovni tipovi materijala su Lambert, Blinn, Anisotropic, Phong i Ramp. Lambert je mat materijal dok je Blinn reflektirajući, odnosno sjajni, Anisotropic je materijal s postepenim prijelazom boja, Phong je sjajni materijal s oštrim osvjetljenjem, Ramp je materijal kojim se kontroliraju prijelazi između više boja i tekstura.

2.2. UV koordinate

UV teksturne koordinate ili UV točke, kako se najčešće zovu, su dvodimenzionalne koordinate koje se razmještaju s komponentama vrhova (engl. Vertex) i dijele informacije za mnogokutne (engl. Polygonal) i podrazredne (engl. Subdivision) plohe mreže. UV točke kontroliraju položaj teksturne mape na 3D modelu tako da razmještaju poziciju 2D teksturne mape prema poziciji vrhova na modelu te se na taj način tekstura pravilno pozicionira. UV točke postoje da bi definirale dvodimenzionalni teksturni koordinatni sustav koji se zove UV teksturni

prostor, a smjerovi u 2D prostoru označavaju se slovima U i V. UV točke su neophodne za povezivanje između mreže ploha modela i slike teksture preslikane na model. UV točke se ponašaju kao oznaka točke koje kontroliraju koje točke na teksturnoj mapi su raspoređene kojom točkom vrha na mreži. Teksture postavljene na mnogokutnu ili podrazrednu plohu, a da ne sadrže UV teksturne koordinate, ne mogu se prikazati.



Slika 1. UV koordinate

2.3. UV preslikavanje

Proces izrade određenih UV točaka za mrežu ploha modela nazivamo UV preslikavanje. UV preslikavanje je proces gdje se izrađuju i uređuju UV točke koje se pojavljuju kao ravne, dvodimenzionalne mreže ploha na dvodimenzionalnoj slici koja se koristi kao tekstura. Prema kompleksnosti i obliku modela, odabire se tip UV preslikavanja i on se primjenjuje na model. UV preslikavanje raspoređuje UV točke pravilno po modelu, a naknadno se UV točke prilagođavaju, spajaju, razdvajaju i premještaju.

UV teksturne koordinate izrađuju se UV tehnikama: Automatsko UV preslikavanje, Ravninsko UV preslikavanje, Cilindrično UV preslikavanje, Sferno UV preslikavanje, UV preslikavanje kamerom. Kod automatskog preslikavanja projicira

se mnogokutna mreža s tri strane modela, te se stvara više UV mreža koje se naknadno spajaju u cjelinu, rotiraju i prilagođavaju teksturi. Ova metoda UV preslikavanja je korisna na kompleksnijim oblicima. Ravninsko preslikavanje projicira UV na mrežu kroz ravnu plohu. Ova projekcija je najbolja za objekte koji su relativno ravni ili su kompletno vidljivi samo iz jednog kuta. Ova projekcija kod kompleksnijih modela stvara preklapanje UV mreže jer ne stvara više komada UV mreža, već su sve UV točke spojene u jednu mrežu. Cilindrično preslikavanje izrađuje UV točke za objekte bazirane za cilindričnu projekciju i mreža se omotava oko objekta. Ova projekcija je najbolja za objekte koji su kompletno zatvoreni i okruglog su, cilindričnog oblika. Kod složenijih oblika dolazi do preklapanja UV točaka, a mreža se sastoji od samo jednog komada. Sferno preslikavanje izrađuje UV mrežu korištenjem projekcije koja je bazirana na kuglastom omotu oko modela. Ova projekcija je najbolja za oblike koji mogu biti cijeli zatvoreni i vidljivi u sklopu okruglog objekta. Preslikavanje kamerom izrađuje UV teksturne koordinate za označeni objekt na trenutačnom pogledu kamere.

2.4. 2D teksture

2D teksture su plošne 2D slike koje se omotavaju oko objekta ili su zalijepljene i drže se plohe na koju su primijenjene. Problem 2D teksture je u usklađivanju i preciznosti namještanja teksture, ako se primijeni više teksturnih slika na jednom modelu. Za što vjerodostojniju teksturu potrebno je usklađivanje s modelom te s obradom u nekom od programa za grafiku. Na objekt se preslikavaju UV koordinate odgovarajućom tehnikom UV preslikavanja, zatim se dodaje materijal na objekt, a materijalu se dodaje željena tekstura. Omogućene su 2D teksture koje se direktno mogu prilagođavati na modelu, a to su predlošci za tkaninu, mreže, tekućine, planine i ostalo.[1]

3. Matematičke osnove u računalnoj grafici

Kako bi mogli shvatiti na koji način preslikavamo sliku na objekt potrebno se je prvo upoznati s notacijom, nekim pojmovima poput točke, vrha, vektora i ravnine te matematičkim operacijama među njima.

Točke se uobičajeno označavaju sa T_X, gdje slovo X predstavlja proizvoljnu točku pravca. Zapis točke T_X po komponentama bit će (T_{X_1} , T_{X_2} , ..., T_{X_n}) pri čemu je T_{X_i} i-ta komponenta točke. Vektori se označavaju slično kao i točke. Vektor pravca bit će označen kao $\vec{v}_p$, odnosno po komponentama $(v_{p_1}, v_{p_n}, \dots, v_{p_n})$. Matrice se označavaju velikim štampanim i podebljanim slovima. Npr. karakteristična matrica pravca nosit će oznaku L.

3.1. Skalarni produkt

Skalarni produkt dvaju n-dimenzijskih vektora $\vec{A} = (a_1, a_2, \dots, a_n)$ i $\vec{B} = (b_1, b_2, \dots, b_n)$ definiran je kao suma umnožaka i-tih parova komponenti obaju vektora. Dakle:

$$\vec{\mathrm{A}} \cdot \vec{\mathrm{B}} = (a_1,\ a_2, \dots, a_n) \cdot (b_1,\ b_2, \dots, b_n) = a_1 \cdot b_1 + a_2 \cdot b_2 + \cdots + a_n \cdot b_n.$$

Od zanimljivijih svojstava skalarnog produkta spomenimo sljedeća dva svojstva. Ako je skalarni produkt dvaju vektora čija je norma različita od nule jednak nula, vektori su međusobno okomiti. Možemo pisati:

$$\vec{A} \cdot \vec{B} = 0 \rightarrow \vec{A} \perp \vec{B}.$$

Skalarni produkt dvaju vektora A i B mjera je kosinusa kuta koji oni zatvaraju. Točnije, vrijedi sljedeće:

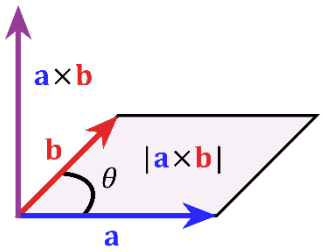
$$\cos(\angle(\mathrm{A}, \mathrm{B})) = \frac{\vec{\mathrm{A}} \cdot \vec{\mathrm{B}}}{\left|\left|\vec{A}\right|\right| \cdot \left|\left|\vec{B}\right|\right|}.$$

Ako je potrebno pronaći bilo koji okomit vektor, postupak prikazan u prethodnom primjeru je sasvim prihvatljiv. Međutim, u praksi se češće umjesto ovakvih "proizvoljnih" ograničenja koristi vektorski produkt opisan u nastavku.

3.2. Vektorski produkt

Vektorski produkt tipično razmatramo u trodimenzijskom prostoru. Vektorski produkt dvaju 3-dimenzijskih vektora $\vec{A} = (a_1, a_2, a_3)$ i $\vec{B} = (b_1, b_2, b_3)$ definiran je kao:

$$\vec{A}\, x\, \vec{B} = \begin{bmatrix} \vec{\imath} & \vec{\jmath} & \vec{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}.$$



Slika 2. Vektorski produkt

što dalje možemo razriješiti kao:

$$\vec{\mathrm{A}} \times \vec{\mathrm{B}} = \vec{\mathrm{i}} \cdot (a_2 \cdot b_3 - a_3 \cdot b_2) - \vec{\mathrm{j}} \cdot (a_1 \cdot b_3 - a_1 \cdot b_1) + \vec{\mathrm{k}} \cdot (a_1 \cdot b_2 - a_2 \cdot b_1).$$

Od zanimljivijih svojstava vektorskog produkta spomenimo sljedeća četiri svojstva. Vektorski produkt $\vec{n}$ dvaju vektora $\vec{A}$ i $\vec{B}$ okomit je i na $\vec{A}$ i na $\vec{B}$ (Slika 2).

$$\left(\vec{A}\ x\ \vec{B}\right) \perp \vec{\mathrm{A}} \quad \wedge \quad (\vec{A}\, x\, \vec{B}) \perp \vec{B}.$$

U ovo se možete uvjeriti tako da pogledate skalarne produkte $\vec{n} \cdot \vec{A}$ i $\vec{n} \cdot \vec{B}$.

Smjer vektora $\vec{n}$ koji predstavlja vektorski produkt dvaju vektora $\vec{A}$ i $\vec{B}$ određen je pravilom desne ruke. Ako prsti desne ruke pokazuju od vektora $\vec{A}$ prema vektoru $\vec{B}$, palac pokazuje smjer vektora $\vec{n}$.

Norma vektorskog produkta $||\vec{n}||$ dvaju vektora $\vec{A}$ i $\vec{B}$ jednaka je:

$$\left|\left|\vec{A}\ x\ \vec{B}\right|\right| = \left|\left|\vec{A}\right|\right| \cdot \left|\left|\vec{B}\right|\right| \cdot \sin(\angle(\vec{A}, \vec{B})).$$

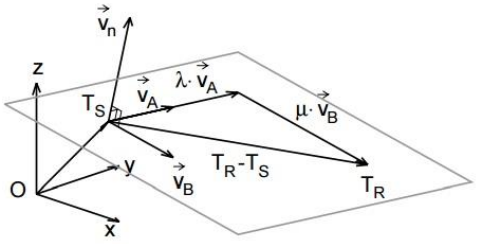
pri čemu se kao kut gleda onaj manji. Geometrijska interpretacija ove činjenice jest da je norma vektorskog produkta vektora $\vec{A}$ i $\vec{B}$ jednaka površini paralelograma što ga razapinju vektori $\vec{A}$ i $\vec{B}$, odnosno jednaka je dvostrukoj vrijednosti površine trokuta koji razapinju ti vektori. Ovu činjenicu kasnije ćemo koristiti na više mjesta, a jedan od primjera će biti i izračun baricentričnih koordinata. Posljedica prethodnog svojstva jest da je vektorski produkt dvaju kolinearnih vektora jednak nul-vektoru. Dakle, ako su vektori paralelni, vektorski produkt ima normu nula.

3.3. Ravnina

Ravnina se kao pojam u općem smislu može definirati u n-dimenzijskom prostoru. Međutim, kako se u računalnoj grafici koriste jedino ravnine u 3D prostoru, razmatranja ćemo ograničiti na taj tip. U 3D prostoru ravnina je određena s dva vektora koji leže u njoj i nisu kolinearni, te jednom točkom ravnine, koja fiksira te vektore. Krenuvši ovakvom definicijom dolazimo vrlo jednostavno do jednadžbe ravnine u parametarskom obliku:

$$T_R = \overrightarrow{v_A} \cdot \lambda + \overrightarrow{v_B} \cdot \mu + T_S.$$

što možemo pročitati i na sljedeći način. Krenuvši od točke T_S svaka točka ravnine može se dobiti pomakom za $\vec{v}_A \cdot \lambda$ i $\vec{v}_B \cdot \mu$, pri čemu su λ i μ realni parametri. Pri tome točka T_R predstavlja bilo koju točku ravnine.



Slika 3. Ravnina i njen vektor

Pretpostavimo da imamo vektor $\overrightarrow{v_n}$ koji je okomit i na vektor $\vec{v}_A$ i na vektor $\vec{v}_B$. Takav vektor zovemo normala ravnine, i ilustriran je na Slika 3. U parametarskoj jednadžbi T_S prebacimo na lijevu stranu:

$$T_R - T_S = \overrightarrow{v_A} \cdot \lambda + \overrightarrow{v_B} \cdot \mu.$$

te razliku točaka na lijevoj strani proglasimo vektorom $\overrightarrow{v_R}$: $\overrightarrow{v_R} = \overrightarrow{v_A} \cdot \lambda + \overrightarrow{v_B} \cdot \mu$.

Pomnožimo sada sve vektorom $\overrightarrow{v_n}$: $\overrightarrow{v_R} \cdot \overrightarrow{v_n} = 0$.

Cijela desna strana jednadžbe je nestala jer su vektori $\overrightarrow{v_A}$ i $\overrightarrow{v_n}$ kao i vektori $\overrightarrow{v_B}$ i $\overrightarrow{v_n}$ međusobno okomiti, pa im je skalarni produkt jednak nuli. Ovo nas vodi na zapis ravnine pomoću njezine normale: $(T_R - T_S) \cdot \overrightarrow{v_n} = 0$.

Vektor $\overrightarrow{v_n}$ naziva se vektorom normale (kraće normala) ravnine iz očitih razloga. Ako znamo vektore $\overrightarrow{v_A}$ i $\overrightarrow{v_B}$, vektor $\overrightarrow{v_n}$ možemo dobiti na različite načine, i taj vektor nije jednoznačan. Naime, postoji beskonačno mnogo vektora koji su okomiti na zadanu ravninu te su svi međusobno kolinearni, a razlika između njih je u njihovoj duljini.[2]

3.4. Matrični račun

Matrični račun temeljni je alat linearne algebre. Prilikom izrade kompleksnih scena, često smo u situaciji da neko tijelo želimo pomaknuti u prostoru, promijeniti mu veličinu ili ga zarotirati oko neke točke. Također nam je potreban zbog pronalaska baricentričnih koordinata. Uporabom matričnog računa i homogenih koordinata svaka se od spomenutih operacija može prikazati kao jedna matrica. Izvođenje takve operacije svodi se na matrično množenje, točku koju želimo pomaknuti u prostoru jednostavno pomnožimo matricom translacije. Iz ovog razloga matrični je prikaz ovih operacija izuzetno pogodan u grafičkim aplikacijama.

Matricu nazivamo m x n matrica ili matrica tipa (redak, stupac) m x n. Ako pišemo A = (a_{ij}) mislimo na cijelu shemu brojeva koji čine matricu A. Ako pišemo a_{ij} mislimo na element koji se nalazi na presjeku i−tog retka i j−tog stupca matrice A. Zbrajanje matrica definira se na slijedeći način:

Neka je $A \in R^{m \times n}$ i $B \in R^{p \times q}$. Ako je m = p i n = q, onda matricu $C \in R^{m \times n}$ s elementima:

$$c_{ij} = a_{ij} + b_{ij}, \qquad 1 \leq i \leq m, \qquad 1 \leq j \leq n$$

nazivamo zbrojem ili sumom matrica A i B i pišemo C = A + B.

Zbrajanje ima u $R^{m \times n}$ ima ova svojstva:

1. A + (B + C) = (A + B) + C (asocijativnost)

2. A + B = B + A (komutativnost)

3. Postoji matrica $0 \in R^{m \times n}$ (neutralni element) sa svojstvom da je A + 0 = A za svaku matricu $A \in R^{m \times n}$. Svi elementi matrice 0 jednaki su nuli.

4. Za svaki $A \in R^{m \times n}$ postoji jedna i samo jedna matrica koju označavamo s −A (inverzni element), takva da vrijedi A + (−A) = 0. Ako je $A = (a_{ij})$, onda je $-A = (-a_{ij})$.

Ako su $a_1, a_2, \ldots, a_n \in R^m$ neki vektori, tada notacija $A = [a_1, a_2, \ldots, a_n]$ znači da je matrica A tako izgrađena da su $a_1, a_2, \ldots, a_n$ njeni stupci u redoslijedu u kojem su napisani. Slično, ako su $a'_1, a'_2, \ldots, a'_m \in R^{1 \times n}$ jednoretčane matrice (vektori redci), tada oznaka:

$$A = \begin{bmatrix} a'_1 \\ \vdots \\ a'_m \end{bmatrix}$$

ukazuje da je $A \in R^{m \times n}$ tako građena da su joj $a'_1, \ldots, a'_m$ redci, u redoslijedu koji je naznačen (vektori stupci).

Neka je $A \in R^{m \times n}$ i $B \in R^{n \times p}$. Umnožak ili produkt matrica A i B je matrica $C = A \cdot B \in R^{m \times p}$ čiji elementi su određeni formulom:

$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} \cdot b_{2j} + \cdots + a_{in} \cdot b_{nj}.$$

Važno je primijetiti da je produkt matrica A i B definiran samo onda kada je broj stupaca matrice A jednak broju redaka matrice B. Operacija množenja matrica · može se gledati kao preslikavanje koje uređenom paru matrica određenih dimenzija pridružuje matricu također određenih dimenzija, prema dijagramu:

$$R^{m \times n} \times R^{n \times p} \rightarrow R^{m \times p}$$

Neka je $A \in R^{n \times n}$ matrica čiji inverz želimo pronaći. Načinimo proširenu matricu

[A | I] primijenimo na nju elementarne retčane transformacije s ciljem da prvi dio matrice, u kojem na početku leži A, svedemo na reducirani oblik A_R. Matrično to možemo opisati pomoću elementarnih matrica:

$$[A_R \mid B] = E_k E_{k-1} \cdot E_2 E_1 [A \mid I].$$

Produkt elementarnih matrica $E_k E_{k-1} \cdot E_2 E_1$ je ne singularna matrica. Označimo ju sa S.

$$[AR \mid B] = S [A \mid I] = S [a_1, a_2, \ldots, a_n \mid e_1, e_2, \ldots, e_n]$$
$$= [S_{a_1}, S_{a_2}, \ldots, S_{a_n} \mid S_{e_1}, S_{e_2}, \ldots, S_{e_n}]$$
$$= [S[a_1, a_2, \ldots, a_n] \mid S[e_1, e_2, \ldots, e_n]] = [SA \mid SI] = [SA \mid S].$$

Izjednačavanjem prvih (zadnjih) n stupaca lijeve i desne strane, dobijemo $A_R = SA$ te također $B = S$. Ako je A regularna, tada je $A_R = I$, pa prva jednadžba daje $I = SA$. Pomnožimo tu jednadžbu s desna s A^{-1} (koja postoji jer je A regularna), dobijemo $A^{-1} = S$. Dakle, ako je A regularna, vrijedit će: $[A_R \mid B] = [I \mid A^{-1}]$.

Drugim riječima, u onom dijelu proširene matrice, na kojem je na početku ležala matrica A, na kraju procesa bit će jedinična matrica I. U drugom dijelu gdje je na početku ležala jedinična matrica I, na kraju procesa bit će inverzna matrica A^{-1}.[3][6]

4. Linearna interpolacija

Postoje dvije vrste linearnih interpolacija, interpolacija između dvije točke te interpolacija između tri točke. Pošto je mrežica građena od poligona odnosno preciznije rečeno trokuta, potrebna nam je linearna interpolacija između tri točke.

4.1. Baricentrične koordinate

Baricentrične koordinate često su korištene u računalnoj grafici pri radu s trokutima. Trokut je dio ravnine omeđen trima točkama koje zovemo vrhovima trokuta. Označimo ih T_x, T_y i T_z. Tri točke koje ne leže na istom pravcu u prostoru određuju ravninu, i u toj ravnini leži trokut $T_xT_yT_z$. Svaku točku ravnine možemo zapisati kao linearnu kombinaciju dvaju nekolinearnih vektora koji leže u toj ravnini, i koja "kreće" iz neke proizvoljne fiksne točke u ravnini.

Neka fiksna točka bude vrh T_x te neka prvi vektor bude onaj između vrhova B i A: $\overrightarrow{v_1}$ = B − A, te neka drugi vektor bude onaj između vrhova C i A: $\overrightarrow{v_2}$ = C − A. Tada se proizvoljna točka T koja leži u toj ravnini može zapisati kao:

$$\vec{T} = \vec{A} + \lambda \cdot (\vec{B} - \vec{A}) + \mu \cdot (\vec{C} - \vec{A}).$$

Krenuvši od ovog izraza, možemo to dalje raspisati i svesti na drugačiji oblik:

$$\vec{T} = \vec{A} + \lambda \cdot \vec{B} - \lambda \cdot \vec{A} + \mu \cdot \vec{C} - \mu \cdot \vec{A} = (1 - \lambda - \mu) \cdot \vec{A} + \lambda \cdot \vec{B} + \mu \cdot \vec{C}.$$

čime dolazimo do konačnog zapisa:

$$\vec{T} = (1 - \lambda - \mu) \cdot \vec{A} + \lambda \cdot \vec{B} + \mu \cdot \vec{C}.$$

Ako parametre uz svaku točku zamijenimo novom oznakom (t_i), dobivamo izraz:

$$\vec{T} = t_1 \cdot \vec{A} + t_2 \cdot \vec{B} + t_3 \cdot \vec{C}.$$

gdje su (t1, t2, t3) baricentrične koordinate točke T. Baricentrične koordinate su drugačiji način za prikaz točaka koje leže u istoj ravnini. Baricentrične koordinate

postoje za svaku točku ravnine koju definiraju tri zadana vrha trokuta. Možemo vidjeti da vrijedi sljedeća jednakost:

$$t_1 + t_2 + t_3 = 1$$

Korištenjem početnih varijabli dobivamo $t_1 = 1 - \lambda - \mu$, $t_2 = \lambda$ i $t_3 = \mu$.

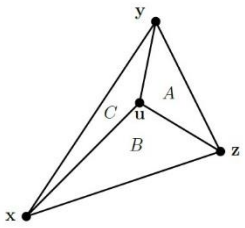
Dokazujemo uvrštavanjem t1 + t2 + t3 = (1 − λ − μ) + λ + μ = 1.

4.2. Linearna interpolacija između tri točke

Neka su T_x, T_y i T_z tri nekolinearne točke te su time vrhovi trokuta T. Točka T_u je težinski prosjek ova tri vrha ako je jednaka:

$$T_u = \alpha \cdot T_x + \beta \cdot T_y + \gamma \cdot T_z.$$

gdje je α + β + γ = 1 tako da su α, β i γ svi ne negativni brojevi. Kao što je prikazano u nastavku, težinski prosjek T_u triju vrhova T_x, T_y, T_z će uvijek biti unutar ili na trokutu T. Varijable α, β i γ zovu se baricentrične koordinate točke T_u.



Slika 4. Baricentrične koordinate

Postoji prikladna karakterizacija baricentričnih koordinata pomoću površina trokuta. Slika 4 prikazuje trokut sa vrhovima T_x, T_y i T_z. Točka T_u dijeli trokut u tri manja trokuta. Površina manjih trokuta je A, B i C, čime je površina cijelog trokuta jednaka zbroju površina manjih trokuta A + B + C. Baricentrične koordinate točke T_u su proporcionalne površinama A, B i C.

$$\alpha = \frac{A}{A+B+C} \qquad \beta = \frac{B}{A+B+C} \qquad \gamma = \frac{C}{A+B+C}$$

4.3. Računanje baricentričnih koordinata

Pretpostavljamo da su točke $T_x = (x_1, x_2)$, $T_y = (y_1, y_2)$, $T_z = (z_1, z_2)$ i $T_u = (u_1, u_2)$ poznate. Tražimo koeficijente α, β i γ koji povezuju točku T_u s ostale tri točke. U prethodnom poglavlju smo rekli da površina paralelograma kojega zatvaraju vektori $\vec{A}$ i $\vec{B}$ je jednaka vektorskom umnošku $\vec{A}$ x $\vec{B}$. Stoga površina trokuta (Slika 4) jednaka je :

$$D = \frac{1}{2}(T_z - T_x)\ x\ (T_y - T_x).$$

Slično tome površina B jednaka je:

$$B = \frac{1}{2}(T_z - T_x)\ x\ (T_u - T_x).$$

Prema formuli odnosa površina dobivamo da:

$$\beta = \frac{(T_z - T_x)\ x\ (T_u - T_x)}{(T_z - T_x)\ x\ (T_y - T_x)}.$$

Slično tome dobivamo:

$$\gamma = \frac{(T_u - T_x)\ x\ (T_y - T_x)}{(T_z - T_x)\ x\ (T_y - T_x)}.$$

Baricentričnu koordinatu α možemo dobiti na isti način ali je brže koristiti jednakost

$$\alpha = 1 - \beta - \gamma$$

Također baricentrične koordinate moguće je dobiti umnoškom vektora točke T_u sa inverzom matrice točaka T_x, T_y, T_z. Zbog optimizacija i jednostavnosti implementacije korišten je upravo ovakav pristup računanju baricentričnih koordinata. [3][5]

$$\begin{bmatrix}\alpha\\ \beta\\ \gamma\end{bmatrix} = \begin{bmatrix}T_{x_x} & T_{y_x} & T_{z_x}\\ T_{x_y} & T_{y_y} & T_{z_y}\\ 1 & 1 & 1\end{bmatrix}^{-1} x \begin{bmatrix}T_{u_x}\\ T_{u_y}\\ 1\end{bmatrix}.$$

5. Metoda montaže

Ideja ovog rada potiče iz već postojećih, ali nedovoljno razvijenih programskih rješenja za kreiranje tekstura iz slika. Preslikavanje imaginarnih ili realnih tekstura na 3D modele uz vrlo malo vizualnih smetnji je težak posao te poradi toga je bilo potrebno pronaći način kako olakšati taj posao, odnosno osigurati visoku preciznost prilikom preslikavanja. Kako su neki postupci montaže odviše kompleksni te nedovoljno brzi, fokusirat ćemo se na montažu uz pomoć projektora (engl. Projector) koji projicira sliku na objekt prema korisnikovom nahođenju.

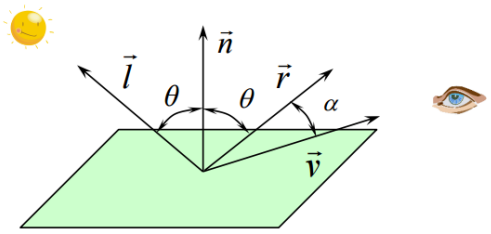
5.1. Montaža uz pomoć projektora

Projektor dozvoljava projiciranje materijala na objekte koji se nalaze u njegovom perspektivnom vidnom polju (engl. Frustum). Kod perspektivne projekcije, perspektivna podjela učinkovito smanjuje daleku geometriju te širi bližu geometriju. Kako bi mogli preslikavati piksele sa slika na točke koje sačinjavaju mrežicu modela, potrebne su informacije o svim točkama, vrhovima, normalama i UV koordinatama.

Za stvaranje objekata u igrama potrebne su konfiguracijske datoteke u našem slučaju to su „.obj“ i „.mtl“ datoteke. Datoteke tipa „.obj“ sadrže informacije o svim definiranim točkama, odnosno njihove x, y i z koordinate (linije koje započinju slovom v), listu UV koordinata (linije koje započinju slovima vt), listu normala vrhova (linije koje započinju slovom vn) te listu polinoma, u našem slučaju trokuta (linije koje započinju slovom f). Kao što vidimo u konfiguracijskoj datoteci nam se nalaze svi potrebni podatci koje smo prethodno naveli što nam veoma pomaže. Prvotno učitamo navedene podatke te predodredimo veličinu generirane teksture. Ta nam veličina treba kako bismo mogli skalirati piksele te da nam izlaz bude što detaljniji. Unutar petlje koja iterira po svim točkama, odabiru se tri po tri točke koje definiraju vrhove trokuta, za svaki vrh potrebno je saznati njegov vektor smjera, normalu, te UV koordinatu. Nakon što smo to saznali određujemo granice trokuta, odnosno njegove minimalne i maksimalne koordinate u kojima se nalazi, time ćemo moći lako obojati trokut tako da se krećemo od minimalnih prema maksimalnim granicama i bojimo samo one točke koje se nalaze unutar ili na trokutu. Da li je točka T unutar

trokuta utvrđujemo uvrštavanjem koordinata točke u jednadžbu vektora te ako je vrijednost manja ili jednaka nuli za sve vektore tada možemo reći da je točka unutar trokuta, ali je potrebno imati sve vektore postavljene u smjeru kazaljke na satu, kako bi vrijednost bila ispravna. Da bi mogli obojati točku T trebamo saznati koju joj boju moramo dodijeliti, to postižemo izračunavanjem baricentričnih koordinata te koeficijenata doprinosa svakog projektora. Matricu M moramo ispuniti x i y koordinatama svih vrhova trokuta te ju zatim invertirati i pomnožiti sa koordinatama točke, nakon čega dobivamo baricentrične koordinate, koje množimo sa svakim vrhom trokuta kako bi dobili odnosni položaj te množimo sa normalom svakog vrha kako bi dobili normalu u toj točki. Nakon toga moramo pronaći matricu pretvorbe iz lokalnog koordinatnog sustava u globalni te ju množimo sa normalom u točki T, koju zatim normaliziramo.

U ovisnosti broja projektora, kojih ima koliko je i slika učitano, određujemo koeficijent kojim on utječe na točku T, određujemo ga tako da podijelimo broj jedan sa brojem projektora. Sada tek možemo odrediti boju, iteriramo po svim projektorima i gledamo pod kojim kutom upadaju zrake svjetlosti na točku T u odnosu na njenu normalu te gdje se nalazi kamera.



Slika 5. Vektori normale, upadne i reflektirane zrake

Ako je skalarni umnožak vektora normale pripadnog poligona i vektora prema promatraču veći od nula, tada kažemo da je poligon prednji, ako je manji od nula, tada je poligon stražnji, inače je degenerirao u liniju. Aplikacija „Obložitelj“ boji samo one točke koje su prednje i koje se nalaze unutar perspektivnog vidnog polja pojedinog projektora, sve ostale točke boje se crno.

6. Implementacija metode montaže

6.1. Grafički pokretač – Unity

Unity je veoma popularan programski paket za razvoj igara i 3D simulacija. Izdan 2005. godine, omogućio je nagli porast broja ljudi i razvojnih timova koji žele kreirati igru ili neku drugu vrstu grafičkog programa. Dobre karakteristike Unity-a su to što je veoma jednostavan i efektivan te ima detaljno opisanu dokumentaciju i veliku podršku na forumu, poradi toga većina grešaka i teškoća u razvoju mogu se otkloniti uz pomoć foruma. Unity je baziran na programskom jeziku C/C++, a za skriptiranje se koriste jezici JavaScript ili C#. Poradi svih prednosti Unity alata, ovaj rad je implementiran u njemu.

6.2. Opis programa „Obložitelj“

„Obložitelj“ je aplikacija za montažu tekstura na trodimenzijske objekte pomoću fotografija i mrežice modela. Primarna zadaća ove aplikacije jest olakšati i ubrzati postupak izrade tekstura. Korisnik ima mogućnost učitavanja bilo koje vrste slika neovisno o modelu te uz fino ugađanje može jednostavno postići željene rezultate. Po zadovoljavajućem rezultatu može pospremiti teksturu te može učitati bilo koju teksturu na izabrani model, tekstura će se aplicirati iako nije možda namijenjena za taj model.

6.3. Opis razvoja

Prilikom izrade aplikacije stvoren je novi projekt u programu Unity. Prvotno je trebalo osmisliti grafičko korisničko sučelje koje će voditi korisnika kroz proces stvaranja teksture. Veoma je bitno imati što jednostavnije, a kvalitetnije grafičko sučelje kako bi korisniku bilo lakše koristiti aplikaciju, također uz smisleno vođenje kroz proces izrade potrebno je imati robustan sustav provjere uvjeta te paziti što korisnik unutar aplikacije radi, odnosno ne dozvoliti korisniku da učitava neispravne datoteke ili da se dovede u nekonzistentno stanje. Da bi se mogla implementirati

sama jezgra aplikacije, montiranje teksture na model, bilo je potrebno detaljno proučiti teoriju u pozadini kako bi se što prije došlo do rezultata.

```
bari = M.inverse * a;
                Vector3 textelPos3 = bari.x * p0 + bari.y * p1 + bari.z * p2;
                Vector3 textelNormal = bari.x * n0 + bari.y * n1 + bari.z * n2;
                Vector4 textelPos4 = new Vector4(textelPos3.x, textelPos3.y, textelPos3.z, 1.0f);
                Matrix4x4 W = target.transform.localToWorldMatrix;
                textelNormal = W * textelNormal;
                textelNormal.Normalize();
```

Slika 6. Dio koda koji preslikava koordinate

Iako je bilo dosta grešaka za vrijeme razvoja aplikacije, Unity forum je puno pomogao te je aplikacija bila relativno brzo gotova.

6.4. Rezultati

Prilikom ulaska u aplikaciju korisnik se susreće sa ulaznim zaslonom, na kojemu može pronaći osnovne informacije vezane uz aplikaciju, klikom na započni, korisnik odlazi na zaslon gdje ima izbor želi li učitati postojeću teksturu ili želi li stvoriti novu iz slika.

Slika 7. Zaslon za odabir

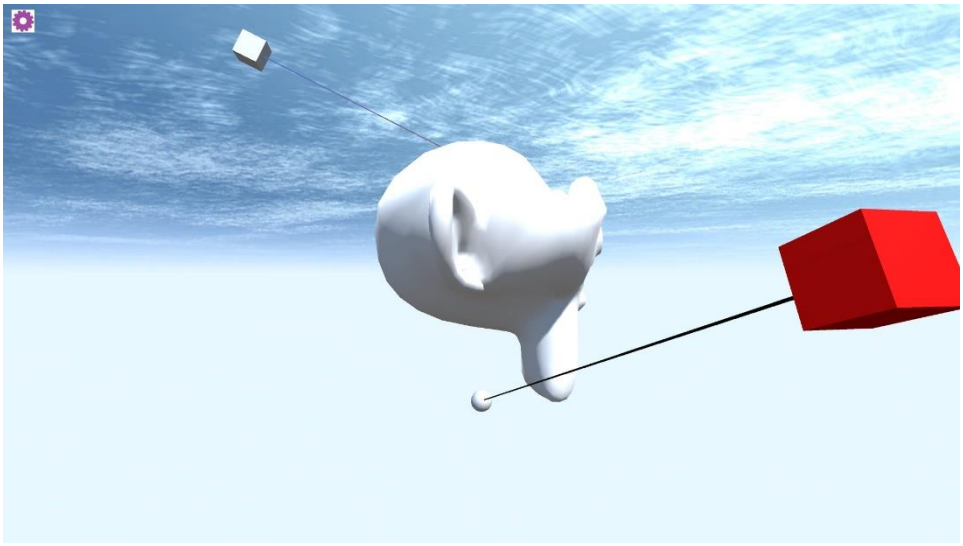
Korisnik u svakom slučaju mora učitati „.obj“ dokument, jer bez njega nije moguće ništa prikazati, uz „.obj“ dokument potreban je i standardni „.mtl“ dokument kojega će program sam učitati u ovisnosti što piše u „.obj“ dokumentu. Nakon toga korisnik učitava jednu ili više slika ili teksturu, ne može učitati oboje. Klikom na nastavi korisnik odlazi na glavni zaslon gdje će mu se odabrani model učitati i automatski skalirati da ne bude niti prevelik niti premalen.

Slika 8. Prethodno generirana tekstura majmuna

Korisnik se može kretati oko predmeta kada klikne i potegne mišem duž prozora, kamera se rotira u onom smjeru u kojem se mišem poteže. Također moguće je približavati se i udaljavati pomoću kotačića na mišu te kretati gore i dolje pomoću W i S.

Kada se odabere stvaranje nove teksture tako što se u prethodnom zaslonu učitaju slike, ostvareno je podešavanje položaja tijela projektora (kockica) te mete projektora (kuglica) u ovisnosti na model. Komponente projektora moguće je pomicati tako da se prvo klikne na njih te onda pomiče tipkama 2, 4, 6, 8, + i -.

Kada se odabere tijelo ili meta, oni zasvijetle crveno kako bi korisnik znao što uređuje, a parovi su međusobno povezani linijama.



Slika 9. Podešavanje položaja projektora

Slika 10. Zanimljiva kreacija majmuna s licem tigra

7. Zaključak

Stvaranje tekstura iz slika nije trivijalan zadatak, ali zato vidjeti jednu vrstu uspjeha je dobar osjećaj i korak u pravom smjeru. Vidim veliki potencijal za ovu vrstu izrade modela, jer unosi jednu komponentu realnosti i detaljnosti u sam doživljaj virtualnog svijeta koji nije moguć konvencionalnim načinima. Unatoč tome što bi takav pristup izrade modela bio uvelike bolji od dosadašnjeg pristupa, ručnog crtanja svakog detalja, još uvijek nije dovoljno zastupljen niti se u njega ulaže dovoljno resursa. Ovaj rad može poslužiti kao dobar temelj za daljnji razvoj tehnologije te kao pregled što je dobro, a što nije. Fokus ovog rada je bio na pitanju može li se ostvariti rješenje, no ne i postoje li bolji algoritmi, zbog toga smatram da ima puno mjesta za napredak. Također velika je prednost što je rad implementiran u Unity razvojnoj okolini, jer ona nudi fleksibilnu i pristupačnu okolinu za nadogradnju, kako se ne bi trebalo opterećivati sa pojedinostima programskog jezika.

Literatura

[1] Bernik, A., Kelnari, D. Vrste i tehnike 3D teksturiranja. Veleučilište u Varaždinu, Varaždin, 2011.

[2] Čupić, M., Mihajlović, Ž. Interaktivna računalna grafika kroz primjere u OpenGL-u. Sveučilište u Zagrebu, Zagreb, 2016.

[3] Hari, V. Linearna Algebra. Sveučilište u Zagrebu, Zagreb, 2005.

[4] Zhou, K., Wang, X., Tong Y., Desbrun, M., Guo, B., Shum H. Y. TextureMontage:

Seamless Texturing of Arbitrary Surfaces from Multiple Images. Caltech, Pasadena, 2005.

[5] Buss, S. R. 3-D Computer Graphics: A Mathematical Introduction with OpenGL. University of California, San Diego: Cambridge University Press, 2003

[6] Vince, J. Calculus for Computer Graphics. Bournemouth University, Bournemouth: Springer, 2013.

[7] Unity Manual, http://docs.unity3d.com/Manual/index.html, 10.6.2016.

Sažetak

Naslov: Montaža teksture na trodimenzijske objekte

Ovaj rad obrađuje temu stvaranja tekstura pomoću fotografija i mrežice modela. Prvi dio sadrži pregled trenutnog postupka izrade tekstura te što je sve za to potrebno. Dan je pregled matematičkih osnova u računalnoj grafici kako bi se kasnije lakše razumjelo na koji način se dobiva tekstura iz fotografija. Rad opisuje postupak linearne interpolacije, sjenčanja, preslikavanja iz lokalnog u globalni koordinatni sustav te rad s projektorima. U sklopu rada razvijena je aplikacija za izradu tekstura. Drugi dio sadrži opis razvoja i način korištenja aplikacije pomoću Unity grafičkog pokretača.

Ključne riječi: 3D model, teksturiranje, Unity, UV koordinate, projektori, montaža

Abstract

Title: Texture montage onto 3D models

The theme of this thesis is texture montage using images and model meshes. First part explains how are textures created nowadays and what is needed in order to make one. In order for you to better understand how the texture is created from images there is also an introduction to basic mathematics for computer graphics. This thesis describes linear interpolation, shading, local to global world mapping and projector handling. An application was developed as a part of this thesis. Second part contains development process and also instructions on how to use the application, which is based upon Unity.

Keywords: 3D model, texturing, Unity, UV coordinates, projectors, montage