

## Toets 2 Graphics

Vrijdag 17 Oktober 2003, 11:00 – 13:00

- Deze toets bestaat uit zes opgaven met in totaal elf sub-opgaven, verdeeld over drie pagina's.
- Het maximaal aantal te behalen punten is 10.5. Het cijfer voor deze toets is  $\min(10, \text{aantal punten})$ .
- Bij deze toets mogen boek, aantekeningen en rekenmachines *niet* gebruikt worden.
- Schrijf op ieder antwoordvel je naam en collegekaartnummer.

### 1 Ray Tracing

We gaan ray tracing doen, waarbij we een straal vanuit het COP door het middelpunt van een pixel schieten. Hierbij raken we een object  $O$  in het punt  $p$ .

- (a) [1 pt] Wat zijn in dit verband *shadow feelers*, en waarom zijn ze relevant voor ray tracing?

### 2 Positieve en Negatieve Halfruimten

Zowel bij clipping met het algoritme van Sutherland-Hodgman als bij het bouwen en gebruiken van BSP-bomen moeten we kunnen bepalen aan welke kant van een lijn of vlak een gegeven punt ligt.

We beschouwen nu een lijn  $\ell$  in  $\mathbb{R}^2$  met normaalvector  $n = \begin{pmatrix} a \\ b \end{pmatrix}$ . De vergelijking van deze lijn is dus  $ax_1 + bx_2 = c$ , voor een zekere waarde van  $c$ . We zeggen dat een punt  $x$  aan de *positieve* kant van  $\ell$  ligt als  $x$  in het door  $\ell$  gedefinieerde halfvlak ligt waar de normaal  $n$  naartoe wijst; het andere halfvlak wordt de *negatieve* kant genoemd.

- (a) [0.5 pt] Geef een functie  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  zodanig dat

- $f(x_1, x_2) > 0$  als het punt  $x = (x_1, x_2)$  aan de positieve kant van  $\ell$  ligt;
- $f(x_1, x_2) = 0$  als  $x$  op  $\ell$  ligt;
- $f(x_1, x_2) < 0$  als  $x$  aan de negatieve kant van  $\ell$  ligt;

- (b) [1.5 pt] Laat zien dat voor een willekeurig punt  $x$  aan de positieve kant van  $\ell$  geldt dat  $f(x_1, x_2)$  inderdaad groter dan nul is voor de functie die je bij vraag (a) hebt gegeven.

### 3 Projectie

Om een polygonale scene af te beelden op een plaatje kunnen we, in plaats van reken-intensieve methodes als ray tracing te gebruiken, ook projectie toepassen. Daarbij transformeren we de scene van *World Coordinates* via *Viewing Coordinates* naar het *Canonical Perspective Frame*. Vervolgens doen we clipping, en projecteren we op de rechthoek  $[-1, 1] \times [-1, 1]$ . Tenslotte beelden we deze rechthoek af op het uiteindelijke  $M \times N$ -plaatje.

Projectie op  $[-1, 1] \times [-1, 1]$  kunnen we impliciet doen door vanuit het *Canonical Perspective Frame* te transformeren naar *Canonical Projection Space*. Dit doen we met de volgende matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Wanneer we het punt  $(20, 30, 9)$  in het *Canonical Perspective Frame* met behulp van bovenstaande matrix transformeren, krijgen we (in homogene representatie) het punt  $(20, 30, 9, 10)$ .

- (a) [1 pt] Wat is het corresponderende *Euclidische 3D-punt* voor het punt  $(20, 30, 9, 10)$  in *Canonical Projection Space*?
- (b) [1 pt] Hoe heet de projectie die we uitvoeren om de scene in *Canonical Projection Space* af te beelden op  $[-1, 1] \times [-1, 1]$ ?

### 4 Back Face Elimination

- (a) [0.5 pt] Wat is *back face elimination*?
- (b) [1.5 pt] We hebben een convex polyhedron  $P$  in 3D, waarvan het oppervlak uit  $n$  driehoeken bestaat. We specificeren een camera zodanig dat het gehele polyhedron binnen het view volume valt. We willen het polyhedron renderen m.b.v. projectie. Wanneer we geen back face elimination toepassen projecteren we dus  $n$  driehoeken.

Hoeveel driehoeken projecteren we *minimaal* bij toepassing van back face elimination, en hoeveel *maximaal*?

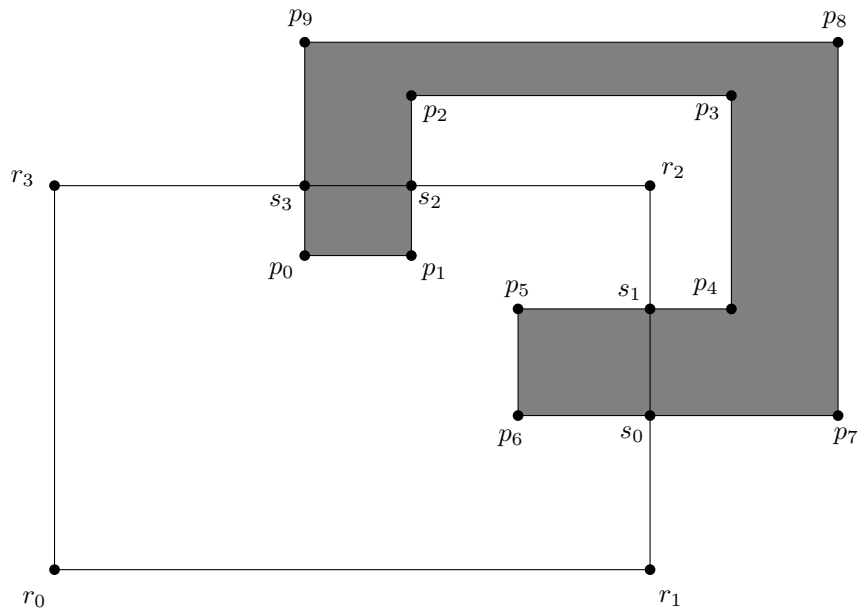
### 5 BSP-bomen

We willen een BSP-boom gebruiken om de projectie-volgorde van de polygonen in een scene te bepalen.

- (a) [0.5 pt] Leg uit hoe de *constructie* van de BSP-boom afhangt van het COP van de camera.
- (b) [0.5 pt] Is het voor dit doeleinde van belang dat de BSP-boom gebalanceerd is? Licht je antwoord toe.
- (c) [1.5 pt] Leg in detail uit hoe BSP-bomen bij ray tracing gebruikt kunnen worden (het bouwen van de boom hoeft je niet uit te leggen, het gebruik wel).

## 6 Clipping

We hebben een polygon  $P = [p_0, p_1, \dots, p_9]$  en een clipping region  $R = [r_0, r_1, r_2, r_3]$ . De snijpunten van  $P$  en  $R$  zijn  $s_0, s_1, s_2, s_3$ , zoals geïllustreerd in onderstaand plaatje:



Wanneer we het Weiler-Atherton-algoritme gebruiken om  $P$  te clippen tegen  $R$  zijn de resulterende polygonen  $[p_0, p_1, s_2, s_3]$  en  $[p_5, p_6, s_0, s_1]$ .

(a) [1 pt] Wat zijn de resulterende polygonen als we clippen met het Sutherland-Hodgman-algoritme?