

Graphics 2013

1. Docenten: **José Lagerberg en Robert Belleman**
2. Assistenten: Koen Koning en Sultan Shabaz
3. Cursusinformatie: <http://www.science.uva.nl/~jose/>
4. Boek:

Fundamentals of Computer Graphics,
Peter Shirley ea
AK Peters, 2005, 2de of 3de uitgave

Waar en wanneer Graphics

1. College Graphics in eerste 3 weken

maandag 15.00-17.00

woensdag 11.00-13.00

2. Practicum Graphics

dinsdag, woensdag en vrijdag

3. Inleiding OpenGL

dinsdag 5 februari om 13.00 uur, zaal G0.10-G0.12 door
Robert Belleman

Afsluiting Graphics eerste deel

1. **Docent** José Lagerberg
2. **Stof** over Graphics H3, H6, H7, H9 en H10
3. **Tussentoets** in 4de week op woensdag 27 februari
(moet voldoende zijn (> 5.5))
4. **Practicum** practicumopgaven inleveren voor deadline

Afsluiting Graphics tweede deel

1. **Docent** Robert Belleman
2. **Stof** later precies opgegeven
3. **Tentamen** in 8ste week op woensdag 27 maart
(moet voldoende zijn (> 5.5))
4. **Practicum** practicumopgaven inleveren voor deadline

Toetsing

1. Het **schriftelijk tentamen** levert een cijfer **T** op een schaal van 10
2. Het **practicum** levert een cijfer **P** op schaal van 10
3. Het **eindcijfer** voor elk onderdeel is dan: **$2T/3 + P/3$**

Computer graphics

1. **Modelling**: creëren van 3D virtuele wereld
2. **Rendering**: creëren van 2D beelden van 3D modellen

Twee basis methoden:

1. Projectieve methoden
2. Ray tracing

Grafische pijplijn is reeks van transformaties die 3D modellen omzetten in pixels op scherm

Raster algoritmen (hoofdstuk 3)

1. Raster displays
2. Rasterizatie van lijnen
3. **Barycentrische** coördinaten
4. Rasterizatie van driehoek
5. **antialiasing**

Raster displays

Raster display is TV of computer monitor die **raster scan** methode gebruikt om beeld te genereren op scherm

- raster display is array van pixels
- pixel is kleinste eenheid op scherm
- scherm opgebouwd uit punten
- **intensiteit** van elke pixel is variabel

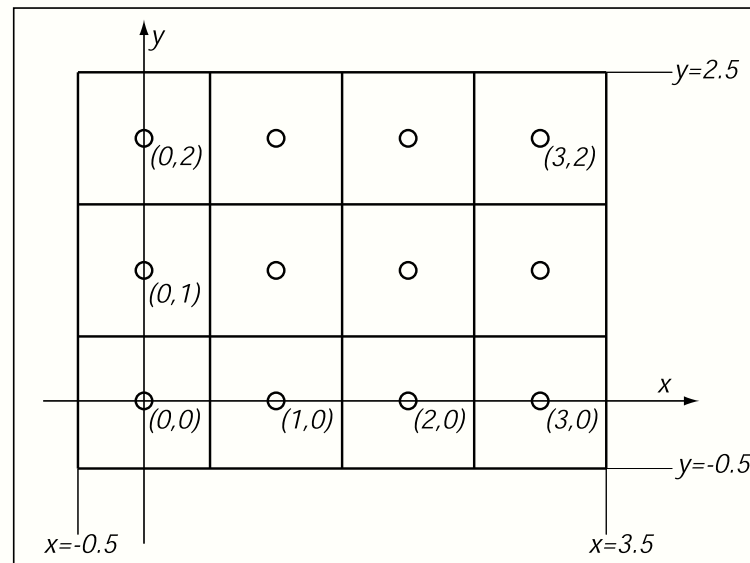
Van 2D model naar pixels op scherm is **rasterizatie**

Scherf coördinaten

Pixels aangegeven door indices voor kolom en rij (i, j)

beeld bevat n_x kolommen en n_y rijen

pixel links onder $(0, 0)$, **pixel** rechts boven $(n_x - 1, n_y - 1)$



Rasterizatie van lijnen

Lijnen tussen twee eindpunten (x_0, y_0) en (x_1, y_1)

x_0, y_0, x_1, y_1 vaak **integers** (pixel centra)

Tekenen gebaseerd op vergelijking lijn, twee typen:

1. **impliciete** vergelijking van lijn
2. **parameter** voorstelling van lijn (doen we later)

Vergelijking van lijn

Expliciete vergelijking lijn: $y = mx + b$

Voorbeeld

1. $y = 2x + 3$

Vraag: wat is m voor $x = 0$?

Impliciete vergelijking lijn: $f(x, y) = Ax + By + C = 0$

Vraag: wat is richtingscoëfficiënt m ?

Voorbeeld

1. Lijn door oorsprong met hoek 45° : $x - y = 0$

2. Lijn door oorsprong en $(2, 3)$: $3x - 2y = 0$

Vergelijking van lijn door (x_0, y_0) en (x_1, y_1)

Vraag: Wat is impliciete vergelijking lijn door (x_0, y_0) en (x_1, y_1) ?

Richtingsvector $\begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix}$, normaalvector $\begin{pmatrix} y_0 - y_1 \\ x_1 - x_0 \end{pmatrix}$

Impliciete vergelijking lijn is

$$f(x, y) = (y_0 - y_1)x + (x_1 - x_0)y + C = 0$$

Bepalen van C door (x_0, y_0) in te vullen:

$$f(x_0, y_0) = (y_0 - y_1)x_0 + (x_1 - x_0)y_0 + C = 0$$

$$C = -x_0y_0 + x_0y_1 - x_1y_0 + x_0y_0 = x_0y_1 - x_1y_0$$

Vergelijking van lijn door (x_0, y_0) en (x_1, y_1)

$$f(x, y) = (y_0 - y_1)x + (x_1 - x_0)y + x_0y_1 - x_1y_0 = 0$$

Vraag: Gaat deze lijn door (x_0, y_0) en (x_1, y_1) ?

Richtingscoëfficiënt: $m = \frac{y_1 - y_0}{x_1 - x_0}$

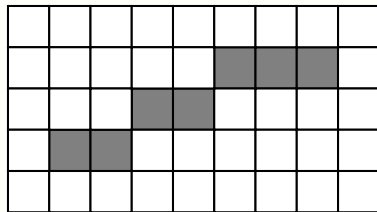
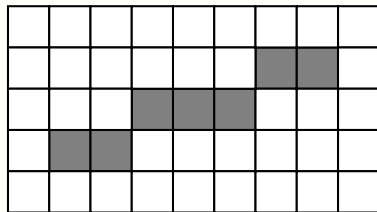
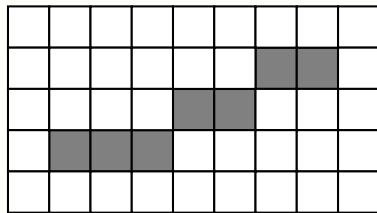
Deze vergelijking gebruiken bij tekenen pixels op scherm

Opgave

Bepaal vergelijking van lijn door $(2, 3)$ en $(3, 1)$

Lijnen zo dun mogelijk

Probleem: gegeven begin- en eindpunt lijn, teken zo'n dun mogelijke lijn. Welke pixels moeten getekend worden?



$$m \in (0, 1]$$

Begin links, teken pixel op dezelfde hoogte of één hoger

Precies **één pixel** per kolom

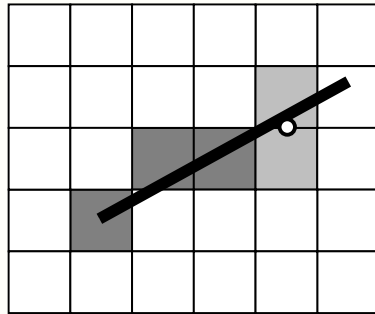
Basisvorm van Middelpunt algoritme

```
y = y0
for x = x0 to x1 do
  draw(x, y)
  if (some condition) then
    y = y + 1
```

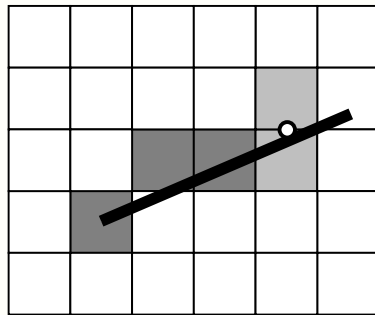
x en y zijn integers

Het gaat om het **if**-statement

Middelpunt $(x, y + 0.5)$ tussen 2 potentiële pixels



Lijn **boven** middelpunt:
top pixel getekend



Lijn **onder** middelpunt:
bottom pixel getekend

Positief boven of onder lijn?

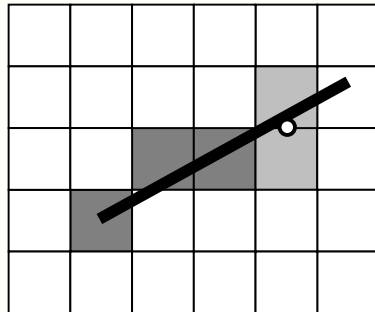
$$f(x, y) = (y_0 - y_1)x + (x_1 - x_0)y + x_0y_1 - x_1y_0 = 0$$

Bepalende term is $(x_1 - x_0)y$

1. $(x_1 - x_0) > 0$
2. $(x_1 - x_0)y$ wordt groter als y groter
3. dus $f(x, +\infty) > 0$

Punten boven lijn zijn **positief**

$f(x, y + 0.5) < 0$ **of** $f(x, y + 0.5) > 0$?

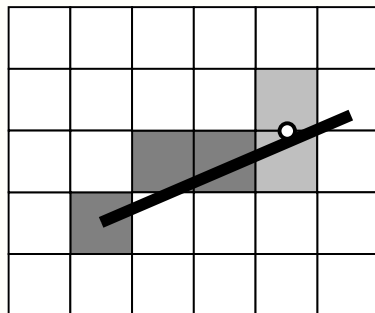


Lijn **boven** middelpunt:

$$f(x, y + 0.5) < 0$$

top pixel getekend

$$y = y + 1$$



Lijn **onder** middelpunt:

$$f(x, y + 0.5) > 0$$

bottom pixel getekend

Middelpunt algoritme

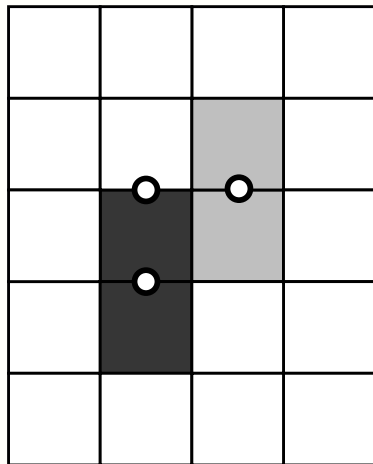
```
y = y0
for x = x0 to x1 do
    draw(x, y)
    if (f(x + 1, y + 0.5) < 0) then
        y = y + 1
```

Geldt alleen voor lijnen met **m** $\in (0, 1]$

Incrementele methode middelpunt algoritme

Incrementele methode hergebruikt berekeningen in loop

Meeste rekenwerk bepalen $f(x + 1, y + 0.5)$



Vorige berekening was:

$$f(x, y - 0.5) \text{ of}$$

$$f(x, y + 0.5)$$

Berekening nieuwe $f(x + 1, y + 0.5)$

$$f(x, y) = (y_0 - y_1)x + (x_1 - x_0)y + x_0y_1 - x_1y_0 = 0$$

Berekening van **f** voor twee pixels in kolom:

$$f(x + 1, y) = f(x, y) + (y_0 - y_1)$$

$$f(x + 1, y + 1) = f(x, y) + (y_0 - y_1) + (x_1 - x_0)$$

Daaruit volgt:

$$1. f(x + 1, y + 0.5) = f(x, y + 0.5) + (y_0 - y_1)$$

$$2. f(x + 1, y + 0.5) = f(x, y - 0.5) + (y_0 - y_1) + (x_1 - x_0)$$

Beslissingsvariabele **d**

Definieer: **d** = $f(x + 1, y + 0.5)$

In **if**-statement beslissingsvariabele **d**:

- startwaarde $y = y_0$
startwaarde **d** = $f(x_0 + 1, y_0 + 0.5)$
- als **d** < 0, **d**_{new} = **d**_{old} + $(x_1 - x_0) + (y_0 - y_1)$
- als **d** ≥ 0, **d**_{new} = **d**_{old} + $(y_0 - y_1)$

Incrementeel middelpunt algoritme

```
y = y0
d = f(x0 + 1, y0 + 0.5)
for x = x0 to x1 do
    draw(x, y)
    if (d < 0) then
        y = y + 1
        d = d + (x1 - x0) + (y0 - y1)
    else
        d = d + (y0 - y1)
```


Incrementeel integer algoritme

Algoritme gebruikt bijna alleen maar **integers**, behalve voor startwaarde van **d** = $f(x_0 + 1, y_0 + 0.5)$

Zelfs **d** kan **integer** gemaakt worden (zie boek)

Middelpunt algoritme meest efficiënt als:

1. incrementeel
2. alleen integer operaties

Barycentrische coördinaten

Barycentrische coördinaten gebaseerd op aantal punten

Veel gebruikt bij Graphics, bijvoorbeeld bij:

- driehoek inclusie test:
ligt punt in driehoek?
- lineaire interpolatie over driehoek:
gegeven 3 hoekpuntkleuren, bepaal kleuren van punt binnen driehoek

Parametervoorstelling van lijn

Gegeven: lijn door twee punten P_0 en P_1

Gevraagd: punt P **op** lijn

Parametervoorstelling van lijn: $P = P_0 + \lambda(P_1 - P_0)$

Anders geschreven: $P = (1 - \lambda)P_0 + \lambda P_1$

Als **som gewichten = 1** ligt punt P op **lijn**

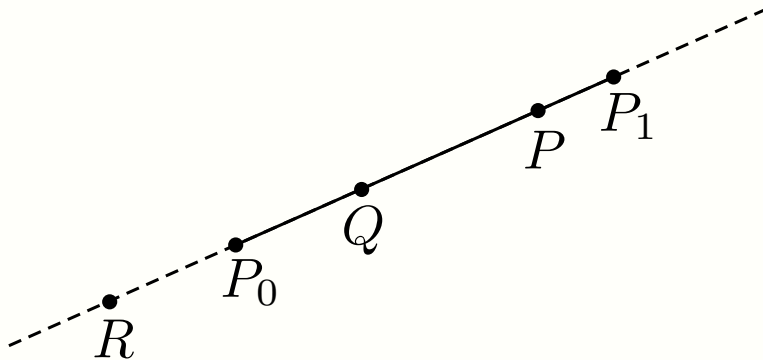
Punten tussen P_0 en P_1 als $0 \leq \lambda \leq 1$

Baryc. coördinaten van P t.o.v. 2 punten

Punt **op lijn** door P_0 en P_1 verkregen door:

$$P = \alpha P_0 + \beta P_1 \text{ met } \alpha + \beta = 1$$

- als $0 \leq \alpha, \beta \leq 1$ dan P **op lijn**segment
- anders erbuiten



$$P = 0.25P_0 + 0.75P_1$$

$$Q = 0.66P_0 + 0.34P_1$$

$$R = 1.33P_0 - 0.33P_1$$

Barycentrische coördinaten van P t.o.v. 3 punten in 2D en 3D

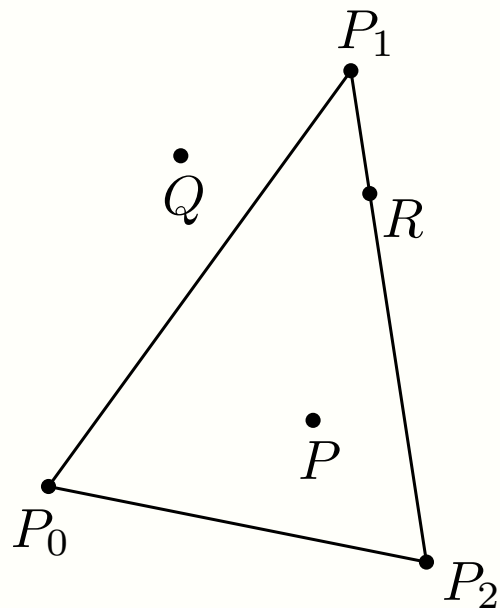
De **barycentrische coördinaten** van punt P t.o.v. de punten P_0 , P_1 en P_2 zijn de getallen α, β, γ zó dat $P = \alpha P_0 + \beta P_1 + \gamma P_2$ met de beperking $\alpha + \beta + \gamma = 1$

Barycentrische coördinaten gedefinieerd voor alle punten **vlak**

- als $0 < \alpha, \beta, \gamma < 1$ dan P **in driehoek**
- als één van weegfactoren gelijk nul, dan op rand
- anders erbuiten

Barycentrische coördinaten van punt t.o.v. 3 punten

$$P = \alpha P_0 + \beta P_1 + \gamma P_2 \text{ met } \alpha + \beta + \gamma = 1 \text{ in vlak}$$



in driehoek, nl. $0 < \alpha, \beta, \gamma < 1$

$$P = 0.25P_0 + 0.25P_1 + 0.5P_2$$

buiten driehoek

$$Q = 0.5P_0 + 0.75P_1 - 0.25P_2$$

op rand driehoek

$$R = 0P_0 + 0.75P_1 + 0.25P_2$$

Berekening barycentrische coördinaten in 2D

Gegeven: P_0 , P_1 en P_2

Bereken: **barycentrische** coördinaten (β, γ) van P ($\alpha = 1 - \beta - \gamma$)

$$P = P_0 + \beta(P_1 - P_0) + \gamma(P_2 - P_0) \Rightarrow$$

$$\beta(P_1 - P_0) + \gamma(P_2 - P_0) = P - P_0$$

$$\beta \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} + \gamma \begin{pmatrix} x_2 - x_0 \\ y_2 - y_0 \end{pmatrix} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

2 vergelijkingen met 2 onbekenden:

$$\begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

Berekening barycentrische coördinaten

$$\begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

Los op: $A\mathbf{x} = b$

1. **bekend:** A 2×2 matrix met coördinaten 3 gegeven punten
2. **bekend:** b vector met coördinaten punt P en P_0
3. **onbekende:** \mathbf{x} met β en γ

Berekening barycentrische coördinaten

Los op: $A\mathbf{x} = b$ met $\mathbf{x} = (\beta, \gamma)$ en $\alpha = 1 - \beta - \gamma$

Verskillende manieren:

1. Inverteren matrix A : $\mathbf{x} = A^{-1}b$
2. Regel van Cramer m.b.v. determinanten
3. **Geometrische** oplossing: $\beta = \frac{f_{20}(x,y)}{f_{20}(x_1,y_1)}$, $\gamma = \frac{f_{01}(x,y)}{f_{01}(x_2,y_2)}$ met

$$f_{20}(x, y) = (y_2 - y_0)x + (x_0 - x_2)y + x_2y_0 - x_0y_2$$

$$f_{01}(x, y) = (y_0 - y_1)x + (x_1 - x_0)y + x_0y_1 - x_1y_0$$

Toepassing 1: ligt punt in driehoek?

Gegeven: punt $P = (4, 4)$ en driehoek door $P_0 = (4, 6)$, $P_1 = (2, 1)$ en $P_2 = (6, 3)$

Gevraagd: Ligt P in driehoek?

Bereken **barycentrische** coördinaten van P t.o.v. P_0 , P_1 en P_2 :

$$P = P_0 + \beta(P_1 - P_0) + \gamma(P_2 - P_0)$$

$$\begin{pmatrix} 4 \\ 4 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} + \beta \begin{pmatrix} -2 \\ -5 \end{pmatrix} + \gamma \begin{pmatrix} 2 \\ -3 \end{pmatrix}$$

$$\beta \begin{pmatrix} -2 \\ -5 \end{pmatrix} + \gamma \begin{pmatrix} 2 \\ -3 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix} \Rightarrow \begin{pmatrix} -2 & 2 \\ -5 & -3 \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \end{pmatrix}$$

$$\begin{pmatrix} \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} -\frac{3}{16} & -\frac{1}{8} \\ \frac{5}{16} & -\frac{1}{8} \end{pmatrix} \begin{pmatrix} 0 \\ -2 \end{pmatrix} \Rightarrow \beta = 0.25 \text{ en } \gamma = 0.25 \Rightarrow \alpha = 0.5$$

$$P = 0.5P_0 + 0.25P_1 + 0.25P_2$$

Som weegfactoren is 1 en alle tussen 0 en 1 \Rightarrow **in driehoek**

Opmerking:

$$\text{Inverse van } \begin{pmatrix} a & b \\ c & d \end{pmatrix} \text{ is } \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Toepassing 2: lineaire interpolatie over driehoek

Gegeven: 3 hoekpuntkleuren (c_0, c_1, c_2)

Bepaal: kleur c van punt P binnen driehoek

1. Bereken **barycentrische** coördinaten (α, β, γ) van P t.o.v. P_0, P_1 en P_2
2. Kleur c van P is: $c = \alpha c_0 + \beta c_1 + \gamma c_2$

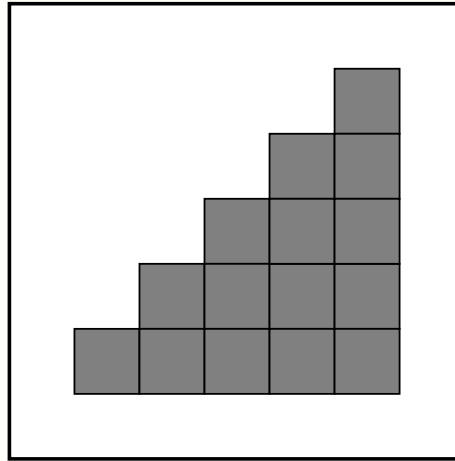
Deze kleurinterpolatie heet **Gouraud** interpolation

Rasterizatie van driehoek

Rasterisatie-algoritme van driehoek

```
for y = ymin to ymax do
  for x = xmin to xmax do
    compute (alpha, beta, gamma) for (x, y)
    if (alpha, beta, gamma tussen 0 en 1) then
      c = alpha c0 + beta c1 + gamma c2
      drawpixel (x, y) with color c
```

Gekleurde driehoek (zie Plate III)



					0.00 1.00 0.00	
				0.25 0.75 0.00	0.25 1.00 0.00	
			0.50 0.50 0.00	0.50 0.75 0.00	0.50 1.00 0.00	
		0.75 0.25 0.00	0.75 0.50 0.00	0.75 0.75 0.00	0.75 1.00 0.00	
	1.00 0.00 0.00	1.00 0.25 0.00	1.00 0.50 0.00	1.00 0.75 0.00	1.00 1.00 0.00	

Veranderingen in elke kleurcomponent **lineair** in rij, kolom en langs rand

Rasterisatie-algoritme van driehoek

```
bepaal bounding box xmin, xmax, ymin, ymax
for y = ymin to ymax do
  for x = xmin to xmax do
    beta  = f20(x, y) / f20(x1, y1)
    gamma = f01(x, y) / f01(x2, y2)
    alpha = 1 - beta - gamma
    if (alpha, beta, gamma > 0) then // dan ook < 1
      c = alpha c0 + beta c1 + gamma c2
      drawpixel (x, y) with color c
```

Practicumopgave: maak algoritme **incrementeel**

Parameterrepresentatie van vlak

Gegeven: vlak door drie punten P_0 , P_1 en P_2

Gevraagd: punt P in vlak

Parameterrepresentatie van vlak:

$$P = P_0 + \lambda(P_1 - P_0) + \mu(P_2 - P_0)$$

Anders geschreven: $P = (1 - \lambda - \mu)P_0 + \lambda P_1 + \mu P_2$

Als **som gewichten = 1** ligt punt P in **vlak**

Punt in driehoek als $0 \leq \lambda, \mu \leq 1$

Berekening barycentrische coördinaten in 3D

Gegeven: P_0 , P_1 en P_2

Bereken: (barycentrische) coördinaten (α, β, γ) van P

$$P = \alpha P_0 + \beta P_1 + \gamma P_2$$

$$\begin{pmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ z_0 & z_1 & z_2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

3 vergelijkingen met 3 onbekenden:

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ z_0 & z_1 & z_2 \end{pmatrix}^{-1} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Berekening barycentrische coördinaten in 3D

Als $\alpha + \beta + \gamma = 1$, dan P in vlak gedefinieerd door 3 gegeven punten

(dan noem je α, β, γ barycentrische coördinaten)

1. Punt P in driehoek $\Leftrightarrow 0 < \alpha, \beta, \gamma < 1$
2. Als één van de bar. coördinaten gelijk 0 is, en andere twee tussen 0 en 1, dan punt op rand driehoek
3. Als twee van de bar. coördinaten gelijk 0, en andere is 1, dan punt is een van de hoekpunten

Ligt 3D punt in driehoek?

Gegeven: punt $P = (1.4, 1.0, 0.8)$ en driehoek door $P_0 = (1, 1, 0)$, $P_1 = (2, 5, 1)$ en $P_2 = (1, -3, 1)$

Gevraagd: Ligt P in driehoek (en dus in vlak door P_0, P_1, P_2)?

Bereken coördinaten van P t.o.v. P_0, P_1 en P_2 : $P = \alpha P_0 + \beta P_1 + \gamma P_2$

$$\begin{pmatrix} 1 & 2 & 1 \\ 1 & 5 & -3 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} 1.4 \\ 1.0 \\ 0.8 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \frac{1}{7} \begin{pmatrix} 8 & -1 & -11 \\ -1 & 1 & 4 \\ 1 & -1 & 3 \end{pmatrix} \begin{pmatrix} 1.4 \\ 1.0 \\ 0.8 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.4 \\ 0.4 \end{pmatrix} \Rightarrow \alpha + \beta + \gamma = 1$$

Punt P in vlak en in driehoek

Ligt 3D punt in driehoek met weinig rekenwerk

Gegeven: punt $P = (1.4, 1.0, 0.8)$ en
driehoek door $P_0 = (1, 1, 0)$, $P_1 = (2, 5, 1)$ en $P_2 = (1, -3, 1)$

Gevraagd: Ligt P in driehoek (en dus in vlak door P_0, P_1, P_2)?

Vlak door P_0, P_1, P_2 is: $P = P_0 + \lambda(P_1 - P_0) + \mu(P_2 - P_0)$

$$\begin{pmatrix} 1.4 \\ 1.0 \\ 0.8 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ -4 \\ 1 \end{pmatrix}$$

3 vergelijkingen met 2 onbekenden

$$1.4 = 1 + \lambda$$

$$1 = 1 + 4\lambda - 4\mu$$

$$0.8 = \lambda + \mu$$

Oplossen levert $\lambda = \mu = 0.4$.

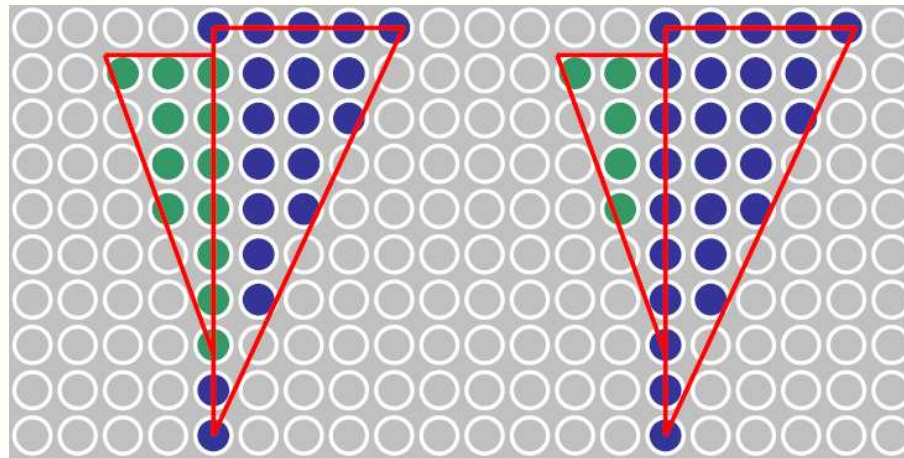
$$P = P_0 + 0.4(P_1 - P_0) + 0.4(P_2 - P_0)$$

$$P = 0.2P_0 + 0.4P_1 + 0.4P_2$$

Dus P ligt in driehoek

Pixels op rand van driehoek

Wat met pixels precies op rand?



Tot welke driehoek behoort rand-pixel?

Beste is pixel toekennen aan **één driehoek**

Problemen bij rasterisatie van driehoek

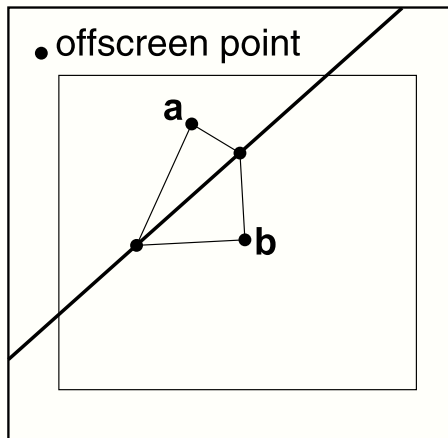
- Welke pixels moeten gekleurd worden?
 - Pixels binnen rand van driehoek
- Wat met pixels precies op rand?
 1. **teken** - maar volgorde van driehoeken maakt uit
 2. **teken niet** - maar gaten tussen driehoeken

Speciale afspraak nodig, b.v.

teken pixels op **linker** of **boven** rand, maar **niet** op rechter of beneden rand

Twee driehoeken met gemeenschappelijke rand

Kies **offscreen point**



rand getekend voor driehoek aan
offscreen kant

dus driehoek met **a** bezit **rand**

Offscreen kant

Lijn door (x_1, y_1) en (x_2, y_2) :

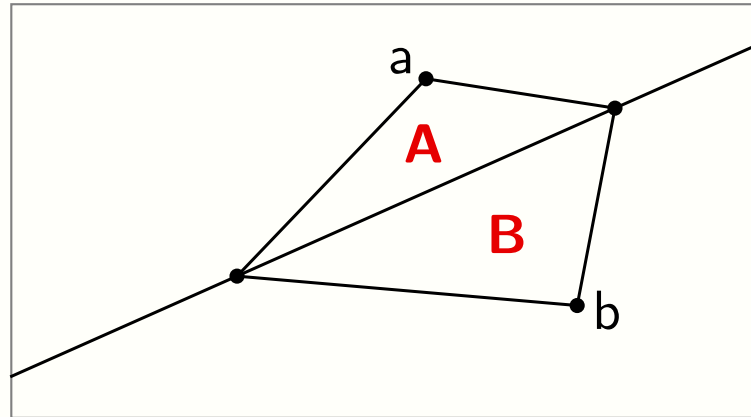
$$f(x, y) = (y_1 - y_2)x + (x_2 - x_1)y + x_1y_2 - x_2y_1 = 0$$

Als **m** $\in (0, 1]$:

1. punt **boven** lijn **positief**
2. punt **beneden** lijn **negatief**

Rand tekenen voor A

- offscreen punt



$f(x_a, y_a) > 0$ en $f(x_{off}, y_{off}) > 0$, omdat $m \in (0, 1]$

Voor alle m:

$f(x_a, y_a)$ en $f(x_{off}, y_{off})$ moeten zelfde **teken** hebben

Rand tekenen

Gegeven: gemeenschappelijke rand door (x_1, y_1) en (x_2, y_2)

Gevraagd: pixel (x, y) **op rand** tekenen?

Lijn door (x_1, y_1) en (x_2, y_2) :

$$f_{12}(x, y) = (y_1 - y_2)x + (x_2 - x_1)y + x_1y_2 - x_2y_1 = 0$$

1. $\alpha = \frac{f_{12}(x, y)}{f_{12}(x_0, y_0)} = 0$

2. Als $f_{12}(x_0, y_0)f_{12}(x_{off}, y_{off}) > 0$ dan tekenen

Algoritme voor randen driehoeken

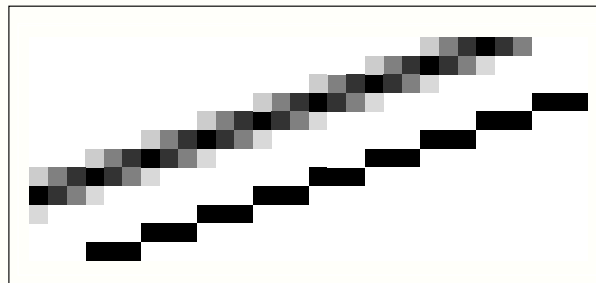
$$(x_{off}, y_{off}) = (-1, -1)$$

```
bepaal bounding box xmin, xmax, ymin, ymax
for y = ymin to ymax do
  for x = xmin to xmax do
    beta  = f20(x, y) / f20(x1, y1)
    gamma = f01(x, y) / f01(x2, y2)
    alpha = 1 - beta - gamma
    if (alpha, beta, gamma >= 0) then
      if (alpha > 0 or f12(x0, y0)f12(-1, -1) > 0) and
        (conditie voor beta) and
        (conditie voor gamma) then
        c = alpha c0 + beta c1 + gamma c2
        drawpixel (x, y) with color c
```

Aliasing

Aliasing is probleem in Computergraphics omdat resolutie van

1. scherm is eindig
2. wiskundige modellen die beeld beschrijven is oneindig



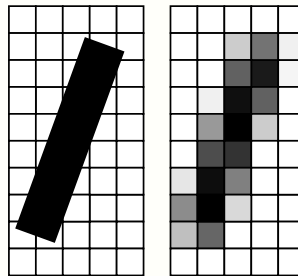
Details van continue lijn hebben hogere frequentie dan pixels

Antialiasing

Probleem van **aliasing** in beelden verminderen

Box filter:

kleur pixel berekenen door overlap tussen **rechthoek** bepaald door lijn en pixel



Real of CG?

<http://area.autodesk.com/fakeorfoto/challenge>