Procesor vrhova je prva programabilna faza koja implementira funkcije transformacije u prostor kamere, sjenčanja vrhova i projekcije.Često se koristi za animaciju, danas rijetko za sjenčanje. Procesor vrhova pokreće se za svaki vrh koji je dan GPU. Ulaz u procesor je pojedini vrh sa koordinatama, položajem vrha, normala, koordinate tekstura,boje. Radi na pojedinom vrhu i nema pristup drugim vrhovima, ne može brisati ni dodavati vrhove. Ti procesori se koriste za manipulaciju UV mapiranja i poziciju i boju svakog vrha. Izlaz. Minimalno: položaj nakon projekcije.
Procesor točaka implementira funkcijsku fazu sjenčanja točaka i potpuno je programabilan. Fragment je ulazni podatak u procesor točaka. Zadatak mu je sječnčanjem odrediti boju fragmenta te je prosljediti na izlaz(u fazu stapanja).U jednoj točki može biti više fragmenata ali samo jedan je vidljiv(najbliži kameri).Nije moguće dobiti podatke o drugim točkama.Izlaz je boja fragmenta.Pokreće se za svaki pixel/točku koja mora biti iscrtana. Koristi se za preslikavanje ravnina, sjena, spekularno sjenčanje. Ulaz: fragment – točka trokuta sa podacima za sjenčanje, operacije: izračun modela osvjetljenja, teksturiranje. Izlaz: boja
Jezici za sjenčanje slični jeziku C: HLSL, GLSL, Cg
Übershader: Zaseban shader za svaku kombinaciju materijala i svjetla, Nije praktično shader za svaku varijantu pisati Zasebno, Piše se jedan složeni übershader: Preprocesorske naredbe za razne kombinacije, Višestruko prevođenje – različite varijante shadera. Dakle to je jedan veliki shader/procesor koji ima implementirane razne opcije kao što su korištenje normalnih mapa, refleksijskih mapa, diffuzije, emisije.
Filtriranje teksture: Ako je tekstura = slika kada se pomoću u,v koordinata iz slike dohvaća vrijednost teksela, javljaju se problemi poduzorkovanja ili naduzorkovanja (ovisno o udaljenosti od kamere). Konačni efekt preslikavanja teksture je pojava slike na zaslonu, a veličina slike na zaslonu ovisi o udaljenostima predmeta s preslikanom teksturom od kamere. Idealno – korespondencija piksel-teksel 1:1, Idealno – korespondencija piksel-teksel 1:1, Slika umanjena (predmet dalje od kamere) → aliasing.
Metoda najbližeg susjeda - Dohvaća se teksel najbliži u,v koordinatama, Naglašen efekt blokova, najjednostavniji način dohvata teksela, jednostavno se uzima vrijednost teksela najbližeg u,v koordinatama uzorkovanja. Koristi boju točke originalne teksture koja je najbliže centru pixela kojeg treba renderirati.
Bilinearna interpolacija – vrši se interpolacija unutar kvadrata 2x2 susjedna teksela odnosno 4 točke najblžie centru točke koju iscrtavamo, između četiri teksela najbližih u,v koordinatama na kojima se uzima uzorak, kao dvije uzastopne linearne interpolacije, prvo po osi u, pa po v.Zamućena slika, ublažen efekt blokova. SKICA: koord sustav, dolje u1,u,u2, gore v1,v,v2. unutar kvadrat T11,T21,T22,T12. U sredini T gore T1 dolje T2.
Mipmap je „puno toga u malom prostoru“, 1 piksel prekriva više teksela. Metoda koja se najčešće koristi za rješenje problema umanjene slike na način da se skup umanjenih i filitriranih slika teksture pripremi unaprijed i pohrani zajedno s originalnom slikom. (svaka 2x manja od prethodne, (datoteka veća za samo 1/3) Ispravno bi bilo filtrom zbrojiti utjecaj svih teksela za svaki piksel, ali to je preskupo realtime. Prilikom preslikavanja teksture koriste se tekseli iz slika čija je veličina trenutno najbolje odgovara veličini piksela, određeno paremetrom d-razina detalja. Parametar d (razina detalja, level od detail) – mjera prekrivanja bloka teksela pikselom, Za uzorkovanje mipmap teksture trebamo u,v i d (veći d → odabire se manja slika), 2 načina izračuna d:1. Najdulja stranice četverokuta-projekcije piksela na teksturu,2. Iz diferencijala u i v duž osi zaslona x i y (du i dv)
Trilinearna interpolacija – funkcija korespondencije ulazi parametri u,v,d. Teksel se dobiva tako da se korištenjem parametra d odrede dvije susjedne mipmap razine teksture, te se potom iz njih dohvate tekseli korištenjem bilinearne interpol. Dobiveni tekseli se linearno interpol po parametru d dobiva se konačni uzorak. Postupak: 1. Odredi vrijednost d,2. Pomoću d, odredi dvije susjedne mipmap

slike,3. Uzorkuj obje slike na u,v (uz korištenje bilinearne interpolacije),4. Linearno interpoliraj rezultat po d

Preslikavanje okoline(environment mapping) je metoda za simuliranje zrcaljenja na oblim površinama. Simuliraju se visoko ssjajni metalni ili platični predmeti, slika okoline gledane iz perspektive predmeta se pohrani u teksturu koja se primjeni na predmet, koordinate teksture se izračunavaju prilikom iscrtavanja, promjena položaja – odsjaj se mjenja, ovisi o položaju promatrača,SKICA:promatrač oko, vektor v, sjajna površina, vektor n gore, vektor r prema tekstura slika okoline

Kuglasto preslikavanje – najjednostavnija metoda, kuglasta tekstura okoline sadržo sliku okoline dobivenu ortografskom projekcijom na površinu savršene kugle koja se nalazi oko predmeta, iscrtavanje se vrši tako da se uzme teksel tamo gje zraka R sječe kuglu.

Kockasto preslikavanje – najšire korištena metoda, kockasta slika generira se projekcijom okoline na 6 strana kovke postavljene oko predmete, rezultat se pohrani u teksturu od 6 djelova kockasta tekstura. Pri sjenčanju predmeta vektor reflektirane trake R koristi se za određivanje teksturnih koordinata za uzorkovanje kockaste teksture okoline. Postupak: 1.vektor R se normalizira, 2.određuje se najveća komponenta od R(x,y iliz) i predznak i određuje se strana kocke za uzorkovanje, ostale dvije komponente od R skaliraju se na raspon [0-1], 3. Uzorkuj odgovarajuću stranu (u,v koordinate – 2 manje komponente r)

Preslikavanje neravnina (bump mapping) – skup tehnika za simuliranje neravnih površina, ideja je da se poremećajem normale u svakoj točki predmeta stvori dojam neravnina bez dodatne geometrije. Postoje dva načina: 1. Koristi se visinska mapa koja govori koji dio površine je višlji od drugih, tada shader izračuna normalu površine prema znanoj visini i izračuna interakciju površine sa svjetlima, 2. umjest ovisinske mape koristi se normalna mapa odnosno koristi se normalno mapiranje. Phongov model osvjetljenja – difuzna i spekularna komponenta ovise o normali

Preslikavanje normala(normal mapping) - u teksturu neravnina izravno se pohranjuju normale u pojedinoj točki površine, dakle tekstura normala. U programu za sjenčanje točaka uzorkuje se tekstura normala korištenjem interpoliranih u,v koord u toj točki te se dobiva normala. Zadaju se u koord sustavu tangente, definiran trima vektorima: vektor tangente, bitangente i geometrijske normale. Uzorkuje se u procesoru točaka, dobivena normala se koristi za proračun osvjetljenja

Preslikavanje pomaka(displacement) – rješava problem kada se površina promatra pri oštrom kutu zbog iluzije promjene geometrije. Ona koristi podatke iz teksture visina za deformaciju same geometrije predmeta pa generira neravnine bez trikova sjenčanja, Predmet se može samosjenčati. Ulaz – „ravna" geometrij, (Vertex shader) Za svaki vrh se uzorkuje tekstura visina, te se vrh pomiče, Nije „pravi" bump mapping (ne povećava detaljnost geometrije), „Pravo" preslikavanje pomaka: (Geometry shader) Uzorkuje se tekstura visina, te se stvaraju dodatni trokuti za neravnine

Program za sjenčanje vrhova - Ulazni parametri su koordinate i normala. Izlazni parametri su transformirane koordinate vrha, vektor svjetla te normala transformirana u globalni koord. Sustav.Računa normalu i vektor svjetla u globalnom koordinatnom sustavu,Transformira vrh u prostor pogleda i projicira ga

Struct vertexIn{ Float3 Position: POSITION; Float3 Normal: NORMAL;}

Struct vertexOut{ Float3 Position: POSITION; Float3 LightVec: TEXCOORD1; Float3 WorldNormal:TEXCOORD2;}

Vertex Out std_us(vertexIn IN){ Vertex OUT; Float4 No) float4(IN.Normal,0); OUT.WorldNormal = mul(N0,WorldITXf).xyz; Float4 Po = float4(IN.Position,1); Float4 Pw = mul(Po,WorldXf); OUT.LightVec) (Lamp0Pos – Pw.xyz); OUT.Hposition = mul(Po,WvpXf); Return OUT;}

Dohvat podataka (uzorkovanje): Kontinuirane teksturne koordinate (0-1),Odgovaraju jednoj ili više memorijskih lokacije, Ako ih je više, sadržaj se interpolira (filtriranje tekstura)

Model izvođenja: 1. U pripremnom koraku: Učitavanje programa za sjenčanje (preko grafičkog API-ja), Postavljanje vrijednosti konstanti, alokacija memorijskih resursa (tekstura, spremnika vrhova...),2. U petlji iscrtavanja: Poziv funkcije za iscrtavanje (engl. draw call), 1 poziv iscrtavanja = 1 prolaz geometrije kroz protočni sustav, Pritom će procesori za sjenčanje izvesti učitane programe na svim primitivima, Rezultat se zapisuje u 1 ili više memorijskih spremnika (npr. spremnik boje, teksturu...)
Procesor vrhova : Ulaz: vrh sa pripadajućim podacima(Minimalno: koordinate, Dodatno: normala, teksturne koordinate, boja..., Rad samo na pojedinom vrhu (nema pristupa drugim vrhovima),) Operacije na ulaznim podacima(Minimalno: transf. u prostor kamere, projekcija, Sve ostalo ovisno o željenom efektu , Ne može brisati niti dodavati vrhove), Izlaz(Minimalno: položaj nakon projekcije, Izlazni registri se interpoliraju u prolazu trokuta, interpolirane vrijednosti za svaku točku ulaze u procesor točaka)
Procesor geometrije: Uporaba je izborna, Ulaz (Objekt (trokut, crta, točka) s pripadajućim vrhovima (+ dodatno susjedni vrhovi)), Izlaz(Nula ili više objekata: procesor može brisati, dodavati i mijenjati primitive (skupa operacija, nije za teselaciju)), Primjeri korištenja(Generiranje čestica raznih veličina i oblika u sustavima čestica, Iscrtavanje siluete u efektu krzna, Generiranje proceduralnih objekata , Generiranje fraktalne geometrije)
Izlaz struje podataka stream output, Izlaz procesora vrhova/geometrije može se zapisati izravno u memorijski spremnik, Već u idućem prolazu se može koristiti kao spremnik vrhova – iterativna obrada podataka!, Faza rasterizacije se može potpuno isključiti (bolje performanse), Korisno za GPGPU
Procesor točaka (fragmenata) : Ulaz: fragment(Fragment: točka trokuta sa podacima za sjenčanje (koordinate, dubina, boja, teksturne koord., normala...), Podaci fragmenta – određeni interpolacijom u prolazau trokuta, Procesor točaka sjenčanjem računa boju fragmenta, Ako fragment preživi određivanje vidljivosti (Z-buffer), boja se zapisuje u spremnik boje (postaje bojom piksela)), Izlaz: boja(Može i izbrisati fragment ili mu promijeniti dubinu (z), Višestruki spremnici (engl. Multiple Render Targets, MRT)), Nema pristupa susjednim točkama, ali ima diferencijalima podataka (ddx, ddy) Zato što se točke obrađuju u blokovima 2x2 – 8x8, Korisno za mipmapping ili anti-aliasing
Stapanje: Zadatak: odrediti konačnu boju točke (piksela) koja će se vidjeti na ekranu, Ulaz: boja više fragmenata koji se preslikavaju u točku, Z-spremnik, spremnik maske..., Određivanje vidljivosti metodom Z-spremnika, Operacije sa spremnikom maske, Nije programabilna, ali velika fleksibilnost u postavkama (rasterske operacije, ROP), Najčešće za miješanje boje, Neovisno miješanje višestrukih spremnika (MRT), Zadatak faze stapanja je odrediti konačnu boju točke koja će se vdijeti na zaslonu, konačna boja je rezultat stapanja boja više fragmenata koji se preslikavaju u tu točku. Na rezultat utječu i drugi spremnici, Z-spremnik i spremnik maske. Jedna od osnovnih operacija faze stapanja je određivanje vidljivosti metodom Z-spremnika. Nije programabilna. Ulaz: boja više fragmenata koji se preslikavaju u točku.
Programski jezici za sjenčanje : Cg – c for graphics, HLSL - High Level Shading Language, GLSL - OpenGL Shading Language
Efekt: skup programa za sjenčanje i svih potrebnih postavki protočnog sustava (npr. uključen/isključen Z-spremnik), Zapisuje se u jedinstvenoj datoteci korištenjem jezika za efekte, Globalni parametri efekta: Kroz njih korisnik upravlja svojstvima efekta, Moguće ih postavljati kroz sučelje alata za razvoj efekata, Npr. efekt Phongovo sjenčanje – parametri materijala, svjetla. Primjer efekata: Goochovo sjenčanje - Vrsta ne-fotorealističnog sjenčanja, Topli tonovi u smjeru svjetla, hladni od svjetla
Shaderi su organizirani u prolaze (engl. pass):1 prolaz iscrtavanja kroz grafički protočni sustav, Sastoji se od programa za sjenčanje vrhova i točaka (te eventualno geometrije) + postavke protočnog sustava, Više prolaza čini tehniku: Slijed prolaza potrebnih za realizaciju efekta,Često može biti više tehnika (npr. za različite generacije grafičkog sklopovlja)

Program za sjenčanje točaka:

```
float4 gooch_PS(vertexOutput IN){ float3 Ln = normalize(IN.LightVec);  float3 Nn = normalize(IN.WorldNormal);  float ldn = dot(Ln,Nn);  float mixer = 0.5 * (ldn + 1.0);  float4 result = lerp(CoolColor, WarmColor, mixer);  return result; }
```

Kombiniranje više svjetala i materijala : 1. Dinamičko grananje (M tipova svjetala, N tipova materijala, L svjetala po predmetu),2. Übershader(Zaseban shader za svaku kombinaciju, Nije praktično shader za svaku varijantu pisati zasebno, Preprocesorske naredbe za razne kombinacije, Višestruko prevođenje – različite varijante shadera),3. Višeprolazno osvjetljenje(Zaseban prolaz iscrtavanja za svaki izvor svjetlosti, Aditivno miješanje rezultata u spremnik boja u fazi stapanja, Broj shadera: MxN, Shaderi su jednostavni, ali performanse su slabije (aditivno miješanje, ponavljanje proračuna)),4. Odgođeno sjenčanje (Sjenčanje se radi nakon određivanja vidljivosti Z-spremnikom, 1. prolaz:Iscrta se scena bez sjenčanja, ali uz uključen Z-spremnik, Daljnji prolazi: Iscrta se jedan pravokutnik koji prekriva cijeli zaslon, a na njega se „lijepi" slika scene, U procesoru točaka – uzorkuju se G-spremnici i obavlja sjenčanje

Metode preslikavanja teksture: Poopćeno teksturiranje, Animacija teksture, Preslikavanje materijala, Preslikavanje prozirnosti (alpha mapping), Preslikavanje svjetlosti (light mapping), Preslikavanje okoline (environment mapping), Preslikavanje neravnina (bump mapping)

Poopćeno teksturiranje, projekcija: Teksturiranje (texture mapping) = preslikavanje uzorka na geometriju predmeta, u,v koordinate obično zadane za svaki vrh (prilikom modeliranja), te se interpoliraju, Više tekstura na predmetu (multitexturing) = više parova u,v koordinata po vrhu

Funkcija korespondencije: Preslikavanje u,v koordinata u prostor teksture (vrijednost ograničena na [0,1]), Koordinate neovisne o dimenzijama – lako zamijeniti sliku (npr. manja razlučivost radi štednje memorije), Funkcije korespondencije:a. Ponavljanje (wrap, repeat, tile),b. Zrcaljenje (mirror),c. Ponavljanje ruba (clamp),d. Ograničavanje (border)

Dohvat teksela: = uzorkovanje teksture, Teksel – točka teksture na u,v koordinatama: RGB – 24-bit (8-bit po komponenti boje), RGBA – 24-bit + 8-bit α kanal, HDR – visoki dinamički raspon; više od 8-bit po komponenti, Uzorkovanje = čitanje memorijske lokacije, Moguće koristiti filtriranje – čitanje susjednih lokacija i interpolacija rezultata, teksel se može i izračunati Kako vrijednost teksela utječe na konačnu boju točke koja se sjenča? Izravna primjena kao boje, Težinsko miješanje s bojom osvjetljenja, Parametar materijala (difuzna ili spekularna tekstura, tekstura sjaja), Normala (v. preslikavanje normala)

Animacija teksture: Promjena slike teksture u vremenu, 1. Vremenski niz slika koje se izmjenjuju: Efekti vatre, dima, eksplozija, Slike pohranjene kao 3D tekstura (w koord. = vrijeme), 2. Animiranje teksturnih koordinata: Translacija, rotacija, skaliranje i smicanje koordinata u programu za sjenčanje, Tok duž površine

Preslikavanje prozirnosti (Alpha mapping): α kanal teksture definira prozirnost u svakoj točki predmeta, 1. Efekt naljepnice: Slika je neki nepravokutan oblik (npr. krug), ostali dijelovi prozirni, Moguće dodavati u scenu dinamički, 2. Primjena prozirnosti na sam predmet: Predmet proziran gdje je α = 0, Prikaz složenog predmeta jednim poligonom, Uz animacijuteksture

Preslikavanje svjetlosti (light mapping): Simulacija efekata difuzne svjetlosti, Osvjetljenje se računa unaprijed (npr. isijavanje) i pohranjuje u teksture svjetlosti (light map), (difuzna svjetlost ne ovisi o položaju promatrača), Teksture svjetlosti koriste se u realnom vremenu

Koordinatni sustav za normale - Globalni koordinatni sustav: Nepraktičan, ne smijemo pomicati niti deformirati predmet, Koordinatni sustavi predmeta: Moguće rigidne transformacije Nije moguća deformacija predmeta (npr. skinning kod likova), Koordinatni sustav tangente: 3 vektora: Vektor tangente T (na površinu predmeta), Vektor bitangente B (na površinu predmeta), Vektor geometrijske normale N(okomit na poligon), T i B – vektori u smjeru rastućih u,v koordinata: 1. Najčešće zadani u svakom vrhu (kao i sama normala), te se interpoliraju duž površine, 2. Mogu se

računati dinamički u proc. točaka iz diferencijala u i v (funkcije ddx i ddy), Koord. sustav tangente varira duž površine –normale ostaju ispravne prilikom deformacija, T i B općenito nisu okomiti → distorzija normale

Zašto je tekstura normala plava - posljedica činjenice da su normale pohranjene u sustavu tangente, budući da su normale gotovo uvijek okrenute u pozitivnom smjeru u odnosu na površinu, njihova z komponenta u sustavu tangente ima pozivitnu vrijednost(između 0 i 1), kako se z komponenta pohranjuje kao plava boja komponenta teksela, u teksturi dominira plava boja

Preslikavanje neravnina i okoline: (Environment bump mapping) - u,v koordinate za uzorkovanje teksture okoline odredimo iz poremećene normale, Rezultat – zrcaljenje na neravnoj površini (npr. vode)

Tehnike panoa (billboard): Za automatsku orijentaciju predmeta prema kameri, Predmet se uvijek vidi iz istog kuta, Na pravokutnik (pano) se lijepi 2D slika predmeta (sprite), Često se koriste preslikavanje prozirnosti i/ili animacija teksture, Pano poravnat sa zaslonom : Uvijek ista orijentacija prema zaslonu, Normala uvijek okomita na ravninu projekcije, Vertikalna os (up axis) = vertikalna os kamere, Globalno orijentirani pano: Vertikalna os djelomično globalno orijentirana –predmet ne može rotirati oko osi pogleda, Za predmete koji su fizički prisutni u sceni, a nisu radijalno simetrični – oblaci, Normala na pano:a. Okomita na ravninu proj.– rubni panoi izgledaju„iskrivljeno",b. Usmjerena prema kameri, Osni pano: Predmet prema kameri rotira oko fiksne osi, Primjer – simulacija drveća

Zrcaljenje na ravnini: jedno zrcalo u sceni, i to ravno, konstruitamo kompletnu kopiju svene zrcaljenjem oko ravnine zrcala, pa crtamo scenu i kopiju, Najjednostavniji slučaj: ravnina y=0 - Zrcaljenje izraženo matricom S(1,-1,1), Za općenitu zrcalnu plohu zadanu točkom P i normalom N: F=T(-P) R(N, (0,1,0)) preslikava plohu na y=0, Konačna zrcalna matrica je M=F S(1,-1,1) F -1, Postupak: 1. Iscrtavanje zrcaljene geometrije (transformirane matricom M),2. Iscrtavanje izvorne geometrije,3. Iscrtavanje zrcalne plohe Faktor prozirnosti djeluje kao faktor zrcaljenja

Problemi zrcaljenja: Geometrija ispod plohe se ne smije zrcaliti(Sve ispod zrcala treba odrezati, direktno ili korištenjem proizvoljne odrezujuće plohe (arbitrary clipping plane)), Odbacivanje stražnjih poligona(Zrcaljenje obrće redoslijed vrhova poligona → odbacivanje daje pogrešan rezultat! Isključiti odbacivanje, ili zadati obrnut redoslijed vrhova)

Sjene: Sjena nastaje jer predmet zaklanja svjetlo Jednostavno, ali: U sceni postoji velik broj potencijalnih zaklanjatelja, Zaklonjenost treba utvrditi u svakoj točki – skupo, Može biti više svjetala, Nomenklatura sjena: Sjene su mekane jer izvori svjetla nisu točkasti, Oštre sjene u prirodi rjeđe, no u grafici se često koriste (bolje išta nego ništa),

Metoda teksture sjena (shadow map) : Ideja – samo točke vidljive iz perspektive svjetla trebaju biti osvijetljene, Postupak: 1. Iscrtati scenu iz perspektive svjetla u Z-spremnik (samo dubine) → tekstura sjena (TS), 2. Iscrtati scenu iz perspektive kamere: U svakoj točki uzorkovati TS, Ako je z > z TS → točka je u sjeni, samo ambijentalno osvj, Samo 1 projekcijska ravnina za TS: za usmjerena svjetla, Za točkasta svjetla potrebna kockasta TS

Problem samosjenčanja: pojava da poligon baca sjenu na samog sebe št se manifestira kao uočljive tamne pruge na površini predmeta, dolazi zbog nedovoljne preciznosti pri usporedbi dubina, zbog koj se za neke točke pogrešno izračuna da su u sjeni, uklanja se uvođenjem korektivnog člana(bias), malog iznosa koji se oduzme od dubine u točki prije usporedbe s dubinom iz teksture sjena