Lösningsförslag till
Tentamen TNM008, 3D datorgrafik och VR

Grupp:	MT2 och NO2MT
Datum:	Måndagen den 10 mars 2003
Hjälpmedel:	inga

Förslag, inte facit

Detta är mina *lösningsförslag*, inget fullständigt facit. Svar med väsentligen samma innehåll som dessa har bedömts med full poäng i rättningen. Andra svar kan emellertid också ge full poäng, förutsatt att jag dels tolkar svaret som korrekt och tillräckligt fullständigt, dels anser att svaret är relevant för den fråga som ställdes.

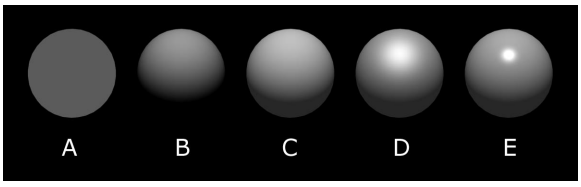
Uppgift 1 (3 p)
Använd *adaptiv sampling*. Aliasing uppstår på de ställen i bilden där det finns kanter och starka kontraster (stor skillnad i ljushet mellan intilliggande pixels). Scenen innehåller stora ytor som har låg kontrast eller ingen kontrast alls, så där kan man klara sig bra med ett sampel per pixel. Längs kanter kan man sampla tätare för att jämna ut taggigheten. Eftersom förhålandevis få pixels täcker en kant blir det inte väsentligt fler än ett sampel per pixel i genomsnitt.

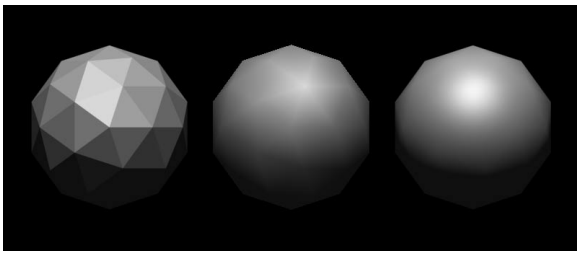
Uppgift 2 (3 p)
Den bästa lösningen i detta fall är att göra om modellen till ett texturmappat plan. Som textur kan användas en renderad bild av ett litet utsnitt av denna noggranna modell. För att få stängslet genomskinligt behöver man dessutom använda en transparenstextur (transparency map, opacity map, alpha map).

Uppgift 3 (7 p)

a) De tre huvudtermerna anger i tur och ordning reflexion av allmänljus (ambient reflection), diffus reflexion och speglande (spekulär) reflexion. I_a, I_d, I_s anger ljusintensiteten för respektive typ av belysning, och k_a, k_d, k_s anger materialets reflexionsegenskaper. $\hat{N}$ är ytnormalen för ytan, $\hat{L}$ anger riktningen till ljuskällan, $\hat{R}$ är den reflekterade ljusstrålens riktning och $\hat{V}$ är riktningen till betraktaren.Skalärprodukten $\hat{N} \bullet \hat{L}$ ger cosinus för vinkeln mellan ytnormalen och ljusets infallsriktning, vilket anger den diffusa reflexionen. Skalärprodukten $\hat{R} \bullet \hat{V}$ ger cosinus för vinkeln mellan riktningen för den perfekta speglande reflexionen och betraktningsriktningen. För att göra beroendet av vinkeln snävare upphöjs denna faktor till ett tal n, vilket är beräkningsmässigt praktiskt och ser rimligt bra ut, även om det inte är fysikaliskt korrekt.

b) Sfär A har endast "ambient reflection" (endast ka>0). Konstant färg över hela objektet.
Sfär B har endast diffus reflexion (endast kd>0). Ett bra matt utseende, men helt svart skuggsida.
Sfär C har är en kombination av ambient och diffus reflexion (ka, kd båda >0). Skuggsidan lättas upp.
Sfär D har dessutom en spekulär komponent (även ks>0). Ett blänk av ljuskällan syns.
Sfär E har ett högre värde på n, vilket ger ett mer koncentrerat blänk ("mer polerad yta").





Uppgift 4 (3 p)
Flat shading beräknar ljusreflexionen utifrån en normal per triangel, vilket ger ett facetterat utseende (bilden till vänster). För att efterlikna mjukt rundade objekt kan man i stället beräkna ljusreflexionen utifrån en normal vid varje hörnpunkt. Riktningen för denna normal efterliknar den krökta ytans normal. Färgen inom varje triangel interpoleras sedan på endera av två sätt. Antingen beräknar man reflexionsfärgen vid varje hörnpunkt och interpolerar färgen mellan hörnpunkterna (Gouraud shading, bilden i mitten), eller så beräknar man en interpolerad normal för varje pixel och använder den för att beräkna reflexionen (Phong shading, bilden till höger). Gouraud shading är matematiskt enklare, men ger ett sämre resultat. Polygonernas kanter är ofta synliga i Gouraud shading, speciellt om modellen har få och stora polygoner. Phong shading är också betydligt bättre än Gouraud shading på att återge noggranna spekulära blänk i ytor.

Uppgift 5 (4 p)
Innan man beräknar ljuset i scenen beräknar man en djupbild (Z-bild) sett från varje ljuskälla som skall kasta skuggor, och sparar dessa bilder. Dessa bilder benämns skuggmappar eller skuggbuffertar, och anger för varje pixel avståndet från ljuskällan till det närmaste objektet i scenen. Detta objekt skyms inte av andra objekt, och ligger därför inte i skugga. När belysningen för en punkt i den slutliga bilden skall beräknas transformerar man denna punkt i scenen till samma koordinatsystem som var och en av skuggbuffertarna, och jämför dess avstånd från ljuskällan med det lagrade värdet i skuggbufferten. Om avståndet är större än skuggbuffertens värde motsvarar punkten ett objekt som ligger bakom det objekt som är närmast ljuskällan, och alltså ligger i skugga. Om djupet är (ungefär) lika med skuggbuffertens djup är punkten belägen på det objekt som ligger närmast ljuskällan, och är alltså belyst.

Uppgift 6 (3 p)
En grafikprocessor är specialiserad för att utföra identiska eller likartade operationer på stora mängder data. Parallell bearbetning och en hög grad av pipelining gör att även en förhållandevis låg klockfrekvens kan ge en mycket stor beräkningskapacitet. En generell CPU måste kunna utföra många olika uppgifter och kan därför varken vara lika specialiserad eller parallelliserad. (Dessutom är huvudprocessorn i en dator upptagen med mycket annat än grafiken, så att låta en särskild enhet rita grafik gör att datorn som helhet arbetar snabbare.)

Uppgift 7 (4 p)

a) När man vrider på huvudet i en VR-hjälm måste bilden uppdateras mycket snabbt för att ge rätt vy för betraktaren. Om bilden släpar efter eller vyn inte är korrekt blir användaren yr och desorienterad. Om bilden i stället projiceras på en stor skärm kan betraktaren se sig om fritt utan att bilden behöver uppdateras alls.

b) Grundtanken är att presentera olika bilder för vänster och höger öga, och därmed utnyttja människors förmåga till stereoskopiskt seende. Den enklaste metoden använder billiga glasögon med ett rött och ett grönt (eller blått) glas, och bilden för vardera ögat presenteras i rött och grönt (eller blått). För att kunna presentera färgbilder i stereo kan man i stället använda sig av tidsmultiplex, så att bilderna för vänster och höger öga presenteras omväxlande i snabb följd. Glasögonen skall i detta fall blockera bilden för ett öga i taget och måste innehålla aktiva slutare. Ett sätt att göra detta är med LCD-teknik (flytande kristaller).

Uppgift 8 (5 p)

a) Invers kinematik (rörelsen hos den den ledade armens slutpunkt, verktyget, är det viktiga)

b) Fysikalisk simulering (enkel fysik, men en rörelse som kan vara krånglig att animera på andra sätt)

c) Keyframing (detaljkontroll över position, hastighet och orientering i varje del av rörelsen)

d) Motion capture (invecklat rörelsemönster, men går bra att spela in från verkliga dansare)

e) Keyframing (enkla rotationer, enkel modell som inte kräver invers kinematik)

Uppgift 9 (4 p)
Java3D är ett högnivå-API. OpenGL arbetar på en lägre och mer detaljerad nivå. Java3D förenklar och döljer mycket detaljer för programmeraren, men är inte lika generellt och snabbt som OpenGL. Java3D använder dock internt OpenGL (eller Direct3D) för att utföra själva ritandet.

Uppgift 10 (8 p)

a) Radiosity modellerar den diffusa ljusreflexionen mellan objekt i en scen, inte bara direkt ljus från ljuskällor. Metoden kan kombineras med vanlig raytracing för att även modellera vissa spekulära effekter, närmare bestämt speglande reflexioner av speglande och diffusa ytor. Att hantera reflexioner mot diffusa ytor från speglande ytor, så kallade “caustics”, är dock ett svårare problem.

b) Radiosity-lösningen anger den diffusa reflexionskomponenten för alla ytor i scenen oberoende av betraktarens position. För statiska scener kan man därför förberäkna radiosity-lösningen och lagra den som texturer, “light maps”, som sedan används för att direkt ange en ytas diffusa färg.

c) B_i anger ljusutstrålning (“radiosity”) från en viss yta i. E_i är egenemissionen från denna yta (alla ytor kan vara ljuskällor). R_i anger reflexionskoefficienten för ytan. Summan anger den ljusmängd som strålar in mot ytan från alla andra ytor. Detta beräknas utifrån utstrålningen B_j från dessa ytor och en uppsättning formfaktorer F_{ij} som anger hur stor rymdvinkel varje annan yta j upptar, sett från yta i.

d) “Gathering” innebär enkelt uttryckt att lösa ekvationen ovan exakt, och i tur och ordning räkna ut varje ytas reflexion genom att beräkna ljusbidragen från alla andra ytor. Denna metod är tämligen beräkningstung, men när man väl fått fram lösningen (vilket i princip är inversen av en stor matris) kan man återanvända den för att snabbt räkna ut resultatet för andra ljusemissioner och reflexionskoefficienter i scenen. “Shooting” innebär att i stället lösa ekvationen iterativt “baklänges”. Först fördelas ljuset från de emitterande ytorna till övriga ytor, och sedan fördelar man efterhand det reflekterade ljuset en reflexion i taget, tills man har tillräckligt litet ljus kvar att sprida vidare i scenen. Detta ger ingen exakt lösning, och resultatet kan inte återanvändas om emissionen eller reflexionen ändras, men man kan bryta iterationen när man är nöjd med det visuella resultatet, i stället för att som vid en exakt lösning vänta på att samtliga ytors utstrålning beräknats.

Uppgift 11 (6 p)

a) Direkt insättning ger $Q(u) = P_0(1-u)^2 + 2P_1u(1-u) + P_2u^2$

b) Ekvationen kan skrivas om enligt $Q(u) = u^2(P_2 - 2P_1 + P_0) + u(2P_1 - 2P_0) + P_0$, vilket på matrisform blir

$$Q(u) = \begin{bmatrix} u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} = P_0(u^2 - 2u + 1) + P_1(-2u^2 + 2u) + P_2u^2$$

c) Kurvans tangent ges av $\frac{d}{du}Q(u) = Q'(u) = (2P_1 - 2P_0) + 2u(P_2 - 2P_1 + P_0)$. Insättning ger:

$Q'(0) = 2P_1 - 2P_0 = 2(P_1 - P_0) = 2(P_0 \to P_1)$

$Q'(1) = 2P_1 - 2P_0 + 2(P_2 - 2P_1 + P_0) = 2(P_2 - P_1) = 2(P_1 \to P_2)$ V.S.V.

För övrigt kan noteras att tangenten i $u = 0.5$ är parallell med sträckan $P_0 \to P_2$:

$Q'(0.5) = (2P_1 - 2P_0) + (P_2 - 2P_1 + P_0) = P_2 - P_0$