Lösningsförslag till omtentamen

TNM077 3D-datorgrafik och animering

samt TNM008 3D-datorgrafik och VR

2004-08-23 kl 8-12

Uppgift 1 (5 p)

a) Materialparametrarna i ekvationen är k_a, k_d, k_s och n, övriga parametrar har att göra med betraktarens position, objektets geometri och ljuskällornas placering och intensitet. Värdet på k_a bör väljas till samma värde som k_d, något beroende på hur I_a definierats. Om det finns tillräckligt med ljuskällor i scenen för att undvika becksvarta slagskuggor sätts ofta I_a till noll, och då är väret på k_a oväsentligt. En grå färg innebär att den diffusa reflexionen är någonstans mitt emellan 0 och 1, så k_d väljs till ungefär 0,5. Det speglande blänket i ytan skall vara ganska framträdande, vilket innebär att k_s skall väljas högt, förslagsvis nära 1. Eftersom ytan är mycket blank och reflexen av ljuskällan skall vara liten så skall parametern n väljas till ett högt värde, kanske omkring 10-20. (3 p)

b) Om Legobiten i stället skall vara vit skall k_d ökas till ett värde nära 1. Även k_a bör ökas till samma värde. Övriga parametrar förblir oförändrade. (1 p)

c) En helt matt yta har endast diffus reflexion, alltså $k_s = 0$. En halvmatt yta kan alltså modelleras genom att välja ett värde på k_s någonstans mitt emellan 0 och 1. Dessutom kan värdet på n minskas för att göra blänket mindre distinkt. (1 p)

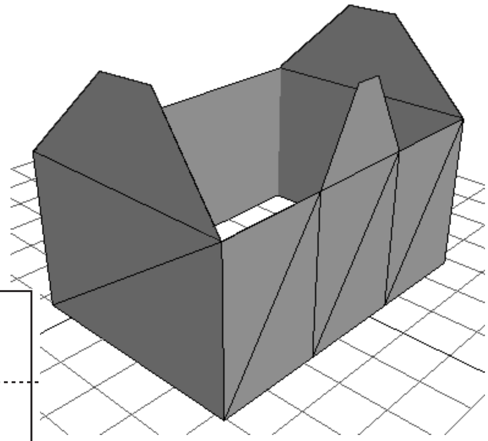
Uppgift 2 (5 p)

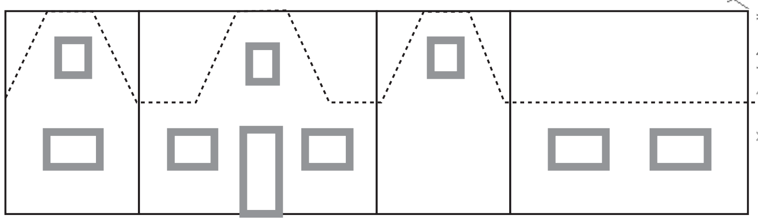
a) I stället för att låta en texturbild bestämma ytans färg så låter man den bestämma ytans genomskinlighet i renderingen, alltså hur mycket av bakomliggande objekt som skall synas genom ytan. Där det skall vara hål har man ett extremvärde (t ex vitt i bilden), där det inte skall vara hål har man ett annat värde (t ex svart). Texturbilden visas till vänster i figuren nedan. (2 p)

b) Man måste skapa en texturbild, vilket i sig är ett extra moment. Texturbilden måste dessutom vara högupplöst för att kunna ge en någorlunda mjukt rundad överkant på fönstret, och även en mycket högupplöst bild kommer att ge en taggig och/eller suddig kant på nära håll (se mitten i figuren nedan). En yta med texturmappade hål är dessutom ett plan utan tjocklek, vilket ser illa ut på nära håll. Om hålet hade gjorts med en polygonmodell så hade även innerkanten av fönstret kunnat modelleras korrekt, något som inte låter sig göras enkelt med den beskrivna texturmappningen (se till höger i figuren). (3 p)

Uppgift 3 (7 p)

a) Fasaderna modelleras lämpligen som enklast möjliga plana ytor, med en textur som bestämmer färgen för att ge ett intryck av detaljer. Texturbilden kan exempelvis tas från foton av den riktiga modellen. Alla ytor är av blank plast med i stort sett samma speglande egenskaper, så endast den diffusa färgen behöver moduleras med texturbilden. Modellen och texturbildens utseende skissas i figurerna till höger och nedan. (3 p)



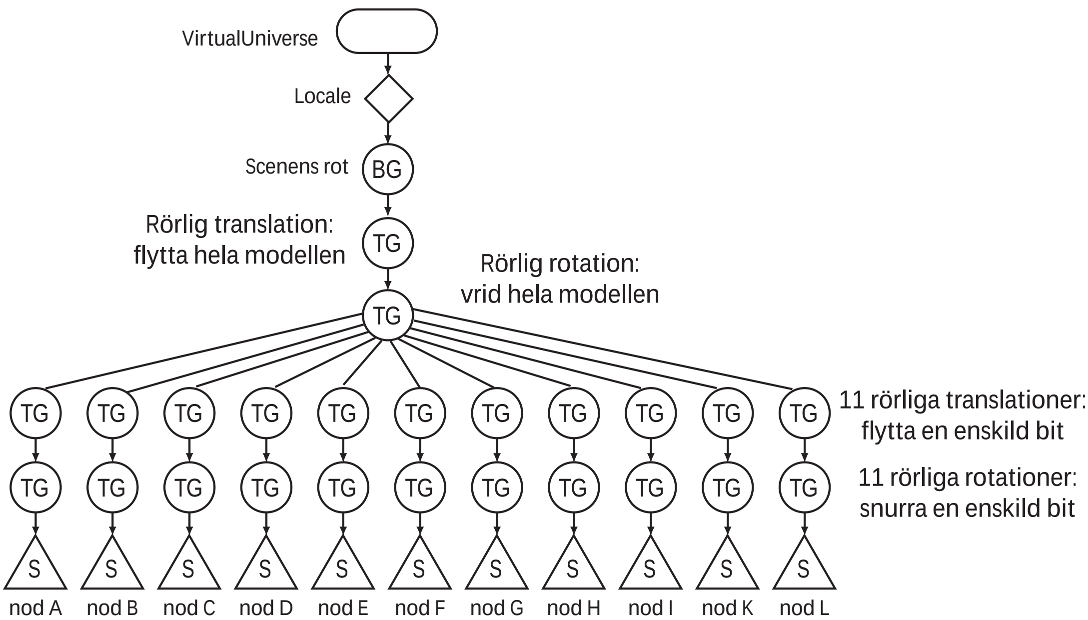


b) På nära håll kan man använda en så kallad *bump map* för att göra ytan lagom skrovlig. Materialegenskaperna i övrigt skall då inte förändras (1 p). På långt håll räcker det med att göra det speglande blänket i ytan mindre distinkt genom att minska parametern n i Phongs modell (1 p).

c) En bildbaserad textur lagras som en pixelbild. Den är enkel och snabb att hantera och att skapa utifrån fotografiskt material, men den har en begränsad upplösning och kräver minne för lagring och rendering. En procedurell textur beräknas utifrån en matematisk modell i den upplösning som behövs för den aktuella situationen. Texturen är litet krångligare att skapa, och den kräver en del beräkningskraft vid renderingen, men bilden kan ha en godtyckligt hög upplösning utan att ta upp mer än marginellt med plats i minnet. (2 p)

Uppgift 4 (5 p)

Varje bit skall kunna flyttas och roteras en i taget, det vill säga oberoende av varandra, och måste alltså ha egna transformationer. Hela samligen av bitar skall sedan kunna flyttas och roteras tillsammans, vilket innebär att de alla dessutom måste vara barn till en och samma överordnade transformation. Att “en bit snurrar” antas vara liktydigt med att den roterar runt sitt lokala origo. (Om ett annat rotationscentrum önskas behövs en ytterligare transformationsgrupp med en fast translation närmast Shape-noden.) Scengrafen kommer att se ut som följer: (5 p)



Uppgift 5 (8 p)

a) Direkt insättning och utveckling ger:

$$\bar{p}(u) = (1-u)^3\bar{p}_0 + 3u(1-u)^2\bar{p}_1 + 3u^2(1-u)\bar{p}_2 + u^3\bar{p}_3$$

Två dimensioner innebär att $\bar{p}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}$ och $\bar{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$. Tecknat för x och y separat får vi:

$$x(u) = (1-u)^3 x_0 + 3u(1-u)^2 x_1 + 3u^2(1-u)x_2 + u^3 x_3$$

$$y(u) = (1-u)^3 y_0 + 3u(1-u)^2 y_1 + 3u^2(1-u)y_2 + u^3 y_3 \text{ (1 p)}$$

b) Kontrollpunkterna $\bar{p}_i$ är konstanter, endast $B_i(u)$ beror av u. Utveckling ger:

$$\bar{v}(u) = \frac{d}{du}\bar{p}(u) = \frac{d}{du}\sum_{i=0}^{3} B_i(u)\bar{p}_i = \sum_{i=0}^{3}\left(\frac{d}{du}B_i(u)\right)\bar{p}_i = \sum_{i=0}^{3} B_i'(u)\bar{p}_i \text{, derivatorna } B'_i(u) \text{ är:}$$

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

$x(u) = (-3 + 6u - 3u^2)x_0 + (3 - 12u + 9u^2)x_1 + (6u - 9u^2)x_2 + 3u^2 x_3$, p.s.s. för y, y_i (3 p).

c) Om $\bar{v} = (v_x, v_y)$ så ges vinkeln av $\varphi = \operatorname{atan}(v_y / v_x)$. (1 p)

Eftersom vektorn kan peka åt vilket håll som helst måste man egentligen även ta hänsyn till tecknen hos täljare och nämnare, samt hantera fallet $v_x = 0$. *Detta görs enkelt med funktionen* $\varphi = \operatorname{atan2}(v_x, v_y)$ *i Java och de flesta andra programmeringsspråk.*

d) Accelerationsvektorn $\bar{a}(u)$ beräknas genom ytterligare en derivering med avseende på u:

$$\bar{a}(u) = \sum_{i=0}^{3} B_i''(u)\bar{p}_i, \quad \begin{array}{ll} B_0''(u) = 6 - 6u & B_1''(u) = -12 + 18u \\ B_2''(u) = 6 - 18u & B_3''(u) = 6u \end{array}$$

En vektor i körriktningen är den tidigare beräknade $\bar{v}(u)$. Normera till enhetsvektor $\hat{v} = \bar{v}/|\bar{v}|$. Den del av $\bar{a}$ som är parallell med $\bar{v}$ ges av $\bar{a}_{||} = (\bar{a} \bullet \hat{v})\hat{v}$, den del som är ortogonal mot $\bar{v}$ ges av $\bar{a}_{\perp} = \bar{a} - \bar{a}_{||}$. Sidoaccelerationens storlek är denna vektors belopp $|\bar{a}_{\perp}|$. (3 p)

För att veta åt vilket håll bilen skall luta måste man dessutom veta vektorns riktning relativt körriktningen (om den pekar "åt höger" eller "åt vänster"). Detta kan beräknas med en enkel kryssprodukt.

Ansätt $\bar{a}_{\perp} = \begin{bmatrix} a_x \\ a_y \\ 0 \end{bmatrix}$, $\bar{v} = \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix}$, *beräkna* $\bar{a}_{\perp} \times \bar{v} = \begin{bmatrix} 0 \\ 0 \\ a_x v_y - a_y v_x \end{bmatrix}$.

Om z-komponenten i denna vektor är positiv accelererar bilen åt höger, annars åt vänster.