

Tentamen EDA 425 Avancerad Datorgrafik

Plats: M-huset

6 uppgifter, max en uppgift per blad.

Ansvarig: Tomas Akenine-Möller, 031-772 5219

2003-05-28, 14:15-18:15

Uppgift 1:Grafikpipeline:n

- a) [2p] Pipelinen för polygonrendering (real-time rendering) är ofta uppdelad i tre konceptuella steg. Vilka, och i vilken ordning exekveras dom?
- b) [2p] Resonera dig fram till vilket av dom två sista stegen ovan som är mest beräkningsintensivt. Redovisa ditt resonemang.
- c) [3p] Vertex shaders och pixel shaders har revolutionerat grafikpipeline:n dom senaste åren. Vad är poängen med dessa två?
- d) [3p] Belysning beräknas ofta i triangelhörnen och interpoleras sedan över triangeln. Belysningsberäkningarna består (i realtidsgrafik) ofta av tre komponenter. Vilka, och vad försöker dom approximera?

Uppgift 2: Ray tracing och global belysning

- a) [2p] I ray tracing används ofta jitterad (eng. jittered) sampling. Hur funkar detta, och varför är det bättre än supersampling (med lika många samples)?
- b) [3p] Tänk dig en swimmingpool, och att du kastat en sten ner i denna som skapat vågor på ytan. Solens strålar bryts i ytan och skapar ett vackert mönster (kallas caustics på engelska) på botten. Klarar vanlig ray tracing av att skapa bilder av caustics? Varför/varför inte?
- c) [3p] Många "global-belysningsalgoritmer" (eng. global illumination) bygger på användandet av Monte Carlo integrering. Förklara principen för detta och hur standardavvikelsen varierar.
- d) [2p] I ljustransportnotation (eng. light transport notation), beskriv vad en global belysningsalgoritm idealt skall klara av, och förklara med ord vad det betyder.

Uppgift 3: Blandade frågor

- a) [3p] Nämn tre ställen där vikning (eng. aliasing) uppkommer i datorgrafik.
- b) [2p] Vad är hierarkisk view frustum culling?
- c) [2p] Antag att man redan har implementerat hierarkisk view frustum culling. Nämn två typer av algoritmer (som du kan implementera i mjukvara) som ytterligare kan förbättra prestanda, och förklara vilka delar i pipeline:n som får mindre att göra.
- d) [3p] Beskriv trilinear mipmapping.

Uppgift 4: Intersektionsberäkningar

- a) [4p] Visa hur man härleder skärningspunkterna mellan en stråle och en cylinder. Förklara när strålen inte träffar cylindern, och när den träffar så visa var (mha av härledd formel).
- b) [3p] Förklara hur separerande axel teoremet (eng. separating axis theorem) används när man vill ta reda på om två tetraedra krockar med varandra.
- c) [3p] Se figur 1. Vid tidpunkten $t = 0$ så finns det en sfär med radien r , och origo placerat i \mathbf{c} . Vid tidpunkten $t = 1$ så har den flyttat sig till \mathbf{e} . Härled en formel för tidpunkten då sfären kolliderar med planet, och ange också villkor för när sfären *inte* kolliderar med planet.

Uppgift 5: Skuggor

- a) [2p] Beskriv kort vad tanken bakom "shadow volume"-algoritmen och "shadow mapping"-algoritmen är, dvs hur skuggan uppkommer i algoritmen.
- b) [4p] Beskriv standardimplementationen av "shadow volume"-algoritmen i detalj.
- c) [3p] Hur kan man ändra på standardimplementationen (uppgift b) så att den automatiskt kan hantera när kameran är inuti skugga?
- d) [1p] Mjuka skuggor: tänk på hur man gör mjuka skuggor i "ray tracing"-baserade algoritmer, och hur man skulle kunna generera mjuka skuggor med hjälp av grafikhårdvara. Vad är den fundamentala skillnaden (och då menar jag inte skillnaden i hårdvara som används)?

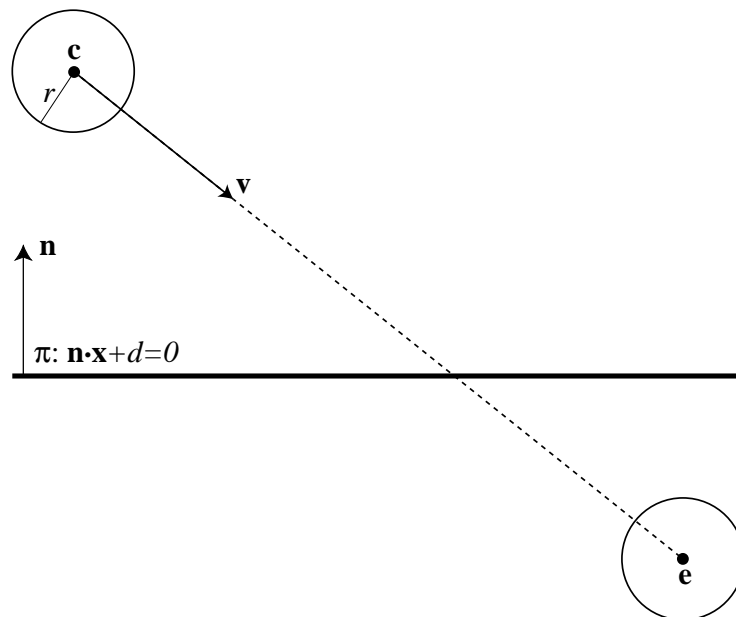


Figure 1: Geometri för uppgift 4c.

Uppgift 6: Grafikhårdvara

a) [3p] Förklara hur perspektiv-korrekt texturering går till. Utgå från ett triangelhorn, \mathbf{v} , och anta att man räknar fram $\mathbf{p} = \mathbf{M}\mathbf{v} = (p_x w, p_y w, p_z w, w)$, där \mathbf{M} är modell-, kamera-, och projektionsmatris sammanslaget. Varje triangel beskrivs av tre sådan hörn, tex \mathbf{v}_0 , \mathbf{v}_1 , och \mathbf{v}_2 .

b) [3p] Antag att du är hårdvarudesigner och att du har råd med 9.6 Gbytes/s i minnesbandbredd (om du vill ha mer blir systemet för dyrt). Vidare vill en tänkt beställare av din hårdvara att den skall klara av så pass komplexa 3D-scener så att 18 Gbytes/s används (givet en naiv hårdvaruimplementation). Nämn tre algoritmer som du kan implementera i hårdvara som minskar på bandbreddskraven, men bibehåller komplexiteten på 3D-scenerna. Förklara också kort varför dom ger minskad bandbredd till minnet.

c) [4p] Förklara hur KYRO arkitekturen fungerar, och vad som skiljer den från standardhårdvara (tex XBOX). Vilka fördelar och nackdelar finns med KYRO?