

Lösningförslag till tentamen

TNM077 3D-datorgrafik och animering 2005-03-17

(även omtentamen TNM008 3D-datorgrafik och VR)

Detta är inte ett facit, utan ett lösningförslag. Lösningen till uppgift 5 är inte fullständig utan hänvisar till kursboken, men de övriga presenterade lösningarna i den form de står här skulle gett full poäng i bedömningen. Även andra lösningar kan dock ge full poäng, och delvis korrekta eller ofullständiga lösningar kan ge delpoäng.

1a) Den första termen avser att modellera scenens "allmänljus", allt ljus som inte kommer direkt från ljuskällor utan reflekteras diffust från andra ytor. Materialparametern k_a bestämmer hur mycket av detta allmänljus som skall reflekteras av ytan, alltså hur mycket ljus som skall falla på de ytor som inte är direkt belysta av någon ljuskälla.

Den andra termen avser att modellera den diffusa reflexionen från ytan, den sorts reflexion som en matt yta ger. Materialparametern k_d anger hur mycket ljus som skall reflekteras diffust, vilket kan sägas vara lika med ytans färg.

Den sista termen avser att modellera den spekulära (speglande) reflexionen av ljuskällan, det som ger ett mer eller mindre skarpt blänk i blanka ytor. Parametern k_s anger färgen och intensiteten för detta blänk, och exponenten n anger hur koncentrerat detta blänk skall vara. Ett stort n ger ett litet, skarpt blänk som om ytan vore polerad, ett litet n ger ett stort, suddigt blänk som om ytan vore mer skrovlig.

1b) Den första termen är en mycket grov förenkling: en komplicerad interreflexion mellan diffusa ytor approximeras med ett ljus som finns överallt i scenen och lyser lika starkt i alla riktningar.

Den andra termen är faktiskt en riktigt bra modell av verkligheten. Den förutsätter att ytan är perfekt diffust reflekterade, och det är inte alla matta ytor som beter sig exakt så, men det är oftast en bra approximation.

Den sista termen är återigen en grov förenkling. Positionen för mitten av blänket är korrekt, men för att det skall vara enkelt att beräkna antar man att det speglande blänket avtar med vinkeln mot mitten som en potensfunktion av cosinus för vinkeln. Detta är inte fysikaliskt motiverat, men ser ofta hyfsat bra ut.

1c) Man summerar helt enkelt de diffusa och spekulära termerna över flera ljuskällor med olika intensitet I_i . Vektorerna L_i och R_i ändrar sig också beroende på ljuskällans position:

$$I = I_a k_a + \sum_i (I_i k_d (N \cdot L_i) + I_i k_s (R_i \cdot V)^n)$$

2a) Ansätt två kurvsegment $\bar{p}(u)$ och $\bar{q}(u)$ med kontrollpunkterna $\bar{p}_0, \bar{p}_1, \bar{p}_2, \bar{p}_3$ respektive $\bar{q}_0, \bar{q}_1, \bar{q}_2, \bar{q}_3$. Kontinuitet innebär att slutpunkten för den första kurvan är lika med startpunkten för den andra kurvan, alltså att

$$\lim_{u \rightarrow 1} \bar{p}(u) = \bar{p}(1) = \bar{q}(0).$$

Direkt insättning i ekvationen för kurvan ger $\bar{p}(1) = \bar{p}_3$ och $\bar{q}(0) = \bar{q}_0$, vilket ger villkoret $\bar{p}_3 = \bar{q}_0$, alltså att den sista kontrollpunkten för det första segmentet skall sammanfalla med den första kontrollpunkten för det andra segmentet.

2b) Kontrollpunkterna är konstanter med avseende på u , så derivatan fås genom att derivera endast polynomen med avseende på u :

$$\frac{d}{du} \sum_{i=0}^3 B_i(u) \bar{p}_i = \sum_{i=0}^3 \bar{p}_i \left(\frac{d}{du} B_i(u) \right) = \sum_{i=0}^3 \bar{p}_i B_i'(u)$$

$$B_0'(u) = -3 + 6u - 3u^2, B_1'(u) = 3 - 12u + 9u^2, B_2'(u) = 6u - 9u^2, B_3'(u) = 3u^2$$

Kontinuitet hos derivatan i skarven innebär att $\frac{d}{du} \bar{p}(1) = \frac{d}{du} \bar{q}(0)$. Insättning ger:

$$\frac{d}{du} \bar{p}(1) = -3\bar{p}_2 + 3\bar{p}_3, \frac{d}{du} \bar{q}(0) = -3\bar{q}_0 + 3\bar{q}_1, \text{ vilket ger villkoret } \bar{p}_3 - \bar{p}_2 = \bar{q}_1 - \bar{q}_0,$$

eller i ord: vektorn mellan den tredje och fjärde kontrollpunkten för det första segmentet skall vara lika med vektorn mellan den första och andra kontrollpunkten för det andra segmentet. Kombinerat med villkoret under a) ger det att de tre punkterna $\bar{p}_2, \bar{p}_3 = \bar{q}_0$ och \bar{q}_1 skall ligga i denna ordning längs en rät linje, och att avstånden \bar{p}_2 till \bar{p}_3 och \bar{q}_0 till \bar{q}_1 skall vara lika.

2c) Det mer avslappnade villkoret innebär matematiskt att $\frac{d}{du} \bar{p}(1) = k \frac{d}{du} \bar{q}(0)$, där k är en godtycklig konstant $k > 0$. Med härledningen ovan ger detta villkoret $\bar{p}_3 - \bar{p}_2 = k(\bar{q}_1 - \bar{q}_0)$, vilket i kombination med villkoret från a) visuellt innebär att punkterna $\bar{p}_2, \bar{p}_3 = \bar{q}_0$ och \bar{q}_1 fortfarande skall ligga i denna ordning längs en rät linje, men att avstånden \bar{p}_2 till \bar{p}_3 och \bar{q}_0 till \bar{q}_1 inte behöver vara lika.

3a) Man skulle i stället kunna använda sig av *texturmapping*, en vanlig standardmetod för att rita mönstrade 3D-objekt. En fördel med den valda metoden är dock att *mönstret alltid är skarpt* oavsett hur nära man zoomar in på objektet. En textur blir däremot antingen suddig eller "pixlig" när man förstorar upp den. Den valda metoden ger dessutom en *enklare representation* av Legobitarna eftersom man endast behöver lagra geometriska data, inga bilder. Renderingen blir förmodligen inte snabbare utan texturer, däremot kan den vara *enklare att programmera*, och om mönstret inte är väldigt komplicerat så behövs faktiskt *mindre data* för några extra polygoner än vad som skulle ha behövts för en högupplöst texturbild.

3b) Insikt: Silhuetten av objektet utgörs av de kanter som på sin ena sida har en triangel som är synlig, och på sin andra sida har en triangel som inte är synlig. Vi kan alltså räkna på i princip samma sätt som för "back face culling", och rita ut endast de kanter som delas mellan en synlig och en icke synlig triangel. För att göra detta behöver vi ytnormaler, som kan beräknas så här:

För alla trianglar:

Bilda en vektor mellan hörnpunkt 1 och 2, samt en mellan hörnpunkt 1 och 3.

Räkna ut triangelns ytnormal genom att ta kryssprodukten mellan dessa två vektorer.

En fullständig algoritm för att rita silhuetten skulle sedan kunna se ut som följer:

För alla "mjuka" kanter:

Beräkna en vyvektor som vektorn mellan en av kantens hörnpunkter och betraktaren.

Sök i triangellistan och hitta de två trianglar som innehåller kanten ifråga.

Beräkna skalärprodukten mellan den ena triangelns ytnormal och vyvektorn.

Beräkna skalärprodukten mellan den andra triangelns ytnormal och vyvektorn.

Om de två skalärprodukterna har olika tecken:

kanten är en del av silhuetten, rita ut den.

Annars:

kanten är inte en del av silhuetten, rita inte ut den.

Klart!

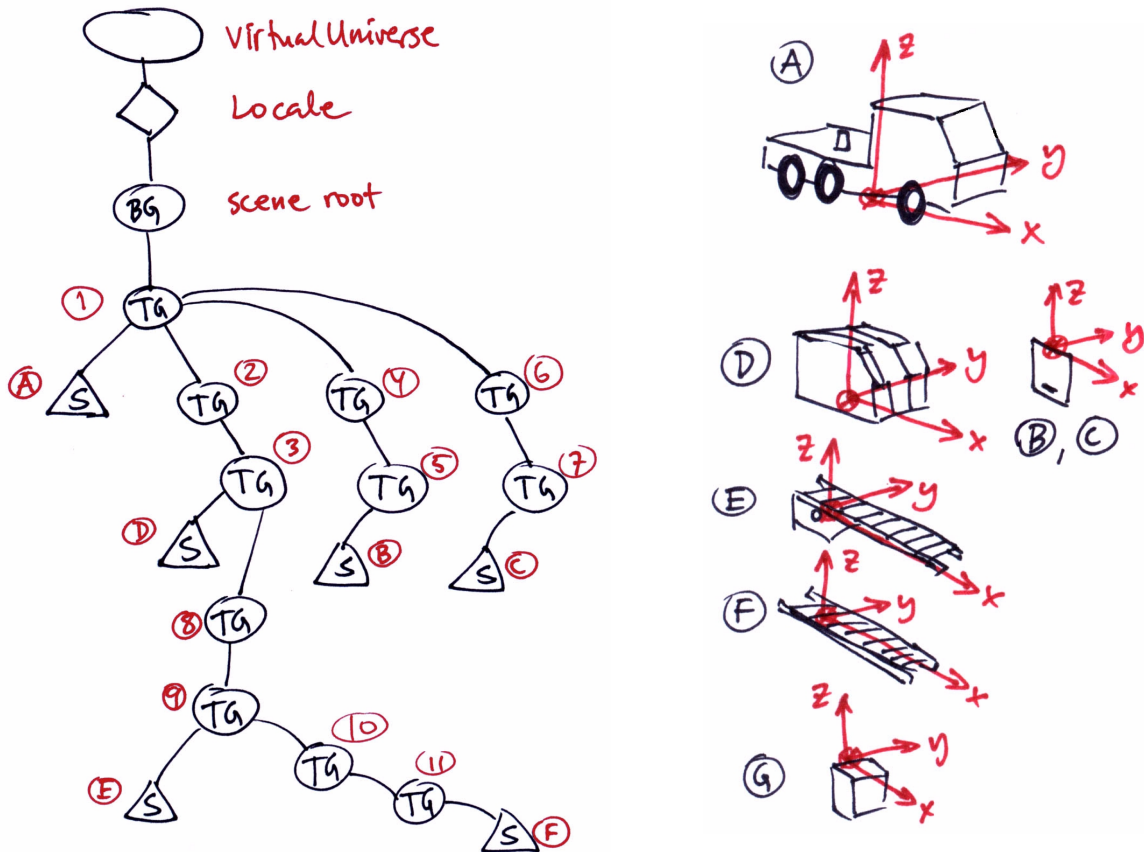
(Uppgift 4 finns på nästa sida)

5a) Raytracing kan tillföra korrekta reflexioner och brytningar mellan speglade och transparenta objekt i en scen, samt geometriska korrekta skuggor.

5b) En fullständig lösning ges inte här, eftersom detta står tydligt förklarat i kursboken. För bästa tydlighet i en översiktlig förklaring bör figurer användas, förslagsvis en bild av en enkel scen med ett eller ett par objekt och några strålars väg, samt gärna en schematisk bild av strålträdet med dess förgreningar. Väsentliga nyckelord i en korrekt och tillräckligt detaljerad förklaring av raytracing är följande: strålarna utgår från ögat, rekursion för nästa nivå, förgrena vid reflexion och brytning (vid träff med genomskinliga objekt). Lokal ljusmodell för ev. diffus reflexion i varje träff. Avsluta rekursionen när ett visst maxdjup uppnåtts eller den kvarvarande intensiteten är tillräckligt låg för att inte ha någon väsentlig inverkan på den slutliga bilden.

5c) En stråle kan reflekteras och brytas många gånger på sin väg genom scenen, och eftersom genomskinliga objekt både reflekterar och bryter ljus förgrenas strålen vid varje träff till två nya strålar. Det kan därför bli väldigt många delstrålar av en enda ursprungsstråle, och det är det som gör att raytracing kan ta så lång tid.

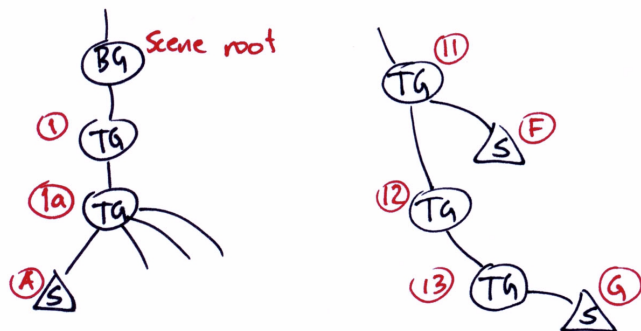
4a) Scengrafens struktur och de aktuella Shape-noderna framgår av figuren nedan.



Transformationsgrupperna har följande roller: 1) Fast transformation för att placera och orientera bilen rätt i scenen. 2) Fast translation för att placera nod D rätt i relation till nod A. 3) Rörlig rotation kring z-axeln för att vrida stegen. 4) Fast translation för att placera nod B rätt i relation till nod A. 5) Rörlig rotation kring x-axeln för att öppna och stänga den ena sidoluckan. 6) Fast translation för att placera nod C rätt i relation till nod A. 7) Rörlig rotation kring x-axeln för att öppna och stänga den andra sidoluckan. 8) Fast translation för att placera nod E rätt i relation till nod D. 9) Rörlig rotation runt y-axeln för att resa upp och fälla ner stegen. 10) Fast translation för att placera nod F rätt i relation till nod E. 11) Rörlig translation i x-led för att dra ut och skjuta ihop stegen.

4b) En rörlig translation i x-led läggs till under TG nummer 1. (1a i figuren till höger).

4c) Ett varv är 2π radianer. Omkretsen för hjulet är $2\pi r$. När det rullar sträckan $2\pi r$ roterar det vinkeln 2π . (Eller enklare: enligt definitionen av en radian roterar hjulet vinkeln 1 när det rullar en sträcka r .) När hjulet rullar en sträcka x roterar det följaktligen vinkeln x/r .



4d) Man lägger till en gren i grafen under TG nummer 11 (se figur ovan). TG nummer 12 är en fast translation som placerar nod G rätt i relation till nod F, och TG nummer 13 är en rörlig rotation kring y-axeln som vrider korgen. Rotationen skall ske med samma vinkel som stegen är upprest (rotationen i TG nummer 9), men med motsatt tecken.