Acceleration af Kollisionsdetektion på Parallelle Computerarkitekturer

Speciale

Andreas Rune Fugl

anfug03@student.sdu.dk

Thomas Frederik Kvistgaard Ellehøj

ththy03@student.sdu.dk

Datateknologi ved Teknisk Fakultet
Syddansk Universitet, Odense

Speciale præsentation – 28. Oktober, 2008

Velkommen

Præsentationen

- Planlagt til 40 minutter
- Spørgsmål til slut

Outline

Introduktion

Algoritmer

Platforme

Resultater

Traditionelle robotsystemer

Traditionelle anvendelser

- Svejsning
- Sprøjtelakering
- Palletering

Kendetegn

- Faste positioner
- Programmer en gang, gentag tusinde
- Hastighed og præcision i højsædet

Traditionelle robotsystemer

Traditionelle anvendelser

- Svejsning
- Sprøjtelakering
- Palletering

Kendetegn

- Faste positioner
- Programmer en gang, gentag tusinde
- Hastighed og præcision i højsædet

Robotsystemer til emnehåndtering

Et nyt problemdomæne

- Flyt emner af vilkårlig form og orientering

Udfordringer

- Emner kan ligge hulter til bulter
- Emner kan være delvist skjulte
- Emner kan være svære at gribe

Robotsystemer til emnehåndtering

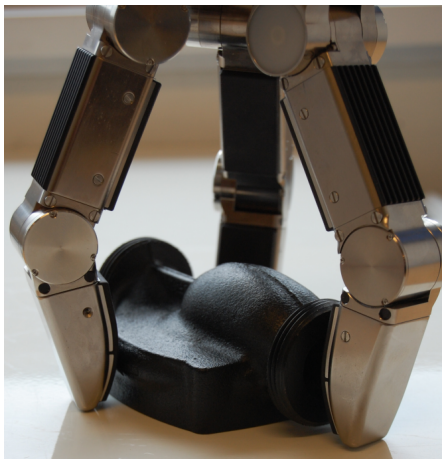
Et nyt problemdomæne

- Flyt emner af vilkårlig form og orientering

Udfordringer

- Emner kan ligge hulter til bulter
- Emner kan være delvist skjulte
- Emner kan være svære at gribe

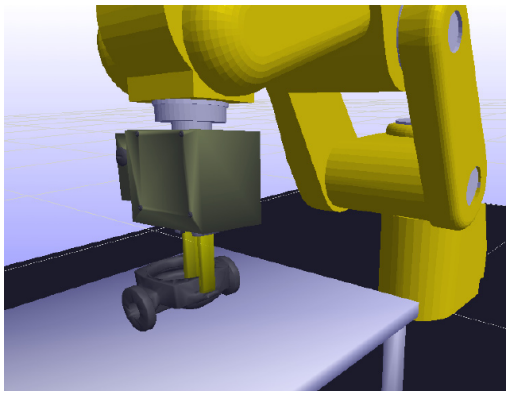
Gribesimulering i emnehåndtering



Robotgruppens fremgangsmåde

1. Genkend og bestem emnets orientering vha. computer vision
2. Opbyg strategi for gribning vha. simulering
3. Automatisk bevægelsesplanlægning af robotten til ønsket position

Gribesimulering



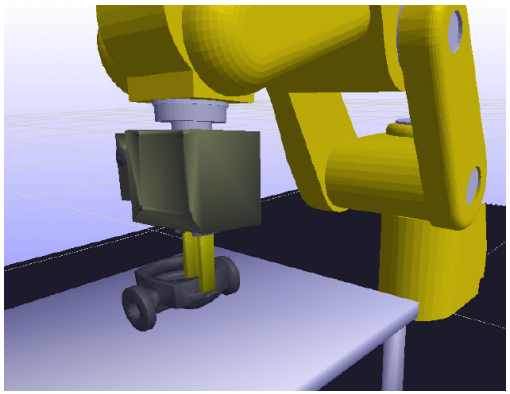
Gribesimulering

- Præcis simulering af gribeprocesser
- Mål: Automatisk bestemmelse af sikre greb

Udfordringer

- Mange forsøg for hvert greb
- Alle kontaktpunkter mellem emnet og griberen ønskes

Gribesimulering



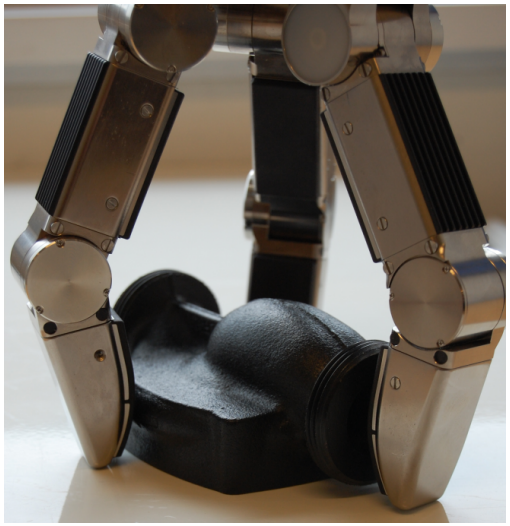
Gribesimulering

- Præcis simulering af gribeprocesser
- Mål: Automatisk bestemmelse af sikre greb

Udfordringer

- Mange forsøg for hvert greb
- Alle kontaktpunkter mellem emnet og griberen ønskes

Modellering af virkeligheden

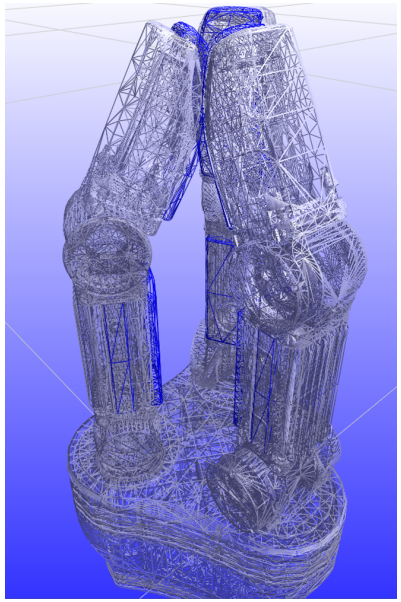


Modellering

- I simuleringen behøves en model af robotten, griberen, emnerne og arbejdscellen
- Mange mulige repræsentationer, men mest benyttet er trekantmodeller

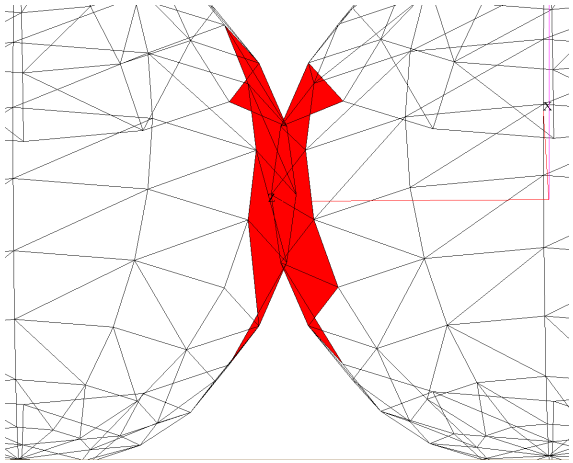
Trekantmodeller

→



- Hvert objekt repræsenteres som sæt af trekanter
- Sættet kan indeholde mange tusinde trekanter

Kollisionstests



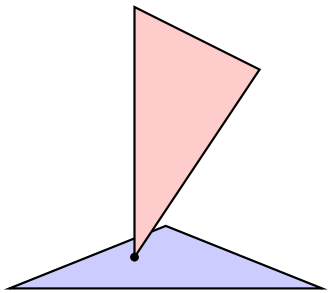
Opgaven

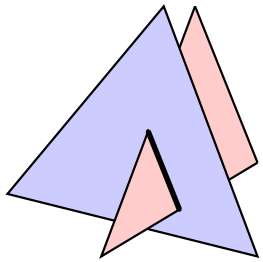
- Givet to trekant sæt A og B, test om de er i kollision
- Hvis A og B er i kollision, hvor kolliderer de så?

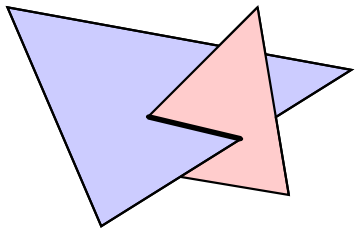
Udfordringer

- Antal tests kvadratisk i antal trekanter
- Hver test kræver mange beregninger

Trekant kollisionstests







Test om to trekanter er i kollision. Dvs. om de to trekanter har fælles punkter.

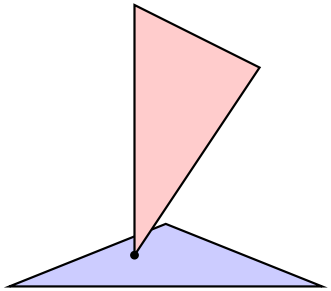
Segment-piercing

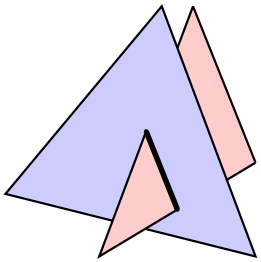
- Ligefrem algoritme
- Baseret på enkle geometriske observationer
- Beregningstung

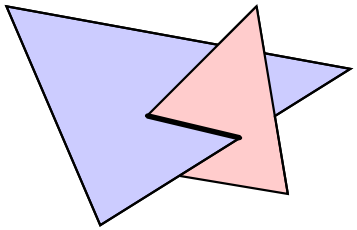
Devillers

- Mere kompliceret algoritme
- Baseret på „sideness tests“
- Terminerer hurtigt hvis der ikke er kollison

Trekant kollisionstests







Test om to trekanter er i kollision. Dvs. om de to trekanter har fælles punkter.

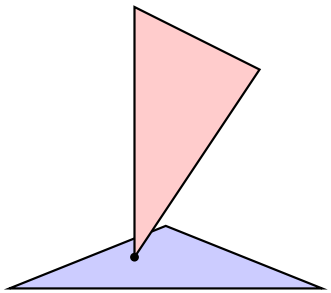
Segment-piercing

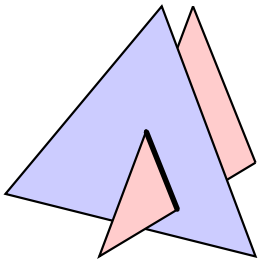
- Ligefrem algoritme
- Baseret på enkle geometriske observationer
- Beregningstung

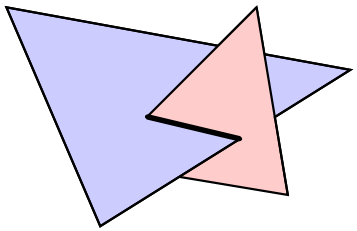
Devillers

- Mere kompliceret algoritme
- Baseret på „sideness tests“
- Terminerer hurtigt hvis der ikke er kollison

Trekant kollisionstests







Test om to trekanter er i kollision. Dvs. om de to trekanter har fælles punkter.

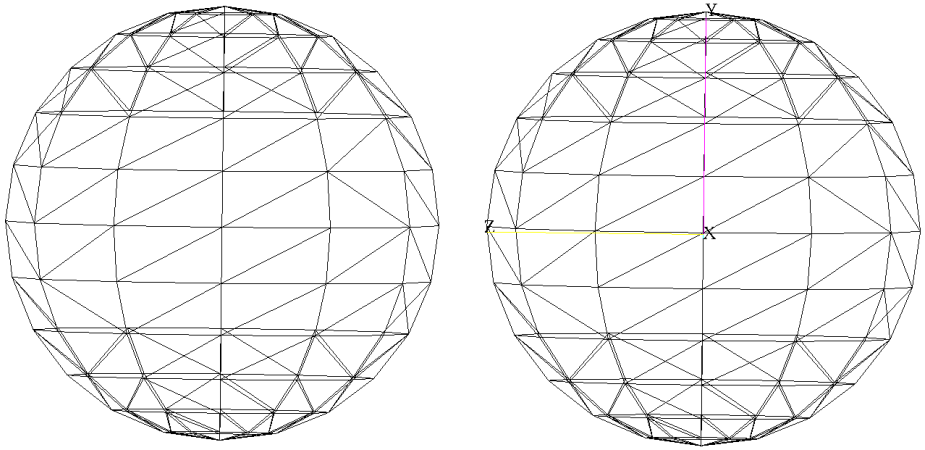
Segment-piercing

- Ligefrem algoritme
- Baseret på enkle geometriske observationer
- Beregningstung

Devillers

- Mere kompliceret algoritme
- Baseret på „sideness tests“
- Terminerer hurtigt hvis der ikke er kollison

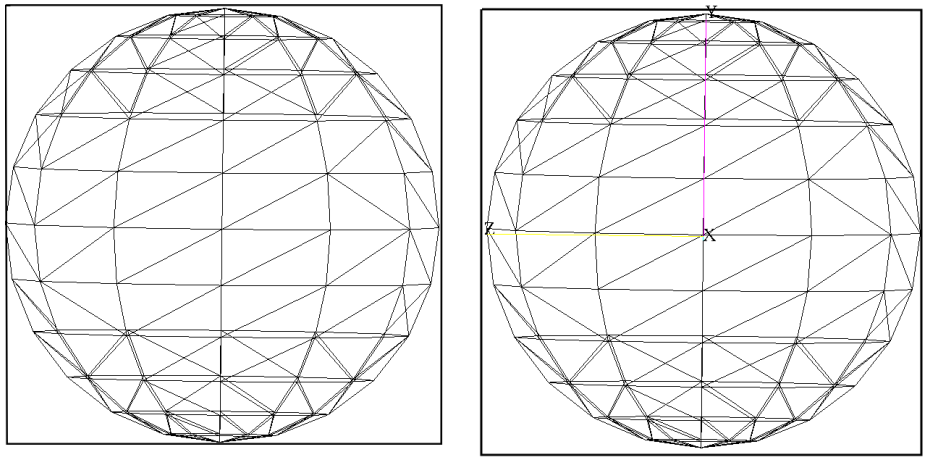
Bounding volumes



Massiv trekant-trekant test

- Alle trekanter skal testes mod hinanden
- Mange tests!

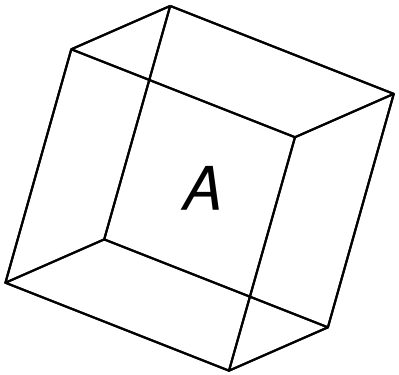
Bounding volumes

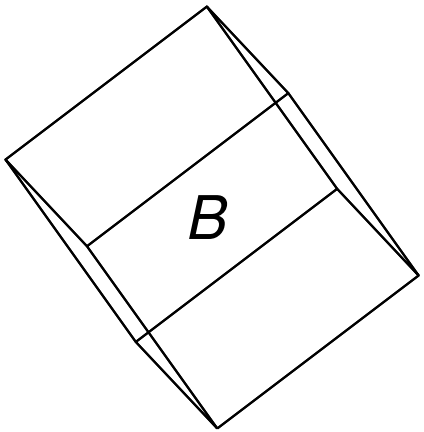


Brug af bounding volumes

- Geometrisk approksimation til oprindeligt objekt
- Benyttes i flere „lag“ (hierarki)
- Kan drastisk reducere antal nødvendige tests
- Oriented Bounding Boxes i dette projekt

Oriented Bounding Box kollisionstests





Test om to OBB'er er i kollision

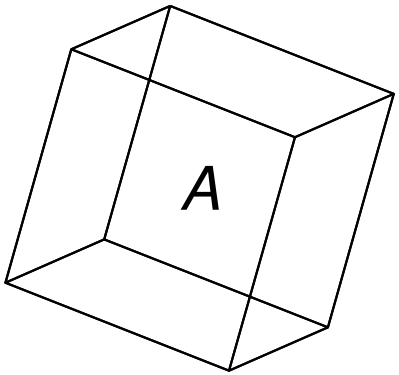
Exhaustive Edge-Face

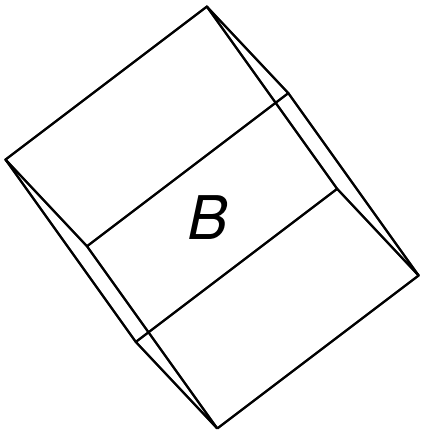
- Ligefrem algoritme
- Baseret på enkle geometriske observationer
- Meget beregningstung, kun anvendelig som reference

Gottschalk

- Baseret på test af separerende akser
- Terminerer hurtigt hvis der ikke er kollision

Oriented Bounding Box kollisionstests





Test om to OBB'er er i kollision

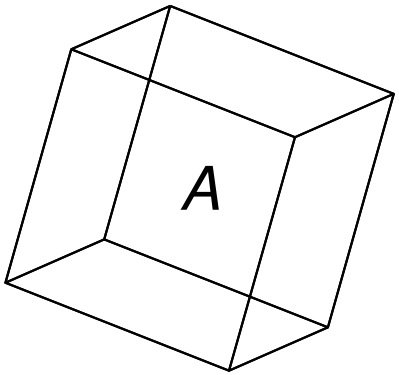
Exhaustive Edge-Face

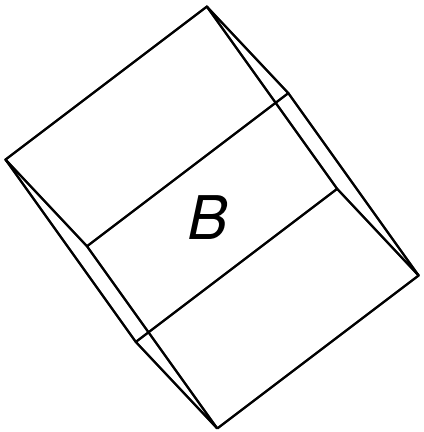
- Ligefrem algoritme
- Baseret på enkle geometriske observationer
- Meget beregningstung, kun anvendelig som reference

Gottschalk

- Baseret på test af separerende akser
- Terminerer hurtigt hvis der ikke er kollision

Oriented Bounding Box kollisionstests





Test om to OBB'er er i kollision

Exhaustive Edge-Face

- Ligefrem algoritme
- Baseret på enkle geometriske observationer
- Meget beregningstung, kun anvendelig som reference

Gottschalk

- Baseret på test af separerende akser
- Terminerer hurtigt hvis der ikke er kollision

Acceleration

Selv med smarte algoritmer er der en nedre grænse for hvor lidt arbejde vi skal udføre!

Algoritme	Antal operationer
Segment piercing	480
Devillers	124
Exhaustive	11520
Gottschalk	228

På en moderne PC (dual core @ 3GHz $\approx$ 24 GFLOPS/s) betyder det at vi kan forvente en teoretisk ydelse på:

- 193 millioner trekant tests (Devillers)
- 105 millioner OBB tests (OBB)

I virkeligheden er disse tal dog typisk en faktor 10 lavere!

Acceleration

Selv med smarte algoritmer er der en nedre grænse for hvor lidt arbejde vi skal udføre!

Algoritme	**Antal operationer**
Segment piercing	480
Devillers	124
Exhaustive	11520
Gottschalk	228

På en moderne PC (dual core @ 3GHz $\approx$ 24 GFLOPS/s) betyder det at vi kan forvente en teoretisk ydelse på:

- 193 millioner trekant tests (Devillers)
- 105 millioner OBB tests (OBB)

I virkeligheden er disse tal dog typisk en faktor 10 lavere!

Acceleration

Selv med smarte algoritmer er der en nedre grænse for hvor lidt arbejde vi skal udføre!

Algoritme	**Antal operationer**
Segment piercing	480
Devillers	124
Exhaustive	11520
Gottschalk	228

På en moderne PC (dual core @ 3GHz $\approx$ 24 GFLOPS/s) betyder det at vi kan forvente en teoretisk ydelse på:

- 193 millioner trekant tests (Devillers)
- 105 millioner OBB tests (OBB)

I virkeligheden er disse tal dog typisk en faktor 10 lavere!

Acceleration

Selv med smarte algoritmer er der en nedre grænse for hvor lidt arbejde vi skal udføre!

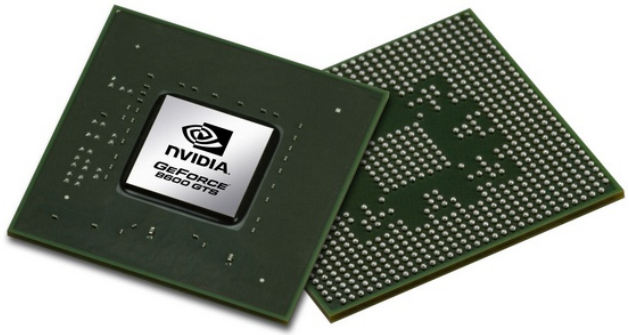
Algoritme	**Antal operationer**
Segment piercing	480
Devillers	124
Exhaustive	11520
Gottschalk	228

På en moderne PC (dual core @ 3GHz $\approx$ 24 GFLOPS/s) betyder det at vi kan forvente en teoretisk ydelse på:

- 193 millioner trekant tests (Devillers)
- 105 millioner OBB tests (OBB)

I virkeligheden er disse tal dog typisk en faktor 10 lavere!

GeForce 8



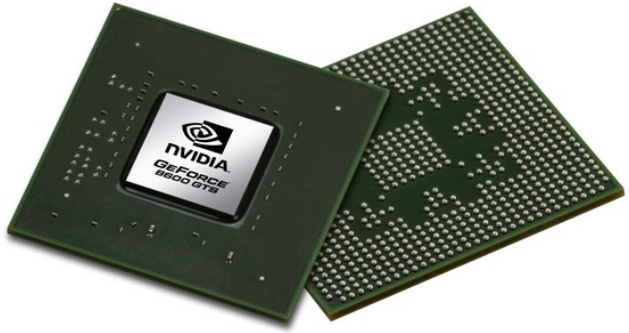
Egenskaber

- 8. generations grafikkort fra NVIDIA
- Unified shader arkitektur

GeForce 8800 GTX

- 16 multiprocessorer @ 1.35 GHz
- 768 MiB RAM, 86.4 GiB/s
- PCI Express, 4 GiB/s

GeForce 8



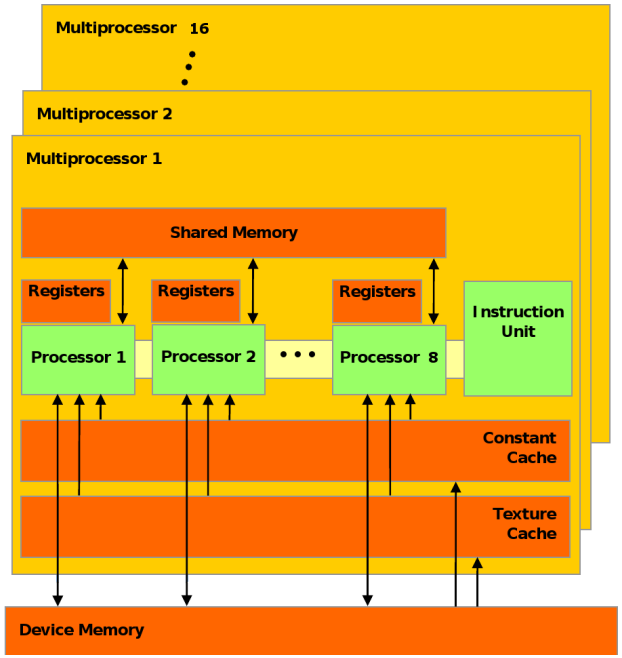
Egenskaber

- 8. generations grafikkort fra NVIDIA
- Unified shader arkitektur

GeForce 8800 GTX

- 16 multiprocessorer @ 1.35 GHz
- 768 MiB RAM, 86.4 GiB/s
- PCI Express, 4 GiB/s

Hvad gør GeForce 8 serien interressant?



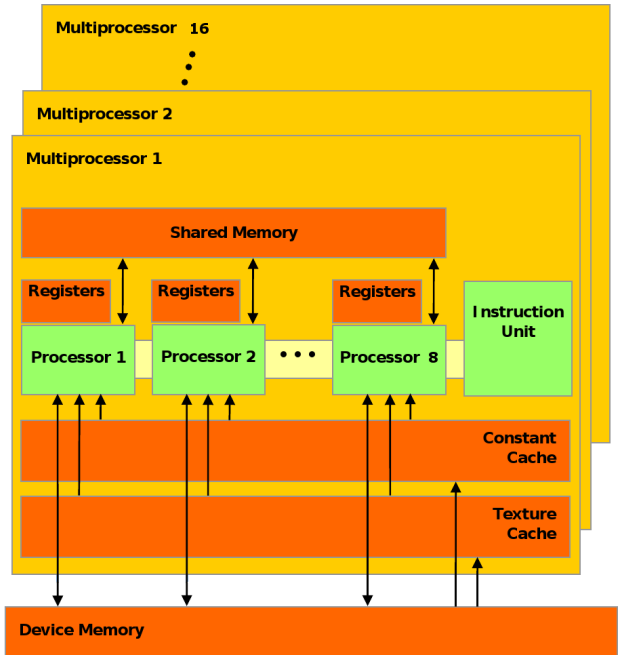
Massiv parallelitet

- 16 styks 8-way SIMD multiprocessorer
- 21.6 GFLOPS/s per multiprocessor

Massiv trådning

- Tusindvis af tråde scheduleres
- Afhjælper memory latency

Hvad gør GeForce 8 serien interressant?



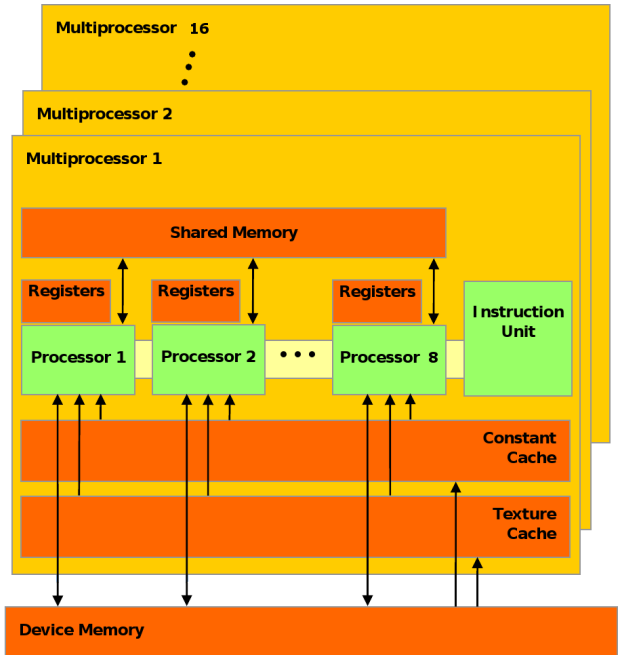
Massiv parallelitet

- 16 styks 8-way SIMD multiprocessorer
- 21.6 GFLOPS/s per multiprocessor

Massiv trådning

- Tusindvis af tråde scheduleres
- Afhjælper memory latency

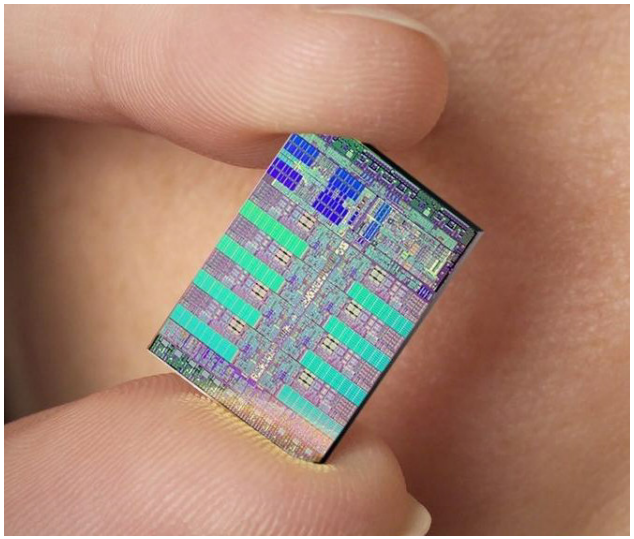
Hvad gør GeForce 8 serien interressant?



Total teoretisk ydelse:

345.6 GFLOPS/s

CELL Broadband Engine



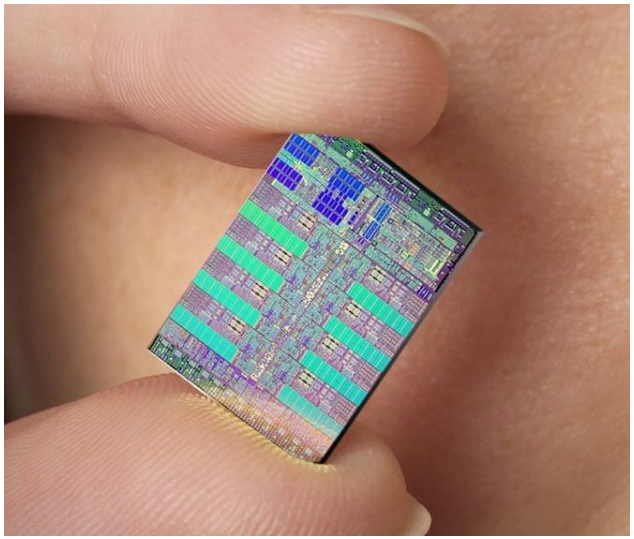
Egenskaber

- Udviklet af IBM, Toshiba og Sony
- Heterogen multiprocessor
- 1 General purpose processor
- 8 Vektor processorer
- High bandwidth interconnect

Sony Playstation 3

- CELL/BE @ 3.2 GHz
- 256 MiB XDR DRAM
- “Kun” 6 Vektor processorer
- Linux (Fedora Core 7)

CELL Broadband Engine



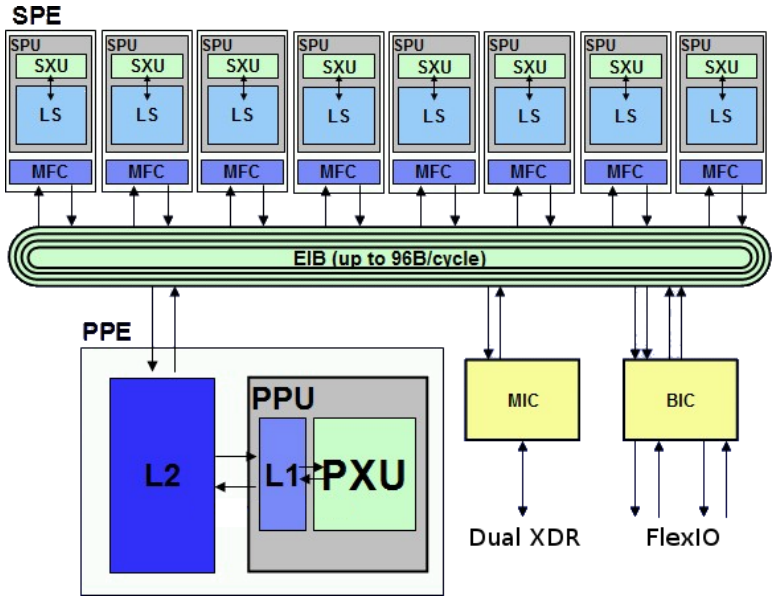
Egenskaber

- Udviklet af IBM, Toshiba og Sony
- Heterogen multiprocessor
- 1 General purpose processor
- 8 Vektor processorer
- High bandwidth interconnect

Sony Playstation 3

- CELL/BE @ 3.2 GHz
- 256 MiB XDR DRAM
- “Kun” 6 Vektor processorer
- Linux (Fedora Core 7)

Hvad gør CELL/BE speciel?



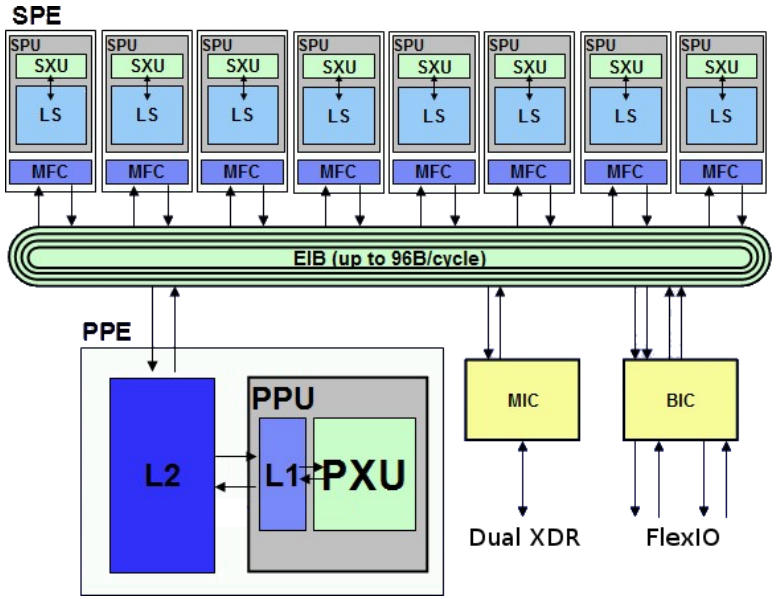
PPE

- 64 bit Power PC (x2)
- RISC med VMX (AltiVec)
- 32 KiB L1/512 KiB L2
- 25.6 GFLOPS/s

SPE

- 128 bit processor
- RISC med SIMD
- 256 KiB Lokal SRAM
- 25.6 GFLOPS/s

Hvad gør CELL/BE speciel?



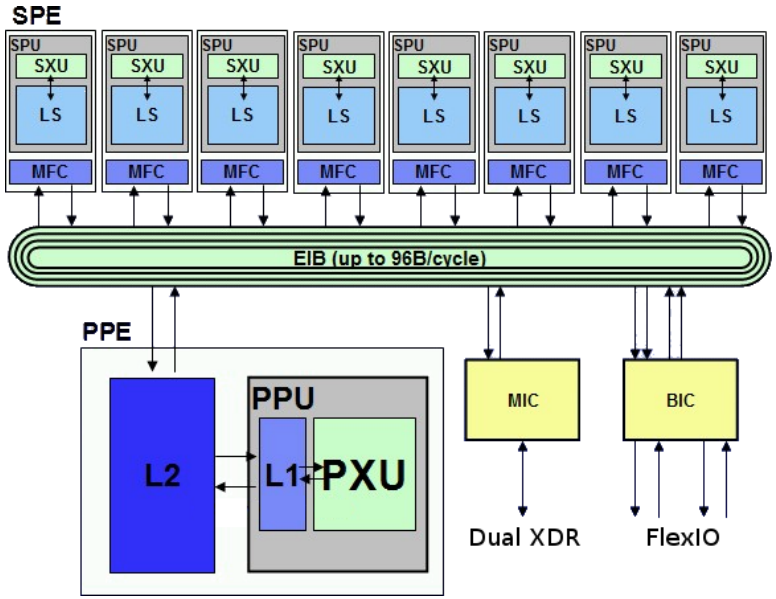
PPE

- 64 bit Power PC (x2)
- RISC med VMX (AltiVec)
- 32 KiB L1/512 KiB L2
- 25.6 GFLOPS/s

SPE

- 128 bit processor
- RISC med SIMD
- 256 KiB Lokal SRAM
- 25.6 GFLOPS/s

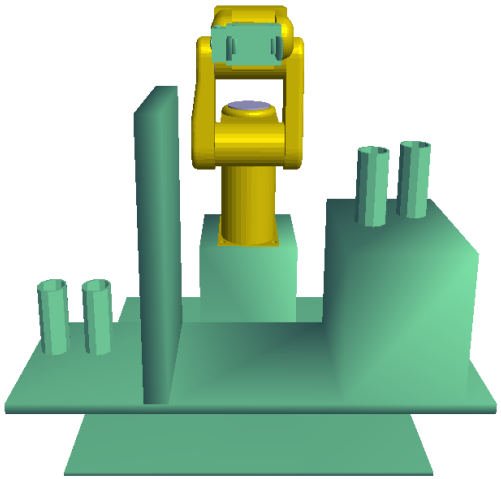
Hvad gør CELL/BE speciel?



Total teoretisk ydelse:

230.4 GFLOPS/s

Algoritme implementation



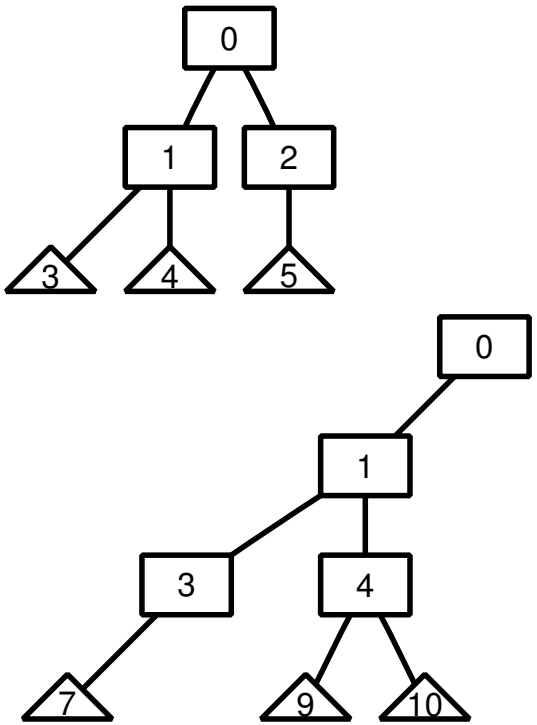
Scenen indeholder

- Hele modellen
- Samling af objekter
- Hvilke objekt par skal testes?

Et objekt

- Stift legeme (Rigid body)
- Samling af trekanter
- Opdeles hierarkisk vha. OBB'er

Algoritme implementation



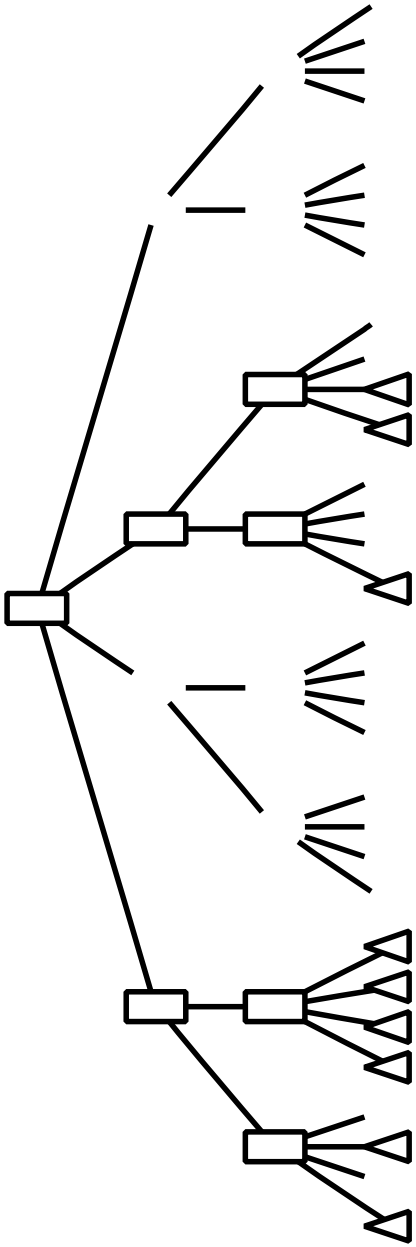
Scenen indeholder

- Hele modellen
- Samling af objekter
- Hvilke objekt par skal testes?

Et objekt

- Stift legeme (Rigid body)
- Samling af trekanter
- Opdeles hierarkisk vha. OBB'er

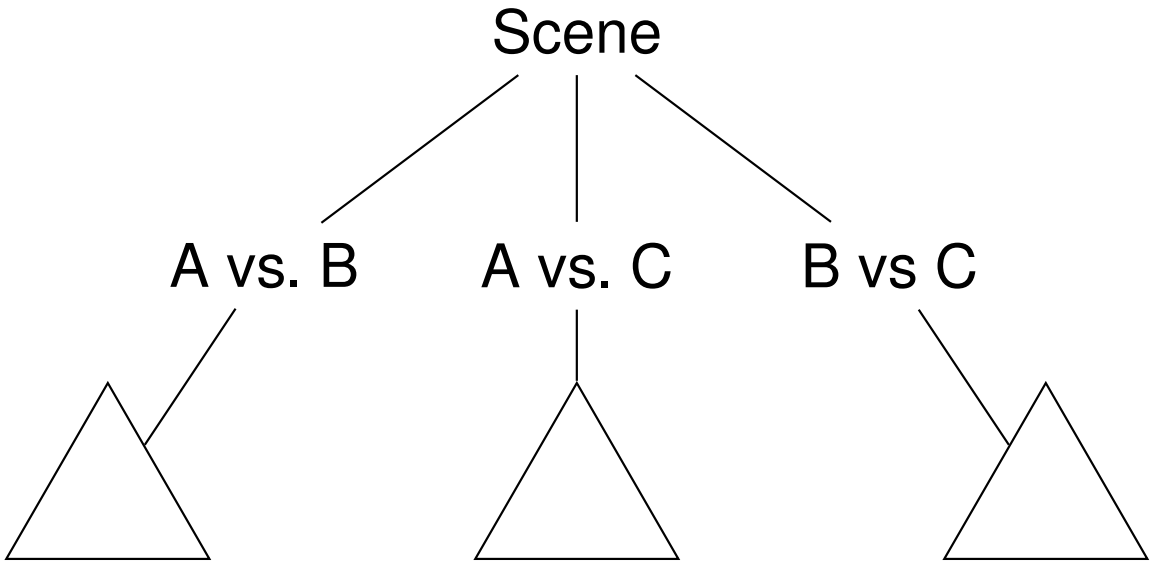
Collision Test Tree



CTT

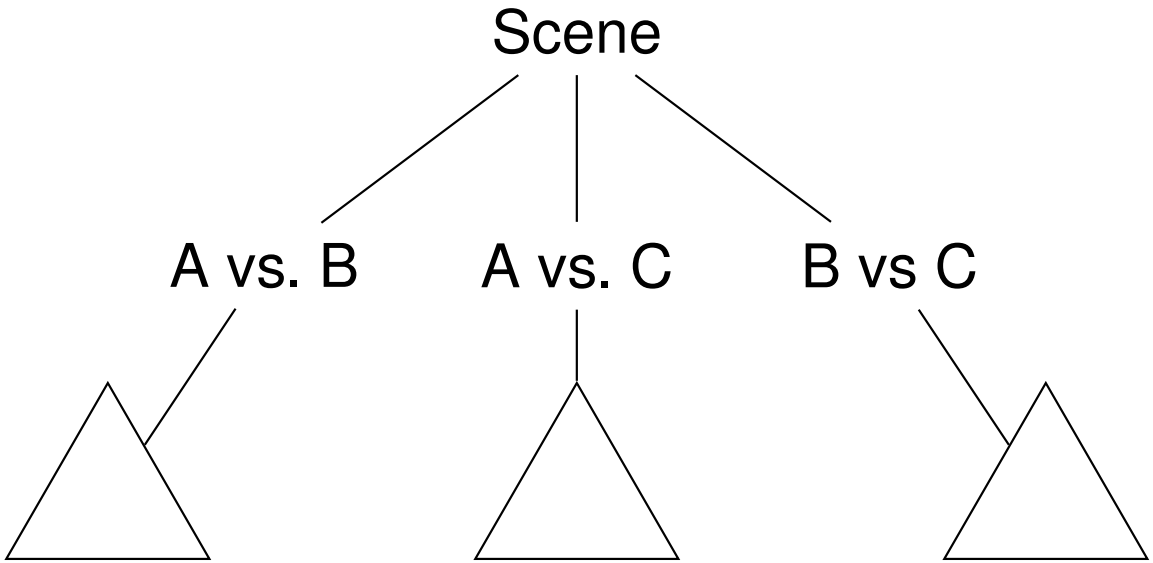
- Beskriver alle tests der skal udføres mellem to objekter
- Hver node svarer til en kollisionstest
- Fastlagt traversering og nedstigning
- Pladskrævende! - Men stadig nyttig abstraktion
- Virtuel repræsentation vha. implicit datastruktur

Generalisering af CTT til hele scenen



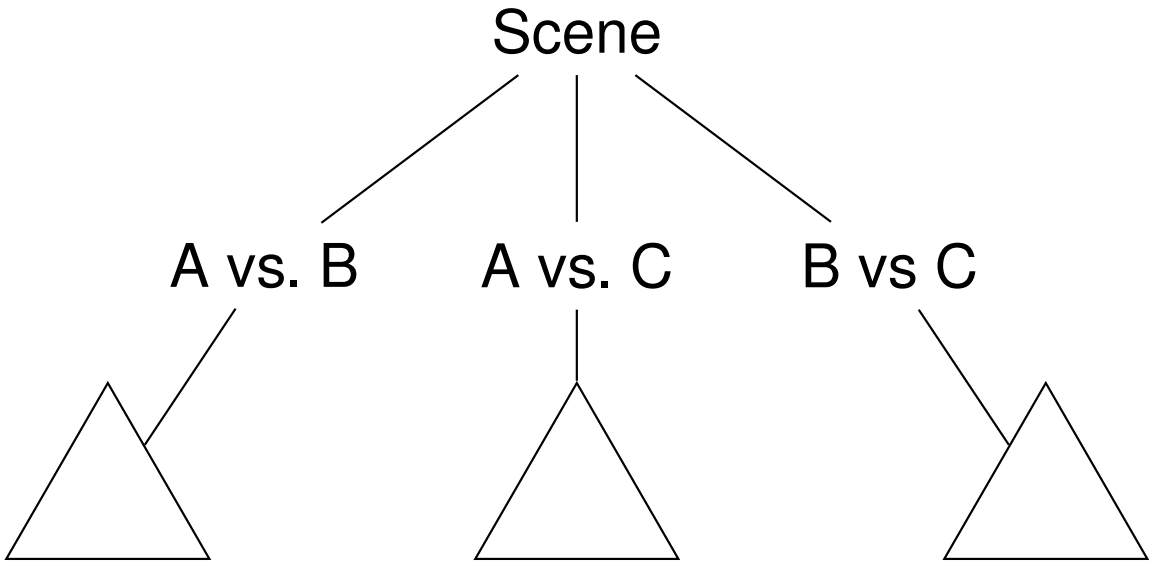
- Omfatter alle objekt par i scenen der skal tests for kollision
- Et træ $\Rightarrow$ Et problem!
- Mange tests! $\Rightarrow$ Mere paralleliserbar
- Dybde-først søgning, kollisionstests udføres parallelt

Generalisering af CTT til hele scenen



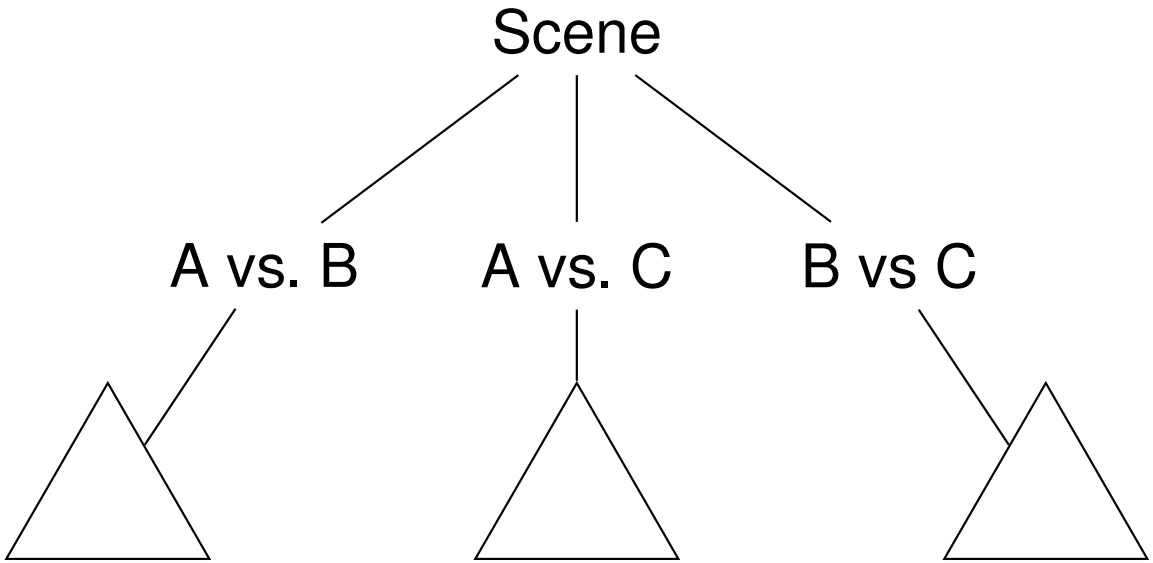
- Omfatter alle objekt par i scenen der skal tests for kollision
- Et træ $\Rightarrow$ Et problem!
- Mange tests! $\Rightarrow$ Mere paralleliserbar
- Dybde-først søgning, kollisionstests udføres parallelt

Generalisering af CTT til hele scenen



- Omfatter alle objekt par i scenen der skal tests for kollision
- Et træ $\Rightarrow$ Et problem!
- Mange tests! $\Rightarrow$ Mere paralleliserbar
- Dybde-først søgning, kollisionstests udføres parallelt

Generalisering af CTT til hele scenen



- Omfatter alle objekt par i scenen der skal tests for kollision
- Et træ $\Rightarrow$ Et problem!
- Mange tests! $\Rightarrow$ Mere paralleliserbar
- Dybde-først søgning, kollisionstests udføres parallelt

Opnåede resultater

GeForce 8

- 71 millioner trekant-trekant tests/s (speedup på 4)
- 164 millioner OBB-OBB tests/s (speedup på 11)
- Speedup på 1.4 ved model tests

CELL/BE

- 52 millioner trekant-trekant tests/s (speedup på 2.9)
- ??? millioner OBB-OBB tests/s
- Speedup på 1.6 ved model tests

Opnåede resultater

GeForce 8

- 71 millioner trekant-trekant tests/s (speedup på 4)
- 164 millioner OBB-OBB tests/s (speedup på 11)
- Speedup på 1.4 ved model tests

CELL/BE

- 52 millioner trekant-trekant tests/s (speedup på 2.9)
- ??? millioner OBB-OBB tests/s
- Speedup på 1.6 ved model tests