

# Hybride kd-bomen voor photonmapping en het versnellen van ray tracing

**Matthias Moulin**



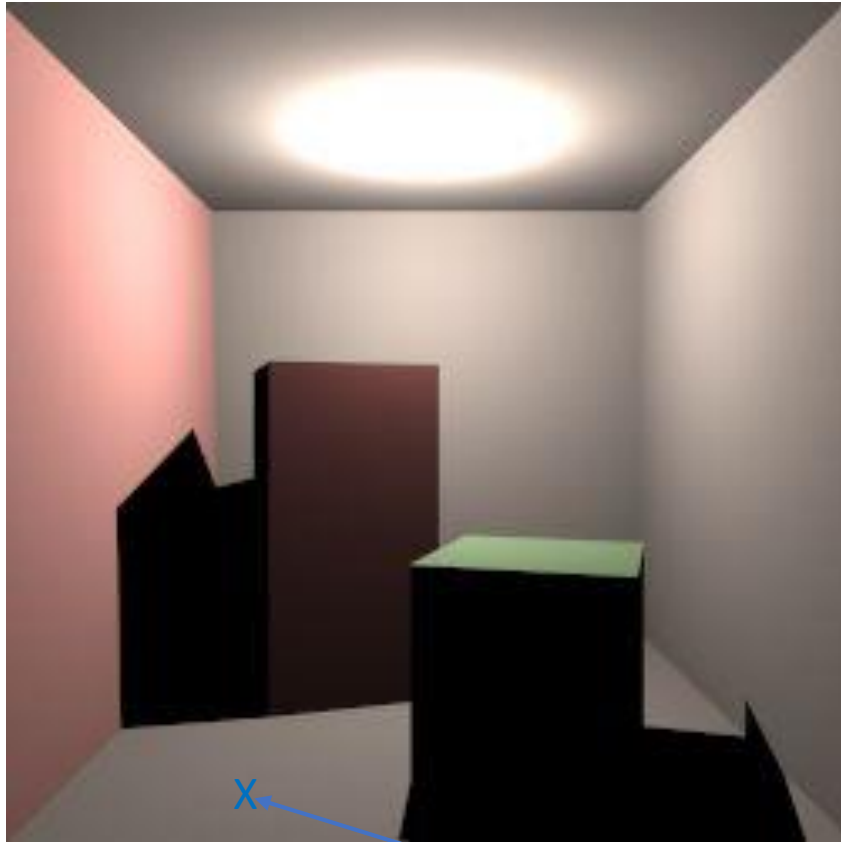
# Ray tracing



# Ray tracing



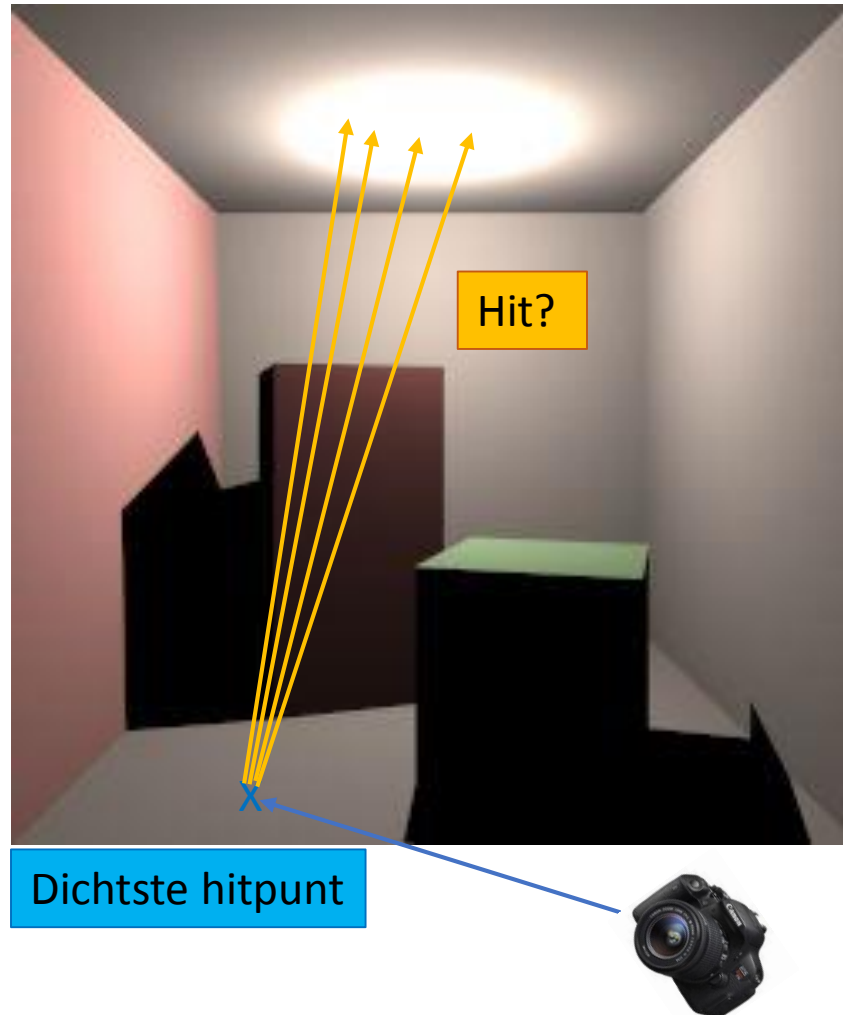
# Ray tracing



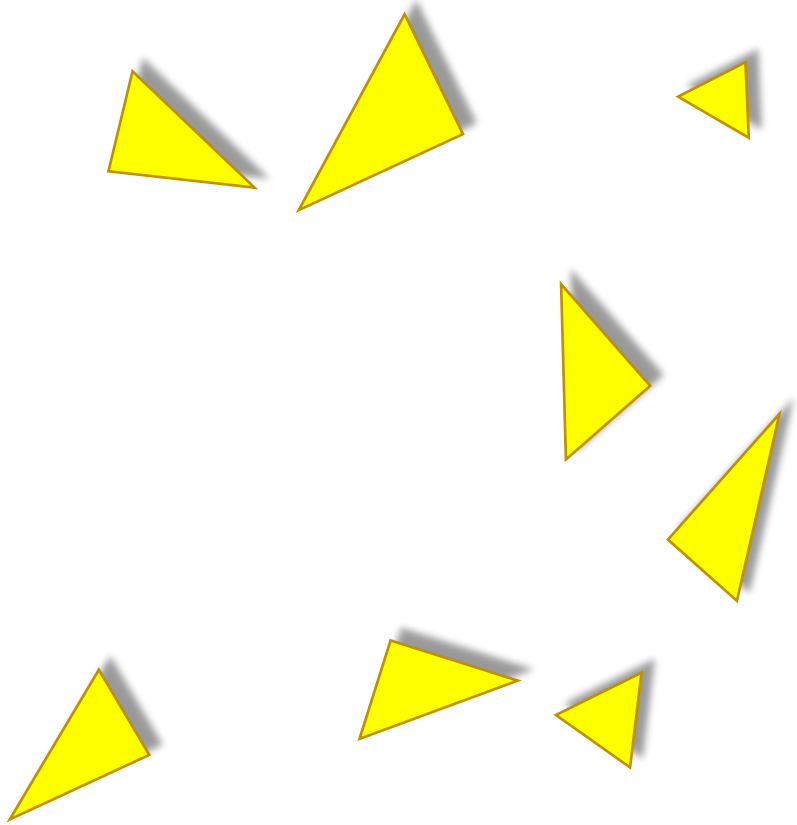
Dichtste hitpunt



# Ray tracing

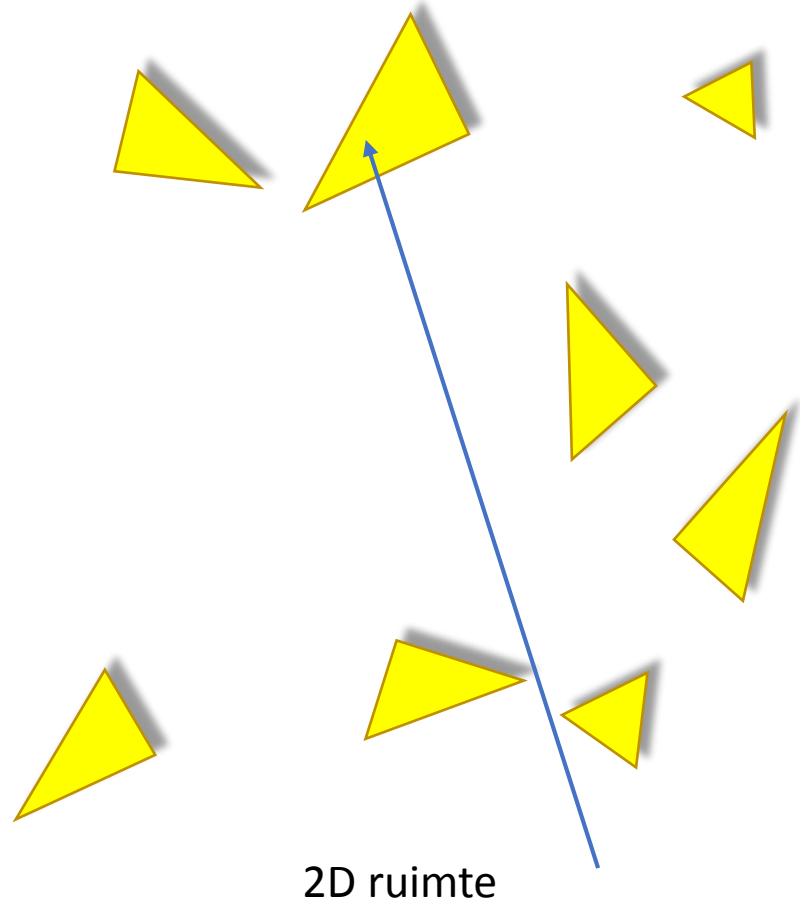


# Acceleratiestructuren – kd-bomen

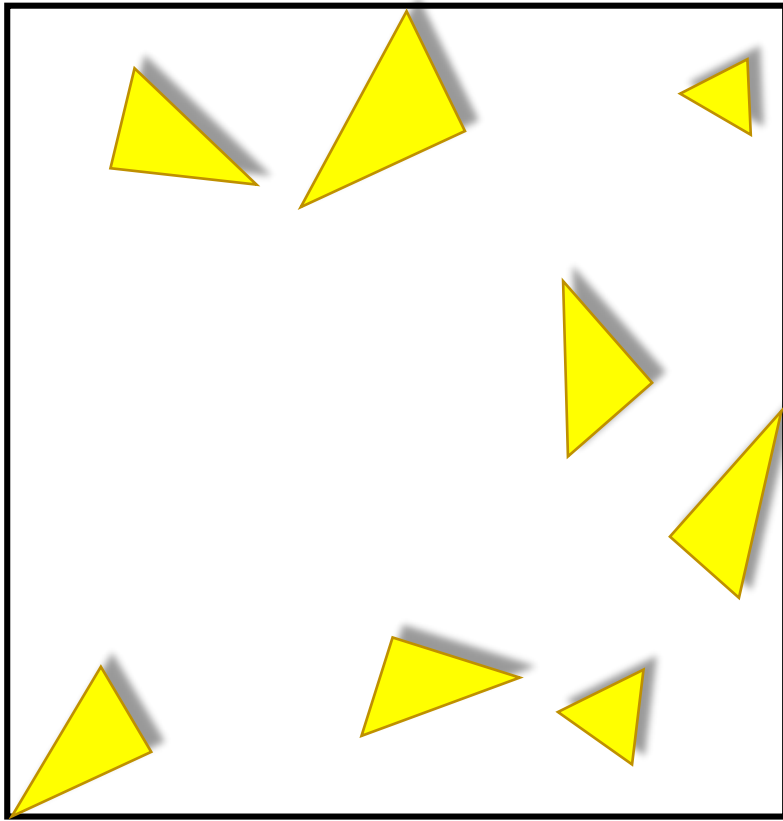


2D ruimte

# Acceleratiestructuren – kd-bomen



# Acceleratiestructuren – kd-bomen



2D ruimte

Kd-boom

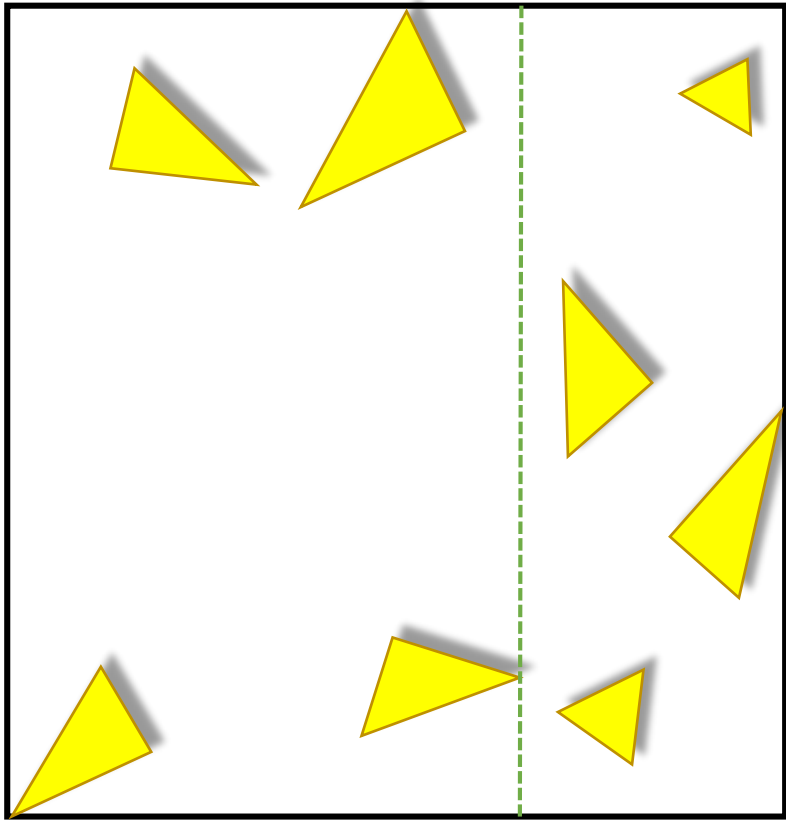
Opsplitsingsheuristiek

- Surface Area Heuristic (SAH)

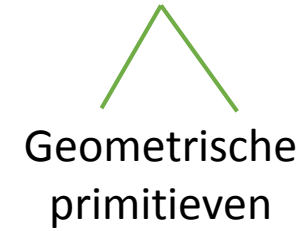
[Goldsmith & Salmon 1987], [MacDonald & Booth 1990]



# Acceleratiestructuren – kd-bomen

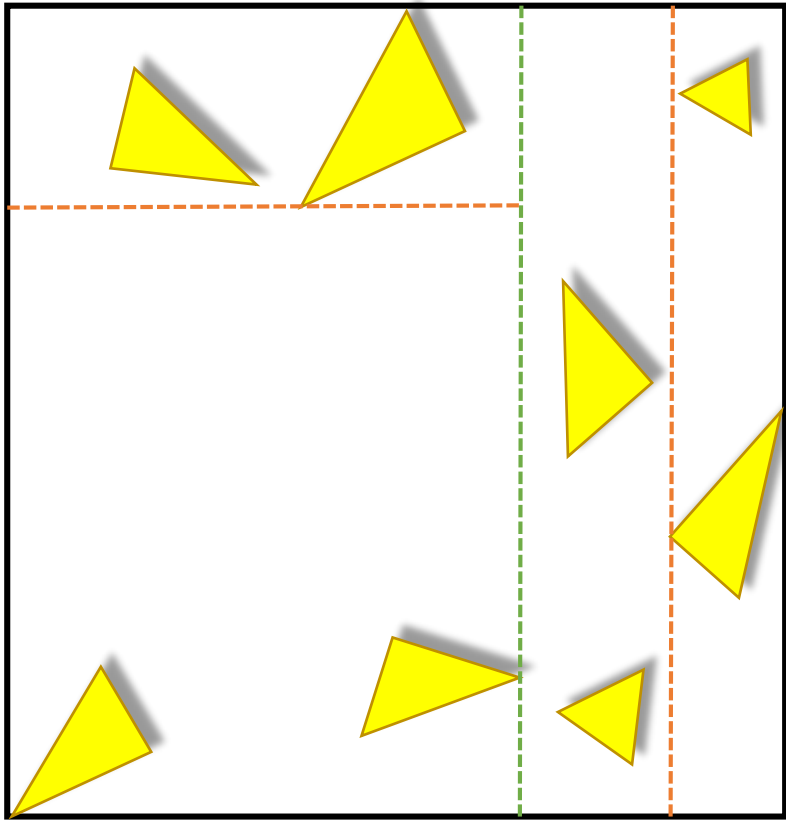


2D ruimte



kd-boom datastructuur

# Acceleratiestructuren – kd-bomen

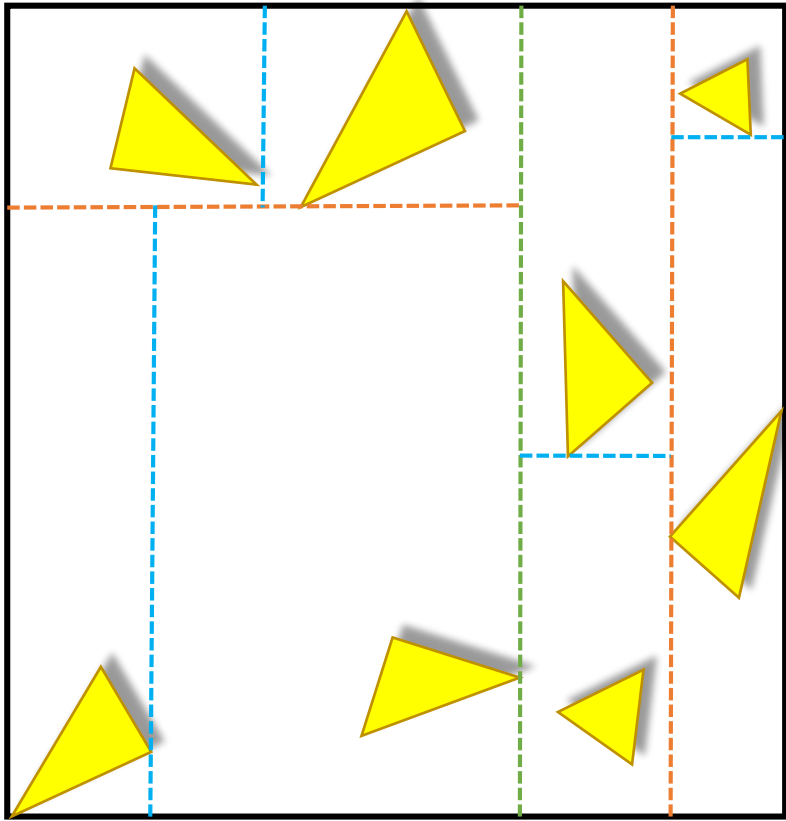


2D ruimte

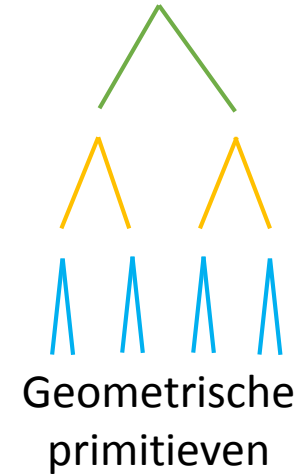


kd-boom datastructuur

# Acceleratiestructuren – kd-bomen

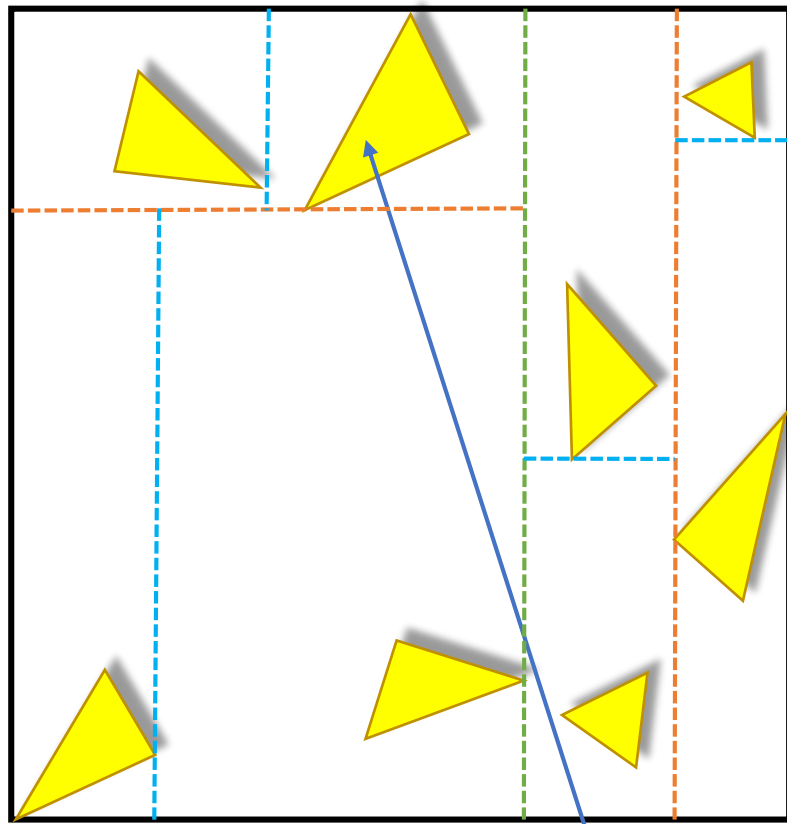


2D ruimte



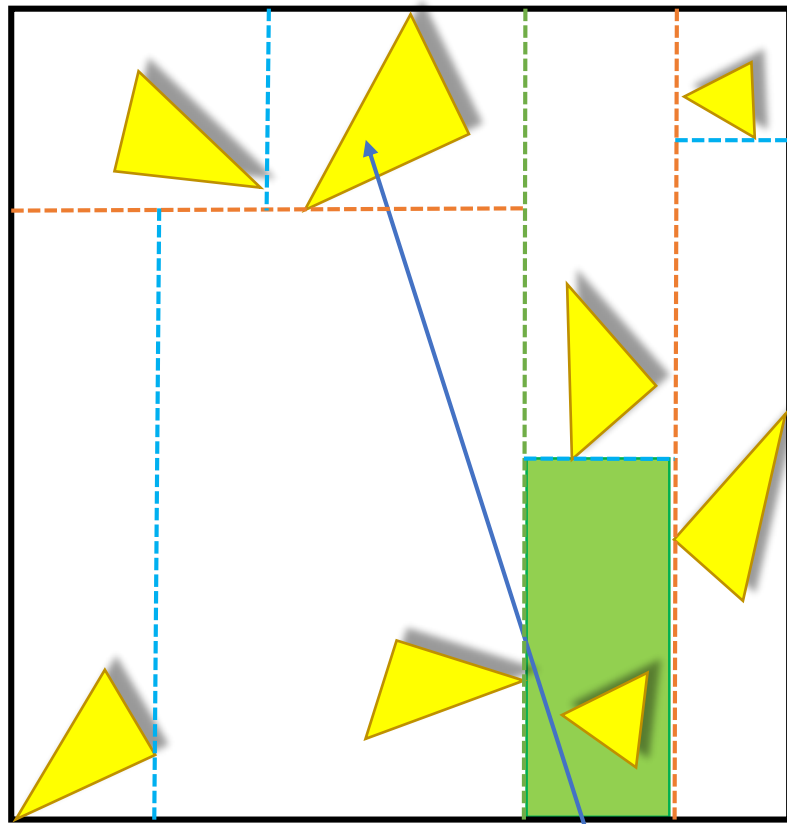
kd-boom datastructuur

# Acceleratiestructuren – kd-bomen



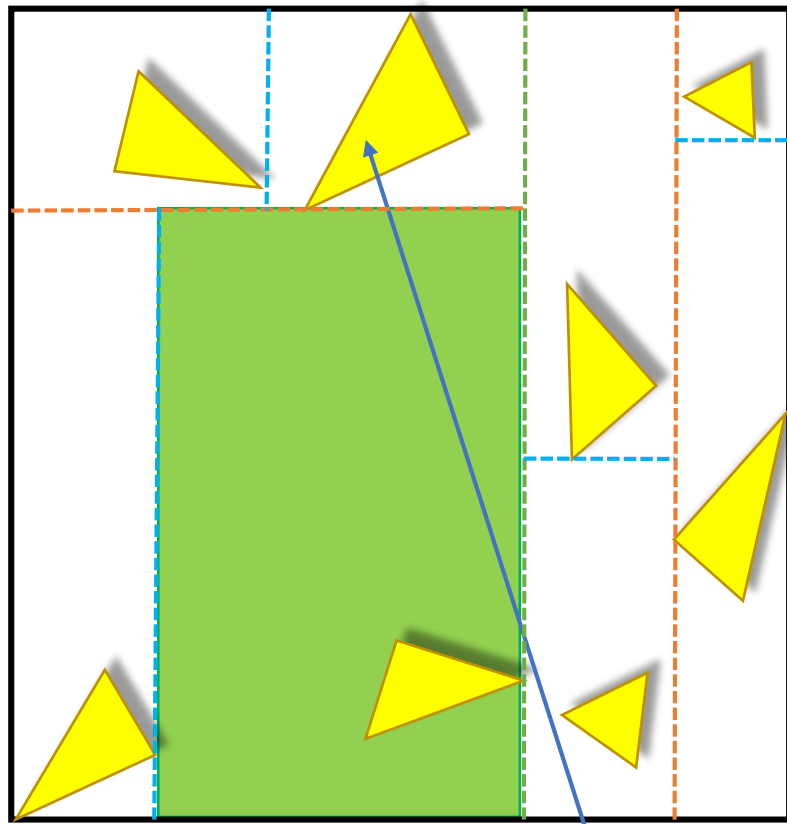
2D ruimte

# Acceleratiestructuren – kd-bomen



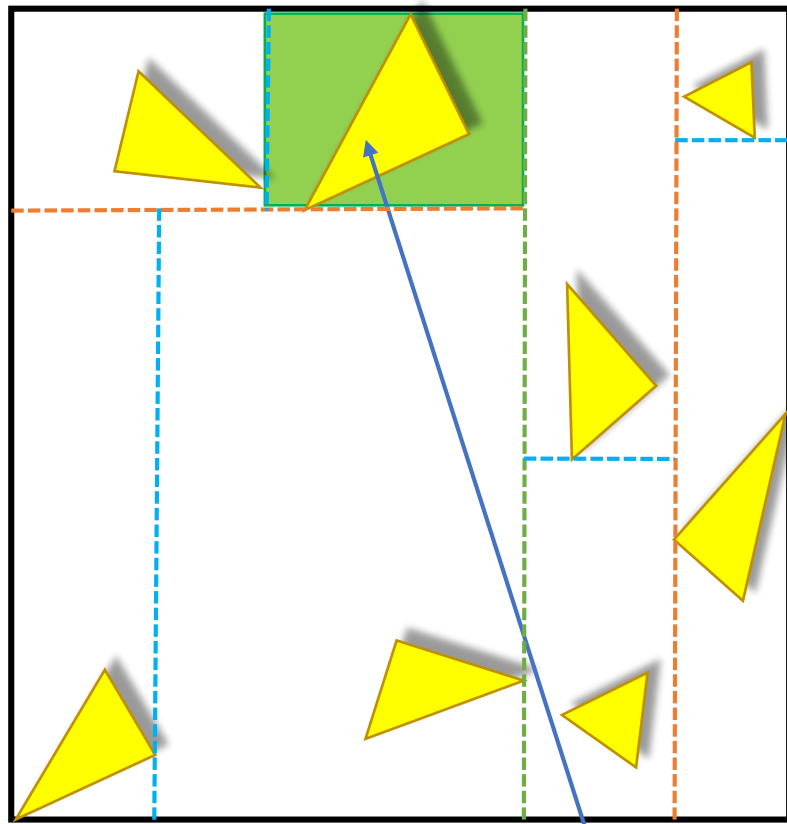
2D ruimte

# Acceleratiestructuren – kd-bomen



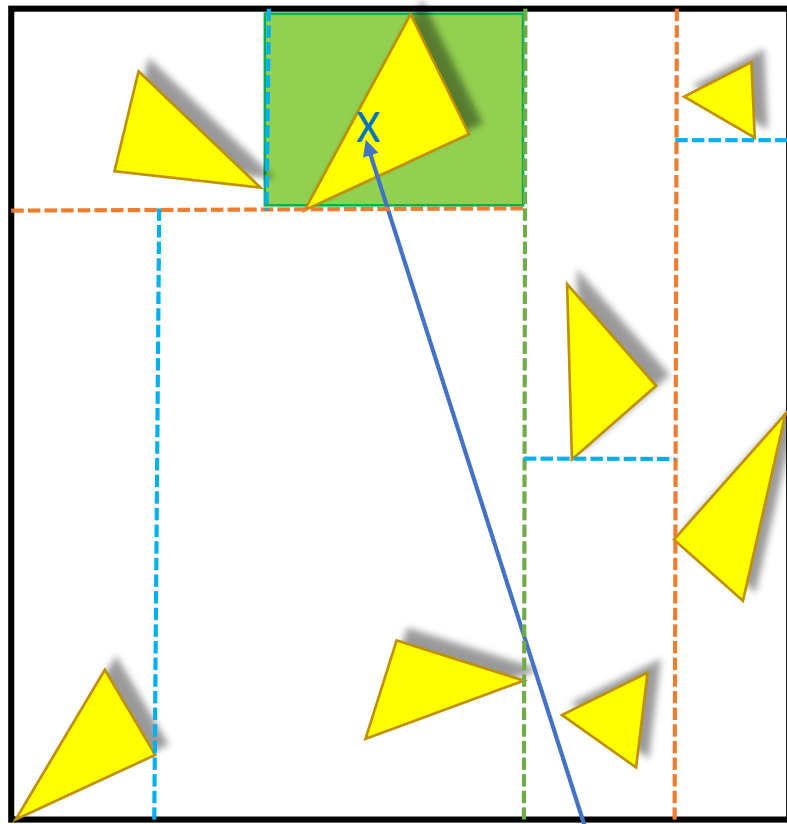
2D ruimte

# Acceleratiestructuren – kd-bomen



2D ruimte

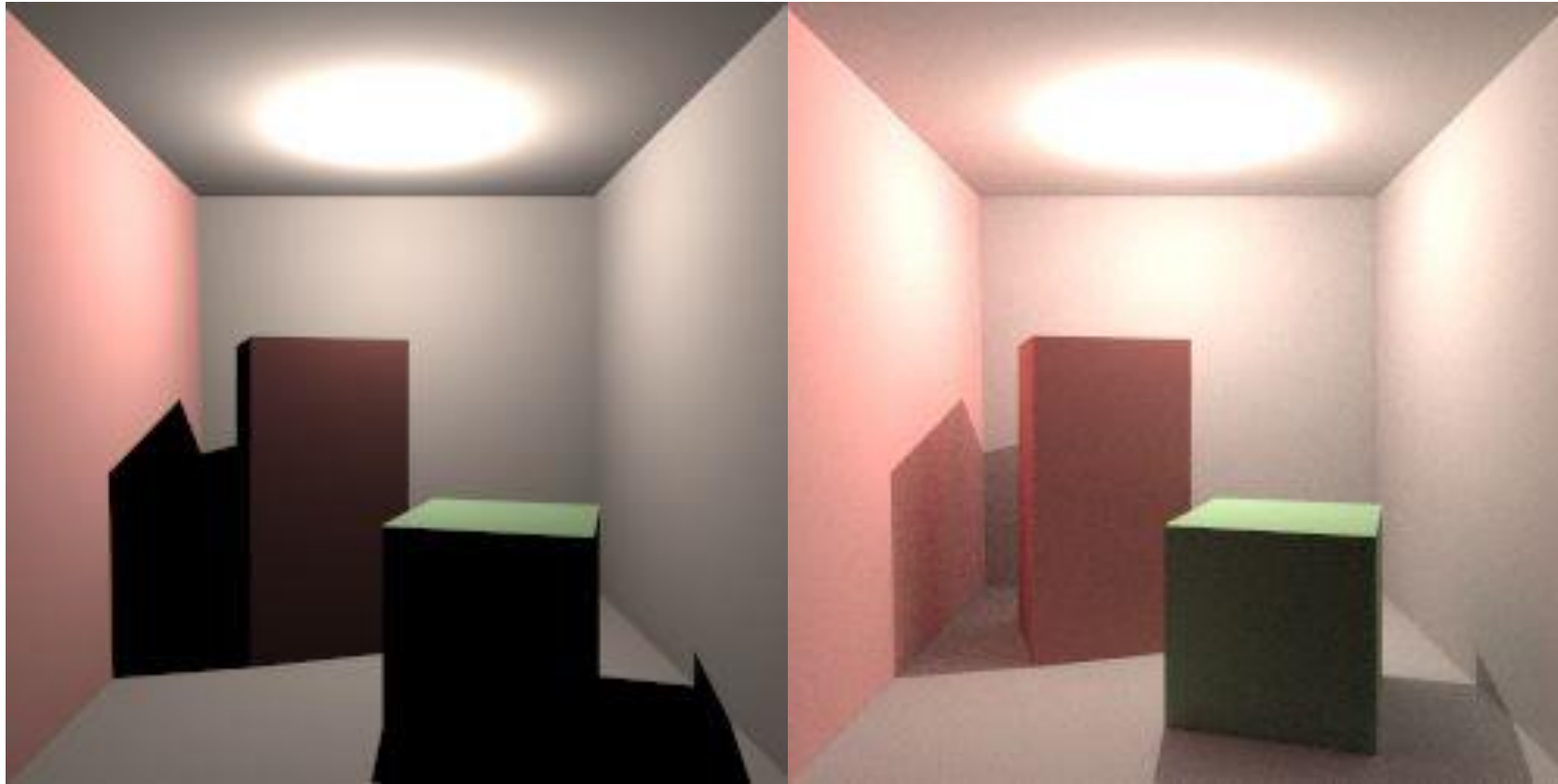
# Acceleratiestructuren – kd-bomen



2D ruimte



# Globale belichting

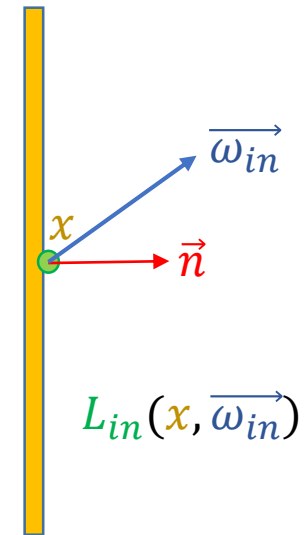
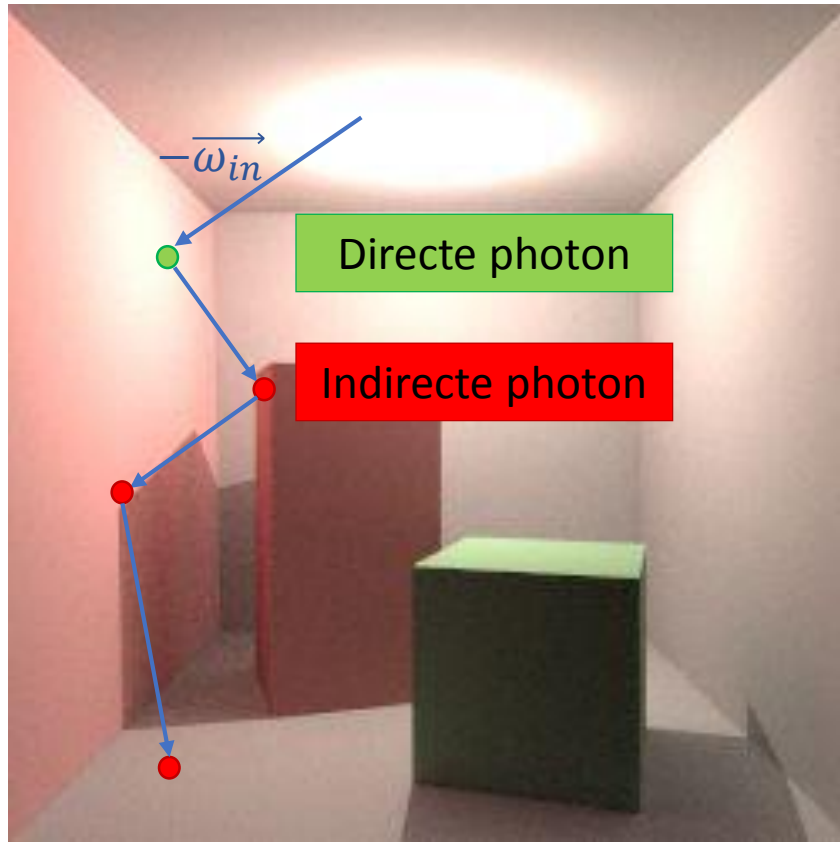


Directe belichting

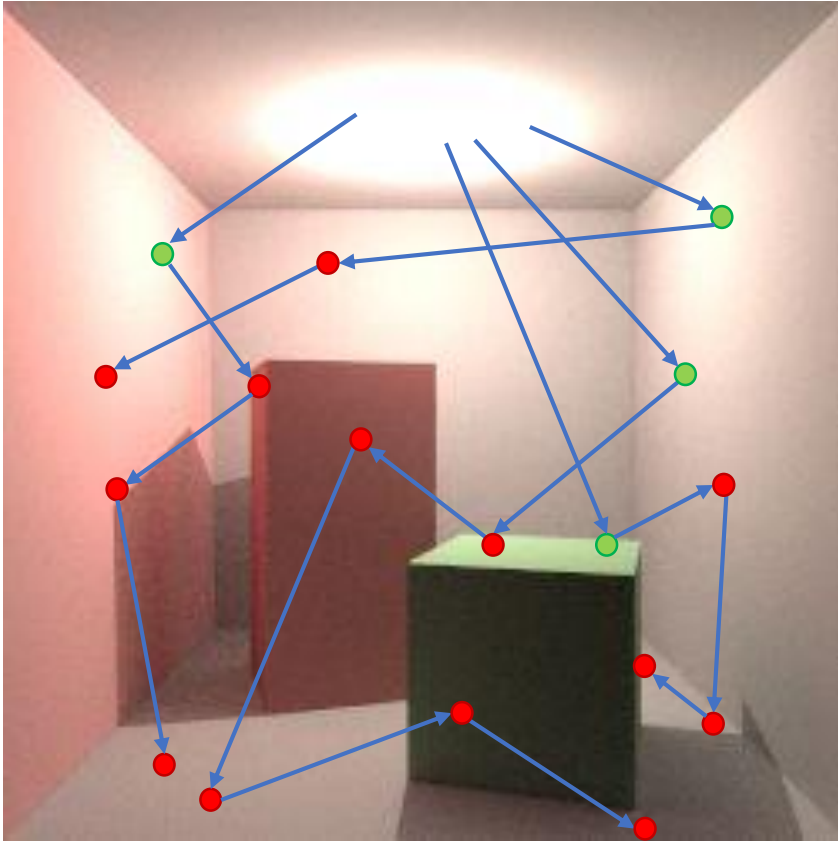
Directe + Indirecte belichting

Photonmapping [Jensen 1996]

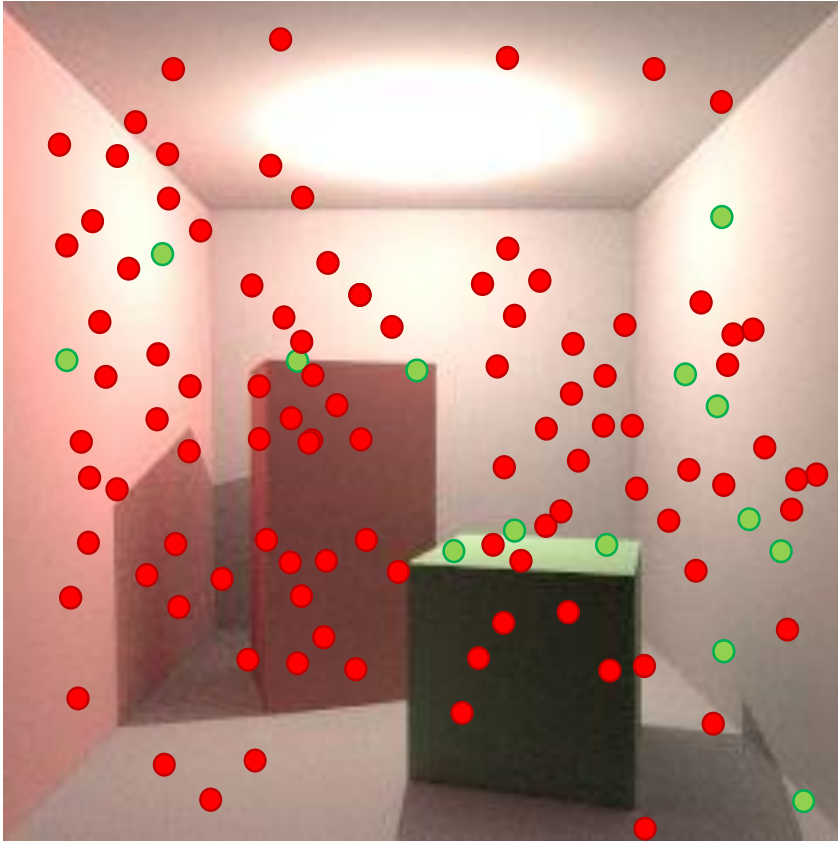
# Photonmapping - Photontracing



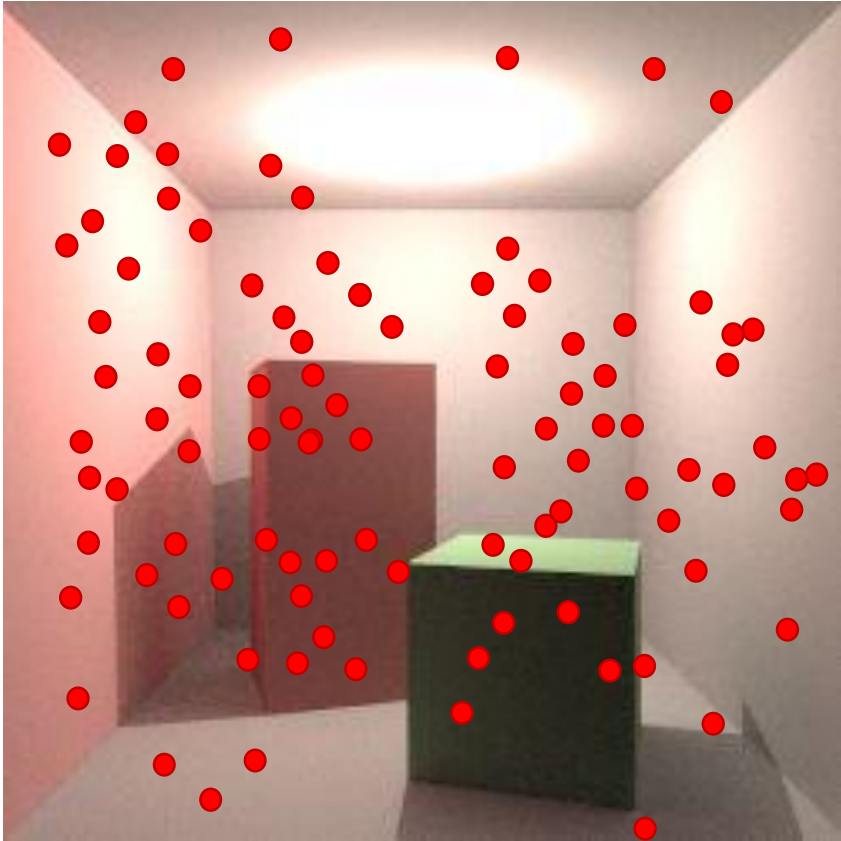
# Photonmapping – Photon tracing



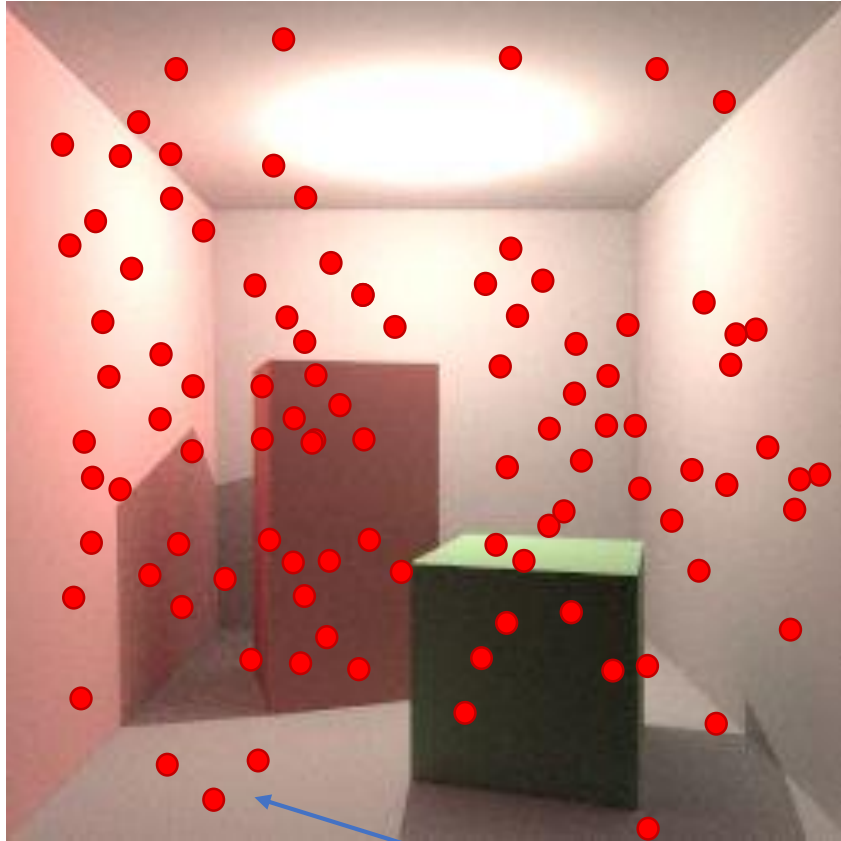
# Photonmapping – Photon tracing



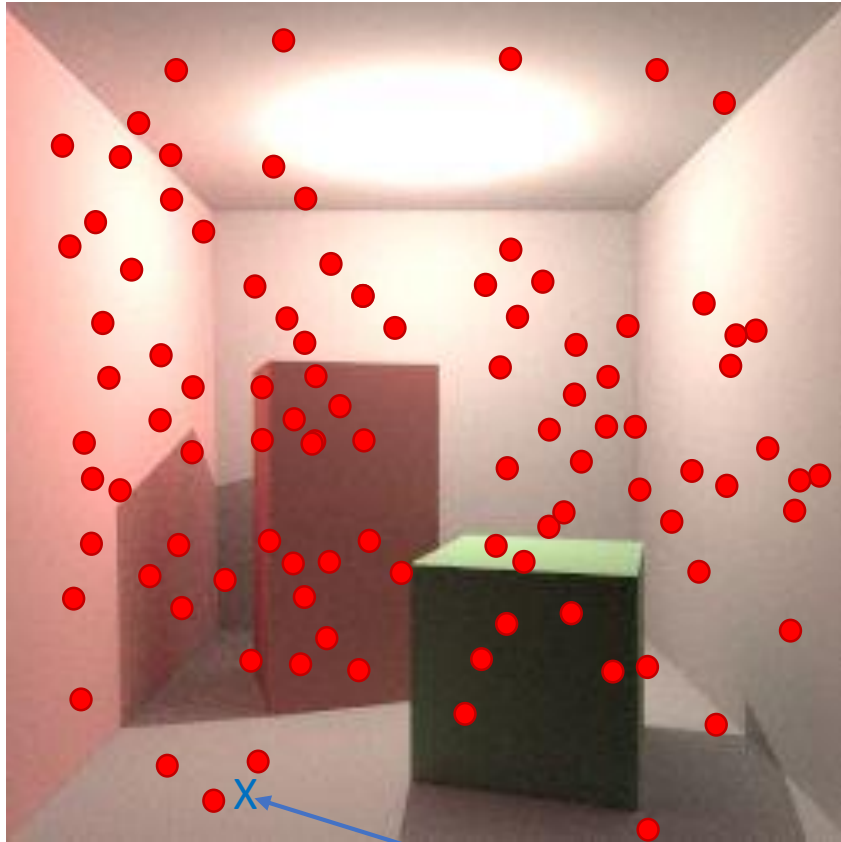
# Photonmapping – Radiance estimate



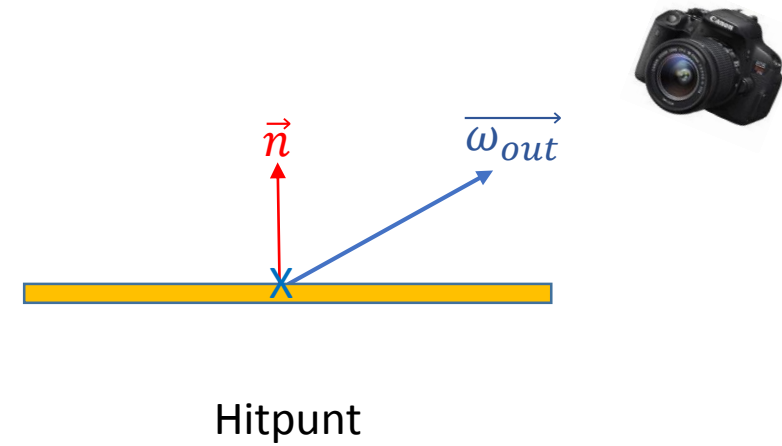
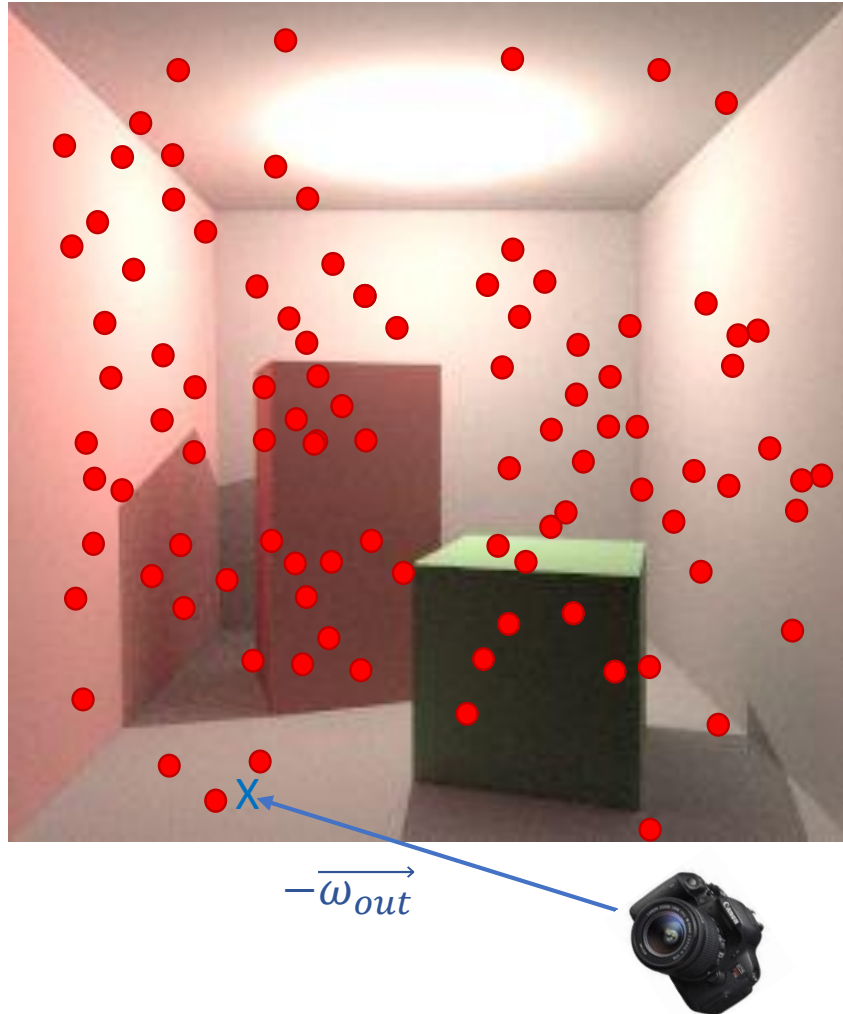
# Photonmapping – Radiance estimate



# Photonmapping – Radiance estimate

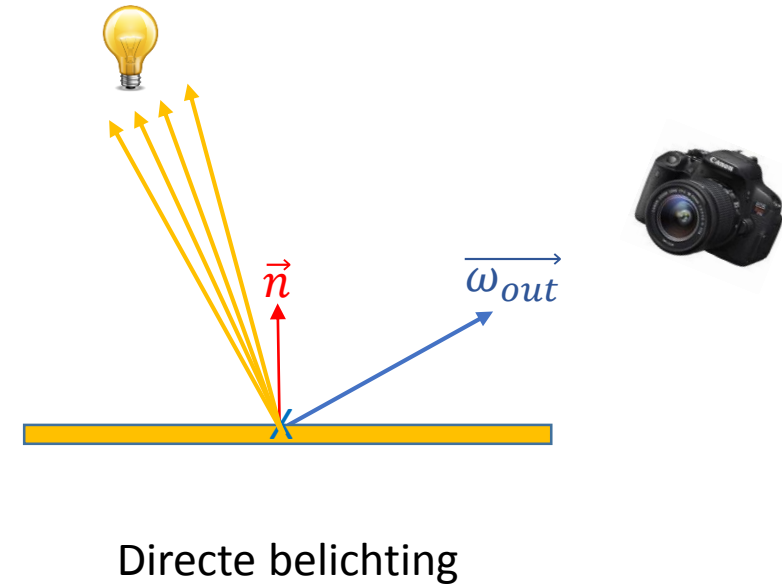
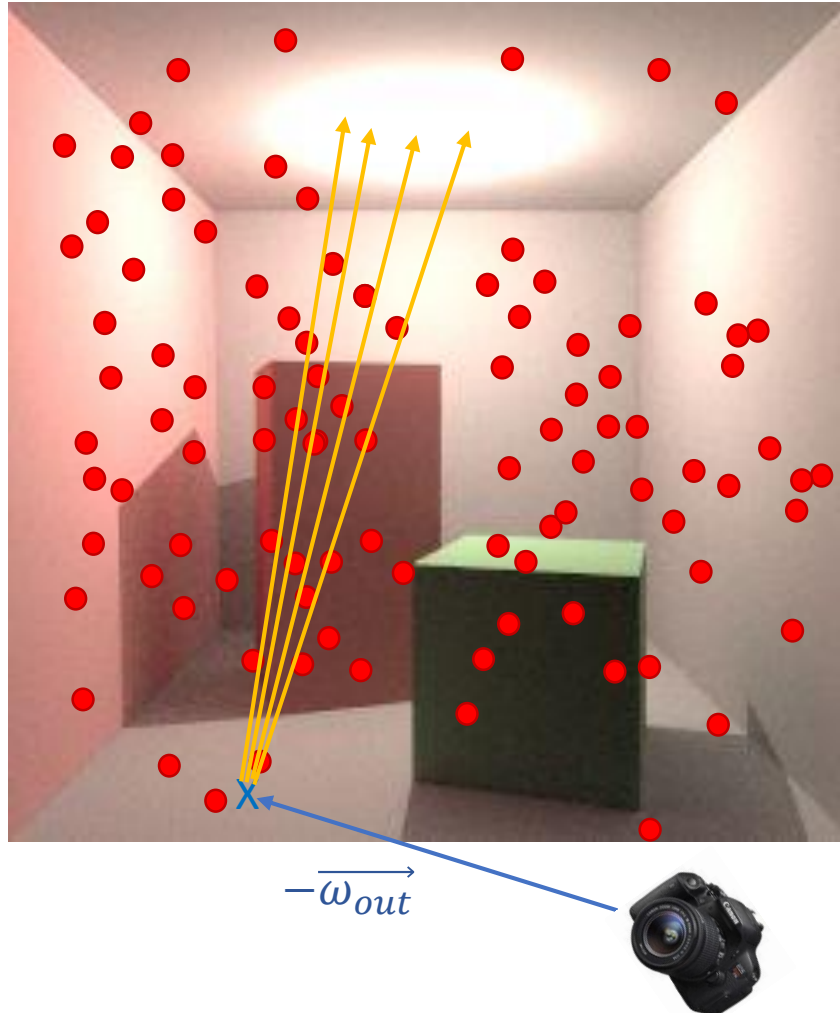


# Photonmapping – Radiance estimate

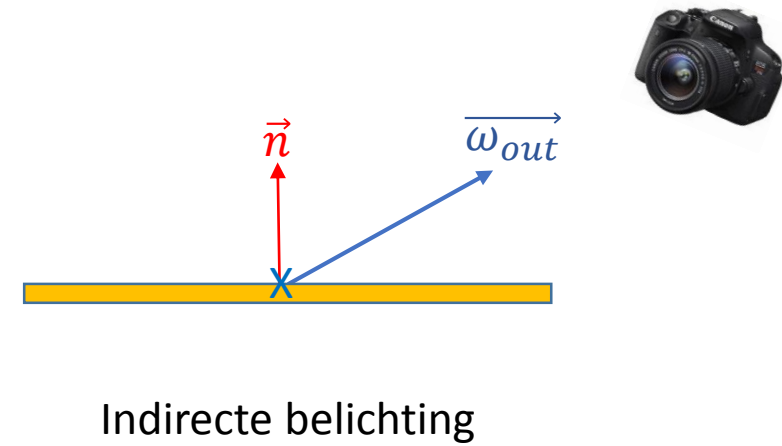
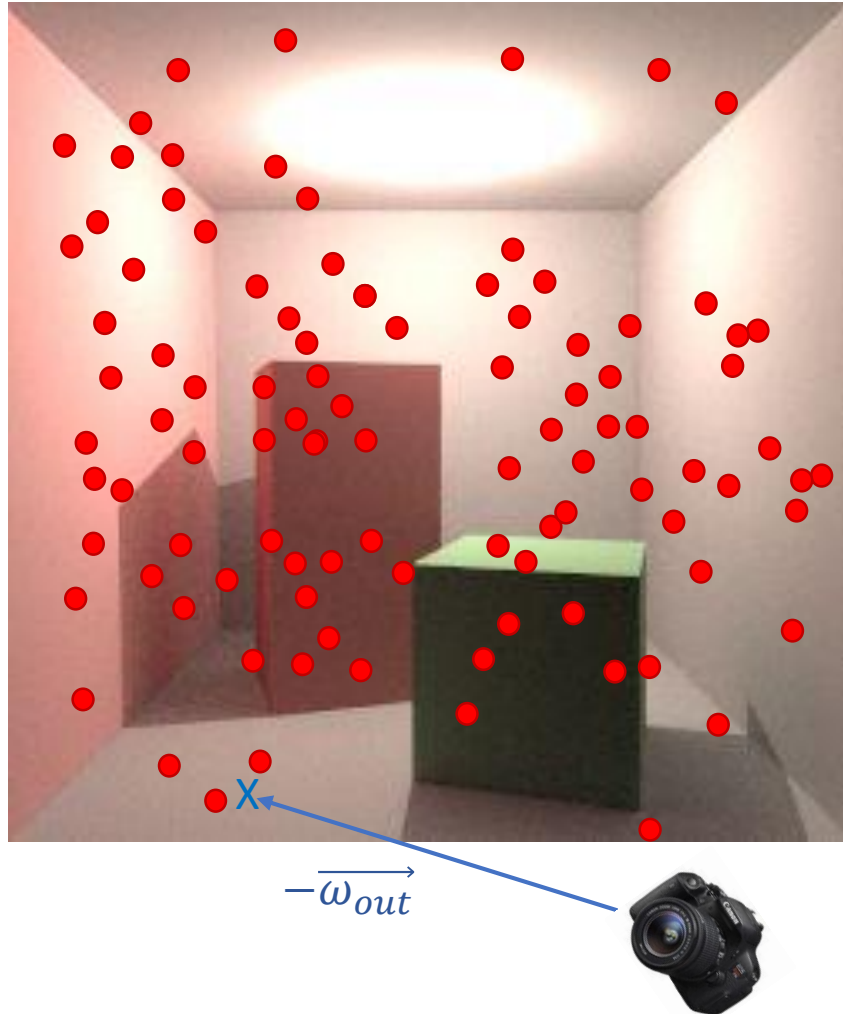




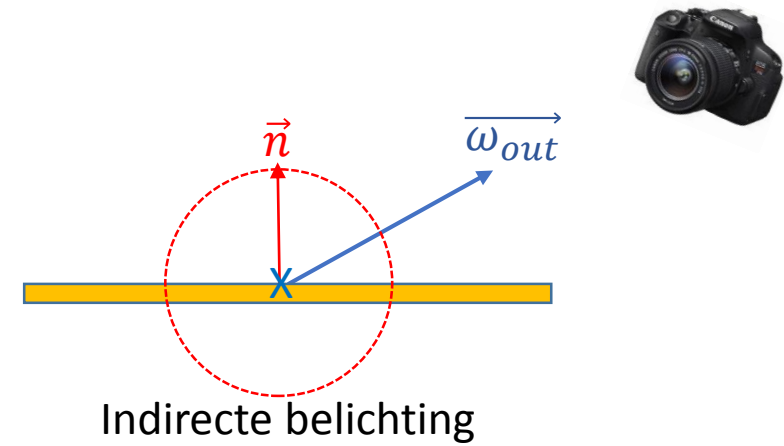
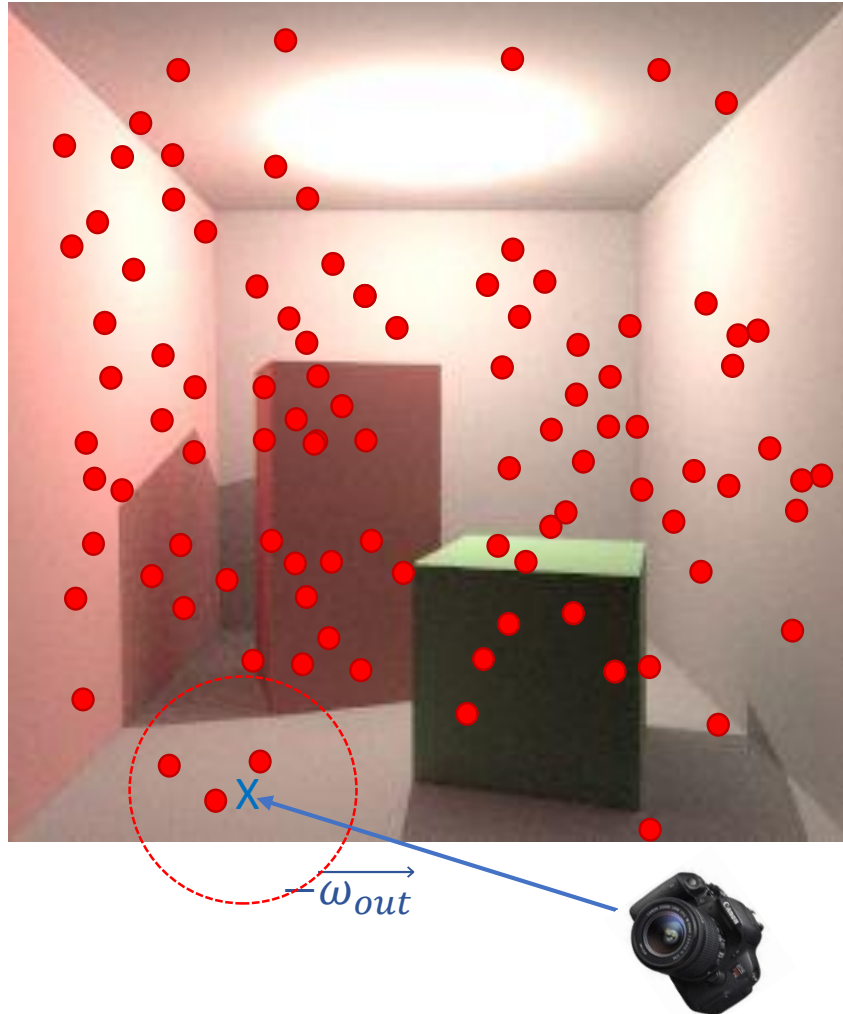
# Photonmapping – Radiance estimate



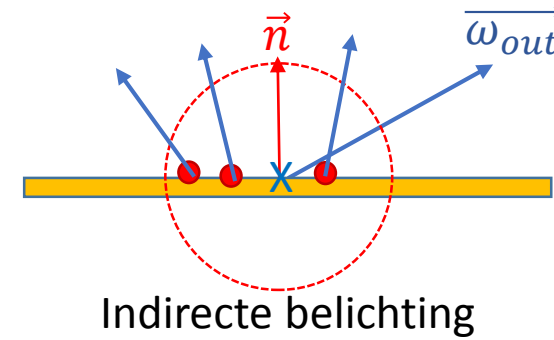
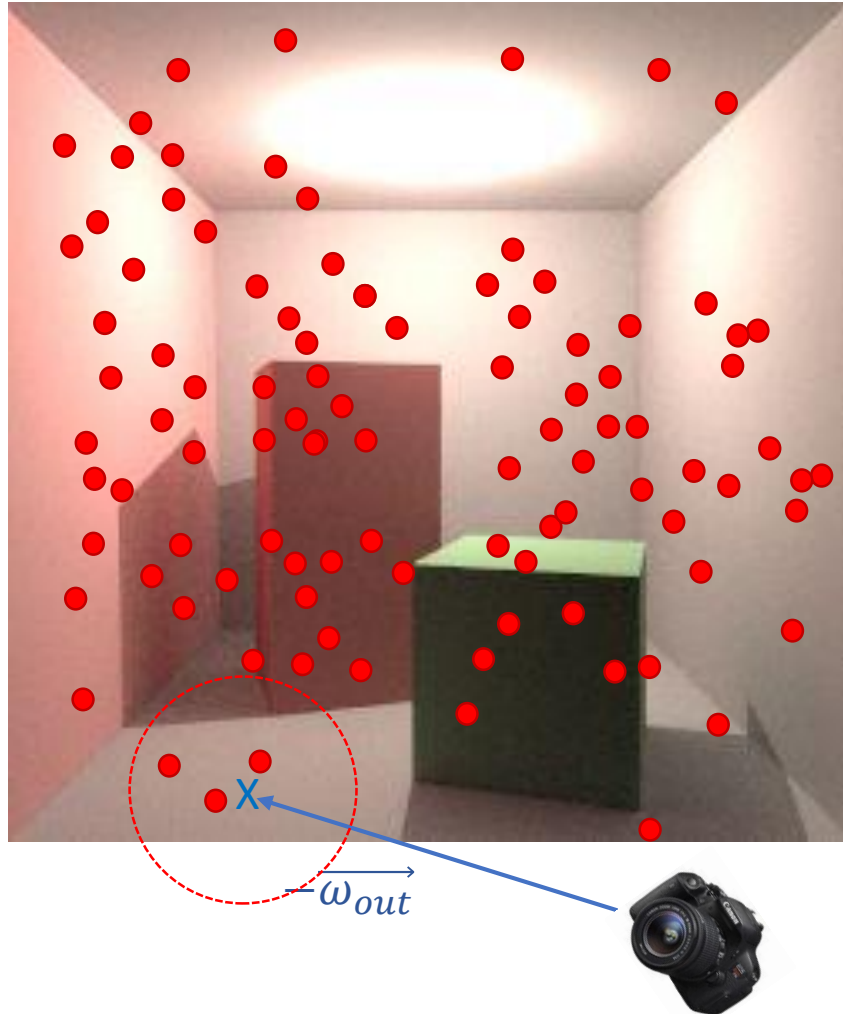
# Photonmapping – Radiance estimate



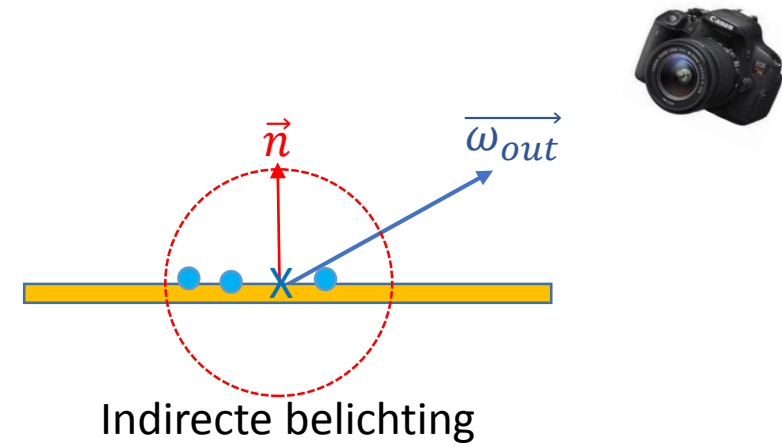
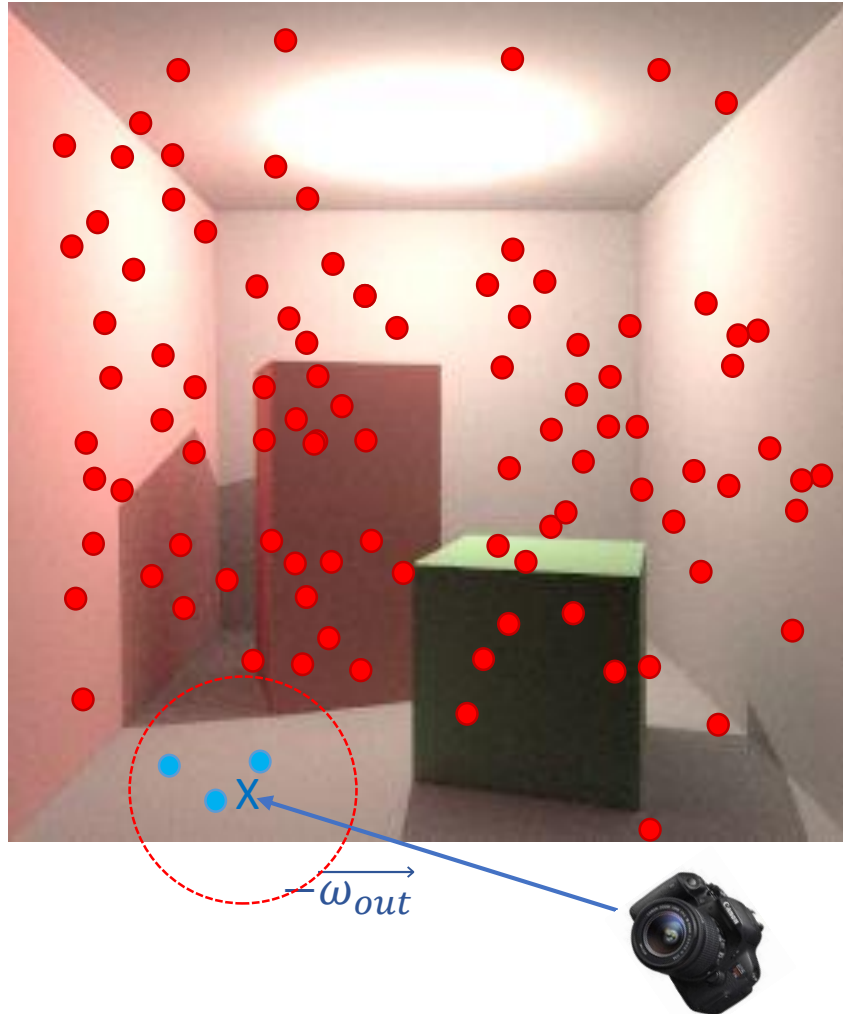
# Photonmapping – Radiance estimate



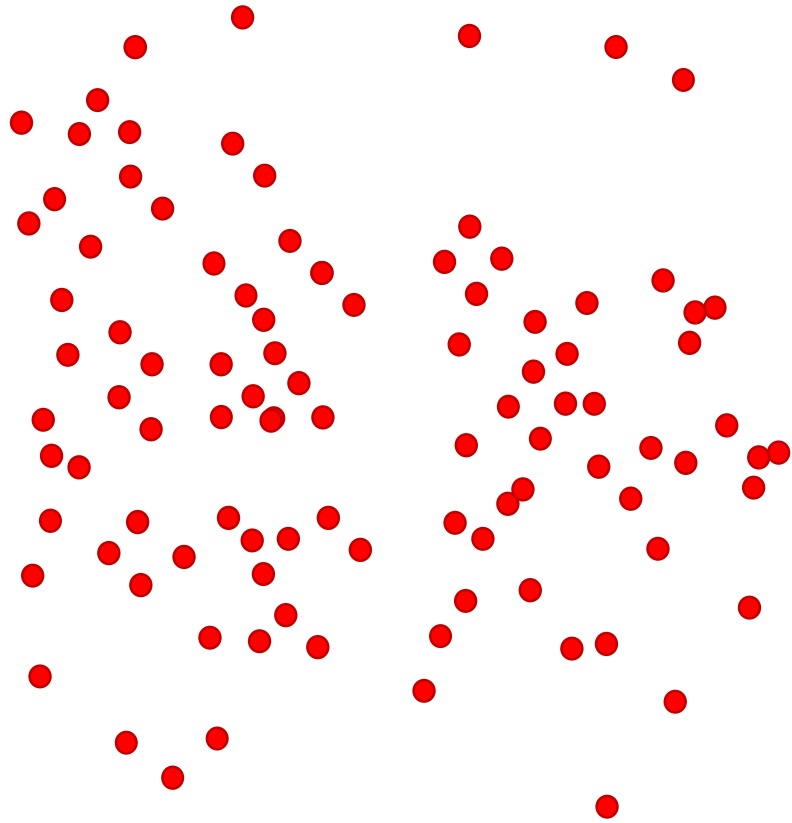
# Photonmapping – Radiance estimate



# Photonmapping – Radiance estimate

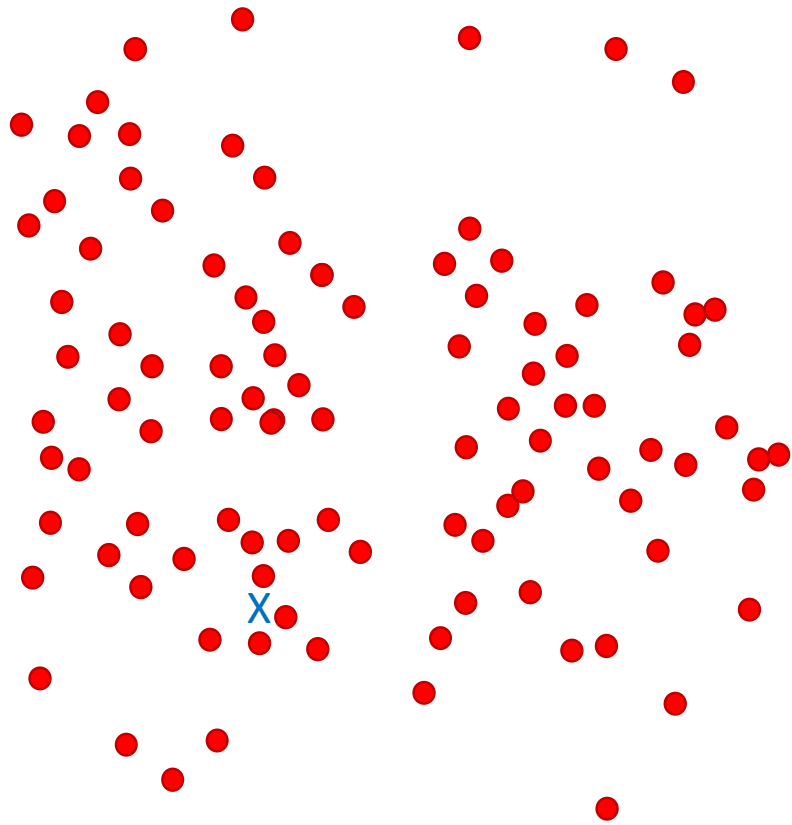


# Photonmapping – Photonmap building



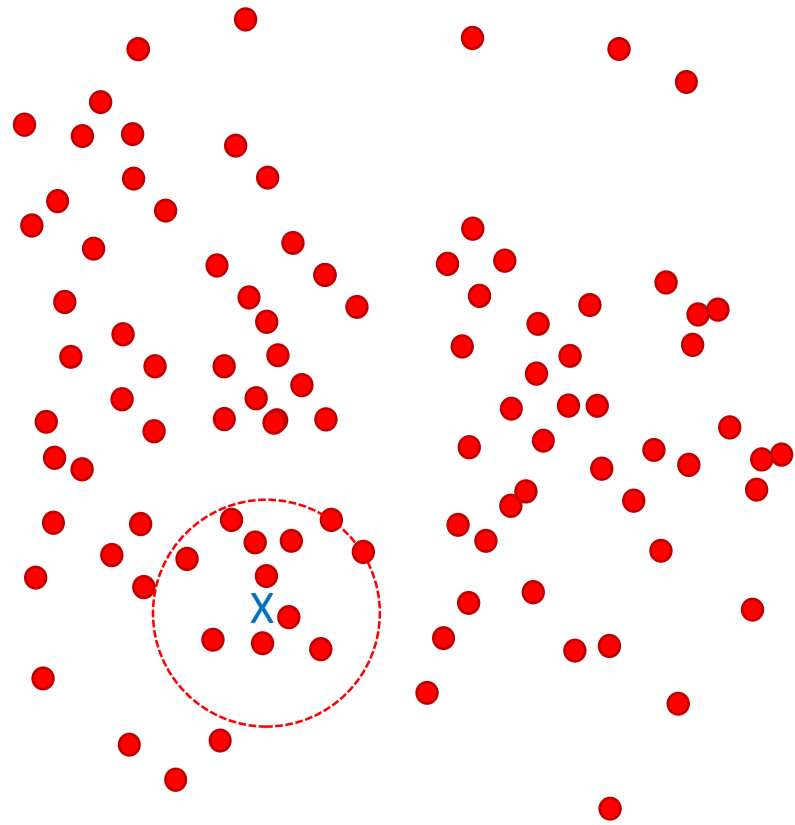
2D ruimte

# Photonmapping – Photonmap building



2D ruimte

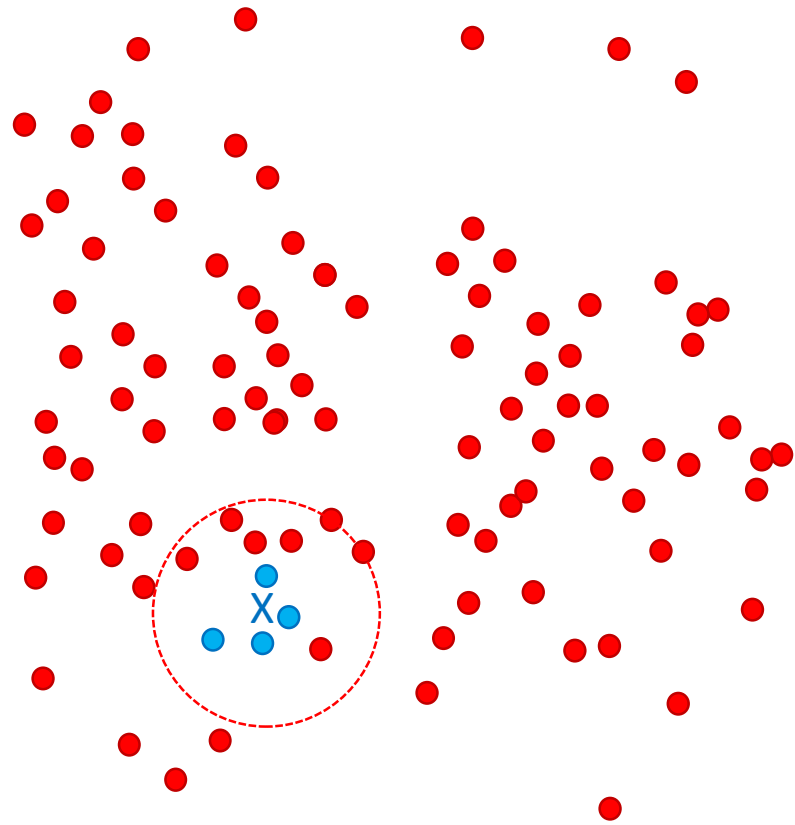
# Photonmapping – Photonmap building



2D ruimte

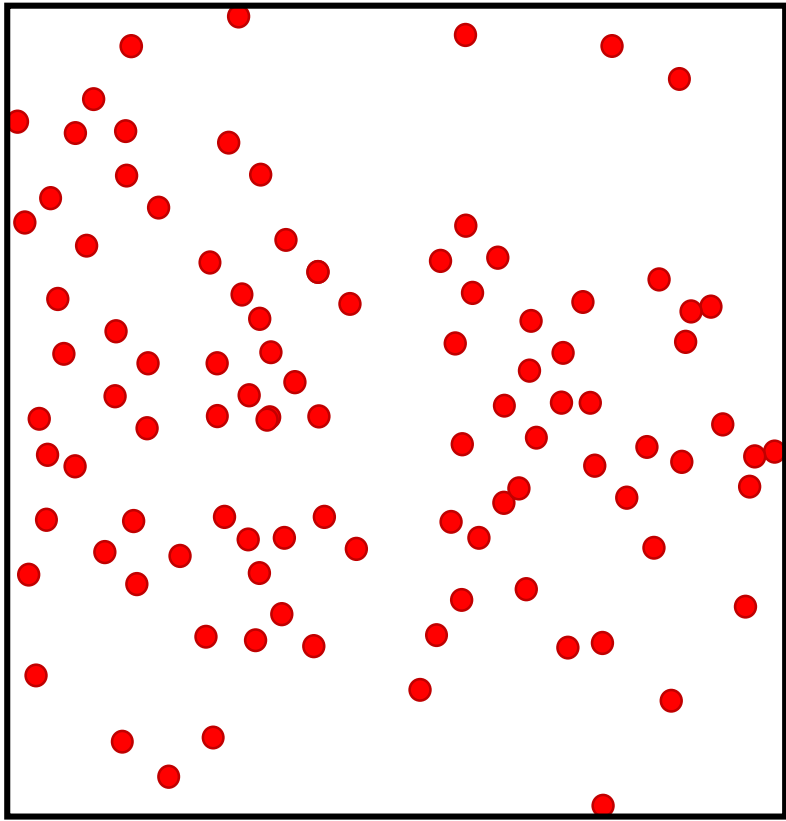


# Photonmapping – Photonmap building



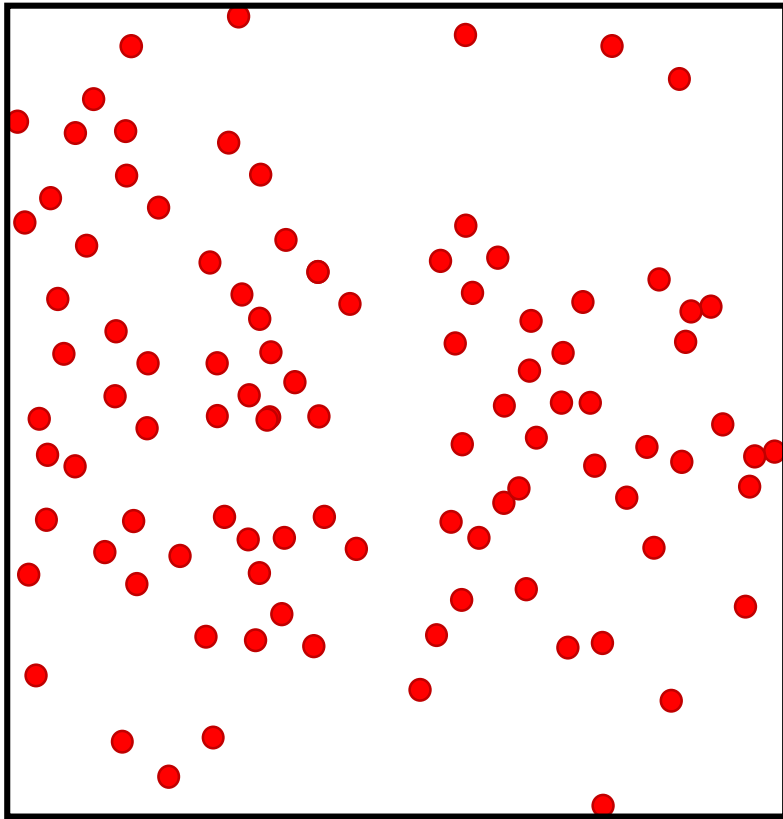
2D ruimte

# Photonmapping – Photonmap building



2D ruimte

# Photonmapping – Photonmap building



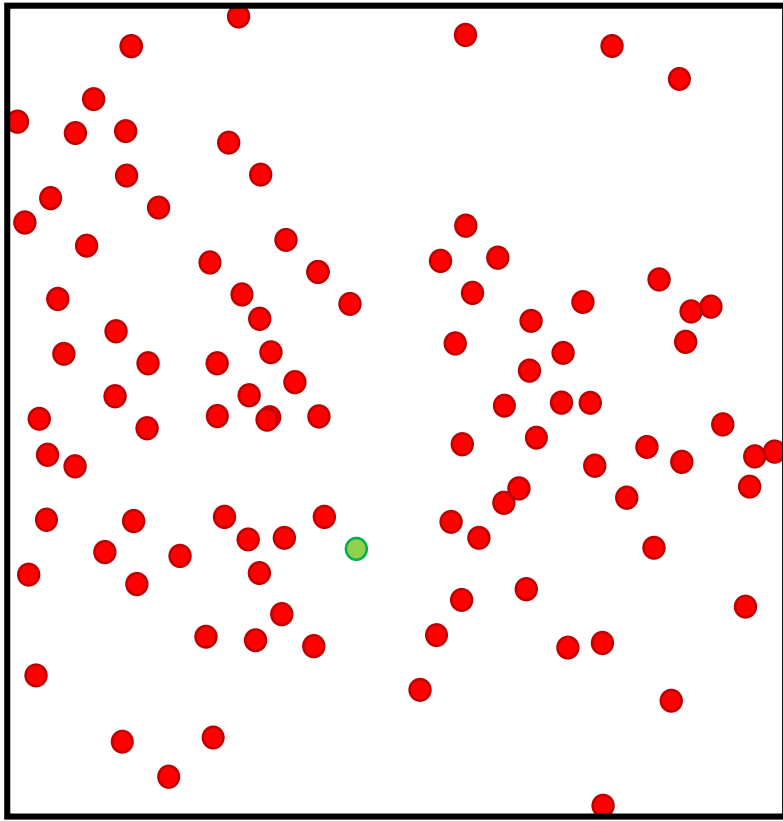
2D ruimte

Kd-boom photonmap

Opsplitsingsheuristiek

- Balanced Heuristic (BH)
- Voxel Volume Heuristic (VVH) [Wald et al. 2004]

# Photonmapping – Photon tracing

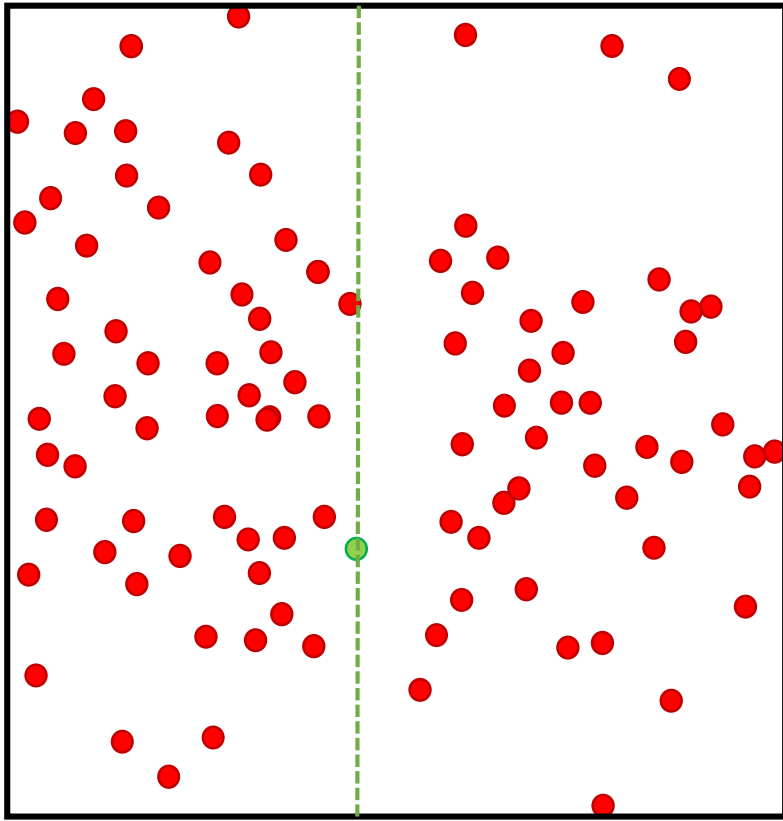


2D ruimte

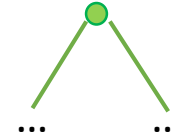


kd-boom datastructuur

# Photonmapping – Photon tracing

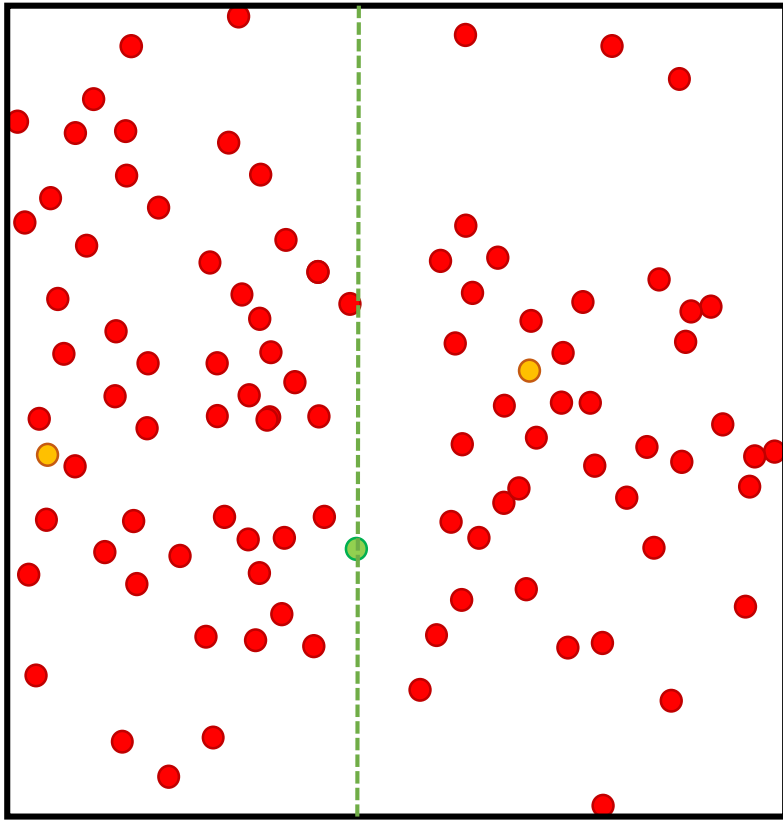


2D ruimte

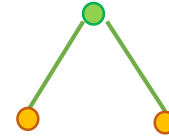


kd-boom datastructuur

# Photonmapping – Photon tracing

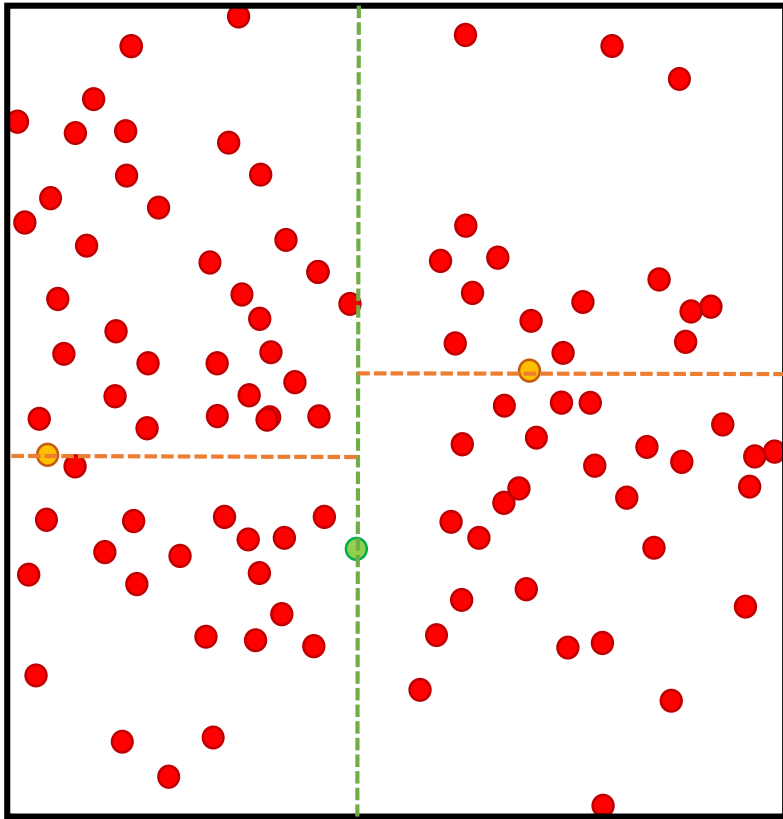


2D ruimte

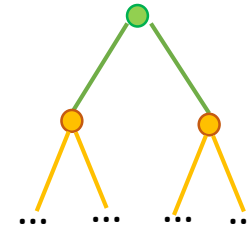


kd-boom datastructuur

# Photonmapping – Photon tracing

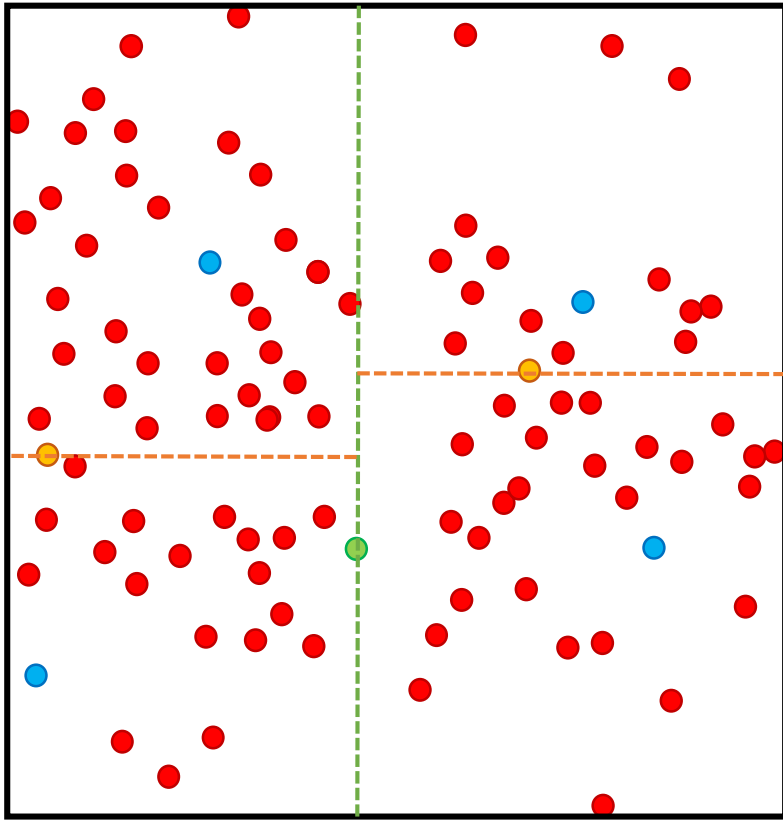


2D ruimte

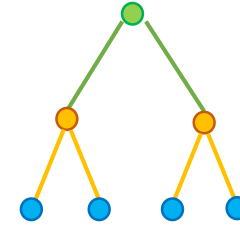


kd-boom datastructuur

# Photonmapping – Photon tracing



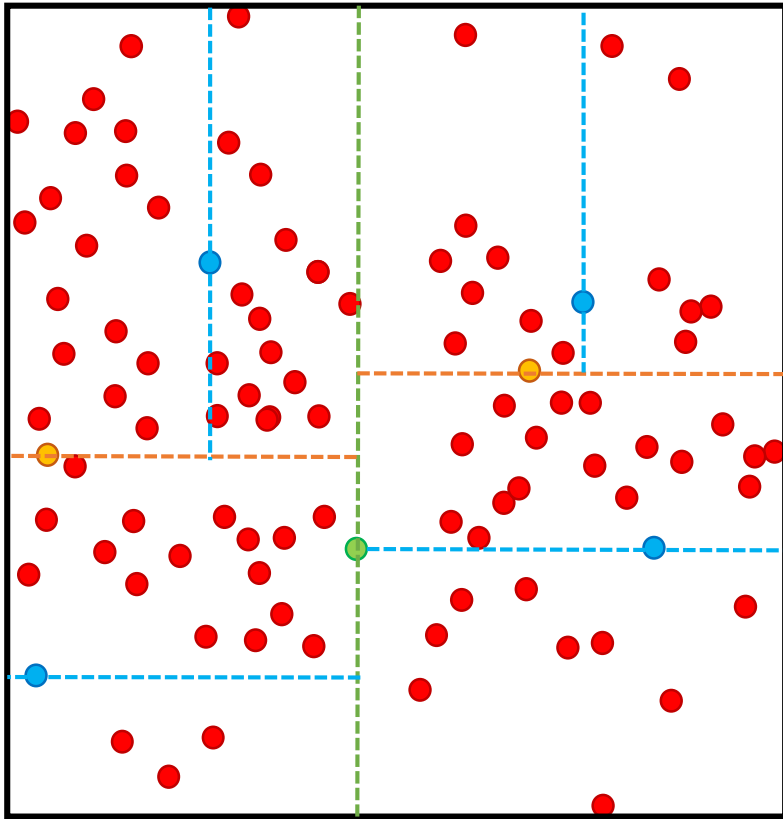
2D ruimte



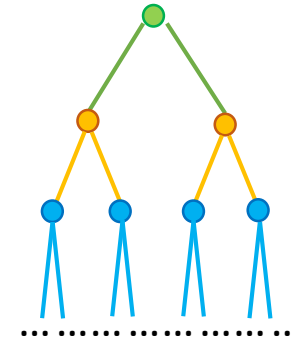
kd-boom datastructuur



# Photonmapping – Photon tracing

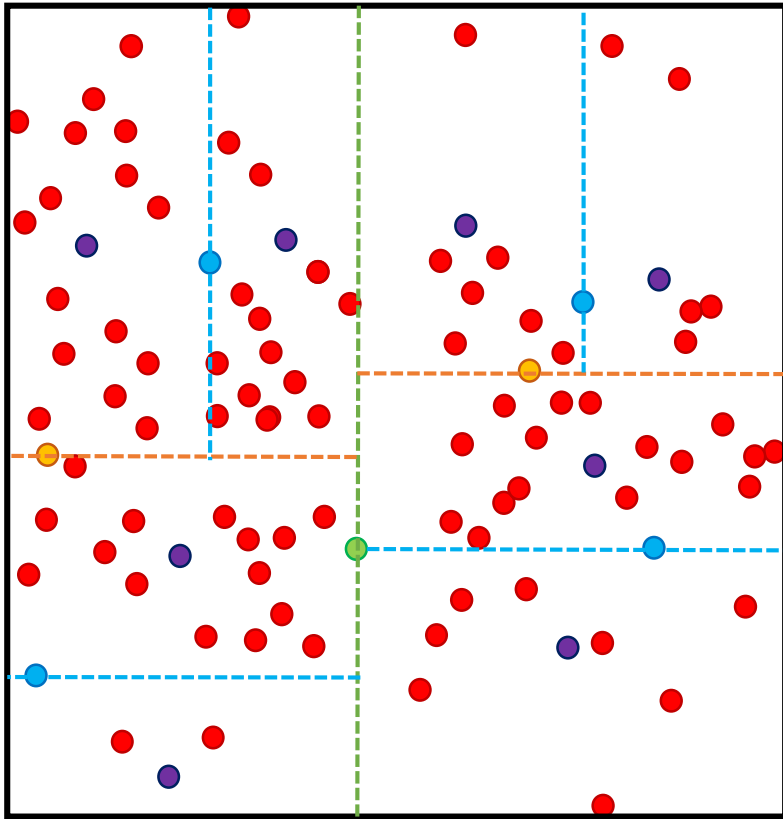


2D ruimte

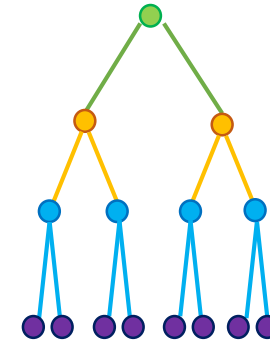


kd-boom datastructuur

# Photonmapping – Photon tracing

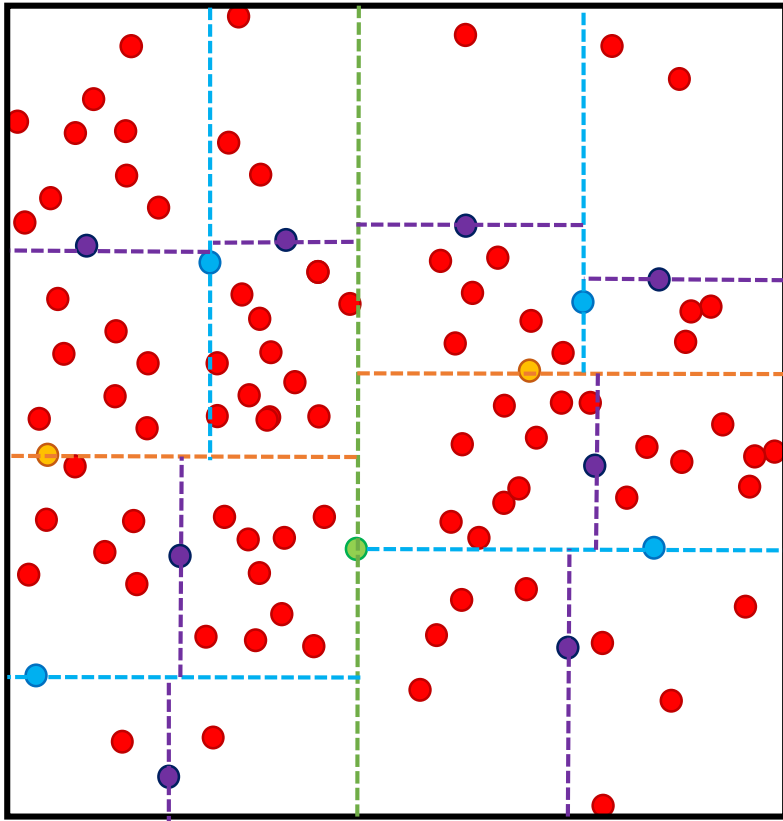


2D ruimte

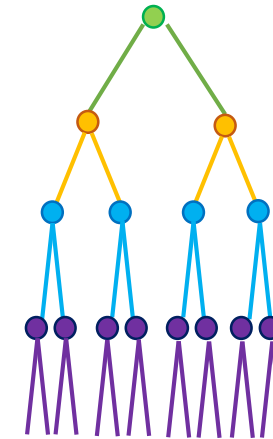


kd-boom datastructuur

# Photonmapping – Photon tracing



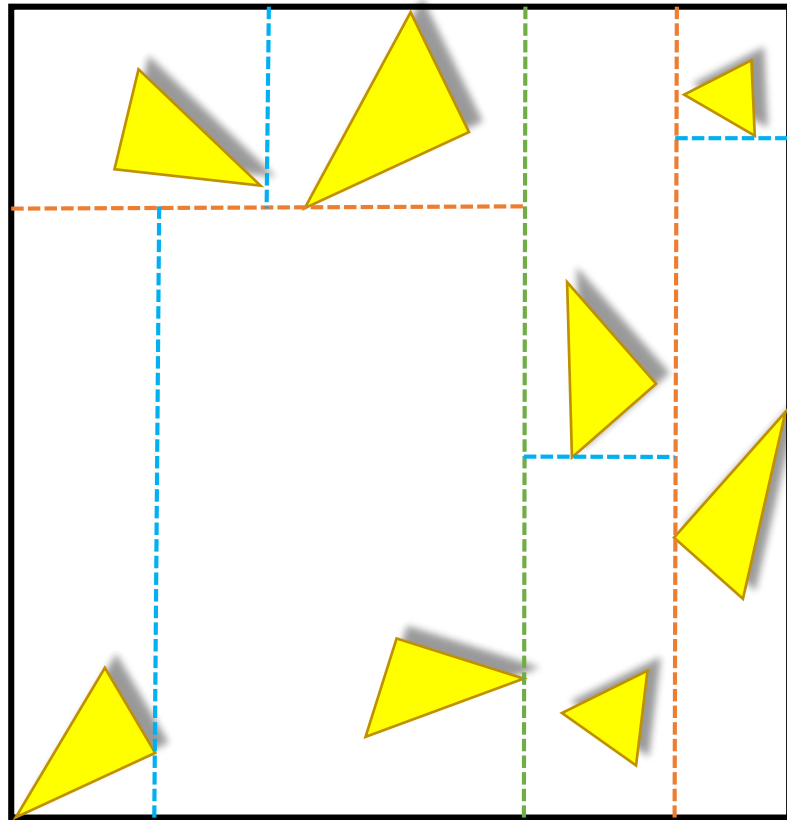
2D ruimte



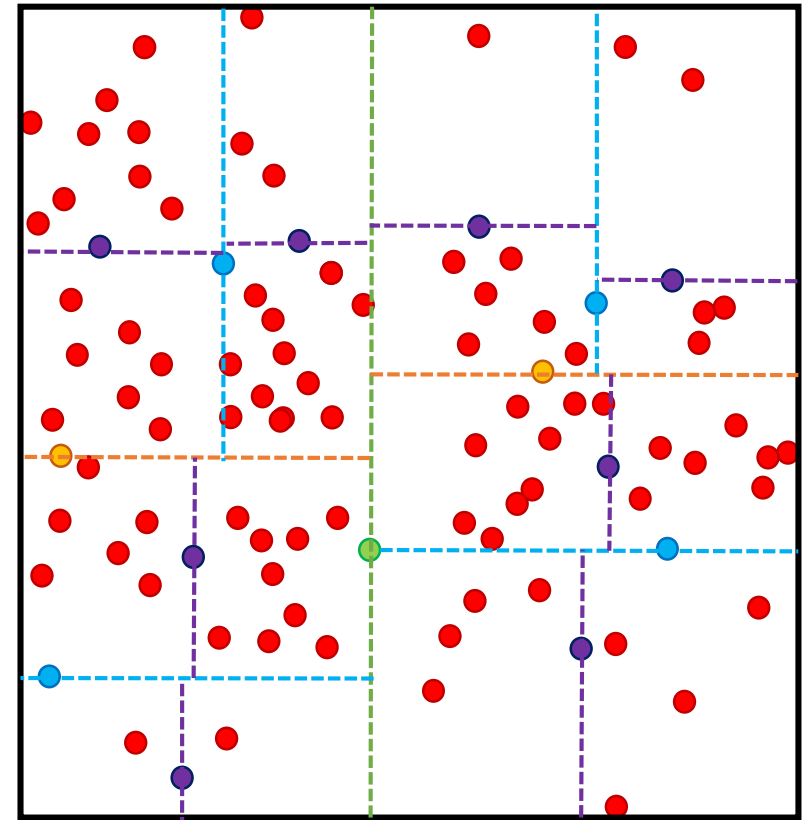
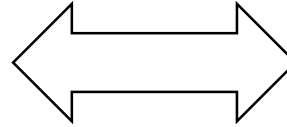
enzovoort

kd-boom datastructuur

# Two aparte kd-bomen

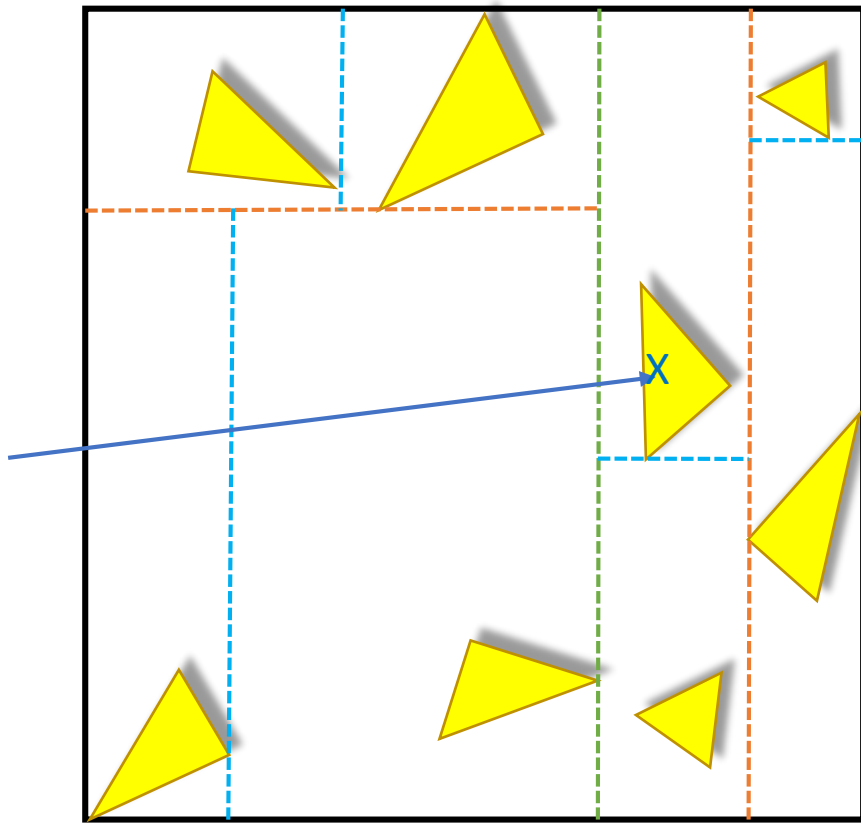


Geometrische primitieven

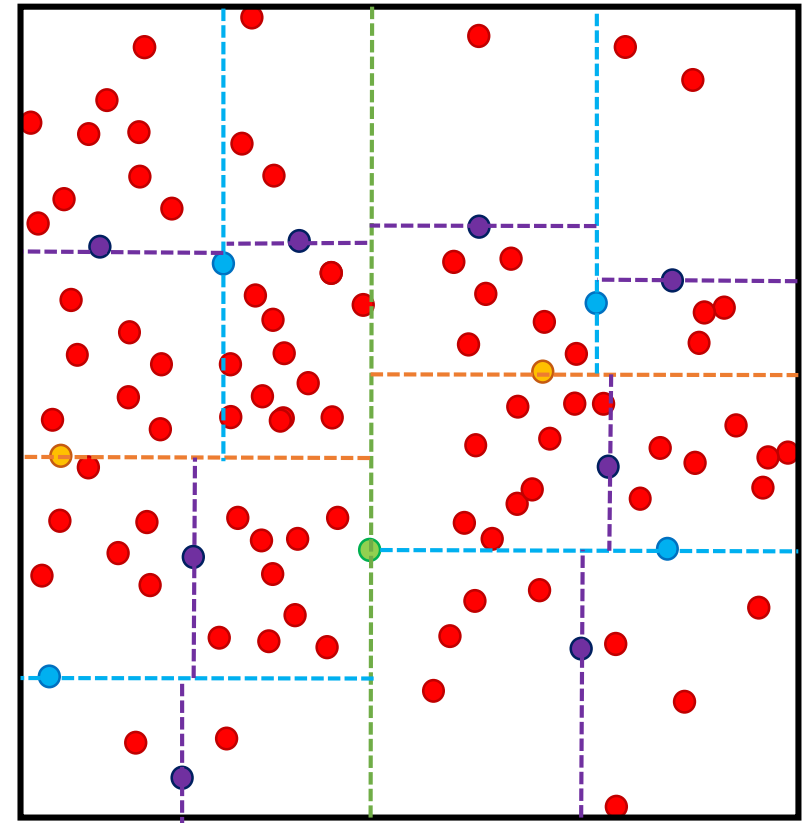
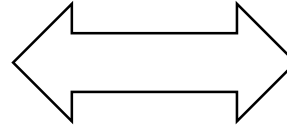


Photons

# Twee aparte kd-bomen

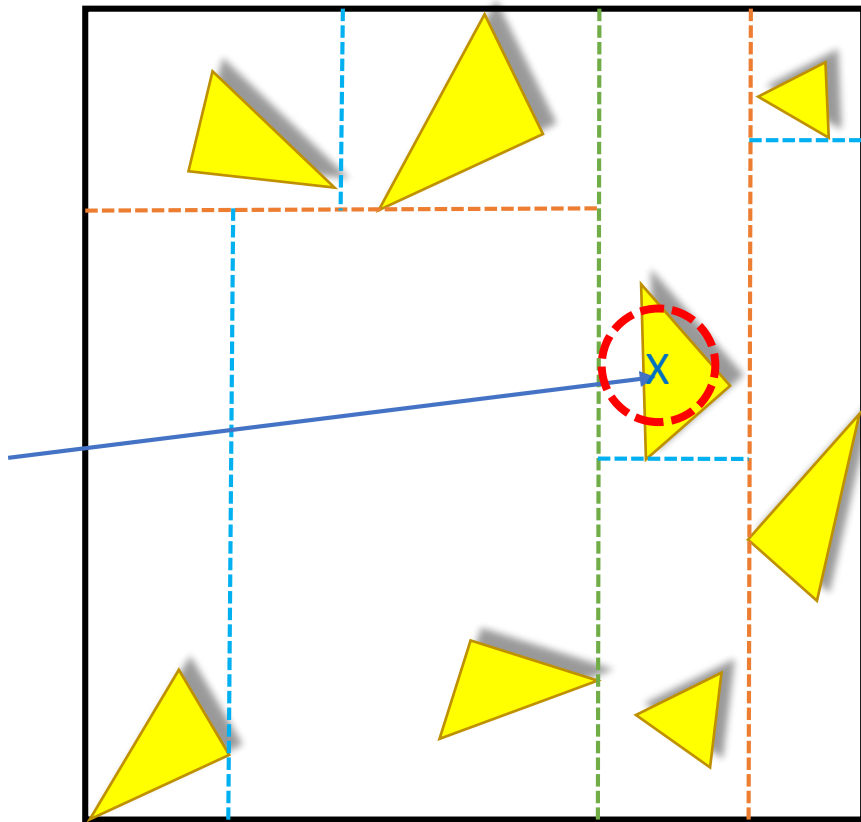


Geometrische primitieven

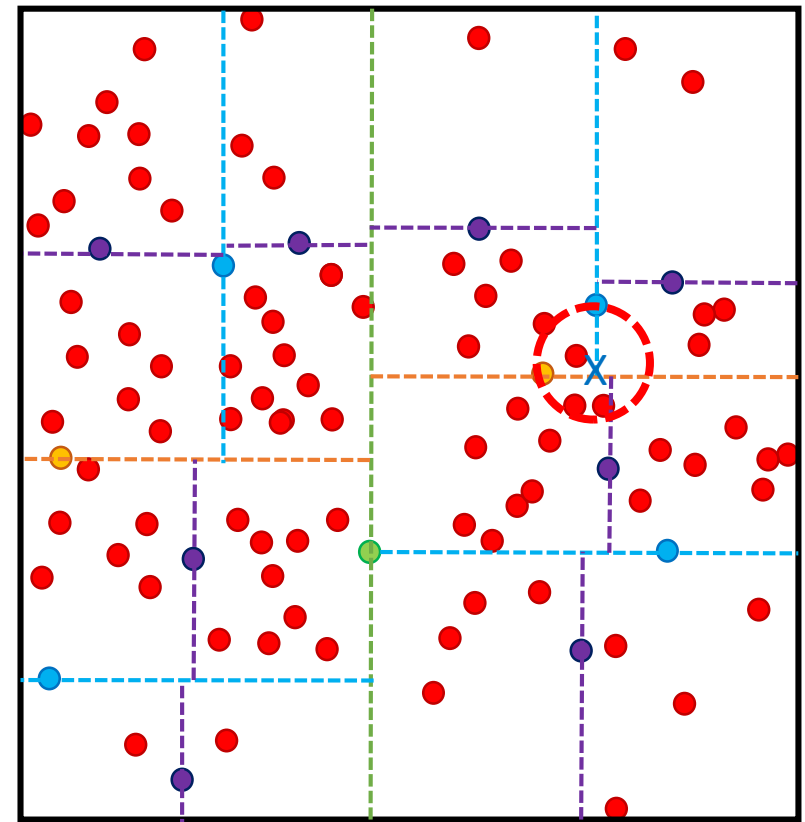


Photons

# Twée aparte kd-bomen

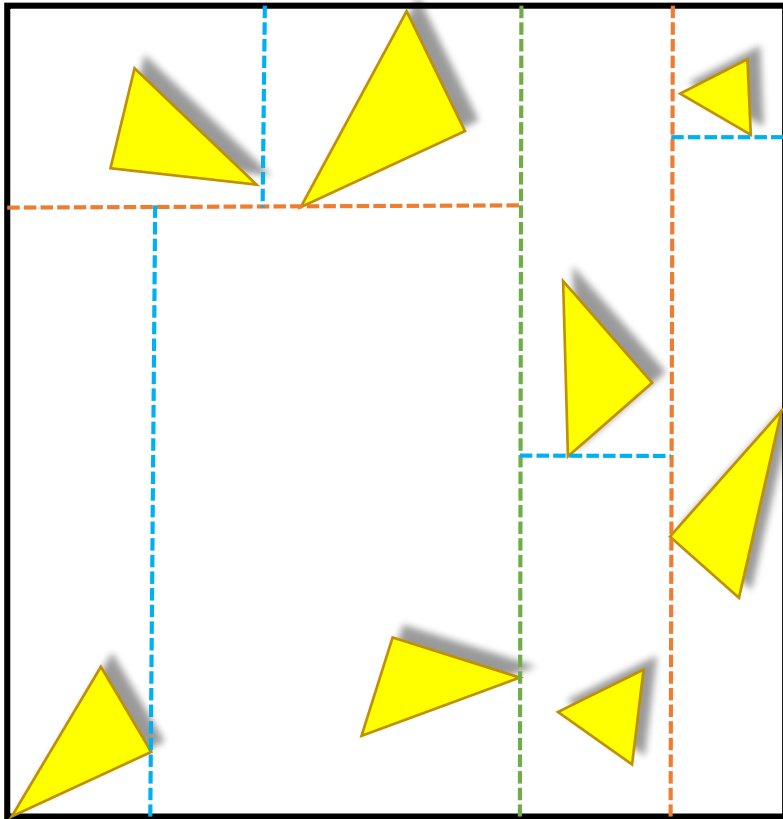


Geometrische primitieven



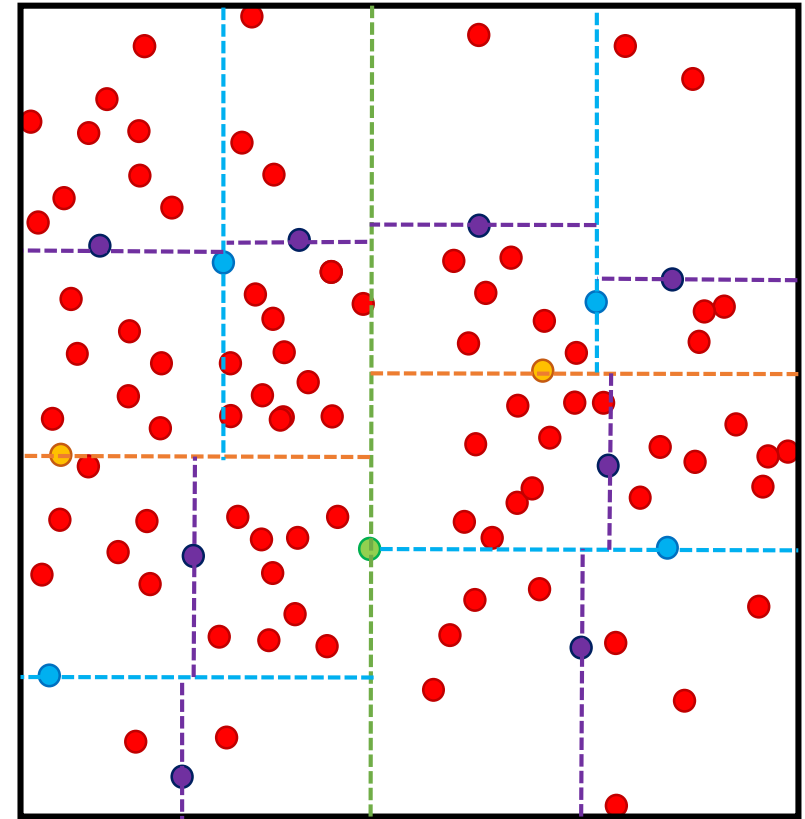
Photons

# Eén kd-boom



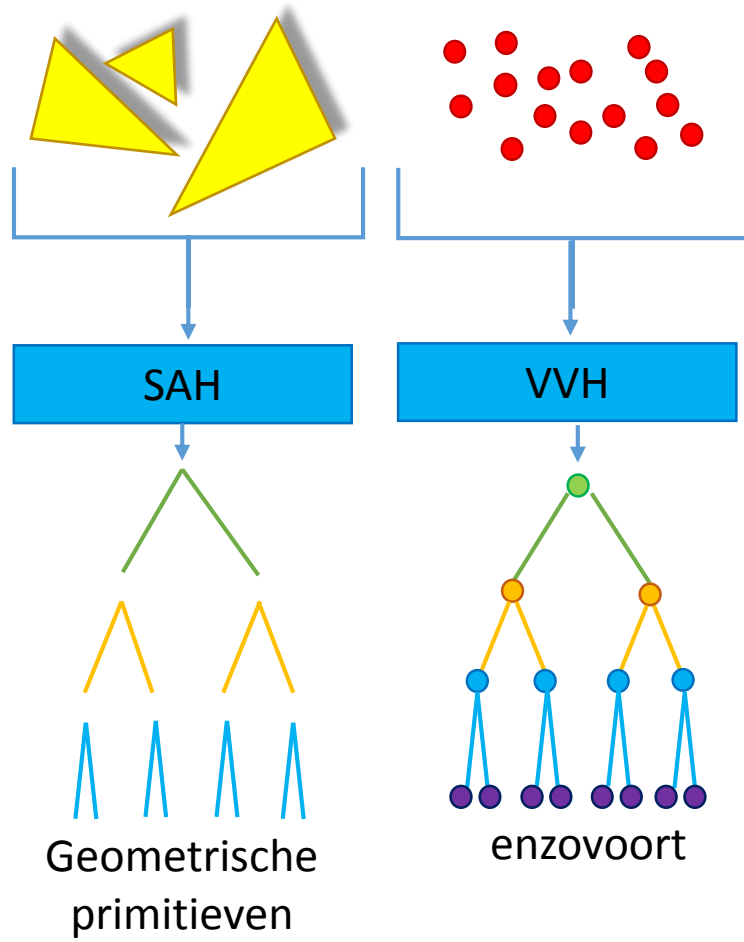
Geometrische primitieven

+



Photons

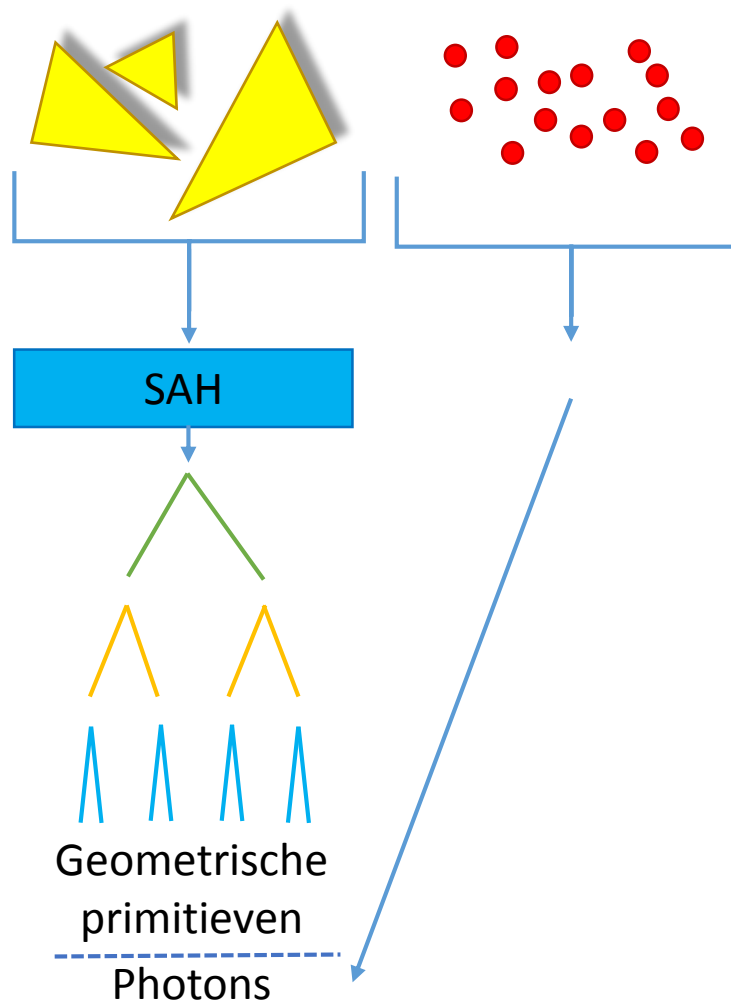
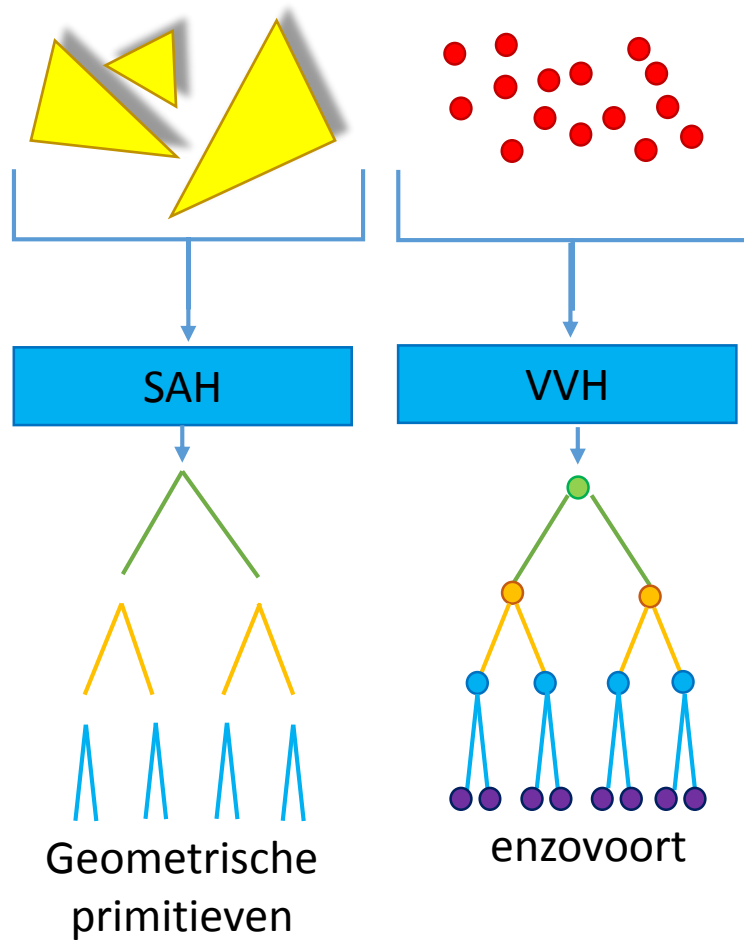
## Photon Map





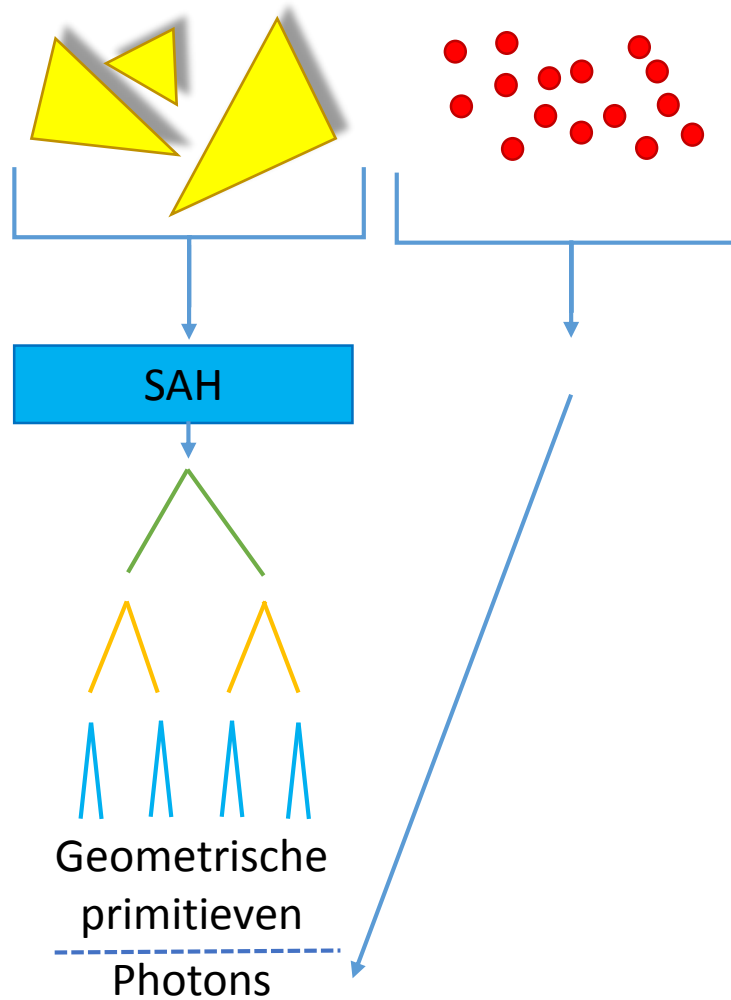
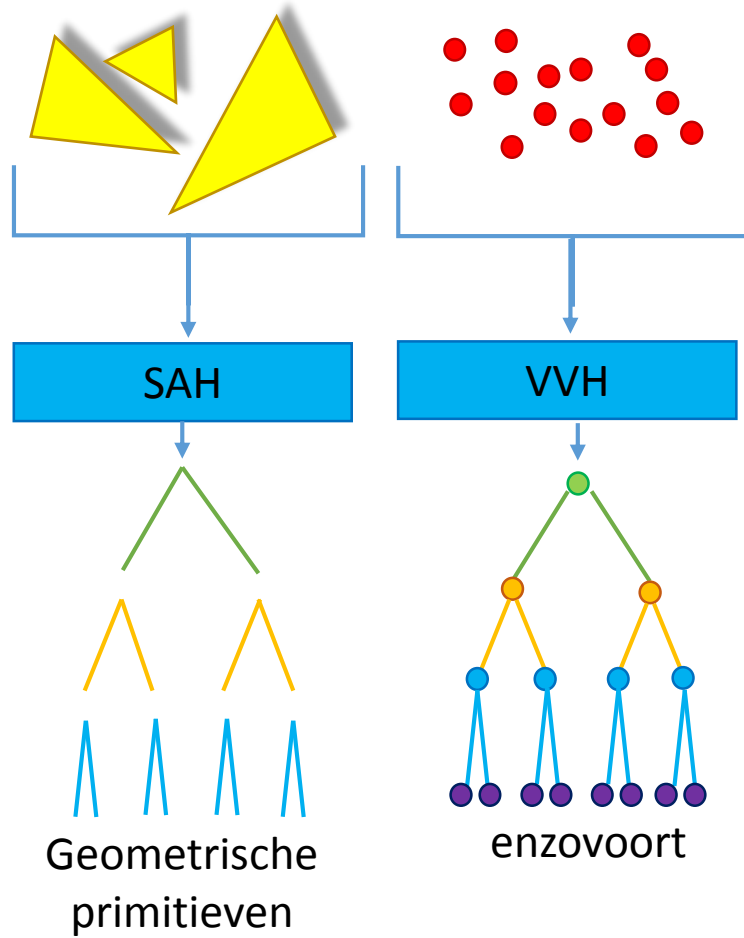
## Photon Map

## Hybrid Additive Photon Map



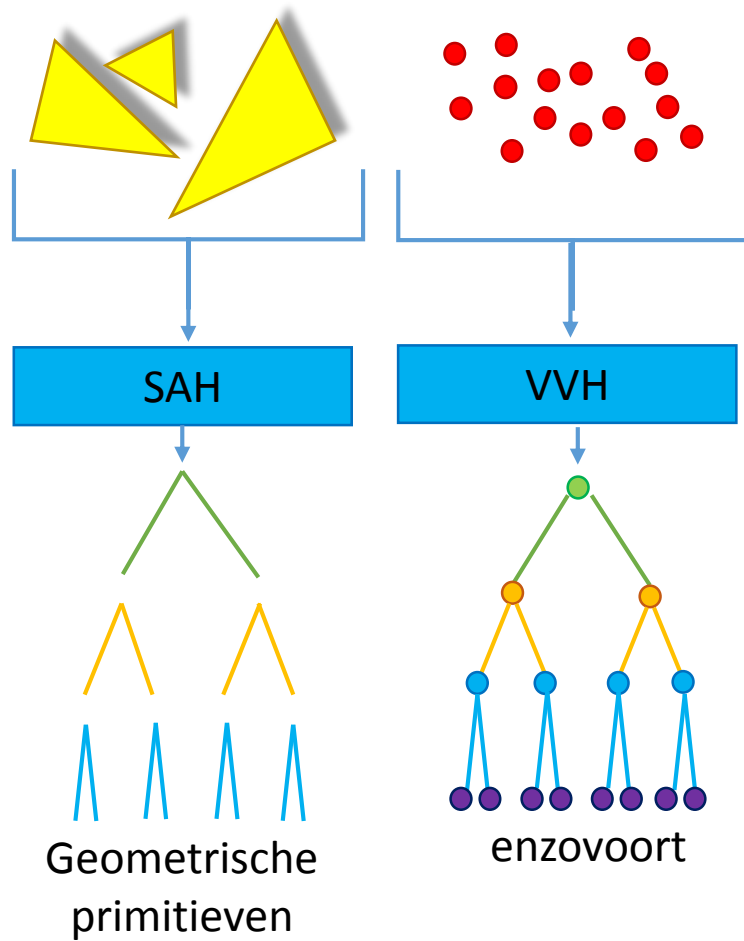
## Photon Map

## Hybrid Additive Photon Map

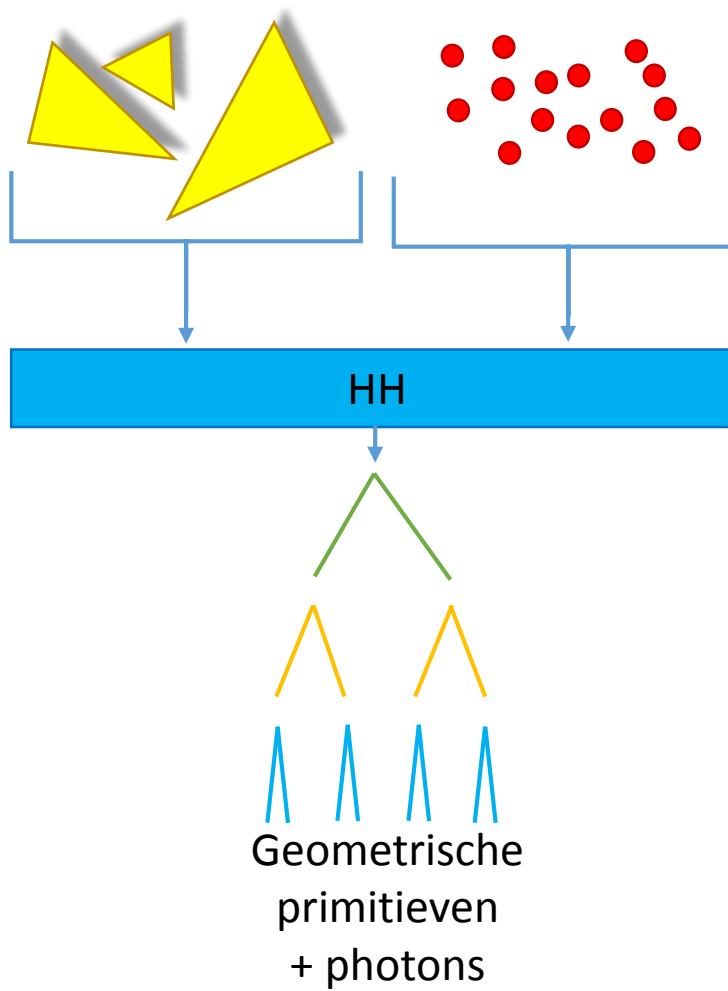


Geen onderscheid tussen ray tracing  
en k-nearest neighbor query's

## Photon Map



## Hybrid Swapping Photon Map

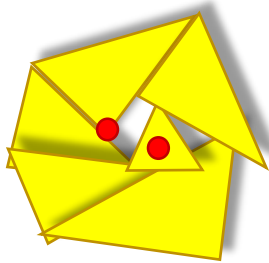


# Hybrid Heuristic (HH)

1. Deel geometrische primitieven en photons op
2. Drop geometrische primitieven en photons

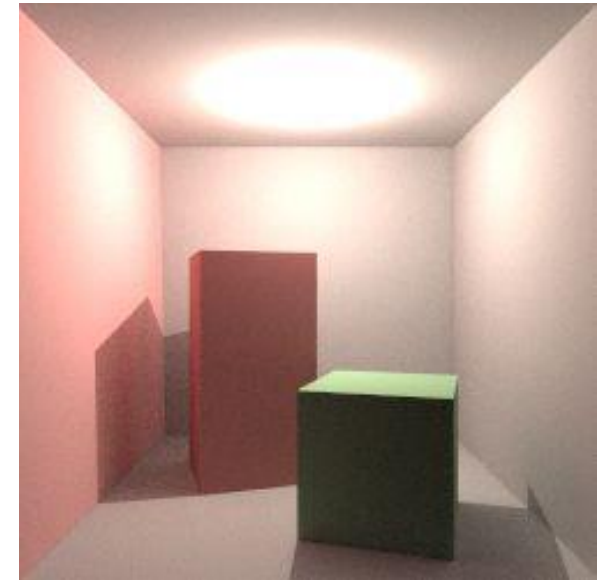
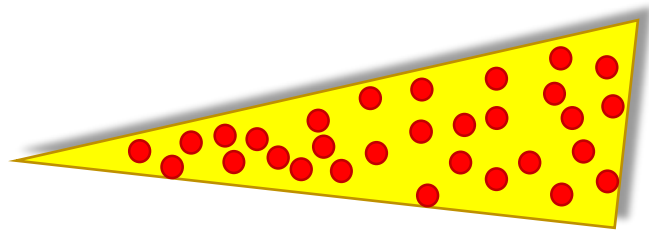
# Hybrid Heuristic (HH)

1. Deel **geometrische primitieven** en **photons** op
2. Drop **geometrische primitieven** en **photons**
3. Deel **geometrische primitieven** op en drop **photons**



# Hybrid Heuristic (HH)

1. Deel **geometrische primitieven** en **photons** op
2. Drop **geometrische primitieven** en **photons**
3. Deel **geometrische primitieven** op en drop **photons**
4. Drop **geometrische primitieven** en splits **photons** op



# Hybrid Heuristic (HH)

1. Deel geometrische primitieven en photons op
  2. Drop geometrische primitieven en photons
  3. Deel geometrische primitieven op en drop photons
  4. Drop geometrische primitieven en splits photons op
- > Ray tracing en k-nearest neighbor query's gekoppeld via hitpunt

# Hybrid Heuristic (HH)

1. Deel geometrische primitieven en photons op
  2. Drop geometrische primitieven en photons
  3. Deel geometrische primitieven op en drop photons
  4. Drop geometrische primitieven en splits photons op
- > Ray tracing en k-nearest neighbor query's gekoppeld via hitpunt
- > Kosten moeten vergelijkbaar zijn



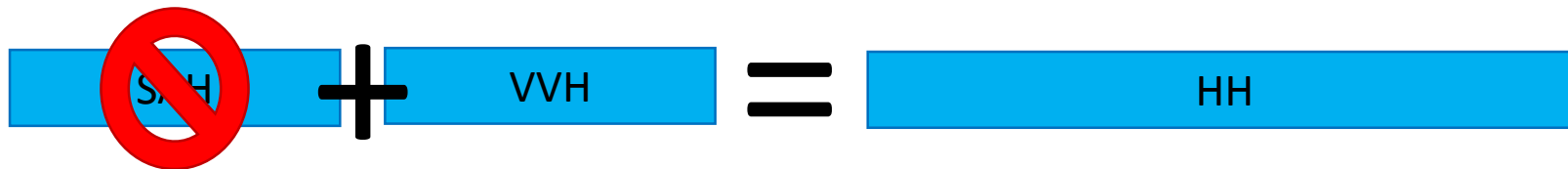
# Hybrid Heuristic (HH)

1. Deel **geometrische primitieven** en **photons** op
  2. Drop **geometrische primitieven** en **photons**
  3. Deel **geometrische primitieven** op en drop **photons**
  4. Drop **geometrische primitieven** en splits **photons** op
- > Ray tracing en k-nearest neighbor query's gekoppeld via hitpunt
- > Kosten moeten vergelijkbaar zijn

$$\text{SAH} + \text{VVH} = \text{HH}$$

# Hybrid Heuristic (HH)

1. Deel **geometrische primitieven** en **photons** op
  2. Drop **geometrische primitieven** en **photons**
  3. Deel **geometrische primitieven** op en drop **photons**
  4. Drop **geometrische primitieven** en splits **photons** op
- > Ray tracing en k-nearest neighbor query's gekoppeld via hitpunt
- > Kosten moeten vergelijkbaar zijn



# Hybrid Heuristic (HH)

1. Deel **geometrische primitieven** en **photons** op
  2. Drop **geometrische primitieven** en **photons**
  3. Deel **geometrische primitieven** op en drop **photons**
  4. Drop **geometrische primitieven** en splits **photons** op
- > Ray tracing en k-nearest neighbor query's gekoppeld via hitpunt
- > Kosten moeten vergelijkbaar zijn

$$\text{RTSAH} + \text{VVH} = \text{HH}$$

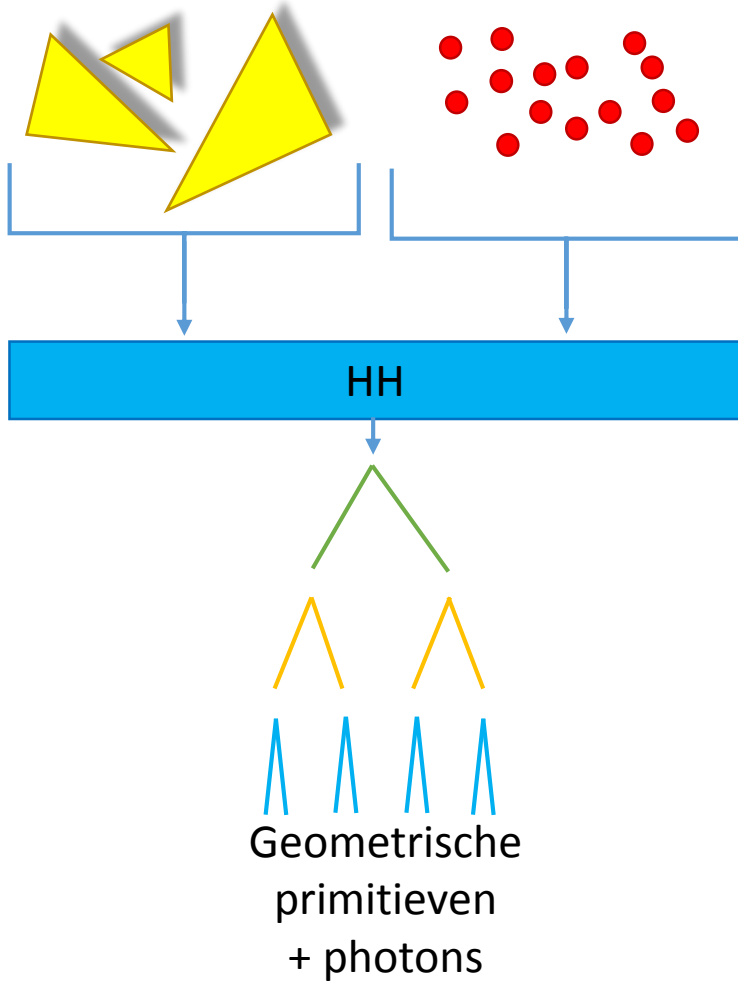


# Efficient visibility heuristics for kd-trees using the RTSAH

**Matthias Moulin, Niels Billen and Philip Dutré**

Department of Computer Science  
KU Leuven, Belgium

## Hybrid Swapping Photon Map



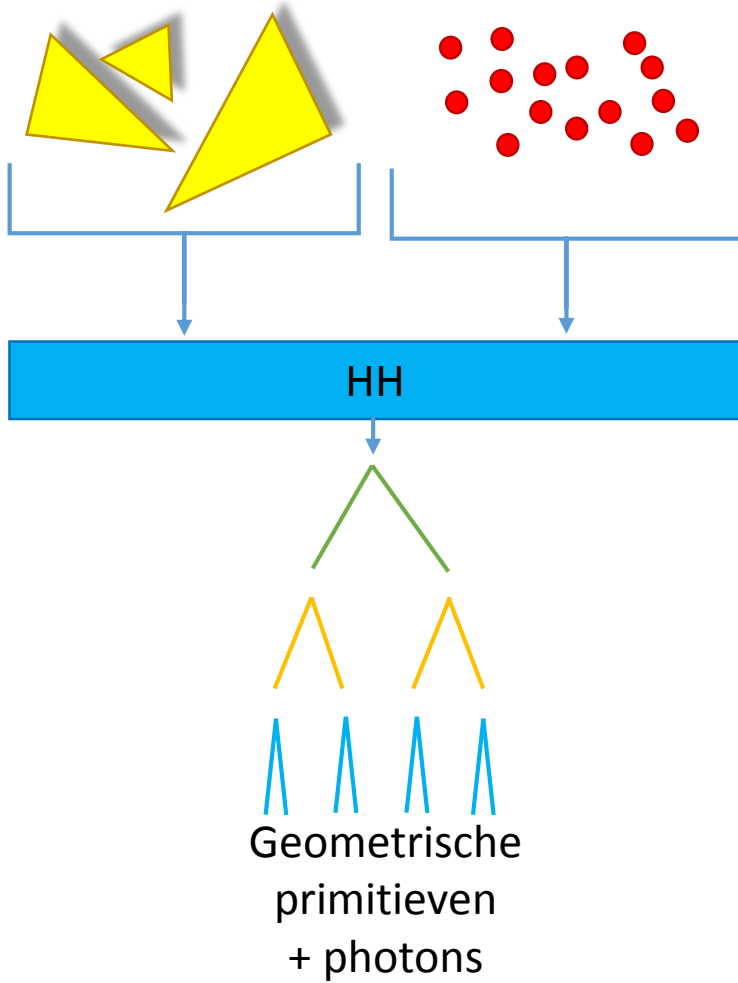
## Speed up

- Vroege terminatie via omsluitende voxel
- Photons opdelen volgens geometrische primitieven

## Helaas...

- Nieuwe acceleratiestructuur bouwen
- Lange bouwtijd
- Hoger geheugengebruik
- Puntdata versus niet-puntdata
- Ongekende informatie tijdens constructie
- Ray tracing versus k-nearest neighbor query's

## Hybrid Swapping Photon Map



## Speed up

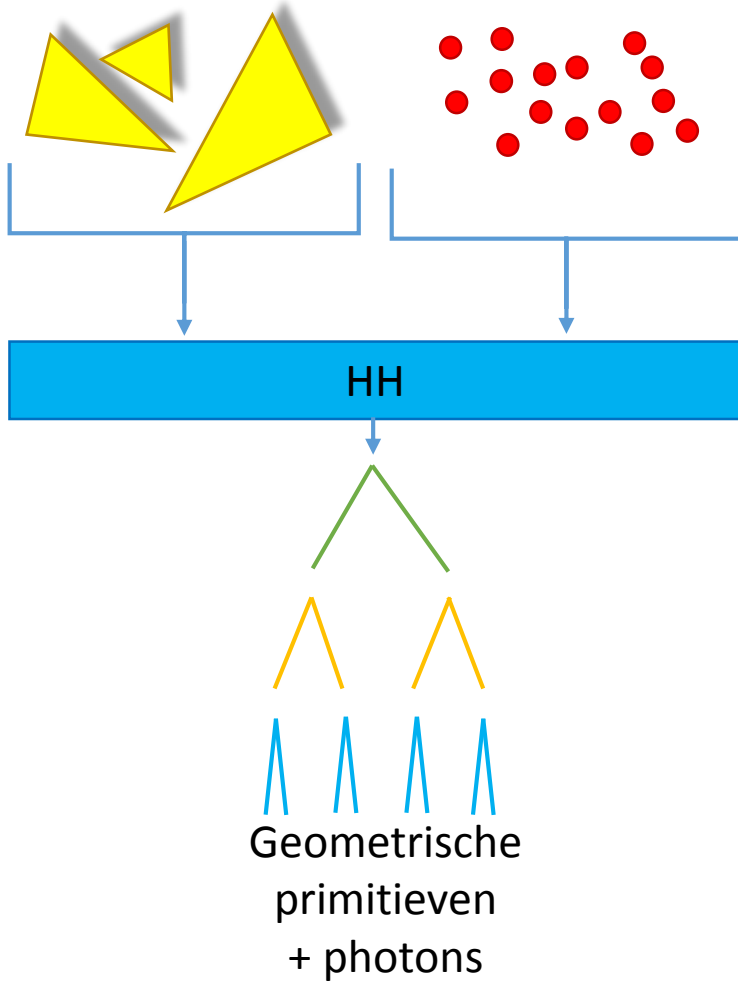
- Vroege terminatie via omsluitende voxel
- Photons opdelen volgens geometrische primitieven

## Helaas...

- Nieuwe acceleratiestructuur bouwen
- Lange bouwtijd
- Hoger geheugengebruik
- Puntdata versus niet-puntdata
- Ongekende informatie tijdens constructie
- Ray tracing versus k-nearest neighbor query's



## Hybrid Swapping Photon Map

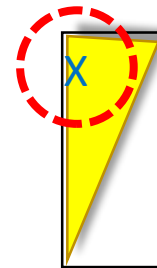


## Speed up

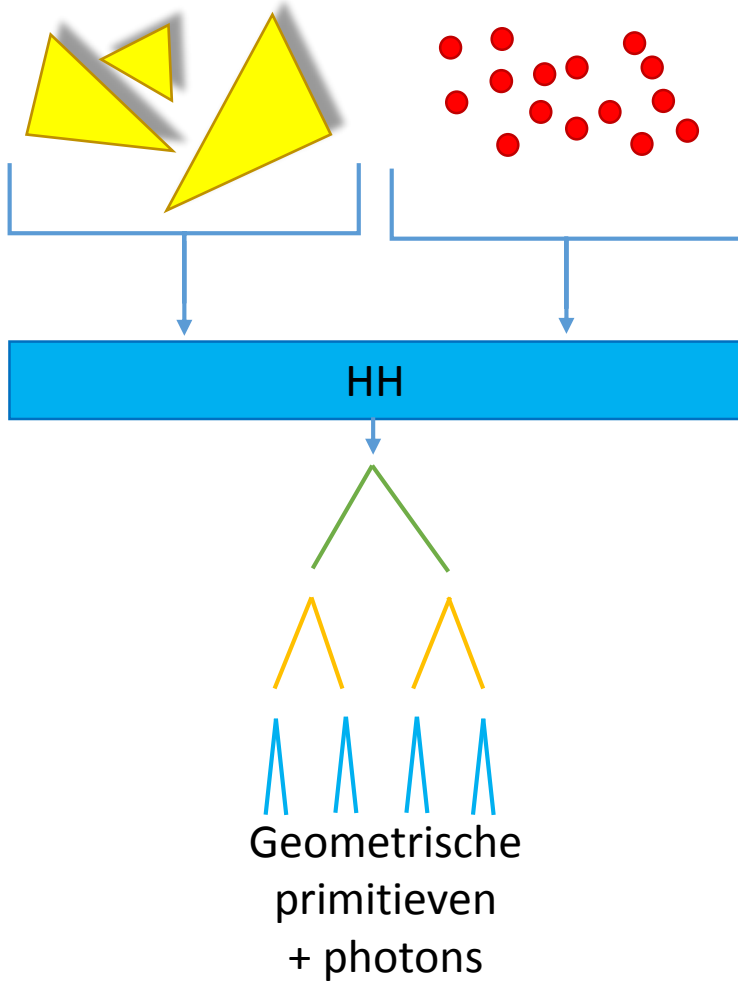
- Vroege terminatie via omsluitende voxel
- Photons opdelen volgens geometrische primitieven

## Helaas...

- Nieuwe acceleratiestructuur bouwen
- Lange bouwtijd
- Hoger geheugengebruik
- Puntdata versus niet-puntdata
- Ongekende informatie tijdens constructie
- Ray tracing versus k-nearest neighbor query's



## Hybrid Swapping Photon Map

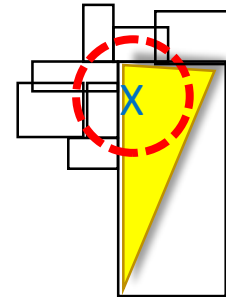


## Speed up

- Vroege terminatie via omsluitende voxel
- Photons opdelen volgens geometrische primitieven

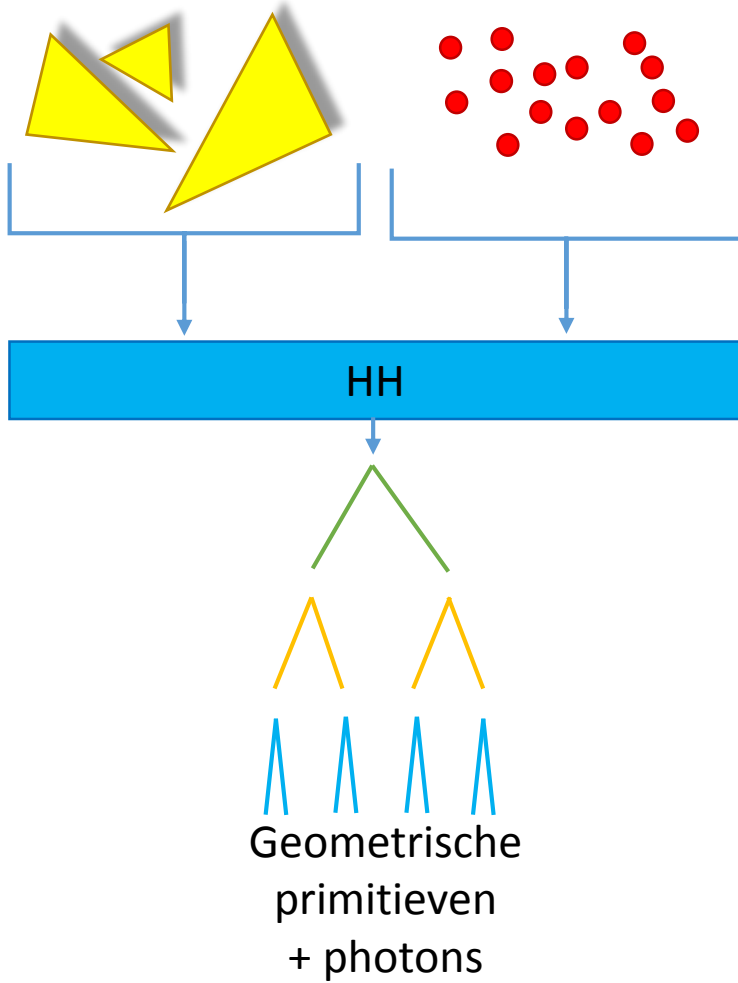
## Helaas...

- Nieuwe acceleratiestructuur bouwen
- Lange bouwtijd
- Hoger geheugengebruik
- Puntdata versus niet-puntdata
- Ongekende informatie tijdens constructie
- Ray tracing versus k-nearest neighbor query's





## Hybrid Swapping Photon Map



## Speed up

- Vroege terminatie via omsluitende voxel
- Photons opdelen volgens geometrische primitieven

## Helaas...

- Nieuwe acceleratiestructuur bouwen
- Lange bouwtijd
- Hoger geheugengebruik
- Puntdata versus niet-puntdata
- Ongekende informatie tijdens constructie
- Ray tracing versus k-nearest neighbor query's
- Correctie voor gulzige bouwheuristiek

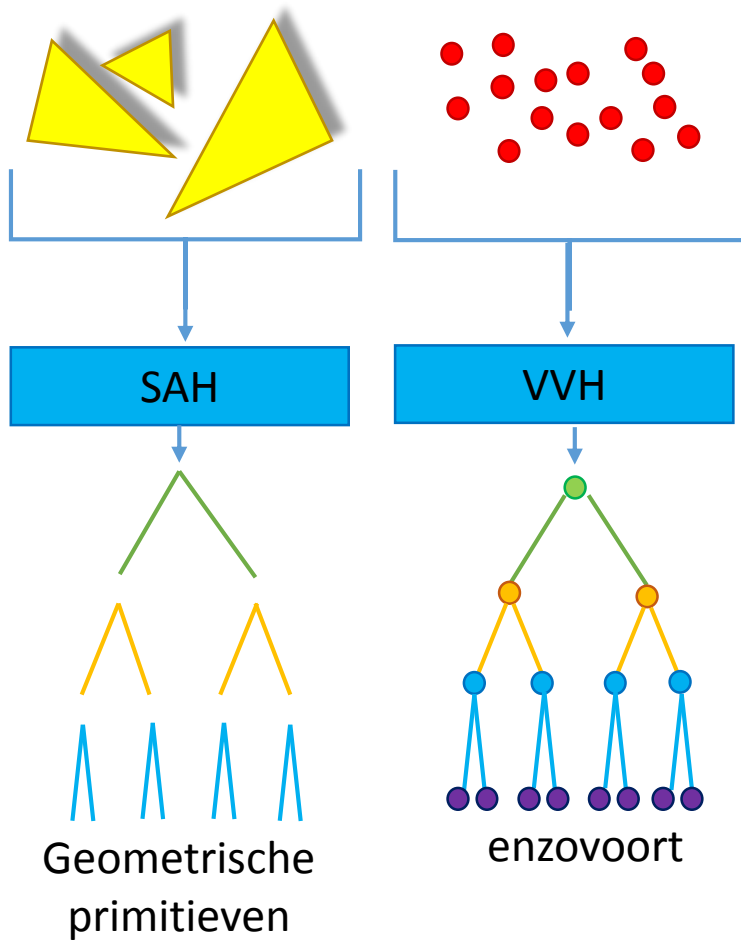
“The **decoupling** of the photon map(s) **from the geometry** is a significant advantage that not only simplifies the representation but also makes it possible to use the structure to represent lighting in very complex models.”

*Henrik Wann Jensen, Realistic image synthesis using photonmaps*

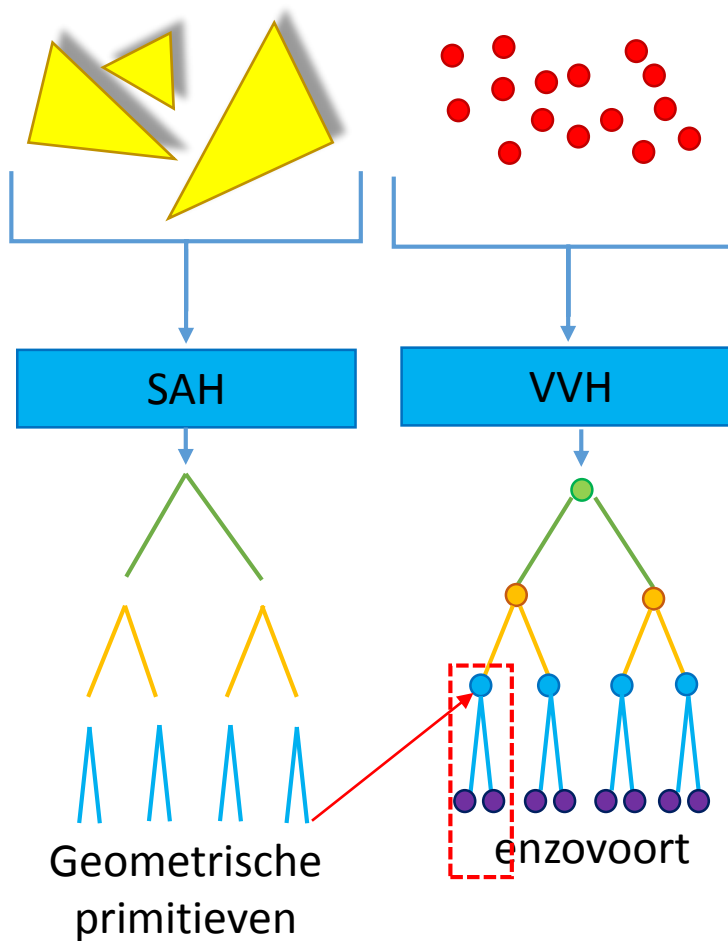
“Two separate kd-trees will always be faster”

*Vlastimil Havran, EGSR15*

## Photon Map



## Hybrid Connected Photon Map



# Zelfreflectie

## Pluspunten

- Regelmatig
- Zelfstandig
- Planmatig
- Autodidactisch
- Automatisatie

## Minpunten

- Balans thesisonderwerp <> paperonderwerp
- Debugging fase
- Volledig design -> implementatie (Luca Fascione)
- Paper georiënteerd



*The End*

