

TINCGR01 — Computer Graphics

dr. Wouter Bergmann Tiest

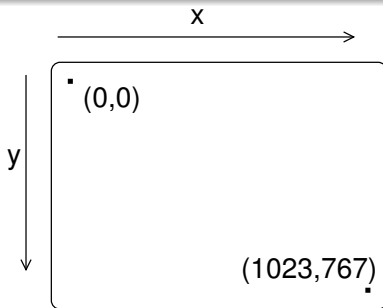
Hogeschool Rotterdam

`W.M.Bergmann.Tiest@hr.nl`

Raster graphics

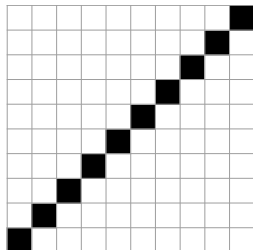
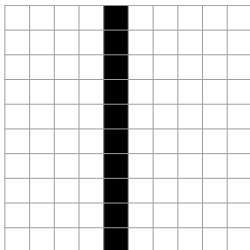
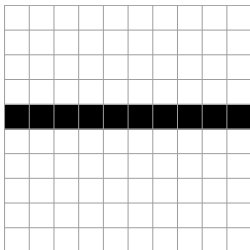
Basisprincipes

- Rooster van puntjes.
- Ieder puntje kan een kleur krijgen.
- $(0,0)$ is meestal linksboven.
- Resolutie (dots per inch, dpi): puntjesdichtheid.



Rasteren

- Omzetten van figuren in raster.
- Horizontale ven verticale lijnen geen probleem.
- Lijnen onder 45° ook niet.
- Andere hoeken dan 0° , 45° of 90° ?



Bresenham-algoritme

- Verbind (x_1, y_1) met (x_2, y_2) .
- Voor hoeken $0 < \alpha < 45^\circ$.
- Helling $a = \frac{y_2 - y_1}{x_2 - x_1}$. ($0 < a < 1$).
- Zet puntje op $(x, \text{round}(a(x - x_1)) + y_1)$ voor $x = x_1, x_1 + 1, \dots, x_2$.
- Andere octanten: x en y verwisselen.

Anti-aliasing

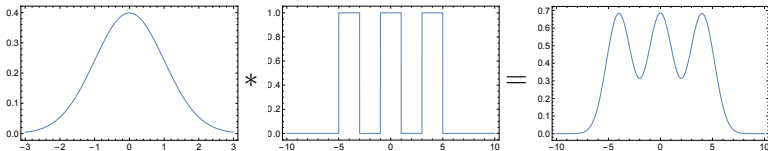
- Bij lage resolutie: rafelrandjes bij schuine lijnen (*aliasing*).
- Oplossing: gebruik *supersampling* (bijv. intern $2\times$ resolutie) en meng voor- en achtergrondkleur (*anti-aliasing*).
- Nadeel: randen worden minder scherp.

Gaussian blur

- Maakt plaatjes “gladder”; artefacten verdwijnen.
- Convolutie met een Gaussische functie.
- Gauss (1D): $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$.
- Convolutie (1D): $(f * g)(x) = \int_{-\infty}^{\infty} f(x')g(x - x')dx'$.

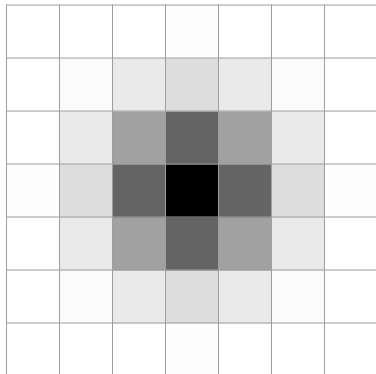
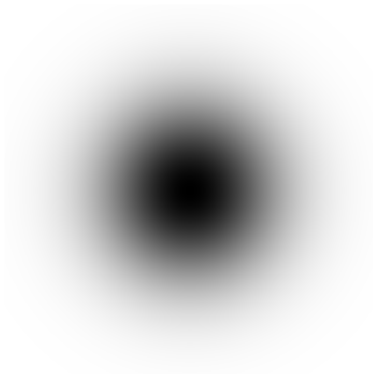
Gaussian blur

- Maakt plaatjes “gladder”; artefacten verdwijnen.
- Convolutie met een Gaussische functie.
- Gauss (1D): $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$.
- Convolutie (1D): $(f * g)(x) = \int_{-\infty}^{\infty} f(x')g(x - x')dx'$.



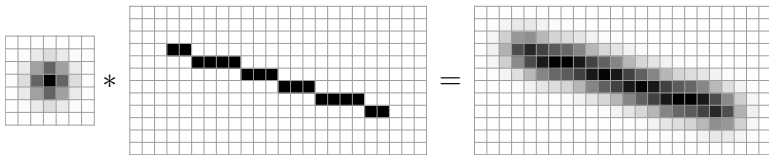
Gaussian blur

- Gauss (2D): $f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$



Gaussian blur

- Convolutie (discreet, 2D): $(f * g)_{i,j} = \sum_{i',j'} f_{i',j'} g_{i-i',j-j'}$.



Gaussian blur

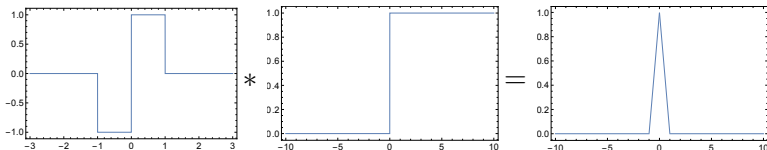
- Convolutie (discreet, 2D): $(f * g)_{i,j} = \sum_{i',j'} f_{i',j'} g_{i-i',j-j'}$.

Edge detection

- Belangrijk voor scheiden voor- en achtergrond in computer vision.
- Neem afgeleide in horizontale en verticale richting en tel kwadratisch op. Boven drempel? Dan rand.
- Discrete afgeleide (horizontaal): $\left(\frac{df}{dx}\right)_{i,j} = f_{i+1,j} - f_{i,j}$.
- Te schrijven als convolutie met $\{-1, 1\}$.

Edge detection

- Belangrijk voor scheiden voor- en achtergrond in computer vision.
- Neem afgeleide in horizontale en verticale richting en tel kwadratisch op. Boven drempel? Dan rand.
- Discrete afgeleide (horizontaal): $\left(\frac{df}{dx}\right)_{i,j} = f_{i+1,j} - f_{i,j}$.
- Te schrijven als convolutie met $\{-1, 1\}$.



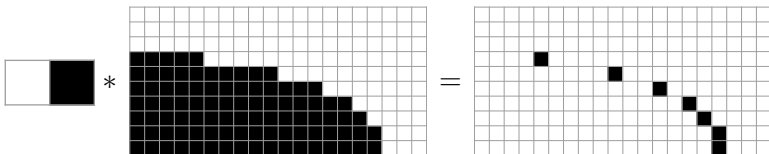
Raster graphics

TINCGR01 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Opdracht les 3



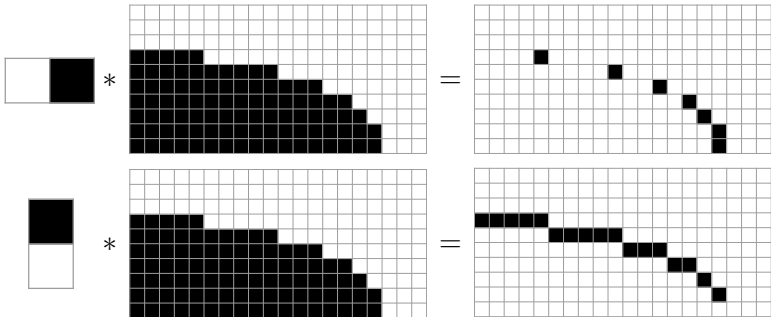
Raster graphics

TINCGR01 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Opdracht les 3



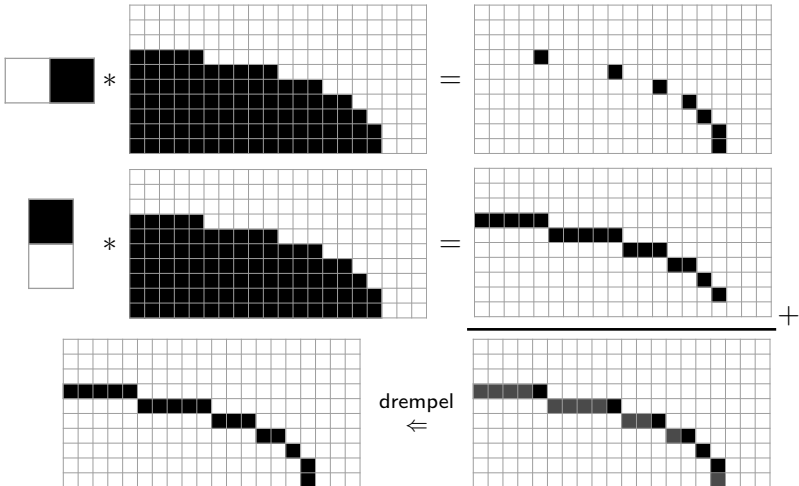
Raster graphics

TINCGR01 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Opdracht les 3



Edge detection

- Meestal in combinatie met Gaussisch filter (in één convolutie).
- σ van Gaussfunctie bepaalt schaal r van randen.

Compressie

- Graphics kosten veel opslagruimte.
- Bijv. $800 \times 600 \times 24 \text{ bits} = 1,44 \text{ MByte}$ voor één plaatje.
- Kan vaak compacter door compressie.
- *Lossless* compressie: originele beeld volledig te reconstrueren, puur wiskundig.
- *Lossy* compressie: weglaten van onbelangrijke delen, dankzij kennis van visuele waarneming.

Lossless compressie

- Bijv. kleur-indexatie:
- In plaatje van 800×600 maximaal 480.000 verschillende kleuren, vaak veel minder.
- In plaats van 24 bits per pixel max. 19 bits per pixel nodig, vaak minder.
- Wel ruimte voor kleurtabel nodig.
- Effectief bij gering aantal verschillende kleuren.

Lossless compressie

- Bijv. *dictionary*:
- Indexeer de meest-voorkomende patronen met hele korte getallen.
- Vergelijk in taal: vervang woorden als *het*, *van*, *een* en *niet* door een enkele letter.

Lossless compressie

- Bijv. *run-length encoding*:
- AAAAAAAAAABBBBBBCDDDD \Rightarrow 9A5BC4D
- Effectief bij grote vlakken van één kleur.
- Formaten als GIF en PNG maken gebruik van combinaties van dit soort technieken.
- Winst in bestandsgrootte varieert, typisch 50–90 % reductie.

Lossy compressie

- Verkleinen kleurreimte, bijv. van 24 bits naar 8 bits per pixel (in combinatie met kleur-indexatie).
- Te sterke reductie geeft *posterization*.

Lossy compressie

- Weglaten van hogere ruimtelijke frequenties.
- Gebruikt in JPEG-compressie (blokjes van 8×8).
- Discrete Cosine Transform (soort Fourier-transformatie).
- Bij hoge compressie artefacten zichtbaar.



Video compressie

- Stuur alleen wat er veranderd is tussen twee opeenvolgende frames.
- Iedere n frames een *key frame*.
- Bij snelle bewegingen artefacten zichtbaar.



(a)



(b)

Opdracht les 3

Rasteren

- Schrijf een functie een lijn rastert tussen (x_1, y_1) en (x_2, y_2) en geef het raster uitvergroot weer.
- `rasterline(x1, y1, x2, y2)`
- Voor alle richtingen, dus check eerst in welk octant je zit.
- In Python: `class Grid` (zie files op N@tschool).
- ```
from grid import *
g = Grid(20, 10)
g.addPoint(1, 1)
:
g.draw()
```
- Broncode voorzien van commentaar inleveren via N@tschool vóór begin volgende les.
- Bonus: gebruik anti-aliasing.