

TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

TINCGR02 — Computer Graphics

dr. Wouter Bergmann Tiest

Hogeschool Rotterdam

W.M.Bergmann.Tiest@hr.nl

TINCGR02 —
Computer
Graphics

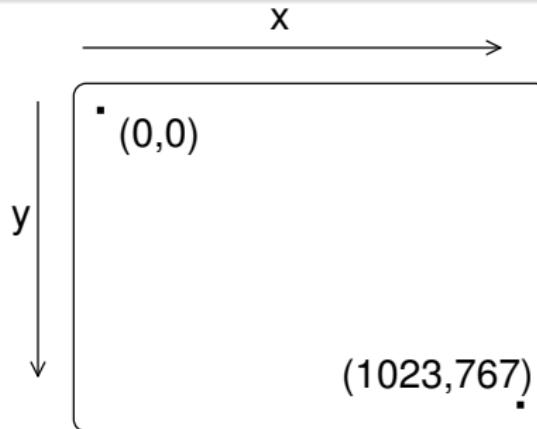
dr. Wouter
Bergmann
Tiest

Raster
graphics

Raster graphics

Basisprincipes

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



Gaussian blur

- Maakt plaatjes “gladde”; 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): $(f * g)(x) = \int_{-\infty}^{\infty} f(x')g(x - x')dx'$.

Raster graphics

TINCGR02 —

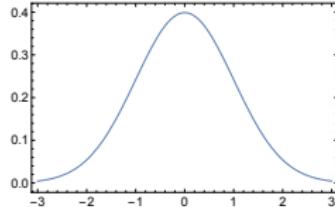
Computer
Graphics

dr. Wouter
Bergmann
Tiest

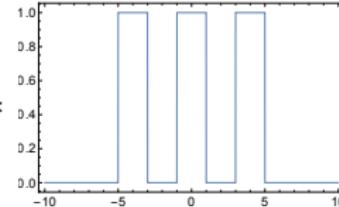
Raster
graphics

Gaussian blur

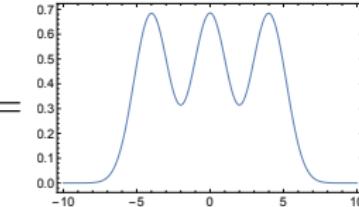
- Maakt plaatjes “gladder”; artefacten verdwijnen.
- Convolutie met een Gaussische functie.
- Gauss (1D): $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$.
- Convolutie (1D): $(f * g)(x) = \int_{-\infty}^{\infty} f(x')g(x - x')dx'$.



*



=



Raster graphics

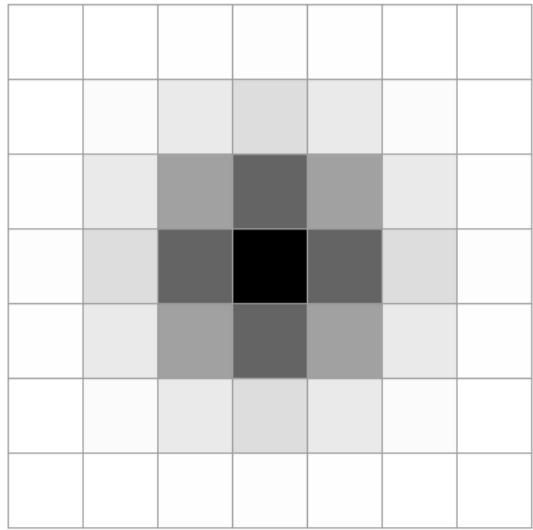
TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Gaussian blur

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



Raster graphics

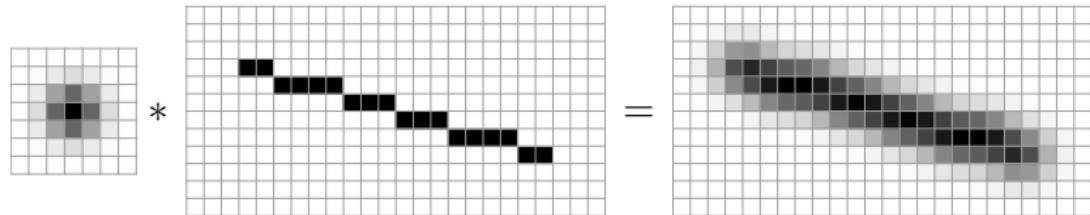
TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Gaussian blur

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



Gaussian blur

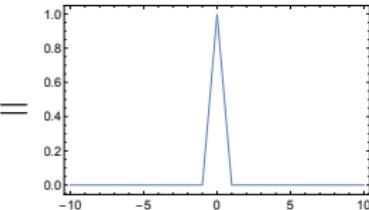
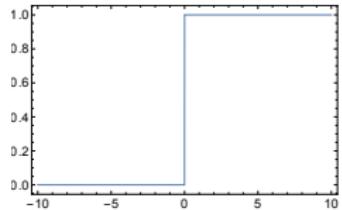
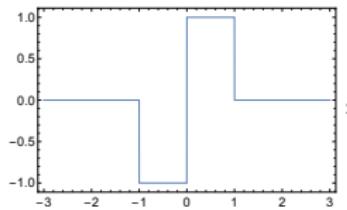
- Convolutie (discreet, 2D): $(f * g)_{i,j} = \sum_{i',j'} f_{i',j'} g_{i-i',j-j'}.$

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 $\{-1, 1\}$.

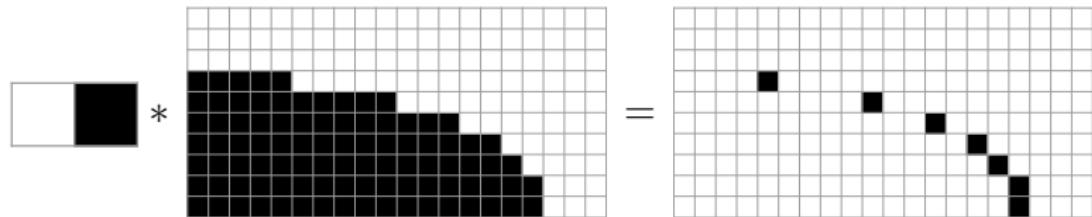


Raster graphics

TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

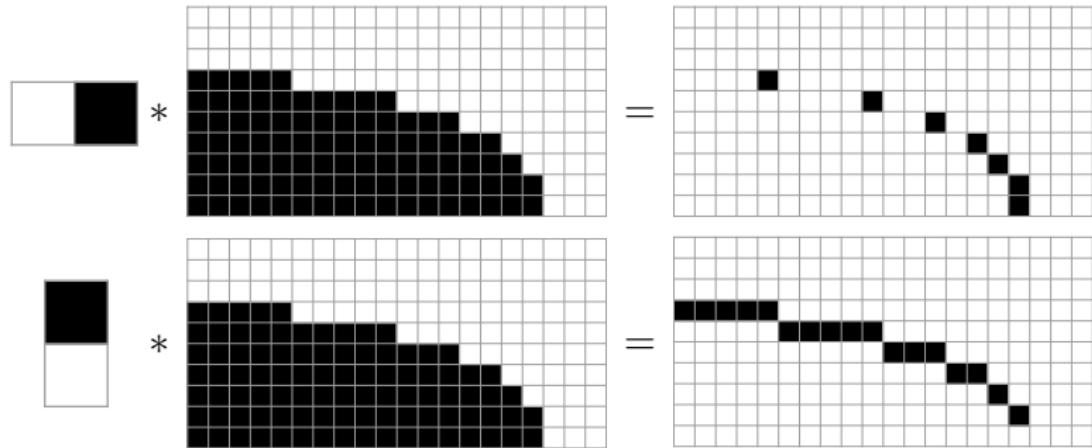


Raster graphics

TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

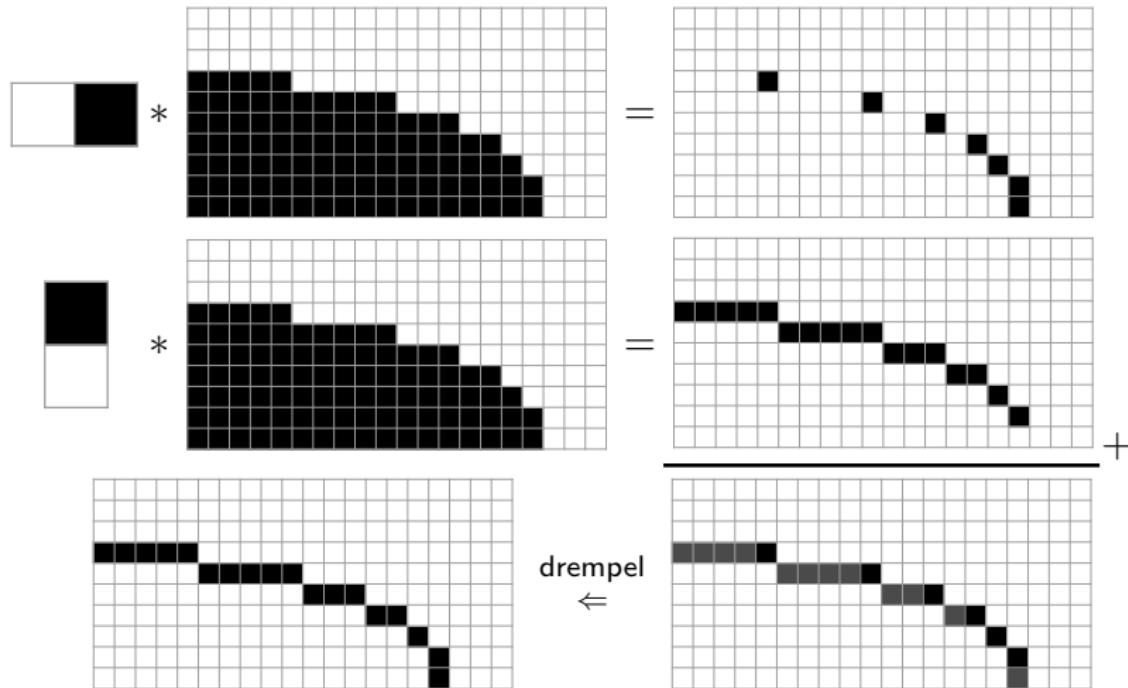


Raster graphics

TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

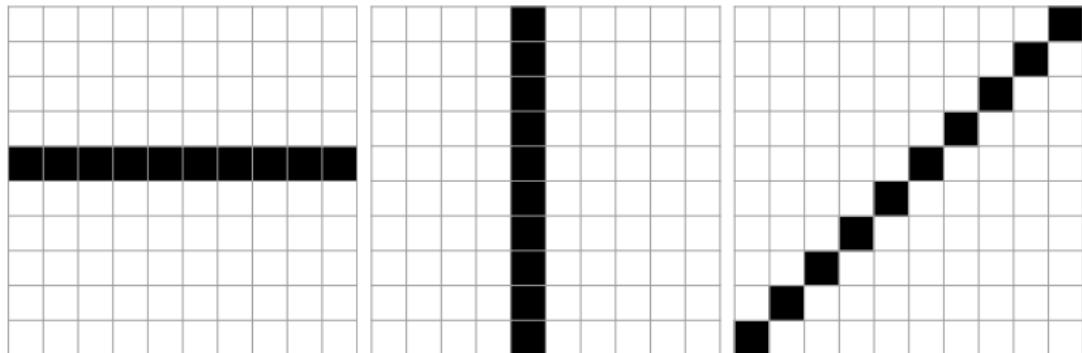


Edge detection

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

Rasteren

- Omzetten van figuren in raster.
- Horizontale en 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 $0 < \varphi < 45^\circ$.
- 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.

Raster graphics

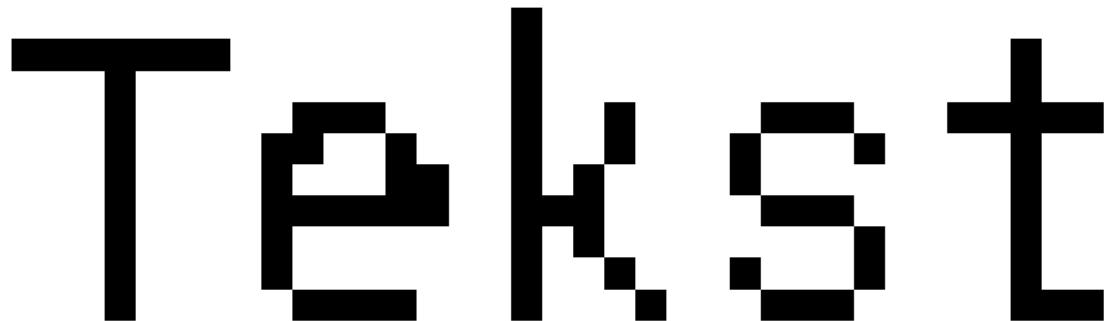
TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Anti-aliasing

- Bij lage resolutie: rafelrandjes bij schuine lijnen (*aliasing*).



Raster graphics

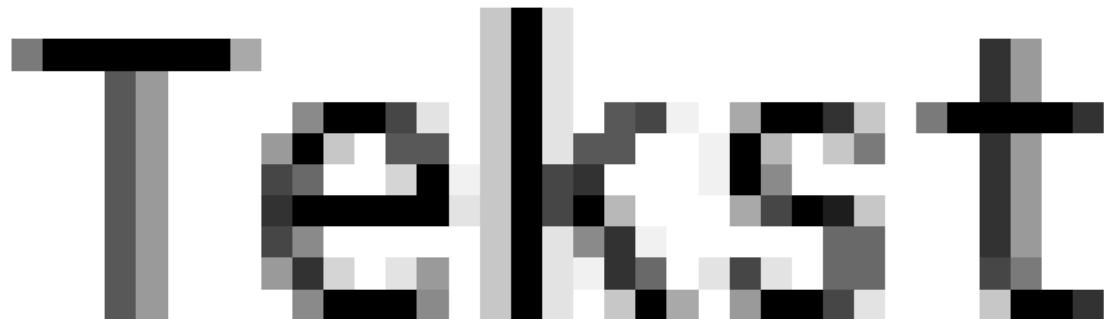
TINCGR02 —
Computer
Graphics

dr. Wouter
Bergmann
Tiest

Raster
graphics

Anti-aliasing

- Bij lage resolutie: rafelrandjes bij schuine lijnen (*aliasing*).



Anti-aliasing

- Oplossing: gebruik *supersampling* (bijv. intern 2× resolutie) en meng voor- en achtergrondkleur (*anti-aliasing*).
- Nadeel: randen worden minder scherp.

Opdracht 3: Rasteren

- Schrijf een functie die 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 Teams).
- ```
from grid import *
g = Grid(20, 10)
g.addPoint(1, 1)
:
g.draw()
```
- Broncode voorzien van commentaar binnen één week inleveren via Teams.
- Bonus: gebruik anti-aliasing.

## Beoordeling

- Het programma maakt geen gebruik van één van de voorgeschreven templates voor Python, Java, C of C++: 0 pnt, ook geen andere punten voor deze opdracht mogelijk.
- (compileert en) draait zonder foutmeldingen: 1 pnt
- richtingen  $0^\circ$  en  $180^\circ$  correct: +1 pnt
- richtingen  $90^\circ$  en  $270^\circ$  correct: +1 pnt
- richtingen tussen  $-45^\circ$  en  $45^\circ$  correct: +1 pnt
- richtingen tussen  $135^\circ$  en  $225^\circ$  correct: +1 pnt
- richtingen tussen  $45^\circ$  en  $135^\circ$  correct: +1 pnt
- richtingen tussen  $225^\circ$  en  $315^\circ$  correct: +1 pnt
- alle getekende lijnen zijn lang genoeg: +1 pnt
- voorzien van informatief commentaar: +1 pnt
- anti-aliasing toegepast op de getekende lijnen: +1 pnt

## 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  $800 \times 600$  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.

# Raster graphics

TINCGR02 —  
Computer  
Graphics

dr. Wouter  
Bergmann  
Tiest

Raster  
graphics

## Lossless compressie

- Bijv. *run-length encoding*:
- AAAAAAAAABBBBBBCDDDD ⇒ 9A6BC4D
- 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 kleurruimte, bijv. van 24 bits naar 8 bits per pixel (in combinatie met kleur-indexatie).
- Te sterke reductie geeft *posterization*.

# Raster graphics

TINCGR02 —

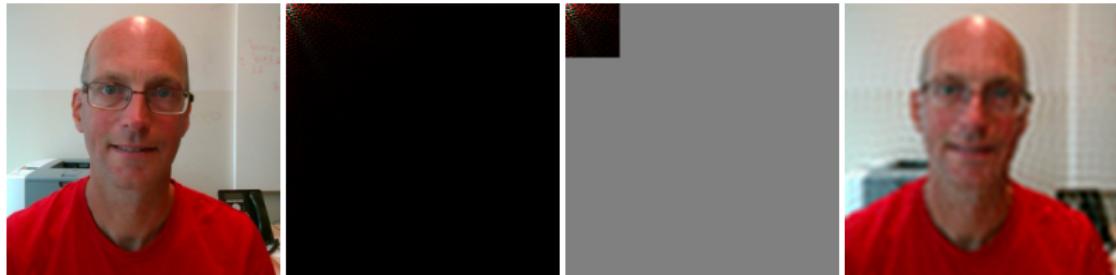
Computer  
Graphics

dr. Wouter  
Bergmann  
Tiest

Raster  
graphics

## Lossy compressie

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



# Raster graphics

TINCGR02 —

Computer  
Graphics

dr. Wouter  
Bergmann  
Tiest

Raster  
graphics

## 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)