

Lösningförslag till tentamen

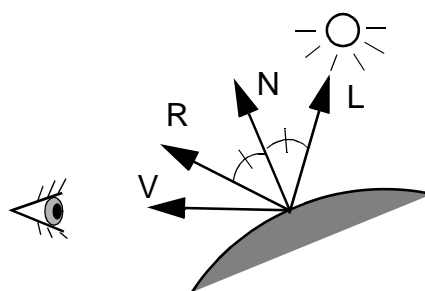
TNM077 3D-datorgrafik och animering

2006-03-17 kl 14-18

Detta är inget facit, utan ett lösningsförslag. Flera av uppgifterna är ganska fria och har en beskrivande karaktär, och där kan även andra svar än just precis dessa vara både korrekta och relevanta och därför tillmätas full eller nästan full poäng. Svaren som redovisas här är tillräckliga för full poäng, men de är alltså inte de enda tänkbara.

Uppgift 1 (10 p)

a) De geometriska parametrarna är vektorerna N , L , R och V . Objektets form bestämmer normalvektorn N för punkten i ytan. Objektets läge i förhållande till ljuskällan ger riktningen till ljuskällan L . Riktningen till betraktaren ger V , och givet N och L kan även den spekulära reflexionsriktningen R beräknas.



Materialets ytrefflexionsegenskaper bestäms av k_a , k_d , k_s och n . Den "allmänna" ljusreflexionen hos objektet (ambient light) bestäms av k_a , den diffusa reflexionen av ljuskällan anges av k_d , och den spekulära reflexionen av ljuskällan anges av k_s . Den spekulära reflexionens snabba avklingning med vinkeln modelleras genom en potensfunktion av $R \cdot V$, man upphöjer den till en exponent n . Ju större exponent, desto mer koncentrerat blänk, varvid ytan ser blankare ut.

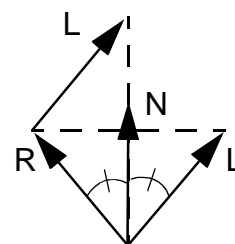
Ljusets intensitet delas upp i "ambient light", diffust ljus och spekulärt ljus. Intensiteterna hos dessa ljusbidrag till scenen anges av I_a , I_d respektive I_s . (I_a är inte kopplat till någon speciell ljuskälla och kan väljas fritt, men I_d och I_s beskriver strängt taget ljus från en och samma ljuskälla och bör därför sättas lika. Modellen i sig kräver dock inte detta.)

b) Sfär nummer 1 är diffust vit utan blänk, och har ganska mycket ljus även på skuggsidan utan att det blir väsentligt urfrätt "mer än vitt" på den ljusa sidan. Sätt förslagsvis $k_a = 0,3$, $k_d = 0,7$, $k_s = 0$, n oväsentligt. Sfär nummer 2 har en mellangrå diffus färg, inte lika mycket ljus på skuggsidan och ett svagt och ganska utbrett blänk. Sätt t ex $k_a = 0,2$, $k_d = 0,5$, $k_s = 0,5$ och $n = 4$. Sfär nummer 3 är svart och har ingen diffus reflexion alls, däremot ett starkt och koncentrerat blänk. Detta görs med värden i stil med $k_a = 0$, $k_d = 0$, $k_s = 1$, $n = 10$.

c) Färgade ljuskällor och färgade ytor hanteras genom att göra samma slags beräkning flera gånger för flera färgkanaler, till exempel rött, grönt och blått, med olika ljusintensitet och reflexionskoefficienter för varje färgkanal. (Såväl I_a , I_d , I_s som k_a , k_d , k_s kan då ses som färgvektorer med tre värden vardera.)

d) Vi observerar att $L \cdot N = R \cdot N$ eftersom reflexionsvinkeln är lika med infallsvinkeln. Vidare inses med hjälp av vidstående figur att $L + R = (L \cdot N)N + (R \cdot N)N = 2(L \cdot N)N$, vilket ger

$$R = 2(L \cdot N)N - L.$$



Uppgift 2 (6 p)

Derivatan beräknas enklast med hjälp av kedjeregeln: $\frac{d}{dt}\bar{p}(u(t)) = \frac{d}{du}\bar{p}(u) \frac{d}{dt}u(t)$.

Kontrollpunkterna \bar{p}_i är konstanta, så $\frac{d}{du}\bar{p}(u) = \sum_{i=0}^2 \bar{p}_i \frac{d}{du}(B_i(u))$.

Derivering ger $\frac{d}{du}B_0(u) = 2u - 2$, $\frac{d}{du}B_1(u) = 2 - 4u$, $\frac{d}{du}B_2(u) = 2u$, $\frac{d}{dt}u(t) = \frac{1}{\pi} \cos \frac{t}{\pi}$.

Insättning i summan följt av ersättning av u med $\sin \frac{t}{\pi}$ ger uttrycket:

$$\bar{v}(t) = \left(\left(2 \sin \frac{t}{\pi} - 2 \right) \bar{p}_0 + \left(2 - 4 \sin \frac{t}{\pi} \right) \bar{p}_1 + 2 \sin \frac{t}{\pi} \bar{p}_2 \right) \frac{1}{\pi} \cos \frac{t}{\pi}$$

Uppdelningen på komponenter (x, y, z) med de givna värdena för kontrollpunkterna blir:

$$\bar{v}(t) = \begin{bmatrix} \left(\left(2 \sin \frac{t}{\pi} - 2 \right) x_0 + \left(2 - 4 \sin \frac{t}{\pi} \right) x_1 + 2 \sin \frac{t}{\pi} x_2 \right) \frac{1}{\pi} \cos \frac{t}{\pi} \\ \left(\left(2 \sin \frac{t}{\pi} - 2 \right) y_0 + \left(2 - 4 \sin \frac{t}{\pi} \right) y_1 + 2 \sin \frac{t}{\pi} y_2 \right) \frac{1}{\pi} \cos \frac{t}{\pi} \\ 0 \end{bmatrix}$$

Komponentuppdelning krävs inte för full poäng. Svar som inte är på slutet form utan fortfarande innehåller en eller flera deriveringsoperatorer ger väsentliga poängavdrag. Enklare räknefel kan slippa lindrigt undan, men felaktiga deriveringar straffar sig. Derivering utan att utnyttja kedjeregeln är mycket krångligt, men ger full poäng om svaret blir rätt.

Uppgift 3 (6 p)

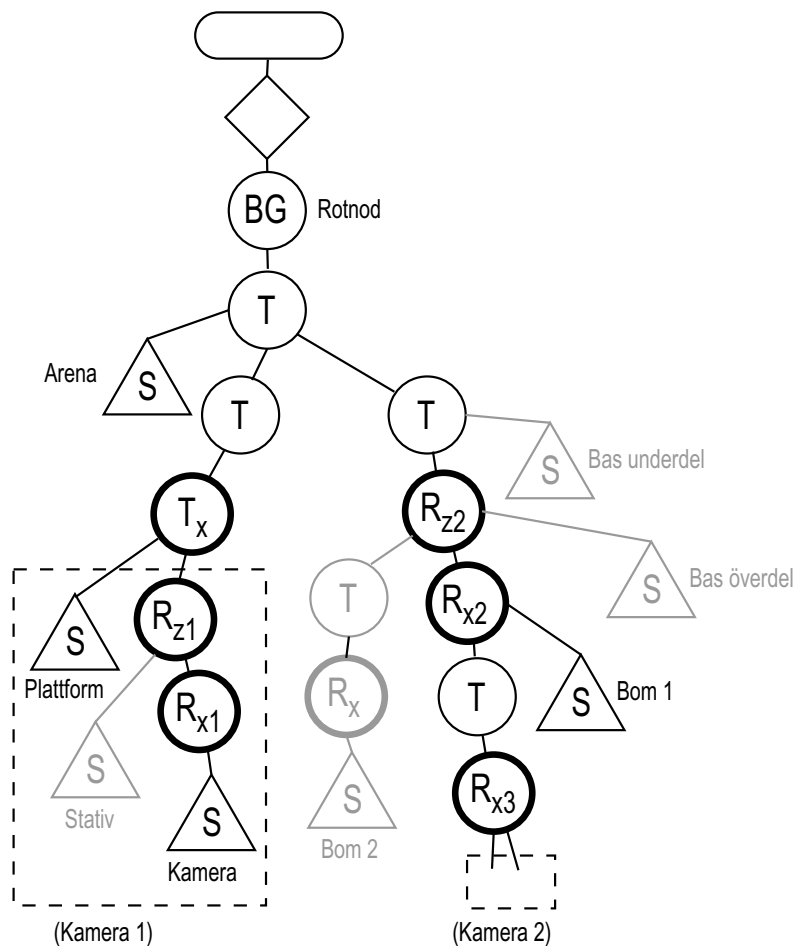
a) Scenen består av ett syntetiskt 3D-objekt lagt ovanpå en fotografisk bild. Tekanan renderas lämpligen mot en genomskinlig bakgrund, sparas för sig till en bild med transparenskanal ("alfakanal"), och läggs samman med fotografiet i efterhand i ett vanligt bildredigeringsprogram. Sammanläggningen kan även göras direkt i 3D-programmet om man tycker det är enklare. För att matcha positionen och storleken hos det syntetiska objektet med fotot så kan fotot användas som bakgrund i vyn vid modelleringen.

Tekanan skall vara spegelblank och behöver därför inte ens belysas i 3D-scenen, det räcker med att använda den framställda reflexionsmappen som "reflection map" (även kallat "environment map") för att ge en illusion av att den reflekterar en verklig omgivning. (Realismen i speglingen kommer helt från bilden, ingen egentlig reflexionsmodell behövs.)

b) Metoden med "reflection mapping" (eller med ett annat ord "environment mapping") baserar sig på en texturbild, men i stället för att använda fasta texturkoordinater för objektet så räknar man ut reflexionsvektorn $R = (R_x, R_y, R_z)$ enligt uppgift 1, beräknar dess riktning i sfäriska koordinater (φ, θ) , tex enligt $\tan \varphi = R_y/R_x$, $\cos \theta = R_z$, och sedan använder koordinatparet (φ, θ) som index till texturbilden. Detta kommer att ge en illusion av att ytan inte har texturans mönster målat direkt på sig, utan i stället reflekterar en omgivning som så att säga ligger tapetserad på insidan av en stor sfär som omsluter scenen.

Uppgift 4 (7 p)

a) Ett förslag till en rimligt fullständig scengraf ges i figuren nedan. Grenarna för de båda kamerorna har samma utseende. Den ena plattformen har sedan en enkel rörlig translation längs en axel som enda ytterligare rörelse, den andra är mer komplicerad med två oberoende och separata rotationer, en fast translation och en avslutande beroende rotation som gör att plattformen alltid är horisontell.



T betecknar translationer, R rotationer. Transformationsnoder med tjock kant är animerade, övriga är fasta translationer för att positionera objekt rätt mot varandra.

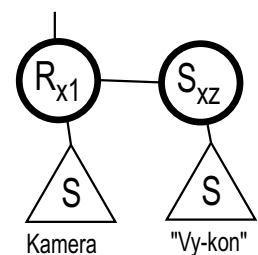
T_x är en translation längs en horisontell axel som styr den första kamerans position på rälsen. R_{z1} är en rotation runt den vertikala axeln som styr kamerans panorering i sidled. R_{x1} är en rotation runt en horisontell axel som styr kamerans lutning uppåt och nedåt.

Den andra kamerans graf har en likadan struktur, men egna oberoende rotationsvinklar.

R_{z2} är en rotation runt den vertikala axeln som styr kamerabommens rotation i horisontalplanet. R_{x2} styr bommens lutning. R_{x3} skall alltid vara lika med inversen av R_{x2} , för att kameraplattformen skall förbli horisontell.

De grå delarna av grafen måste inte tas med för full poäng. Det räcker med att de stora delarna finns med, att det är rätt slags objekthierarki och att den slutliga rörelsen för kameran är korrekt. Scengrafer som ger eller tillåter felaktiga rörelser ger poängavdrag, liksom scengrafer som inte medger de efterfrågade rörelserna. Villkoret att kameraplattformen alltid är horisontell skall finnas med på något sätt.

b) Det önskade beteendet kan lösas med en olikformig skalning, där man inte förändrar konens längd (t ex i y-led), men ökar och minskar dess bredd och höjd (t ex x- och z-led) samtidigt och med samma faktor. För att få konen med denna extra skalning att dessutom följa med kamerans rörelser skall den sitta som barn till kamerans transformation i grafen, se figur.



Uppgift 5 (6 p)

a) Radiosity modellerar diffus reflexion och ljustransport i en scen. Speglande och direkt ljus modelleras inte alls av metoden, så den kan inte ensam rendera alla slags bilder. Däremot är den en fysikaliskt motiverad och bra metod för att rendera naturtrogna bilder av objekt utan speglade ytor i en mjuk, diffus belysning.

För att beräkna en bild med radiosity måste man först avgöra synligheten mellan alla ytor i scenen, och beräkna hur mycket av det ljus som sänds ut från en yta som landar på andra ytor. Dessa data brukar kallas formfaktorer, och behövs för att kunna räkna på ljustransporten i scenen. Därefter sätter man startvillkor genom att ange vilka ytor som avger ljus, och tilldelar diffusa reflexionsegenskaper (färg) till alla ytor. Sedan räknar man ut energibalansen i scenen vid jämvikt, antingen genom att iterativt söka sig fram till en tillräckligt noggrann lösning eller genom att lösa en stor matrisekvation och invertera en matris som bland annat beror av samtliga formfaktorer. Lösningen ger den reflekterade intensiteten för alla ytor i scenen, oberoende av betraktarens position.

(Svar som förklarar detta tydligt med utgångspunkt i korrekta ekvationer ger också full poäng.)

b) Den största nackdelen med radiosity är att metoden är beräkningstung. Även om detta spelar mindre roll allt eftersom datorer blir snabbare så tar det väsentligt längre tid att rendera en bild med radiosity än med andra, mindre korrekta men oftast helt visuellt acceptabla metoder. Formfaktorerna beror dessutom på ett komplicerat sätt av scenens geometri, så rörliga objekt är extra krångliga att beräkna. För varje rörelse i scenen måste man egentligen räkna om alla formfaktorer från början. Det inledande steget med att beräkna formfaktorerna kan i själva verket vara det mest beräkningstunga. En annan uppenbar nackdel är att endast diffusa reflexioner behandlas, men speglingar och brytningar är viktiga komponenter i många verkliga scener. Inte ens skarpa skuggor eller riktade ljuskällor kan hanteras rakt av. Det finns också avsevärda problem med att dela upp sin scen i lagom stora ytelement. Hur man skall göra uppdelningen beror på ett icke-trivialt sätt av ljussättningen i scenen och behöver komplicerade adaptiva metoder för att göras på ett tillräckligt bra sätt.

Det finns dock flera fördelar också. Den goda realismen är en klar fördel, radiosity ger snygga bilder som har en god överensstämmelse med verkligheten. Om man räknat fram en uppsättning formfaktorer för en scen så är större delen av jobbet redan gjort, man kan sedan till exempel ändra belysningen, inklusive att ändra vilka ytor som sänder ut ljus, utan att man måste göra om alla beräkningar från början. Sist men inte minst så är den lösning man får fram helt vyoberoende, vilket gör att man kan förberäkna den och använda den som texturinformation i en realtidsrendering, en så kallad "light map".

(De flesta av dessa för- och nackdelar skall finnas med i svaret för full poäng.)

Uppgift 6 (5 p)

En samplad texturbild är en enkel och rättfram metod som sedan länge har bra stöd i realtidsgrafik. Sådana texturer renderas snabbt och effektivt och har dessutom använts sedan lång tid tillbaka, så alla verktyg för modellering har stöd för dem. En samplad bild har dock alltid en begränsad upplösning. När man zoomar in på texturen blir den därför förr eller senare suddig, se bilderna till vänster i figuren.

Om man vill kunna detaljstudera ytan behöver man lagra texturbilden i hög upplösning, och detta kostar både möda och minnesutrymme. Minnet på ett modernt grafikkort är stort, men begränsat. Även om man gör bilden väldigt stor så finns det ändå en risk att betraktaren vill gå ännu närmare än man tänkt, och då brister det i realism. Verkligheten har inte begränsad upplösning, i alla fall inte innan man kommer ner till atomnivå.

Om man i stället lagrar texturen som en konturbeskrivning så gör man på samma sätt som man länge gjort i 2D-grafik: man går från en representation med begränsad, fast upplösning till en representation som är skalningsoberoende, som tål att renderas i godtycklig upplösning. Då kan man zooma in extremt mycket utan texturen blir suddig eller "pixlig" (se bilderna till höger i figuren). Ändå tar texturen inte upp extra stor plats i minnet, de högupplösta detaljbilderna beräknas nämligen utifrån exakt samma data som den lågupplösta avståndsbilden.

Nackdelen med detta är att texturen blir mer komplicerad att rendera. I stället för att bara läsa ett pixelvärde i minnet så måste man beräkna värdet för varje pixel med en mer komplicerad algoritm. Dagens grafikkort kan visserligen göra dessa beräkningar snabbt, som det sägs i uppgiften, men beräkningskraften kan behövas även till annat. En annan nackdel är att detta är en ny teknik i realtidssammanhang, och därför just nu saknar stöd i 3D-modelleringsprogram. Äldre datorer och datorer utan speciell 3D-grafikhårdvara kan också få problem med avancerade metoder som denna.

(Även något mindre fylliga svar än ovanstående kan ge full poäng.)

För mer detaljer om denna metod, se <http://staffwww.itn.liu.se/~stegu/GLSL-conics/>