

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1382

PRIKAZ POVRŠINSKIH NERAVNINA

Karlo Jež

Zagreb, lipanj 2010.

Sadržaj

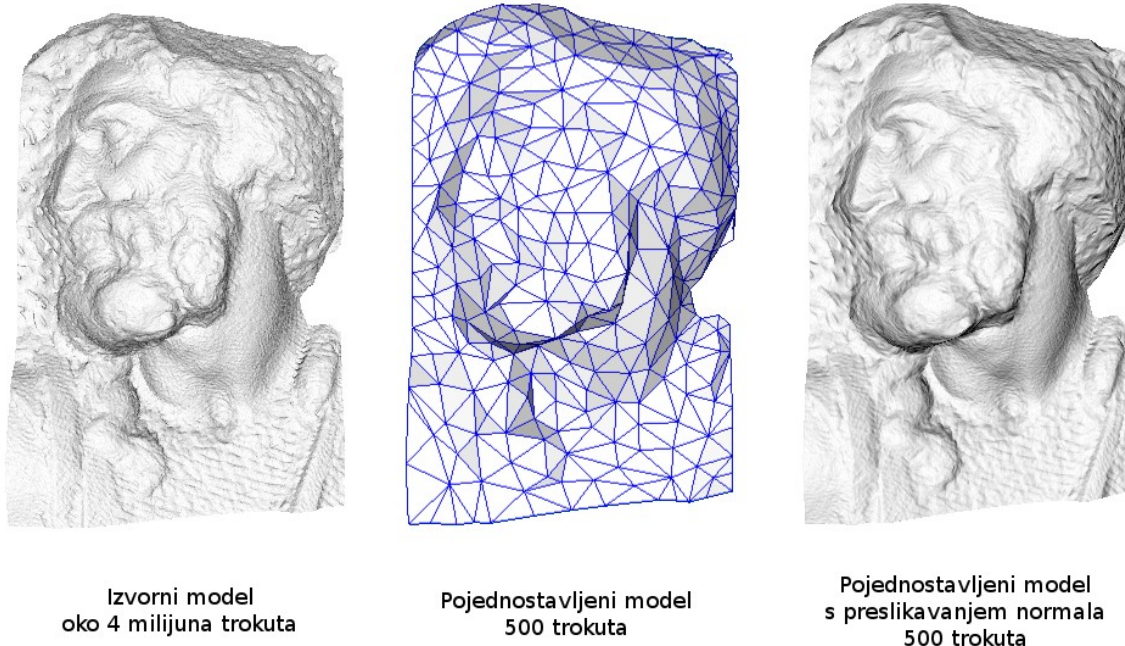
Uvod.....	1
Preslikavanje neravnina.....	2
Phongov model osvjetljenja.....	3
Phongovo sjenčanje.....	5
Preslikavanje normala.....	6
Preslikavanje paralakse.....	11
Praktični dio.....	13
Zaključak.....	15
Literatura.....	16
Sažetak.....	17
Summary.....	18

Uvod

Jedan od glavnih problema prikaza složenijih trodimenzionalnih objekata na računalu jest količina detalja koju treba prikazati. Ako se svaka neravnina, bora i izbočina modelira kao dio samog objekta, njegova složenost, vrijeme potrebno za izradu i prikaz modela eksponencijalno rastu. Ako se pak ti detalji izostave, objekti izgledaju umjetno, neprirodno i plastično. Rješenje tog problema su razne metode simuliranja neravnih površina čije korištenje je ekonomičnije za prikaz i izradu. Rad na području metoda prikaza neravnina počeo je 70-ih godina prošlog stoljeća te su takve metode bile korištene za prikaz unaprijed ostvarenih prikaza, ali je do pravog razvoja došlo početkom 21. stoljeća kada su komercijalno dostupne grafičke kartice počele podržavati programabilne sklopove za sjenčanje te omogućile prikaz neravnih površina u realnom vremenu. Počeo je nagli razvoj novih metoda prikaza neravnina u želji da se neravnine prikažu čim vjernije. Moderne računalne igre intenzivno koriste te metode i na taj način tjeraju proizvođače sklopovske opreme da snagom i mogućnostima grafičkih kartica prate potrebu za sve realnijim metodama prikaza objekata u prostoru.

Preslikavanje neravnina

Preslikavanje neravnina je tehnika prikaza predmeta kod koje se za svaki slikovni element površine objekta provjerava odstupanje od normale površine, te se koristi za izračun osvjetljenja objekta. Preslikavanje neravnina se koristi kao proširenje Phongovog sjenčanja, a odstupanje od normale dobivene Phongovom interpolacijom se računa pomoću slike koja opisuje neravnine objekta. Metode prikaza neravnih površina omogućuju smanjivanje broja poligona objekta bez većeg gubitka detalja. (slika 1.)



Slika 1: Usporedba objekta načinjenog velikom brojem trokuta i objekta gdje je površina ostvarena tehnikom preslikavanja normala.

Takvo preslikavanje se još naziva i lažno preslikavanje neravnina jer se ne iskrivljuje sama geometrija objekta, nego samo normale na površinu. Razlika između lažnog i stvarnog preslikavanja se vidi kod malih kutova prikaza; primjerice, silueta objekta kojem su neravnine dobivene lažnim preslikavanjem izgleda potpuno glatko, dok će ona dobivena stvarnim preslikavanjem biti neravna. Glavna prednost lažnog preslikavanja neravnina je učinkovitost i brzina, grafički procesori mogu normale računati u realnom vremenu, dok se stvarno preslikavanje koristi za unaprijed ostvarene prikaze. Postoji više vrsta preslikavanja neravnina, koje se razlikuju po vjernosti prikaza, vektorima koje uzimaju u obzir te brzini i učinkovitosti izvođenja.

Phongov model osvjetljenja

Osvjetljenje objekta čine tri komponente: ambijentna, difuzna i zrcalna.

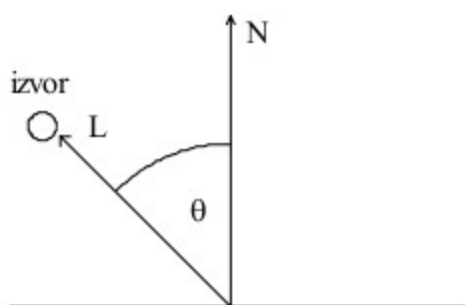
Ambijentna komponenta predstavlja reflektiranje svih poligona okolnog prostora i aproksimira globalno osvjetljenje. Ukoliko nije prisutna, poligoni koji su stražnji s obzirom na izvor svjetla će biti crni.

$$I = K_a I_a$$

K_a – koeficijent reflektirane ambijentne svjetlosti, ovisi o materijalu

I_a – intenzitet ambijentne svjetlosti

Difuzna komponenta se temelji na Lambertovom zakonu. Količina svjetlosti koju vidi promatrač neovisna je o položaju promatrača. Proporcionalna je kosinusu kuta između normale na površinu i vektora prema izvoru.



$$I_d = k_d I_p \cos(\theta)$$

K_d – koeficijent difuzne refleksije

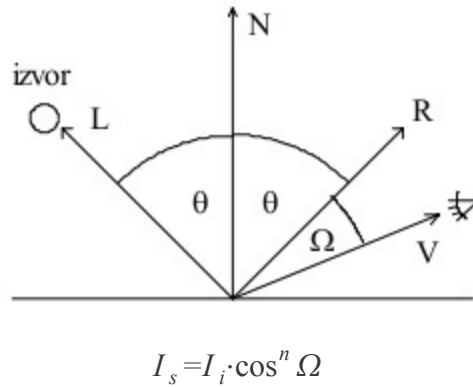
I_p – intenzitet točkastog izvora

θ – kut između normale na površinu i vektora prema izvoru

Još se u obzir uzima udaljenost izvora od površine d_l te se dobiva formula:

$$I_d = K_a I_a + K_d I_p \frac{\vec{l} \cdot \vec{n}}{d_l^2}$$

Zrcalna komponenta predstavlja blještavilo i obojena je bojom upadne svjetlosti. Zrcalnom komponentom određujemo reflektivnost materijala od kojeg je objekt sastavljen.



Vektor \mathbf{R} predstavlja vektor reflektirane zrake i nalazi se u ravnini koju tvore vektori \mathbf{L} i \mathbf{N} , od istom kutom što ga zatvara vektor \mathbf{L} s vektorom \mathbf{N} . Vektor \mathbf{V} je vektor iz promatrane točke prema promatraču. Vektori \mathbf{R} i \mathbf{V} su jedinični, što znači da se kut Ω može opisati skalarnim produktom $\mathbf{R}\mathbf{V}$, pa formula prelazi u:

$$I_s = I_i \cdot k_s \cdot (\vec{R} \cdot \vec{V})^n$$

Ukupan utjecaj iskazuje se kao linearna kombinacija svih triju komponenata.

$$I = I_a \cdot k_a + I_i \cdot \left(k_d \cdot (\vec{L} \cdot \vec{N}) + k_s \cdot (\vec{R} \cdot \vec{V})^n \right)$$

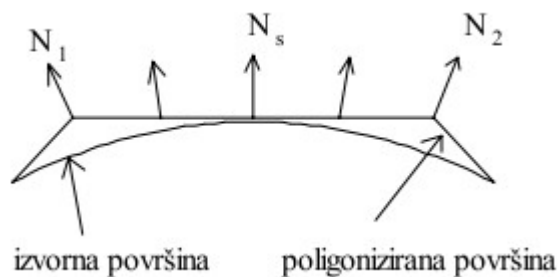
Difuzna i reflektirajuća komponenta modela mogu se nadopuniti s oblicima koji u obzir uzimaju udaljenost.

Osobitosti Phongovog modela osvjetljenja:

- Svjetlosni izvori su točkasti.
- Svjetlosni izvori i promatrač nalaze se u beskonačnosti.
- Difuzna i reflektirajuća komponenta modeliraju se lokalno.
- Koristi se empirijski model simulacije rasipanja svjetlosti oko refleksijskog vektora čime se modelira blještavilo.
- Za boju rasipane svjetlosti uzima se boja izvora.
- Ambijent je modeliran konstantom

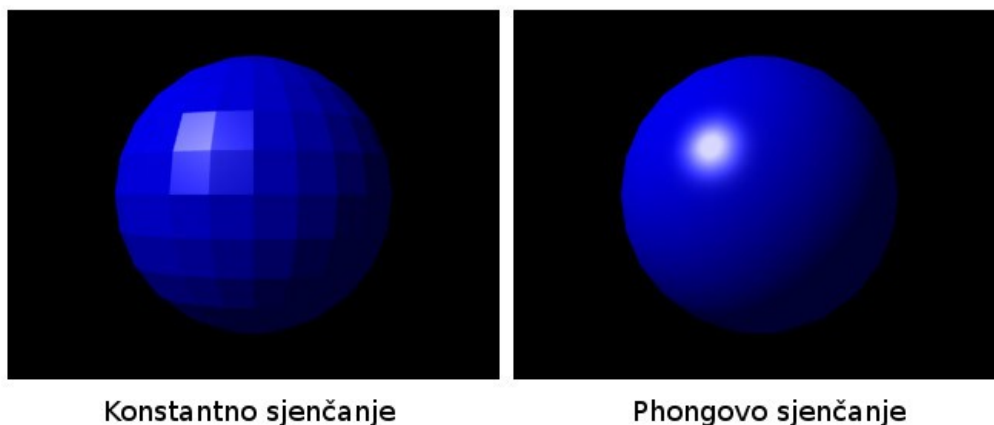
Phongovo sjenčanje

Sjenčanje se zasniva na Phongovom refleksijskom modelu. Temelji se na bilinearnoj interpolaciji normala te se time stvara privid da normala slijedi željenu zakrivljenost površine iako se radi o ravnom poligonu.



Slika 2: Normale se dobivaju linearnom interpolacijom normala na vrhovima.

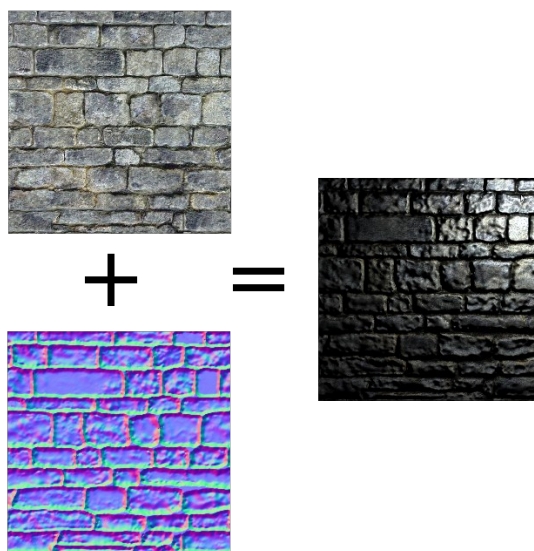
Prvi korak postupka je računanje normala vrhova, nakon toga se izvodi bilinearna interpolacija normala. Kod bilinearne interpolacije prvo se interpoliraju normale duž bridova, a potom se dobivene normale na bridovima interpoliraju duž ispitnih linija. Na kraju se za svaku točku računa intenzitet svjetlosti.



Slika 3: Usporedba konstantnog i Phongovog sjenčanja

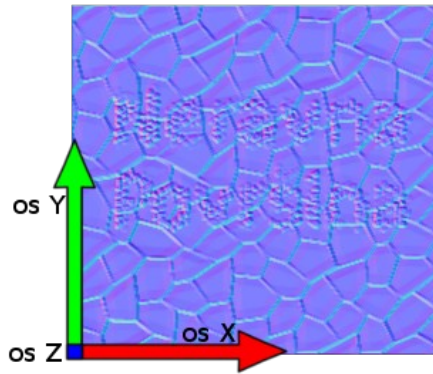
Preslikavanje normala

Tehnika preslikavanja normala je jedna od najkorištenijih metoda prikaza neravnih površina u realnom vremenu. Temelji se na Lambertovom zakonu i koristi difuznu komponentu osvjetljenja, a može koristiti i zrcalnu komponentu. Preslikavanje normala ne uzima u obzir međusobni položaj promatrača i objekta, nego ovisi samo o međusobnom položaju objekta i izvora svjetlosti. Zbog toga je glavni nedostatak metode neprirodan izgled neravnina pod manjim kutovima, osobito kad se promatrač nalazi blizu objekta. Ovaj problem rješava metoda preslikavanja paralakse.



Slika 4: Kombinacija teksture i mape normala omogućuje privid neravne površine

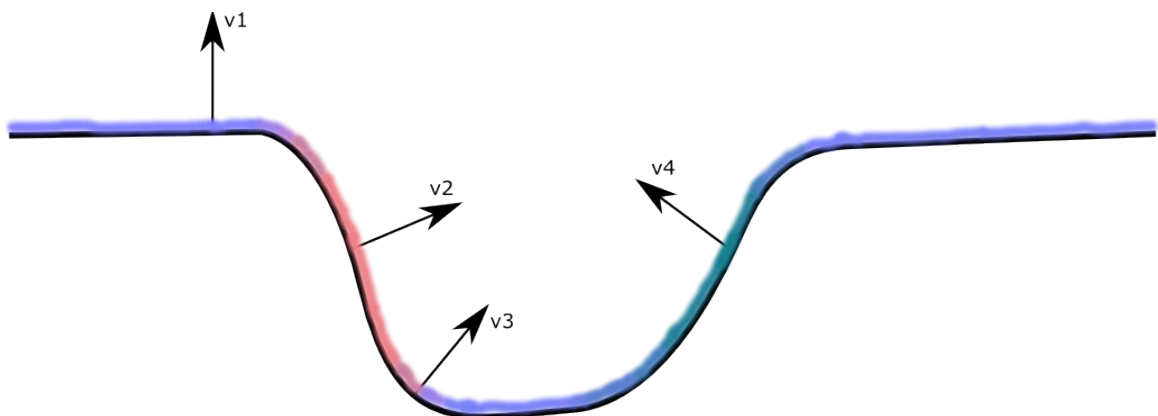
Mapa normala je najčešće 24-bitna RGB slika, pri čemu svaki kanal (boja) odgovara jednoj koordinati trodimenzionalnog koordinatnog sustava. Crvena označava x os, zelena y, a plava boja z os. Svaki slikovni element mape ima određenu količinu svake boje, koja određuje vektor normale tog elementa. Kod 24-bitne RGB slike, svaka elementarna boja određena je nizom od 8 bita. U taj niz moguće je spremiti 256 razina nijansi boje. Budući da vektori mogu po osima imati i pozitivne i negativne vrijednosti, a vrijednosti u nizu su u rasponu od 0 do 255, vrijednost niza 128 se prihvaća kao 0 za vrijednost vektora. Pozitivne vrijednosti osi vektora su od 128 do 255, a negativne od 127 do 0.



Slika 5: Boje kao koordinate na mapi normala

Kao što se vidi na slici, kod većine slikovnih elemenata prevladava plava boja. Razlog tome je smjer plavog, tj. Z, vektora, koji je okomit na površinu tekstone. Ako bi vrijednost plave boje bila manja od polovice maksimalne vrijednosti, normala elementa bila bi okrenuta prema unutrašnjosti površine, što bi stvorilo vizualne artefakte.

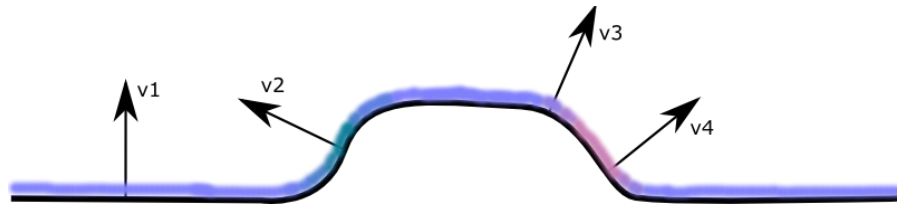
Prikaz izbočina i udubina je na ovaj način vrlo jednostavan, izbočine su s gornje i desne strane izrazito zelene i crvene, a s donje i desne strane su vrijednosti zelene i crvene komponente manje od polovine. Udubine se prikazuju smanjenim vrijednostima boja s gornje i desne strane udubine, a povećanim s donje i lijeve.



Slika 6: Prikaz udubine s bojama pripadajuće mape normala

Na slici se vidi udubina s odabranim pripadajućim vektorima i bojama mape normala. Radi jednostavnosti dvodimenzionalnog prikaza izbačena je jedna od osi paralelnih s ravninom.

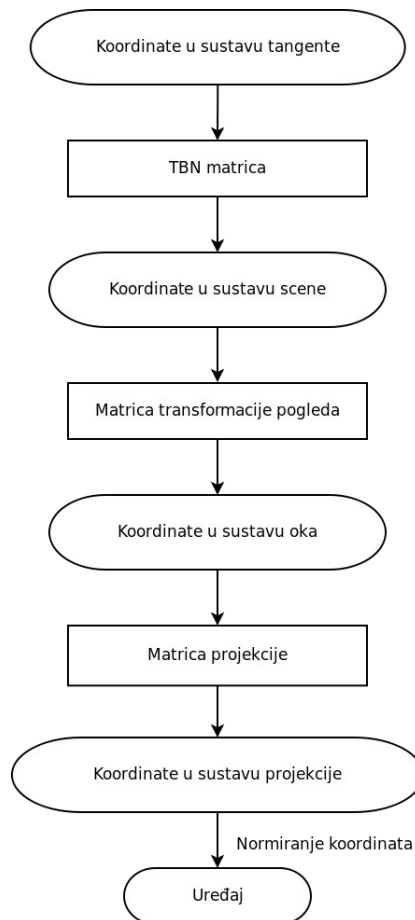
Vidi se da svi vektori imaju pozitivan iznos z osi, te je stoga slika pretežno plava. Lijevi rub udubine je crvenkast, jer je vektor pozitivan po x osi, dok je desni zelenkast (zbog smanjene količine crvene boje), jer je vektor negativan po x osi.



Slika 7: Prikaz uzvisine s bojama pripadajuće mape normala

Kod prikaza izbočine na desnom rubu imamo izraženiju crvenu boju (pozitivan smjer vektora), a na lijevom je crvena manje izražena (negativan smjer vektora).

Vektori dobiveni iz mape normala nalaze se u sustavu tangente. Sustav tangente je trodimenzionalni prostor čije dvije osi leže na površini objekta kojeg mapa opisuje, treća os je normala na tu površinu. Na slici 4, os x je S-tangenta, os y se naziva T-tangenta ili binormala, a os z je normala na ravninu. Te se osi označavaju redom slovima T, B i N. Matrica koja koordinate sustava scene pretvara u koordinate sustava tangente naziva se TBN matrica.



Slika 8: Transformacije koordinati kod prikaza preslikavanja normala

Vektor zrake svjetlosti potrebno je prebaciti u prostor tangente da bi ga se moglo pomnožiti s normalama neravne površine. Transformacija se obavlja inverznom TBN matricom.

$$\text{TBN matrica: } \begin{bmatrix} t_x & b_x & n_x \\ t_y & b_y & n_y \\ t_z & b_z & n_z \end{bmatrix}$$

$$|M_{TBN}| = (t_x b_y n_z - t_z b_y n_x) + (b_x n_y t_z - b_z n_y t_x) + (n_x t_y b_z - n_z t_y b_x)$$

$$\text{Inverzna TBN matrica: } M = \frac{1}{|M_{TBN}|} \begin{bmatrix} (B \times N)_x & -(B \times N)_y & (B \times N)_z \\ -(N \times T)_x & (N \times T)_y & -(N \times T)_z \\ (T \times B)_x & -(T \times B)_y & (T \times B)_z \end{bmatrix}$$

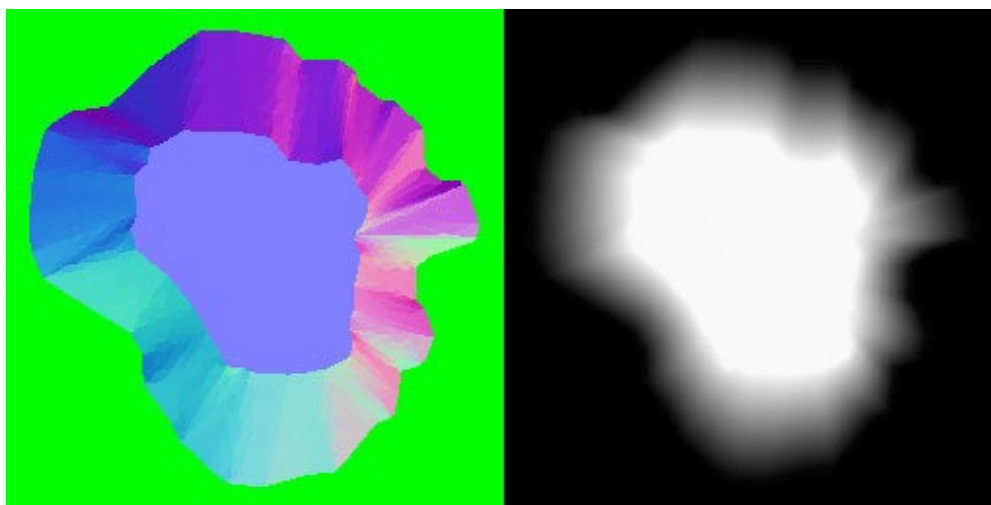
Ako su T i B okomiti TBN matrica je ortonormirana što znači da je za dobivanje inverza dovoljno samo transponirati matricu. To vrijedi kad se normale računaju za pojedine trokute, tj. kad se radi o konstantnom sjenčanju, ali ako je potrebna interpolacija normala onda je potrebno računati normale za svaki vrh, što znači da TBN prostor više nije ortonormiran, te je potrebna općenitija inverzna matrica.

Postupak preslikavanja normala:

1. Izračunati inverz TBN matrice za sve trokute
2. Izračunati vektor zrake svjetlosti te ga transformirati iz sustava scene u sustav tangente pomoću inverzne TBN matrice.
3. Izračunati vektore normala iz mape normala.
4. Skalarnim produktom vektora svjetlosti i normale dobiti kosinus kuta između njih, te ga pomnožiti s bojom svjetla i bojom materijala površine.
5. Ponoviti proces za svaki slikovni element površine

Preslikavanje paralakse

Paralaksa je prividni pomak objekta kod promjene položaja promatrača, pri čemu se objekti koji su udaljeniji od promatrača pomiču sporije od onih koji su blizu. Preslikavanje paralakse je nadopuna preslikavanja normala. Uz mapu normala koristi se i dubinska mapa, crno-bijela slika koja opisuje izbočenost i udubljenost pojedinog slikovnog elementa.



Slika 9: Mapa normala i dubinska mapa (West, M. 2007.)

Kod preslikavanja paralakse koordinate teksture se pomiču s obzirom na položaj promatrača i vrijednost dubinske mape slikovnog elementa. Kod oštrijih kutova pogleda na površinu, pomak koordinata teksture je veći, pri čemu se stvara iluzija dubine zbog efekta paralakse kod pomaka kuta gledanja. Time se rješava glavni problem metode preslikavanja normala – neprirodan izgled površine kod manjih kutova gledanja.

Nedostatak osnovnog algoritma preslikavanja paralakse je nemogućnost prekrivanja dijelova teksture; iako se dijelovi teksture mogu suziti i proširiti te time stvoriti privid dubine, dio teksture se ne može sakriti. Posljedica toga je pojava artefakata kod površina koje zahtijevaju veću dubinu prikaza. Razvijena su brojna proširenja osnovnog algoritma koja rješavaju problem prekrivanja. Jedno od njih je strmo preslikavanje paralakse (steep parallax mapping) koje omogućuje prekrivanje dijelova teksture te koristi sjene za dodatni privid dubine.



Slika 10: Usporedba metoda prikaza neravnih površina

Postupak preslikavanja paralakse sličan je postupku preslikavanja normale, s nekoliko dodatnih koraka koji prethode koracima za preslikavanje normale:

1. Iz dubinske mape se učitava dubina na koordinatama slikovnog elementa (u, v)
2. Vektor od promatrača na površinu se pomoću inverzne TBN matrice transformira u sustav tangente.
3. Vektor promatrača se normalizira te se skalira s dobivenom dubinom i konstantom dubine.
4. Komponente vektora x i y se dodaju koordinatama u i v , te se koriste nove koordinate, z koordinata vektora se zanemaruje.
5. Obavljaju se koraci za preslikavanje normala, ali s novim koordinatama.

Nedostatak svih metoda preslikavanja paralakse je nemogućnost prikazivanja neravnina preko ruba teksture. Zbog toga se metoda najčešće koristi za velike površine kao što su podovi i zidovi, a na rubovima zidova se koristi stvarno modeliranje da se izbjegne nerealni prikaz. Za prikaz sitnijih detalja na manjim površinama prikladnije je preslikavanje normala, jer nema veće razlike u prikazu, a metoda preslikavanja normala se brže izvodi.

Praktični dio

Praktični dio ovog rada je prikaz metode preslikavanja normala i ostvaren je u jeziku C++. Korišteno je grafičko programsko sučelje OpenGL i jezik GLSL za opisivanje programa za sjenčanje. GLSL (*OpenGL Shading Language*) je viši jezik za sjenčanje i baziran je na programskom jeziku C te omogućuje izravno upravljanje grafičkim cjevovodom bez korištenja assemblera ili drugih jezika koji ovise o pojedinoj grafičkoj platformi.

Program za sjenčanje sastoji se od potprograma za obradu vrhova i potprograma za obradu fragmenata. U potprogramu za obradu vrhova obavlja se transformacija vektora u sustav tangente korištenjem TBN matrice, a u potprogramu za obradu fragmenata se množenjem vektora svjetlosti i vektora dobivenog iz mape normala računa osvjetljenje svakog fragmenta.

Za prikaz neravnina korišten je model čajnika koji dolazi sa GLUT-om (*GL Utility Toolkit*). Čajnik bez korištenja preslikavanja normala ima glatku površinu. (slika 11.)



Slika 11: Model čajnika bez korištenja prikaza neravnina

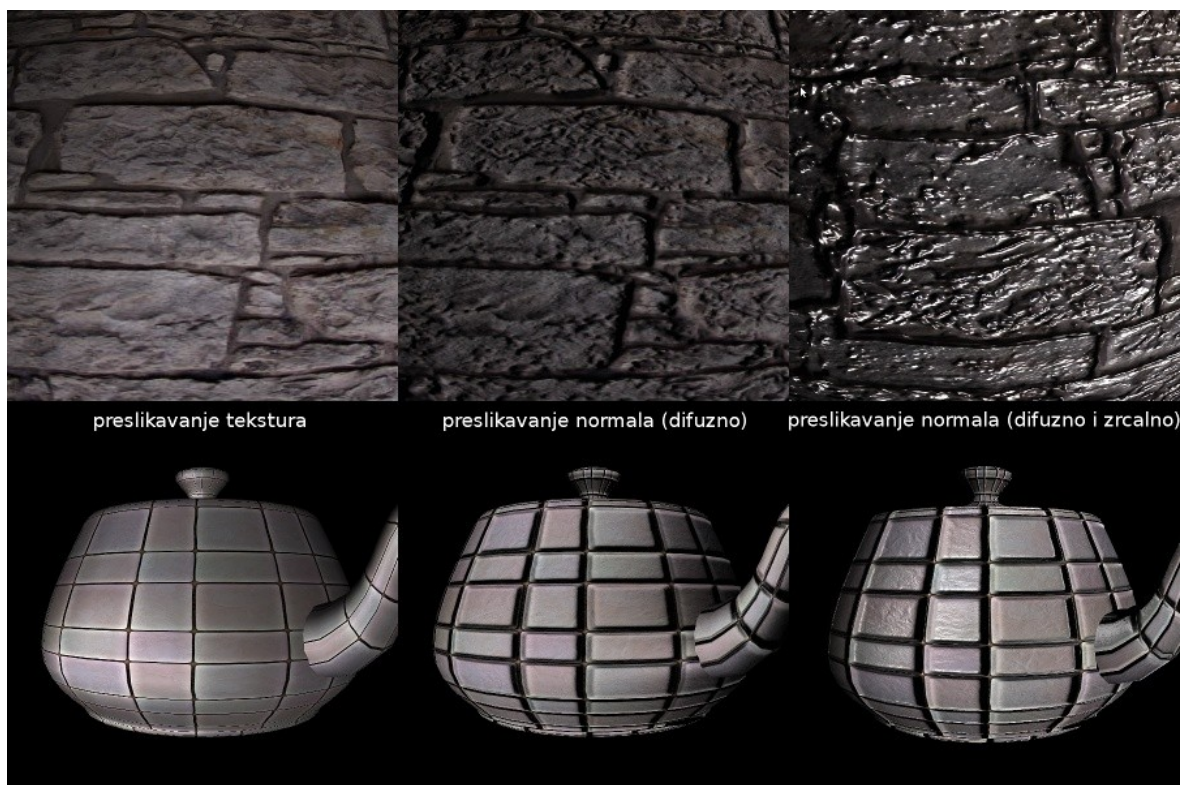
Prikaz neravnina preslikavanjem normala može se napraviti na difuznoj i zrcalnoj komponenti u Phongovom modelu osvjetljenja. Ako se prikaz neravnina radi samo na difuznoj komponenti ne smije se koristiti prikaz zrcalne komponente, jer neravna difuzna komponenta izgleda neprirodno uz korištenje glatke zrcalne komponente (slika 12.). Zbog toga se prikaz neravnina na difuznoj komponenti koristi za površine koje nisu sjajne, kao što su zidovi, papir, pamučne tkanine ili nelakirano drvo. Zrcalnu komponentu potrebno je

koristiti za reflektivne površine poput plastike, metala ili lakiranih površina.



Slika 12: Neprirodni prikaz neravnina difuzne komponente s glatkom zrcalnom komponentom.

Prikaz preslikavanja normala je puno manje uočljiv na statičnim slikama nego u pokretu, osim ako se koriste vrlo izražene neravnine, poput onih u donjem redu slike 13. Na slici 13. prikazani su rezultati praktičnog dijela ovog rada.



Slika 13: Usporedba neravnih površina dobivenih preslikavanjem normala s glatkim površinama.

Zaključak

U ovom radu su prikazane neke od danas najčešće korištenih metoda prikaza neravnina. S obzirom na brzinu napretka ovih metoda te na korištenje sve novijih metoda u računalnim igrama zatvorenog koda bilo bi vrlo teško dati potpuni pregled svih metoda. Ipak, sve se uglavnom temelje na ovdje opisanim konceptima. Napredak tehnologije omogućio je jednostavnije pisanje programa za sjenčanje te se svaka metoda može prilagoditi posebnim potrebama.

Praktični dio rada sastoji se od prikaza preslikavanja normala na čajniku Utah. Za opis preslikavanja koristi se jezik GLSL koji je dio otvorenog OpenGL standarda. Pisanje programa koji se izvode na grafičkoj kartici mnogo je zahtjevnije od onih koji se izvode na glavnom procesoru, najviše zbog nemogućnosti jednostavnog otkrivanja mjesta greške u programu bez korištenja posebnih programa za otkrivanje pogrešaka (*debuggera*). S druge strane, logičke pogreške u programu poput korištenja pogrešnih vektora ili transformacijskih matrica često proizvedu neočekivane, ali vizualno vrlo atraktivne rezultate.

Literatura

- [1] Mihajlović, Ž. Nastavni materijali iz predmeta Interaktivna računalna grafika.
- [2] Čupić, M. Skripta iz računalne grafike.
- [3] Dreijer, S. Bump Mapping Using CG (3rd Edition). Blacksmith Studios.
http://www.blacksmith-studios.dk/projects/downloads/bumpmapping_using_cg.php
- [4] West, M. Parallax mapped bullet holes. Cowboy Programming.
<http://cowboyprogramming.com/2007/01/05/parallax-mapped-bullet-holes/>
- [5] Blinn, J.F. Simulation of wrinkled surfaces. ACM SIGGRAPH Computer Graphics. 12/3 1978. Stranice 286-292.
- [6] Cohen, J. Olano, M. Manocha, D. Appearance-preserving simplification. Proceedings of the 25th annual conference on Computer graphics and interactive techniques. 1998. Stranice 115-122.

Sažetak

Prikaz površinskih neravnina

Preslikavanje neravnina je tehnika prikaza predmeta kod koje se za svaki slikovni element površine objekta provjerava i računa odstupanje od uobičajene boje elementa s ciljem postizanja prividne neravnosti objekta. Podaci o neravninama objekta spremljeni su u posebnim teksturama u kojima su bojama slikovnih elemenata označene neravnine. Najčešće korištene tehnike su preslikavanje normala i preslikavanje paralakse, iako postoje brojne druge tehnike koje se uglavnom temelje na navedenima. Prikaz neravnina ostvaruje se pisanjem programa za sjenčanje koji omogućavaju prikaz ovako složenih transformacija u realnom vremenu.

računalna grafika, preslikavanje neravnina, preslikavanje normala, preslikavanje paralakse

Summary

Display of bumpy surfaces.

Bump mapping is a method in computer graphics in which illumination of each pixel of a given surface is calculated to simulate roughness of its surface. Surface perturbations are stored in special textures in which surface changes are coded using colors. Most commonly used methods are normal mapping and parallax mapping. There are other methods, but they are usually based on normal and parallax mapping. Display of bumpy surfaces is implemented using vertex and fragment shaders, which enable such complex transformations to be calculated in real time.

computer graphics, bump mapping, normal mapping, parallax mapping