KU LEUVEN

Het gebruik van de normalen bij het bouwen van BSP acceleratie-structuren

Thesisverdediging

Jesse Hoobergs
KU Leuven
Juni 2019

Promotor:
Prof. dr. ir. Ph. Dutré

0 Overzicht

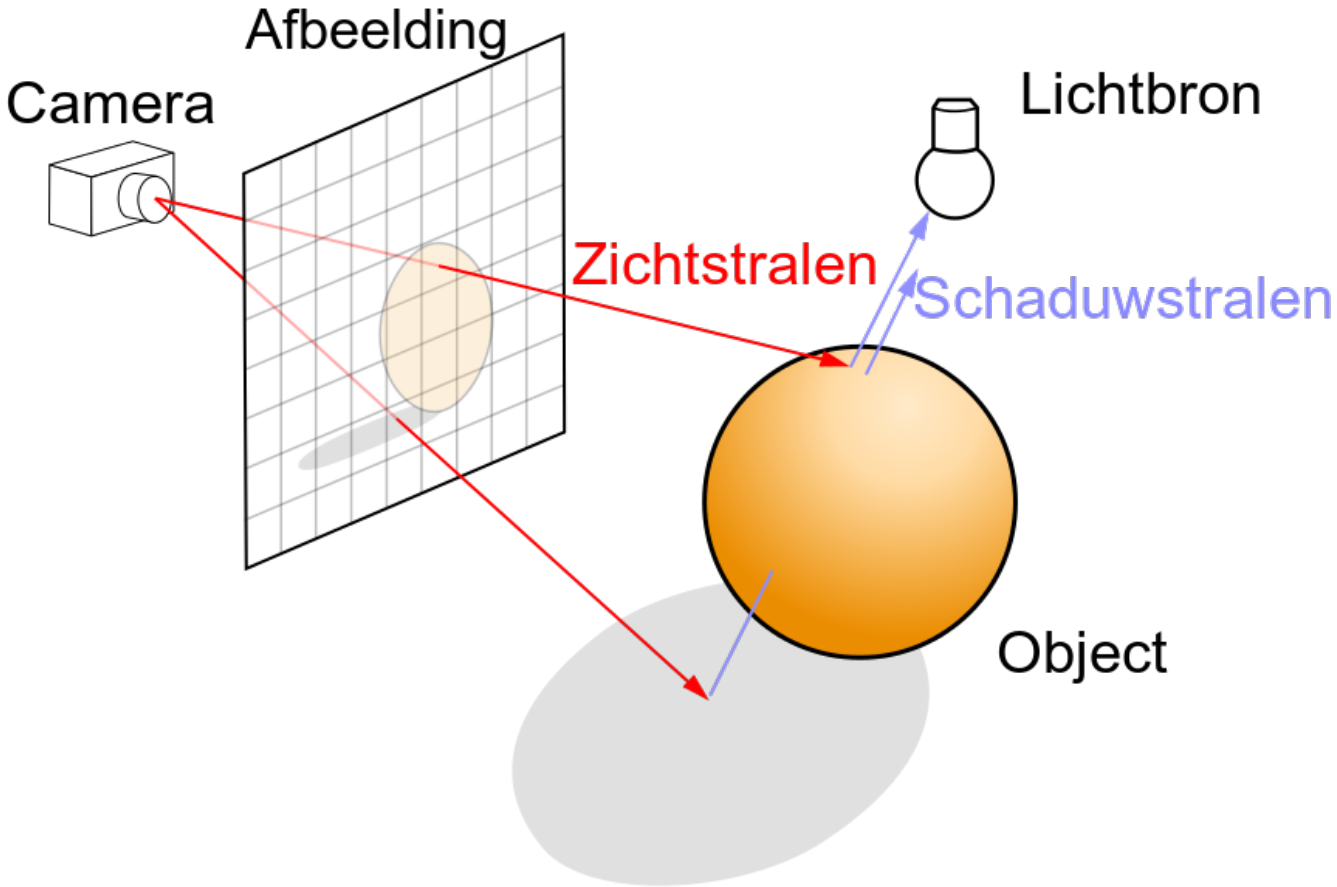
1 Outline

1 Ray tracing

- Fysisch gebaseerd renderen
- Stralen volgen door een scene

1 Ray tracing

- Fysisch gebaseerd renderen
- Stralen volgen door een scene



1 Ray tracing

- Praktische aantallen:
  - 1 miljoen pixels
  - 1 miljoen driehoeken (mogelijks veel meer)
  - 100 stralen per pixel

1 Ray tracing

- Praktische aantallen:
  - 1 miljoen pixels
  - 1 miljoen driehoeken (mogelijks veel meer)
  - 100 stralen per pixel

$\Longrightarrow$ 10^{14} straal-driehoekintersecties

1 Ray tracing

- Praktische aantallen:
  - 1 miljoen pixels
  - 1 miljoen driehoeken (mogelijks veel meer)
  - 100 stralen per pixel

$\Longrightarrow$ 10^{14} straal-driehoekintersecties

$\Longrightarrow$ Acceleratiestructuren

1 Acceleratiestructuren

- Doel:
  - Totale rendertijd minimaliseren
  - Straal-driehoekintersecties te verminderen

1 Acceleratiestructuren

- Doel:
  - Totale rendertijd minimaliseren
  - Straal-driehoekintersecties te verminderen
- Simpelste versie
  - Omhullende volume van scene (balk, bol, etc)
  - Test intersectie met omhullend volume
    - `Intersectie`: Test alle driehoeken
    - `Geen intersectie`: Test nul driehoeken
  - Recursief opdelen tot boomstructuur

1 Acceleratiestructuren

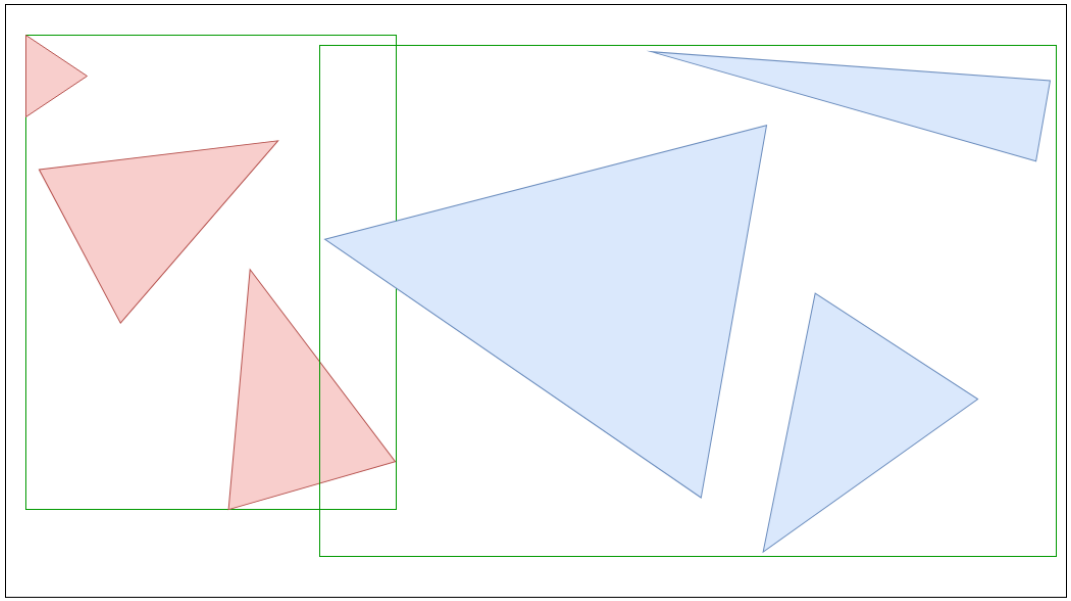
- Doel:
  - Totale rendertijd minimaliseren
  - Straal-driehoekintersecties te verminderen
- Simpelste versie
  - Omhullende volume van scene (balk, bol, etc)
  - Test intersectie met omhullend volume
    - `Intersectie`: Test alle driehoeken
    - `Geen intersectie`: Test nul driehoeken
  - Recursief opdelen tot boomstructuur
- Twee manieren van opdelen:
  - Volgens objecten
  - Volgens volume

1 Opdelen volgens object

- Driehoeken opgedeeld in disjuncte groepen
- Kindvolumes = omhullende volumes groepen
- Elke driehoek in exact één kindvolume
- Kindvolumes kunnen overlappen in de ruimte

1 Opdelen volgens object

- Driehoeken opgedeeld in disjuncte groepen
- Kindvolumes = omhullende volumes groepen
- Elke driehoek in exact één kindvolume
- Kindvolumes kunnen overlappen in de ruimte

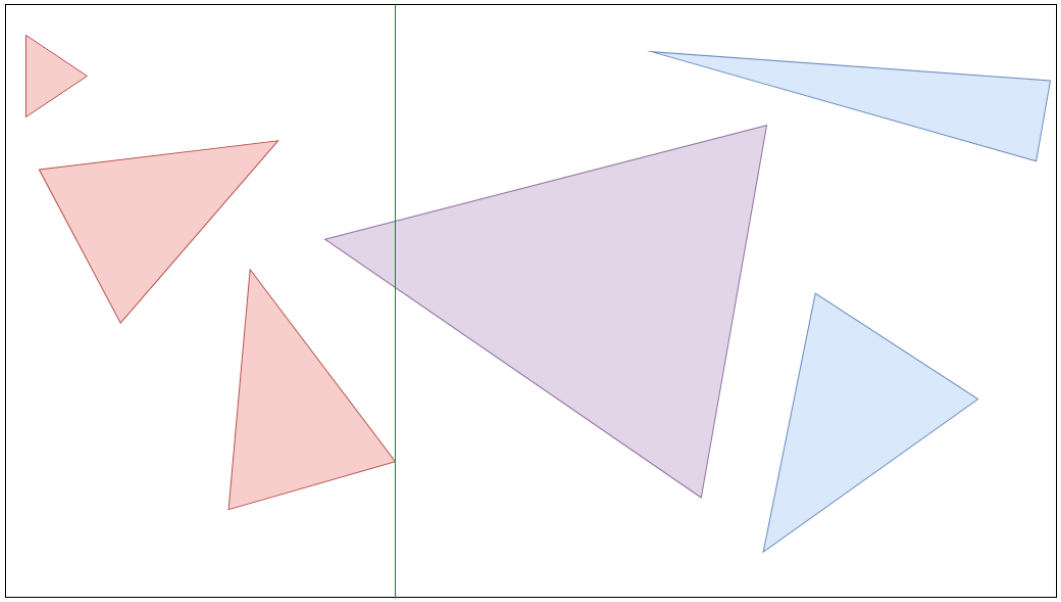


1 Opdelen volgens ruimte

- Volume opgedeeld in disjuncte groepen
- Splitsing via *splitsingsvlakken*
- Elke driehoek in minstens één kindvolume
- Kindvolumes overlappen niet in de ruimte

1 Opdelen volgens ruimte

- Volume opgedeeld in disjuncte groepen
- Splitsing via *splitsingsvlakken*
- Elke driehoek in minstens één kindvolume
- Kindvolumes overlappen niet in de ruimte



1 BSP bomen

- *Binary Space Partitioning* bomen
- Delen volgens ruimte
- Delen steeds in 2 kindvolumes
- Splitsing via willekeurige vlakken in de ruimte
- Kd boom:
  - Enkel asgealigneerde vlakken
  - Computationele voordelen bij bouwen en renderen
  - Meest gebruikte BSP boom
- Algemene BSP boom:
  - Veel meer mogelijke splitsingsvlakken
  - Moeilijk om goede te vinden

1 Doel thesis

- Algemene BSP boom
- Normalen van driehoeken
- Goede splitsingsvlakken

- BSP_{SWEEP} boom

2 Outline

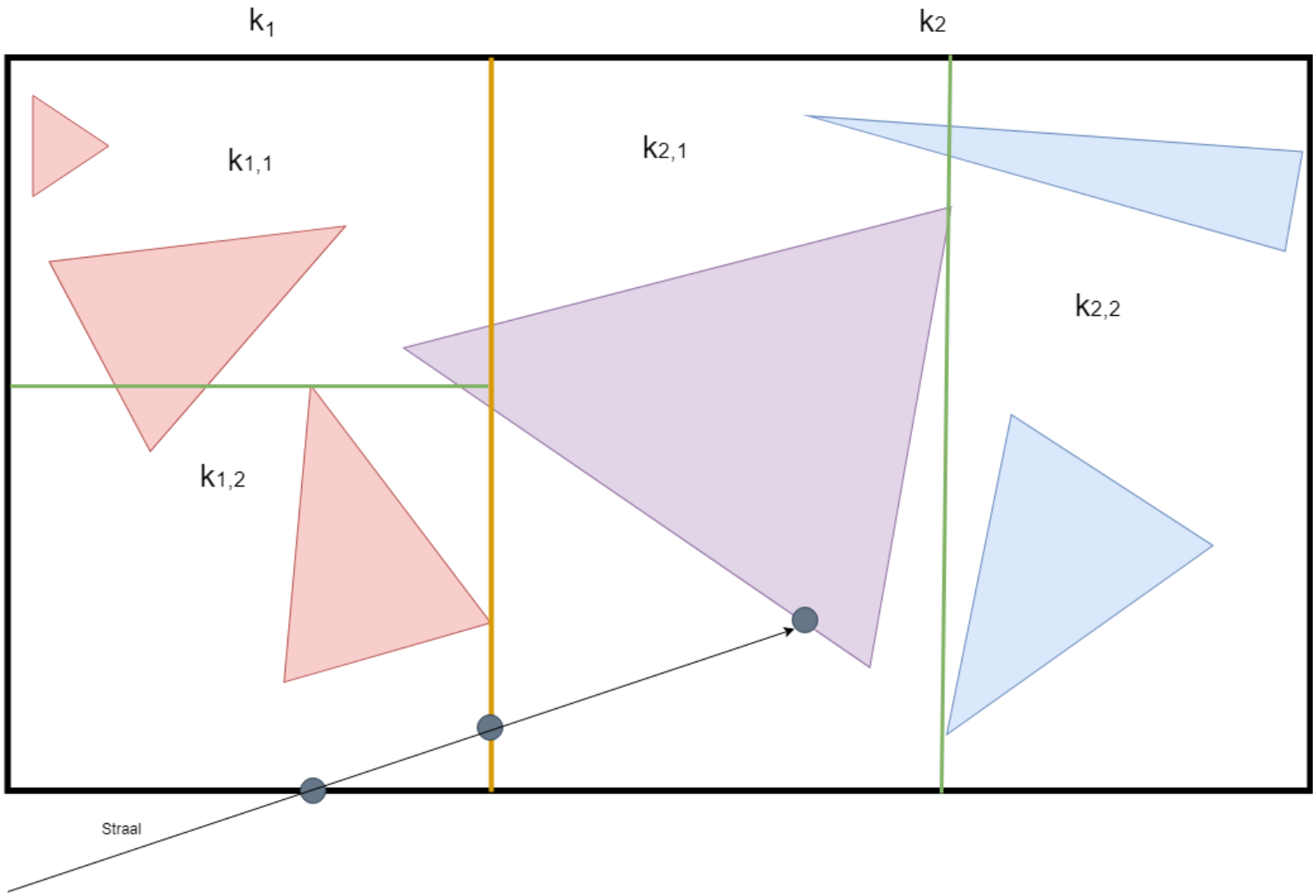
2 BSP bomen

- Bouwen
- Intersecteren
- Bestaande BSP bomen

2 Bouwen BSP boom

- Wortelknoop
  - Volume = omhullende balk scene
  - Bevat alle driehoeken
- Splits in twee kindknopen
  - Splits het volume volgens een splitsingsvlak
  - Maak twee kindknopen, één voor elk volumedeel
  - Bepaal voor elke kindknoop de driehoeken
- Splits kindknopen recursief tot stopconditie

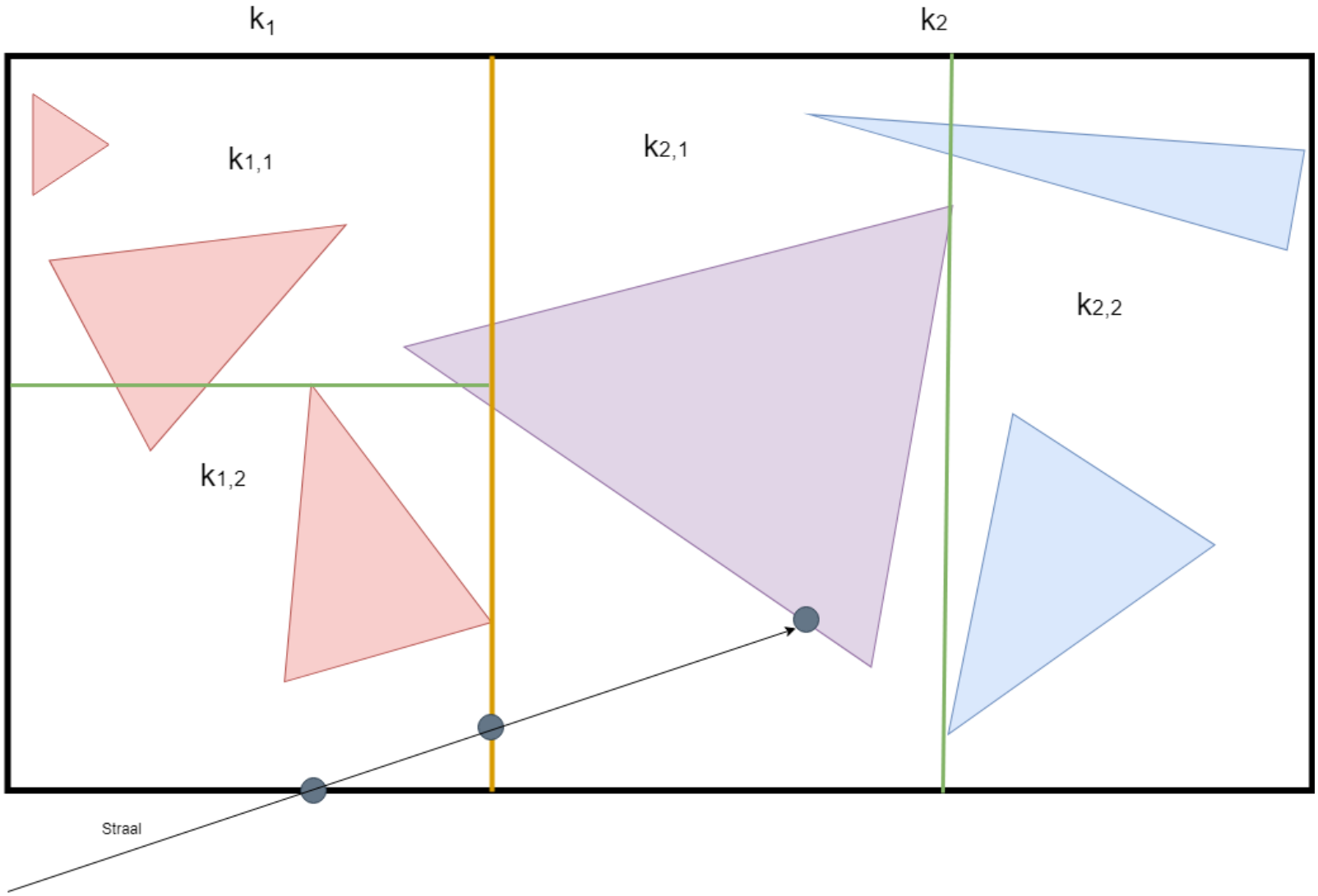
2 Bouwen BSP boom



2 Intersecteren BSP boom

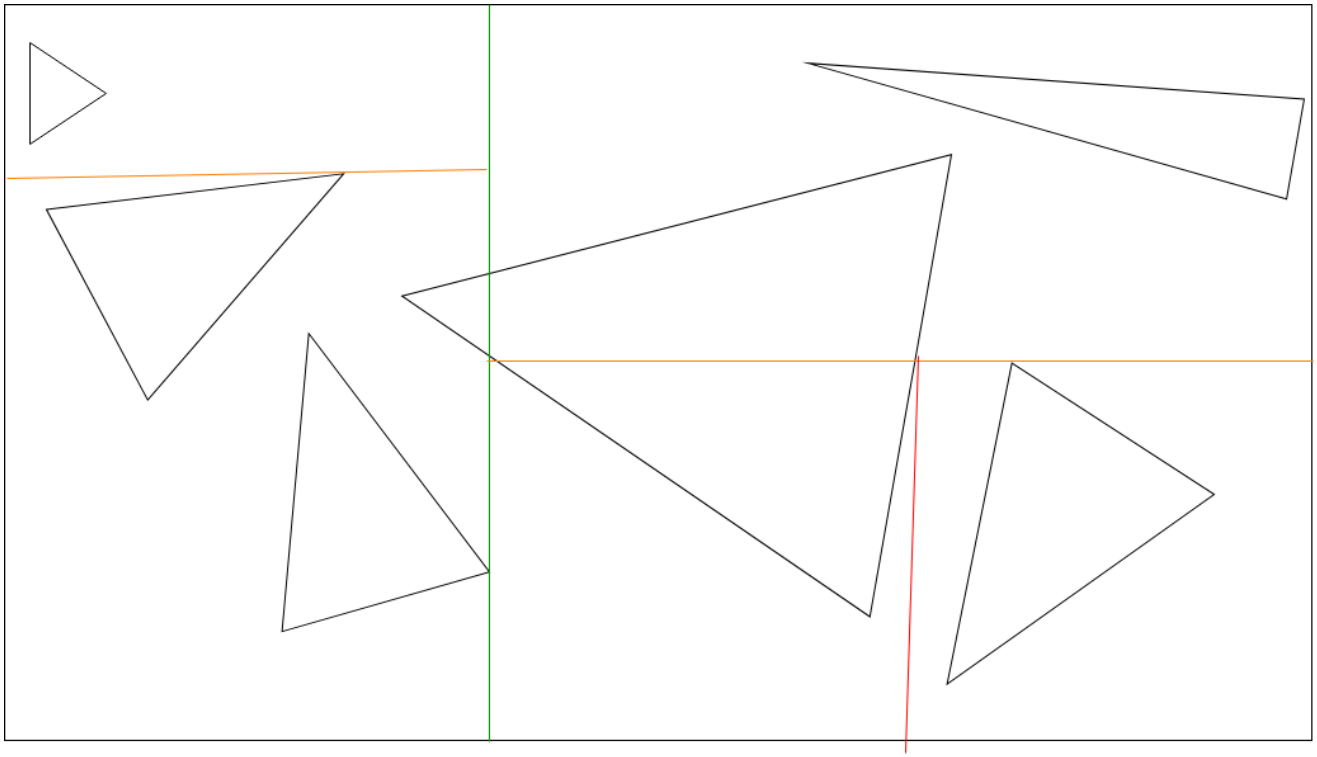
- Bepaal intersectie straal met wortelknoop:
  - geen intersectie: geen intersecterende driehoek
  - anders: doorkruis de wortelknoop
- Doorkruisen knoop
  - Inwendige knoop
    - Bepaal intersectie straal met splitsingsvlak
    - Bepaal de volgorde waarin de straal door de kindknopen gaat
    - Doorkruis de kindknopen in deze volgorde
  - Bladknoop
    - Bepaal voor elke driehoek de straal-driehoek intersectie

2 Intersecteren BSP boom



2 Kd boom

- Enkel asgealigneerde splitsingsvlakken
  - Volume elke knoop = asgealigneerde balk
  - Goedkoper doorkruisen inwendige knoop
  - Kunnen zich minder goed aanpassen aan scene

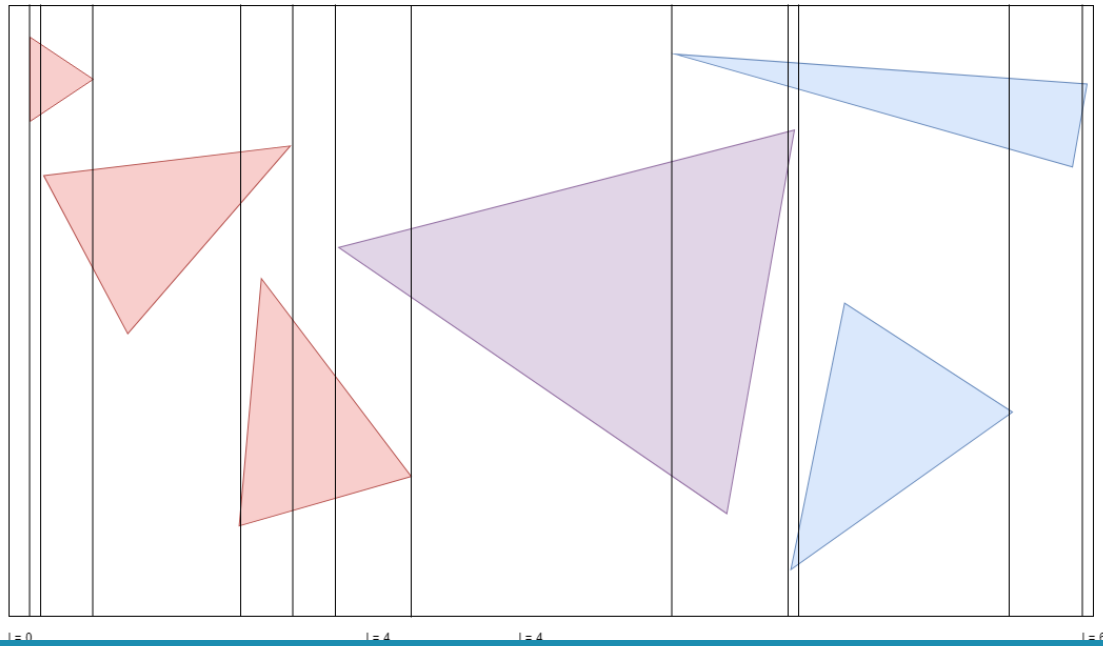


2 Bepalen beste splitsingsvlak

- Bepaal kost na splitsen volgens vlak
- Kies beste splitsingsvlak of splits niet
- *Surface Area* heuristiek
  - Kans om kindknoop te doorkruisen is evenredig met oppervlakte
  - Kost knoop evenredig met aantal driehoeken
  - Na splitsing zijn beide kindknopen bladknopen
  - $\mathcal{K}_p = \frac{SA(l)}{SA(p)} * n_l * \mathcal{K}_i + \frac{SA(r)}{SA(p)} * n_r * \mathcal{K}_i + \mathcal{K}_d$
  - Kost om niet te splitsen: $n_l * \mathcal{K}_i$
- Alle asgealigneerde vlakken testen is onhaalbaar
  - Havran: slechts $2n$ mogelijke splitsingsvlakken per richting
  - SA kost stijgt/daalt monotoon tussen eindpunten driehoeken langs die richting
  - Enkel asgealigneerde vlakken door eindpunten testen

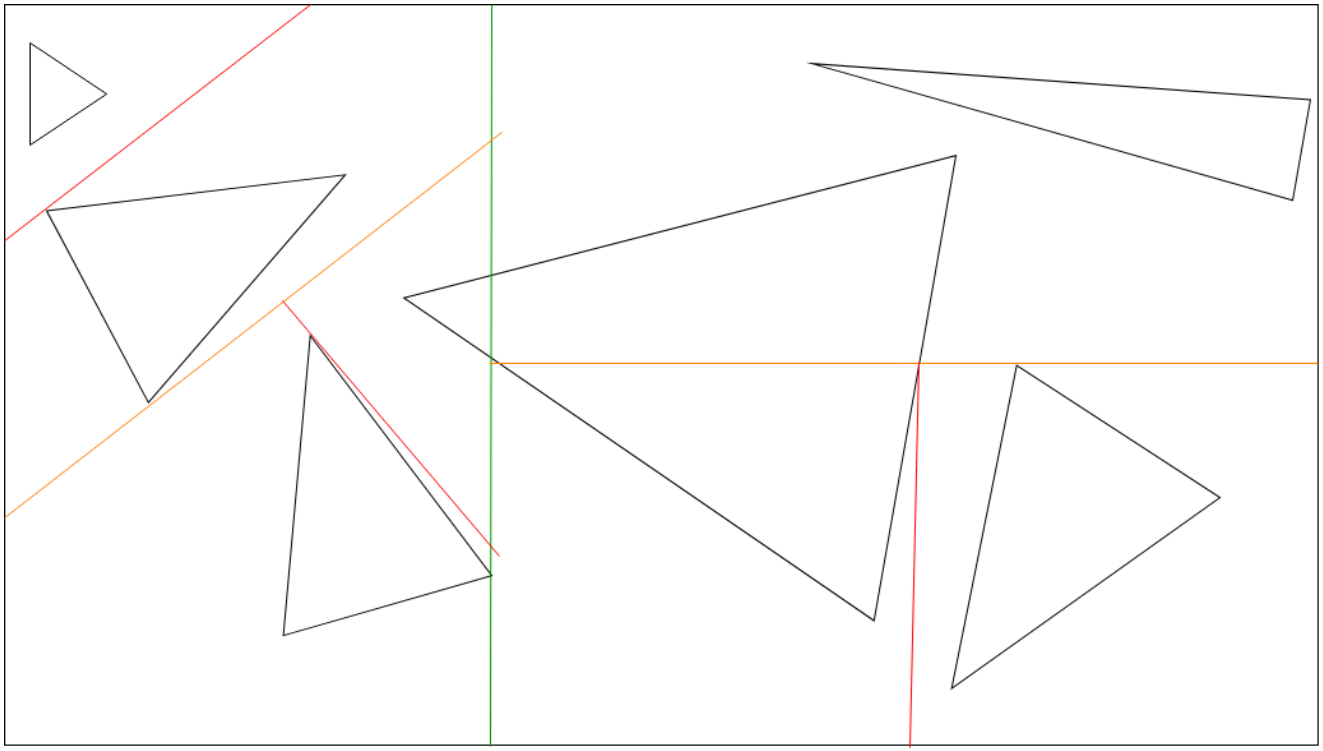
2 Bepalen beste splitsingsvlak

- SA kost berekenen
  - Aantal driehoeken in beide kindknopen nodig
  - Oppervlaktes beide kindknopen nodig
- *Sweeping*
  - Sorteer driehoeken volgens eindknopen langs as
  - 'Veeg' over de as en update n_l en n_r



2 $RBSP$ boom

- Enkel splitsingsrichtingen uit vaste verzameling van k richtingen
  - Volume elke knoop $= k - DOP$
  - Duurder doorkruisen inwendige knoop
  - Kunnen niet alle niet-intersecterende driehoeken scheiden



2 Praktisch

- Bepalen vaste verzameling splitsingsrichtingen
  - Belangrijk dat ze samen de eenheidsbol goed bedekken
- SA kost kan gebruikt worden, inclusief *sweeping*
- Oppervlakte $k - DOP$ berekenen is duurder
- Ten opzichte van Kd boom
  - Minder straal-driehoekintersecties en doorkruisingen
  - Tragere inwendige knoopdoorkruising
  - Tragere rendertijd

2 BSP_{IZE} boom

- Enige bestaande algemene BSP boom bij rendering
- Geometrie-afhankelijke splitsingsvlakken
  - De asgealigneerde vlakken van Kd boom
  - Vlak door elke driehoek
  - Drie vlakken door zijde driehoek en loodrecht op driehoek
- Volume elke knoop $=$ convex veelvlak
- Sweeping
  - Mogelijk voor de Kd richtingen
  - Niet mogelijk voor de vier andere vlakken per driehoek
    - BVH hulpstructuur nodig om n_l en n_r efficiënt te berekenen
    - Tragere bouwtijd

2 BSP^{Kd}_{IZE} boom

- Optimalisatie
- Inwendige Kd knopen bevoordelen
  - Sneller te doorkruisen dan BSP knopen
  - Lagere doorkruiskost in SA kost dan inwendige BSP knopen
- Aanpassing SA heuristiek
  - Aparte $\mathcal{K}_{d,Kd}$ en $\mathcal{K}_{d,BSP}$
  - Rechtstreeks gebruiken in SA kost werkt niet
    - SA kost varieert lineair in aantal driehoeken
    - BSP knopen splitsen beter
    - Bijna enkel BSP knopen gebruikt
  - $\mathcal{K}_{d,BSP}$ lineair afhankelijke van aantal driehoek
    - $\mathcal{K}_{d,BSP} = \alpha * \mathcal{K}_i * (n-1) + \mathcal{K}_{d,Kd}$
    - Beste splitsingsvlak zoeken
    - Indien niet gevonden, vaste $\mathcal{K}_{d,BSP}$

2 Vergelijking

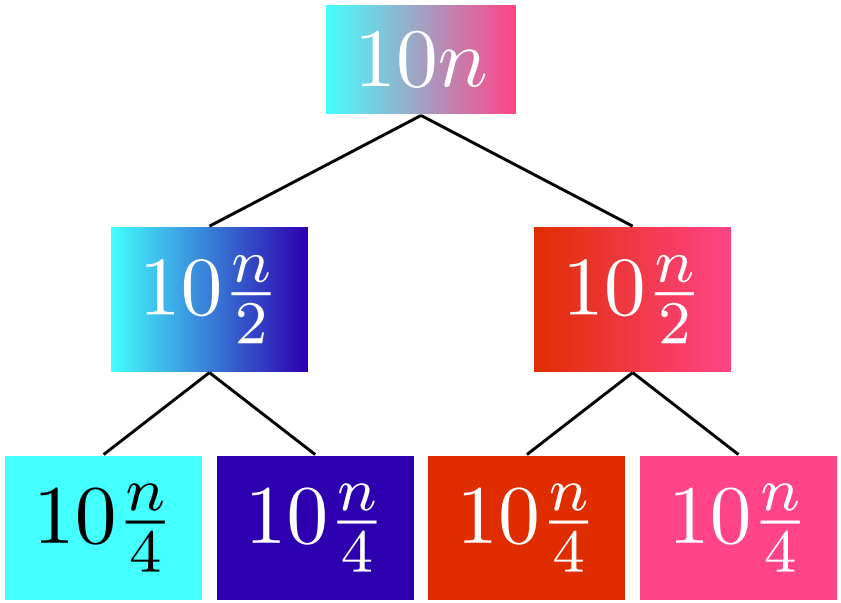
- Ten opzichte van Kd boom
  - Geen volledige sweeping mogelijk
  - Minder straal-driehoekintersecties en doorkruisingen
  - Gemiddeld lichtjes tragere inwendige knoopdoorkruising
  - Lichtjes snellere rendertijd
- Ten opzichte van $RBSP$ boom
  - Duurdere bouwtijd
  - Beide kunnen snelle Kd doorkruising gebruiken

2 Vergelijking

- Aantal splitsingsvlakken per niveau
  - Kd: $6n$
  - $RBSP$: $2kn$
  - BSP_{IZE}: $10n$
- Totaal aantal verschillende geteste splitsingsvlakken
  - Kd: $6n$
  - $RBSP$: $2kn$
  - BSP_{IZE}: $10n$
- Zelfs BSP_{IZE} gebruikt niet volledige vrijheid

2 Aantal splitsingsvlakken

- Zelfde splitsingsvlakken op elk niveau (bv BSP_{IZE})



$\#$	$\# \neq$
$10n$	$10n$
$10n$	$10n$
$10n$	$10n$

- Driehoeken die in het bovenste niveau niet gesplitst kunnen worden
  - Kunnen in geen enkel niveau van elkaar gesplitst worden

3 Outline

3 Concept

- Algemene BSP boom
- Geometrie-afhankelijke splitsingsrichtingen
  - In elke knoop k richtingen bepaald
  - Richtingen kunnen afhankelijk zijn van driehoeken in knoop
  - *Sweeping* over deze richtingen
  - Geen hulpstructuur nodig
- Drie ontwerpbeslissingen
  - Methode gebruikt om de k-richtingen te bepalen
  - Waarde van k
  - Kd richtingen altijd gebruiken of niet ?
- $RBSP$ boom is BSP_{SWEEP} boom met steeds zelfde richtingen

3 Bepalen k-richtingen

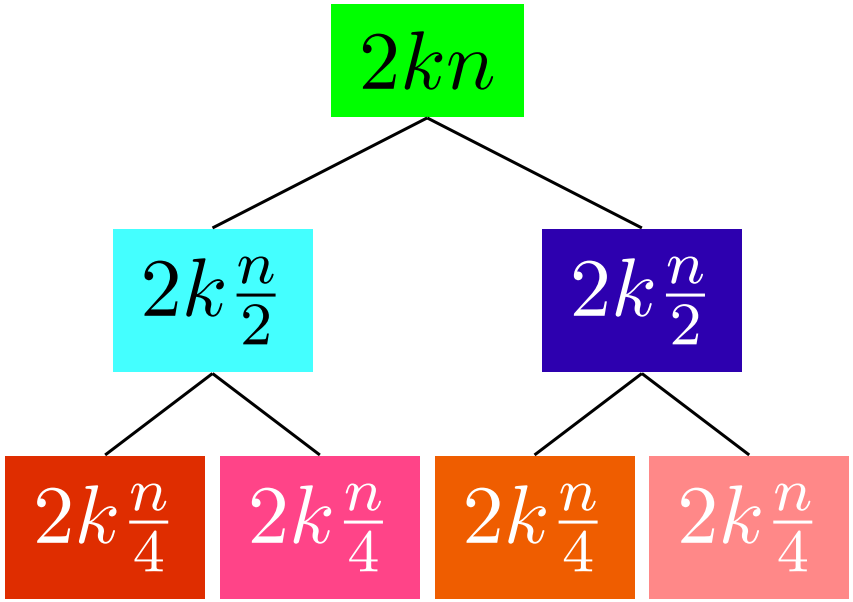
- BSP_{random}
  - Willekeurige richtingen (uniform op hemisfeer)
  - Idee: met veel verschillende (mogelijks slechte) vlakken proberen te splitsen
  - Kans op splitsing door willekeurige richting even groot als door Kd richting
- BSP_{wn}
  - Normalen van willekeurige driehoeken in de knoop
  - Idee: splitsen volgens oriëntatie driehoeken
  - Maakt gebruik van welke driehoeken samen in een knoop zitten
- BSP_{cn}
  - Clustercentra van normalen van de driehoeken in de knoop
  - Idee: splitsen volgens veelvoorkomende oriëntaties
  - Maakt gebruik van welke driehoeken samen in een knoop zitten

3 Kd richtingen gebruiken ?

- k richtingen genereren
  - BSP_{random}
  - BSP_{wn}
  - BSP_{cn}
- Kd richtingen en $k-3$ richtingen genereren
  - Kd richtingen behandelen als BSP richtingen
    - $BSP_{random+}$
    - BSP_{wn+}
    - BSP_{cn+}
  - Kd richtingen apart behandelen
    - $BSP^{Kd}_{random+}$
    - BSP^{Kd}_{wn+}
    - BSP^{Kd}_{cn+}

3 Aantal splitsingsvlakken

- Andere splitsingsvlakken op elk niveau



$\#$	$\#\neq$
$2kn$	$2kn$
$2kn$	$4kn$
$2kn$	$6kn$

- Driehoeken die in het bovenste niveau niet gesplitst kunnen worden
  - Kunnen op lagere niveaus misschien wel gesplitst worden
  - $\mathcal{O}(nlog(n))$ verschillende splitsingsvlakken ipv $\mathcal{O}(n)$

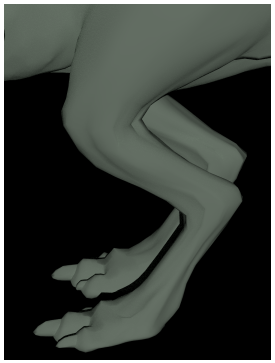
4 Outline

4 Implementatie

- Pbrt-v3 renderer
- Enkel op de CPU
- Bouwen is niet geparallelliseerd
- Geïmplementeerde BSP bomen:
  - Kd
  - $RBSP$ en $RBSP^{Kd}$
  - BSP en BSP^{Kd}_{IZE}
  - $BSP^{(Kd)}_{random(+)}$, $BSP^{(Kd)}_{wn(+)}$ en $BSP^{(Kd)}_{cn(+)}$

5 Outline

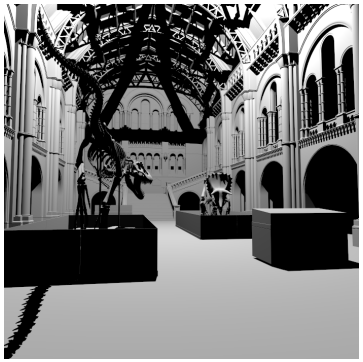
5 Scenes



(a) Killeroo Been scene

(b) Sponza scene

(c) Conference scene



(d) Museum scene

- Killeroo: 33264 driehoeken
- Sponza: 227309 driehoeken
- Conference: 123651 driehoeken
- Museum: 1462840 driehoeken

5 Acceleratiestructuren

5 Acceleratiestructuren

6 Outline

6 Acceleratiestructuren

6 Acceleratiestructuren

6 Acceleratiestructuren