

# Distributiefunctie van de geometrische normalen gebruiken voor het bouwen van BSP acceleratiedatastructuren

Jesse Hoobergs

Thesis voorgedragen tot het behalen  
van de graad van Master of Science  
in de ingenieurswetenschappen:  
computerwetenschappen, hoofdoptie  
Mens-machine communicatie

**Promotor:**

Prof. dr. ir. Philip Dutré

**Assessoren:**

Ir. W. Eetveel  
W. Eetrest

**Begeleiders:**

Ir. M. Moulin  
Ir. P. Bartels

© Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot het Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 of via e-mail [info@cs.kuleuven.be](mailto:info@cs.kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# Voorwoord

Dit is mijn dankwoord om iedereen te danken die mij bezig gehouden heeft. Hierbij dank ik mijn promotor, mijn begeleider en de voltallige jury. Ook mijn familie heeft mij erg gesteund natuurlijk.

*Jesse Hoobergs*

# Inhoudsopgave

<b>Voorwoord</b>	<b>i</b>
<b>Samenvatting</b>	<b>iv</b>
<b>Lijst van figuren en tabellen</b>	<b>v</b>
<b>Lijst van afkortingen en symbolen</b>	<b>vi</b>
<b>1 Inleiding</b>	<b>1</b>
1.1 Ray tracing	1
1.2 Doelstelling	1
1.3 Methodologie	1
1.4 Contributie	1
1.5 Overzicht	1
<b>2 Voorgaand werk</b>	<b>3</b>
2.1 Acceleratiestructuren	3
2.2 <i>BSP</i> bomen	3
2.3 <i>Kd</i> Bomen	4
2.4 <i>RBSP</i> -bomen	4
2.5 Algemene <i>BSP</i> -Bomen in de praktijk	4
2.6 Hiërarchie	5
<b>3 <i>BSP<sub>SWEEP</sub></i></b>	<b>7</b>
3.1 Algemeen idee	7
3.2 Gebaseerd op geometrische normalen	7
3.3 Gebaseerd op random richtingen	7
<b>4 Implementatie</b>	<b>9</b>
4.1 Outline <i>BSP</i> -algoritmes	9
4.2 <i>Kd</i> boom	9
4.3 <i>RBSP</i> boom	9
4.4 <i>BSP<sub>IZE</sub></i>	9
4.5 <i>BSP<sub>SWEEP</sub></i>	9
<b>5 Resultaten</b>	<b>11</b>
5.1 Praktische aspecten	11
5.2 Afhankelijkheid van aantal richtingen	11
5.3 Vergelijking	11

**Bibliografie**

**13**

# Samenvatting

In dit **abstract** environment wordt een al dan niet uitgebreide samenvatting van het werk gegeven. De bedoeling is wel dat dit tot 1 bladzijde beperkt blijft.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# Lijst van figuren en tabellen

## Lijst van figuren

## Lijst van tabellen

- 5.1 Statistieken over de rendertijd voor  $BSP_{random+}^{Kd}$  voor verschillende waarden van  $K$ . Voor elke waarde van  $K$  is het algoritme 6 keer uitgevoerd. 11

# Lijst van afkortingen en symbolen

## Afkortingen

$BSP$	Binary Space Partitioning
$BVH$	Bounding Volume Hierarchy
$RBSP$	Restricted Binary Space Partitioning Tree
$SA$	Surface Area

## Symbolen

$BSP_{wn}$	$BSP$ boom die per knoop random normalen als splitsrichtingen gebruikt.
$BSP_{wn+}$	$BSP_{wn}$ boom waarbij de drie richtingen volgens de hoofdassen altijd deel uitmaken van de splitsrichtingen.
$BSP_{wn+}^{Kd}$	$BSP_{wn+}$ boom waarbij $Kd$ knopen efficiënter doorkruist worden dan $BSP$ knopen.
$BSP_{wn}$	$BSP$ boom die per knoop een clustering van de normalen berekend en de centruns van deze clusters als splitsrichtingen gebruikt.
$BSP_{wn+}$	$BSP_{wn}$ boom waarbij de drie richtingen volgens de hoofdassen altijd deel uitmaken van de splitsrichtingen.
$BSP_{wn+}^{Kd}$	$BSP_{wn+}$ boom waarbij $Kd$ knopen efficiënter doorkruist worden dan $BSP$ knopen.
$BSP_{random}$	$BSP$ boom die per knoop random richtingen als splitsrichtingen gebruikt.
$BSP_{random+}$	$BSP_{random}$ boom waarbij de drie richtingen volgens de hoofdassen altijd deel uitmaken van de random splitsrichtingen.
$BSP_{random+}^{Kd}$	$BSP_{random+}$ boom waarbij $Kd$ knopen efficiënter doorkruist worden dan $BSP$ knopen.
$\mathcal{K}_{d,BSP}$	De kost om een $BSP$ knoop te doorkruisen.
$\mathcal{K}_{d,Kd}$	De kost om een $Kd$ knoop te doorkruisen.
$RBSP^{Kd}$	$RBSP$ waarbij $Kd$ knopen efficiënter doorkruist worden dan $BSP$ knopen.



# Hoofdstuk 1

## Inleiding

In dit hoofdstuk wordt het werk ingeleid. Het doel wordt gedefinieerd en er wordt uitgelegd wat de te volgen weg is (beter bekend als de rode draad).

Als je niet goed weet wat een masterproef is, kan je altijd Wikipedia eens nakijken.

### 1.1 Ray tracing

**Stralen volgen** Door elke pixel één (of meerdere stralen), kleur intersectiepunt is kleur pixel -> path tracing.

**Acceleratiestructuren** Doel: aantal straal-driehoek intersecties verminderen.

### 1.2 Doelstelling

Betere Acceleratiestructuur bouwen door een algemene BSP te maken die gebruik maakt van de geometrische normalen bij het splitsen. Aantal intersecties nog doen dalen, traversals stijgen, rendertijd dalen.

### 1.3 Methodologie

### 1.4 Contributie

### 1.5 Overzicht



## Hoofdstuk 2

# Voorgaand werk

Raytracing vereist acceleratiestructuren om efficiënt driehoeken in de scene te kunnen zoeken. Dit hoofdstuk start met een algemene uitleg over acceleratiestructuren.

### 2.1 Acceleratiestructuren

Het doel van acceleratiestructuren is om het aantal straal-driehoek intersecties te verminderen. De simpelste acceleratiestructuur bestaat uit het omhullende volume van de scene. Testen op intersectie met de driehoeken in de scene gebeurt dan enkel als dit omhullende volume intersecteert met de straal. Dit kan worden uitgebreid tot een boomstructuur door dit volume recursief op te delen in kindvolumes. Binaire bomen delen elk omhullend volume op in twee nieuwe volumes, andere acceleratiestructuren zoals bijvoorbeeld octrees delen het volume op in meer dan twee volumes.

Het volume kan worden opgedeeld op twee manieren: volgens objecten of volgens ruimte. Bij opdeling volgens objecten worden de objecten binnen het volume opgedeeld in meerdere disjuncte groepen en de kindvolumes zijn de omhullende volumes van deze groepen. Na deze opdeling zit elk object in exact één van deze nieuwe volumes, maar de volumes kunnen overlappen. Een voorbeeld van een acceleratiestructuur waarbij de opdeling volgens objecten gebeurt is de *Bounding Volume Hierarchy (BVH)*. Opdeling volgens ruimte betekent dat de ruimte in het volume wordt opgedeeld in meerdere delen. Na deze opdeling overlappen deze nieuwe volumes niet, maar een object ligt nu in minstens één (en mogelijk in meerdere) kindvolume(s). De Binary Space Partitioning (*BSP*) boom deelt de ruimte van het volume steeds op in twee kindvolumes.

### 2.2 BSP bomen

De *BSP* boom deelt de ruimte van het omhullende volume recursief op door te splitsen volgens een willekeurig georiënteerd vlak totdat een bepaald stopconditie bereikt is. Het algoritme splitst een knoop niet op als het aantal driehoeken in de knoop lager is dan een vastgelegde limiet, bijvoorbeeld als er maar één driehoek in

de knoop zit. Een heuristiek kan ook tonen dat het beter is om de knoop niet te splitsen. Meerdere stopcondities worden vaak tegelijk gebruikt. Het feit dat de *BSP* volgens willekeurig georiënteerde vlakken splitst, is zowel een voor- als een nadeel. Het zorgt ervoor dat de *BSP* zich heel goed kan aanpassen aan de scene en alle niet intersecterende driehoeken in principe kan scheiden. Maar het zorgt er ook voor dat het heel moeilijk is om deze goede splitsingsvlakken te vinden.

### 2.3 *Kd* Bomen

In de praktijk wordt vaak een specifieke soort *BSP* boom gebruikt. De *Kd* boom is een *BSP* boom waarbij alle splitsingsvlakken asgealigneerd zijn. Hiermee wordt bedoelt dat de splitsingsrichting (de normaal op het splitsingsvlak) evenwijdig is aan één van de drie hoofdassen. Dit zorgt ervoor dat elke knoop van de boom een asgealigneerde balk voorstelt. Het doorkruisen van een knoop uit een *Kd* boom is daardoor goedkoper dan het doorkruisen van een knoop uit een algemene *BSP* boom. De reden hiervoor is dat het goedkoper is om het intersectiepunt van een straal en een asgealigneerd vlak te vinden (verschil en vermenigvuldiging) dan het intersectiepunt van een straal en een willekeurig vlak (scalair product en deling). *Kd* bomen kunnen zich minder goed aanpassen aan de scene en kunnen niet alle niet intersecterende driehoeken scheiden.

Ize et al [1] geven drie (volgens hun foute) ruimverspreide aannames over algemene *BSP* bomen. Ten eerste wordt er aangenomen algemene *BSP* bomen nooit sneller kunnen zijn dan *Kd* bomen omdat het doorkruisen van een *BSP* boom beduidend duurder is. De beperkte precisie van vlottende komma getallen wordt gezien als tweede probleem want het zou *BSP* bomen numeriek onstabiel maken. Door de grotere flexibiliteit is het veel moeilijker om een *BSP* boom te bouwen dan om een *Kd* boom te bouwen.

Meest onderzoek naar deze en gebruikt Beperkte versie: as-gealigneerd -> balken  
Voordelen: simpele intersectie splitsingsvlak SAH (inclusief nulbonus) -> sweeping

### 2.4 *RBSP*-bomen

Algemener dan *Kd* -> meer richtingen -> richtingen gekozen volgens *kDOPs* ipv balken  
Praktische problemen: worst of both worlds -> trage doorkruising, houd geen rekening met geometrie, kan nog steeds niet goed splitsen

### 2.5 Algemene *BSP*-Bomen in de praktijk

Paper Ize et al. 3 *Kd* richtingen (sweepen) plus 4 richtingen afhankelijk van geometrie  
Snelle *Kd* traversal, aanpassing SAH Geen sweeping -> BVH als hulpstructuur

Voordelen Knopen kunnen beter worden opgesplitst Nadelen Duurder om te traversen Duurder om te bouwen (berekening SA van veelvlak vs SA van balk) Grote zoekruimte om beste splitsingsvlak te vinden

## 2.6 Hiërarchie



# Hoofdstuk 3

## *BSP<sub>SWEEP</sub>*

### 3.1 Algemeen idee

De bekeken richtingen zijn verschillend per node Richtingen kunnen rekening houden met lokale geometrie (kan niet bij RBSP, wel bij BSPize) Sweeping van richtingen Kd-richtingen + aantal richtingen of puur die richtingen Snelle traversal voor kd-richtingen

### 3.2 Gebaseerd op geometrische normalen

#### 3.2.1 Willekeurige normaal

Extra richtingen door random normalen te kiezen Autopartitie van Ize maar gesweept Waarom zou dit werken ? : ...

#### 3.2.2 Geclusterde normalen

(Extra) richtingen via K-means clustering Sweepen volgens die richtingen Waarom zou dit werken ...

### 3.3 Gebaseerd op random richtingen

(Extra) richtingen door random richtingen te kiezen Ter controle dat de richtingen met behulp van normalen, nuttige richtingen zijn Sweeping Waarom zou dit werken ? : Random per node itt vast bij Kd Driehoeken proberen te worden gesplitst volgens meer verschillende richtingen, Kd probeert steeds hetzelfde Kd heeft maar 3 opties, als het volgens geen kan -> nooit mogelijk Kans splitsbaar door Kd richting of random is even hoog in uniform geval. Scenes hebben wel veel asgealigneerde delen, dus daarom die extra.





# Hoofdstuk 4

## Implementatie

In dit hoofdstuk wordt het werk ingeleid. Het doel wordt gedefinieerd en er wordt uitgelegd wat de te volgen weg is (beter bekend als de rode draad).

Als je niet goed weet wat een masterproef is, kan je altijd Wikipedia eens nakijken.

### 4.1 Outline BSP-algoritmes

#### 4.1.1 Bouwalgoritme

#### 4.1.2 Intersectie-algoritme

### 4.2 $Kd$ boom

### 4.3 $RBSP$ boom

### 4.4 $BSP_{IZE}$

### 4.5 $BSP_{SWEEP}$

#### 4.5.1 Algemeen

#### 4.5.2 $BSP_{random}$

#### 4.5.3 $BSP_{wn}$

#### 4.5.4 $BSP_{wn}$



# Hoofdstuk 5

## Resultaten

In dit hoofdstuk wordt het werk ingeleid. Het doel wordt gedefinieerd en er wordt uitgelegd wat de te volgen weg is (beter bekend als de rode draad).

Als je niet goed weet wat een masterproef is, kan je altijd Wikipedia eens nakijken.

### 5.1 Praktische aspecten

### 5.2 Afhankelijkheid van aantal richtingen

### 5.3 Vergelijking

K	Feet		Sponza		Conference Hall		Museum	
	Mediaan	Stdev	Mediaan	Stdev	Mediaan	Stdev	Mediaan	Stdev
4								
5								
6								
7								
8								
9								
10								

TABEL 5.1: Statistieken over de rendertijd voor  $BSP_{random+}^{Kd}$  voor verschillende waarden van K. Voor elke waarde van K is het algoritme 6 keer uitgevoerd.



# Bibliografie

- [1] T. Ize, I. Wald, and S. G. Parker. Ray tracing with the bsp tree. In *2008 IEEE Symposium on Interactive Ray Tracing*, pages 159–166, Aug 2008.

## Fiche masterproef

*Student:* Jesse Hoobergs

*Titel:* Distributiefunctie van de geometrische normalen gebruiken voor het bouwen van BSP acceleratiedatastructuren

*Engelse titel:* Using the distribution function of the geometric normals to build BSP acceleration data structures.

*UDC:* 621.3

*Korte inhoud:*

Hier komt een heel bondig abstract van hooguit 500 woorden.  $\text{\LaTeX}$  commando's mogen hier gebruikt worden. Blanco lijnen (of het commando `\par`) zijn wel niet toegelaten!

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Thesis voorgedragen tot het behalen van de graad van Master of Science in de ingenieurswetenschappen: computerwetenschappen, hoofdoptie Mens-machine communicatie

*Promotor:* Prof. dr. ir. Philip Dutré

*Assessoren:* Ir. W. Eetveel  
W. Eetrest

*Begeleiders:* Ir. M. Moulin  
Ir. P. Bartels