

Avaruusjako tietokonegrafiikassa

Timo Heinonen
LuK-tutkielma
tietojenkäsittelytiede
Turun yliopisto

Lokakuu 2016

Sisältö

1	Johdanto	2
2	Tietokonegrafiikan peruskäsitteitä	2
2.1	Avaruus \mathbb{R}^3 , objektit ja polygonit	2
2.2	Ray-Tracing tekniikka	2
3	Avaruusjakopuut	3
3.1	Binary Space Partitioning	3
3.2	Bounding Volume Hierarchy	3
4	Renderoinnin optimoiminen avaruusjakopuiden avulla	3
4.1	Aliavaruuspuuta käyttävä rekursiivinen Ray-Tracing algoritmi	3

Lista algoritmeista

1	Ray-Tracing -algoritmi	2
---	----------------------------------	---

1 Johdanto

Hello World!

2 Tietokonegrafiikan peruskäsitteitä

2.1 Avaruus \mathbb{R}^3 , objektit ja polygonit

2.2 Ray-Tracing tekniikka

```
Input:
Kuvataso:  $x * y$  kokoinen taulukko pikseleitä
Maisema: joukko valonlähteitä ja polygoneihin jaettuja objekteja
Output:
Kolmiulotteinen maisema projisoituna kuvatasolle
1 foreach pikseli  $(x, y)$  näytöllä do
2   foreach polygoni maisemassa do
3     Ammu säde  $\vec{R} = O + t\vec{D}_1$  kamerasta pikselin läpi maisemaan
4     if säde osui polygoniin pisteessä  $P$  then
5       foreach Valonlähde  $L$  do
6         Ammu varjostussäde  $\vec{R} = L - P$  valonlähdettä kohti
7         Kasvata pisteeseen  $P$  kohdistuvaa valosummaa
8       end
9       Aseta pikselin  $(x, y)$  väri valosumman mukaisesti
10    else
11      Aseta pikseli  $(x, y)$  taustan väriksi
12    end
13  end
14 end
```

Algoritmi 1: Ray-Tracing -algoritmi

Ray-Tracing -tekniikan pseudokoodi on esitetty algoritmissa 1

Algoritmin suoritusta hidastaa se, että jokaista sädettä kohti on käytävä läpi kaikki maiseman polygonit ja testattava osuuko säde niihin. Säteiden ja polygonien yhteistörmäyksien määrittämiseen joudutaan joissain tapauksissa käyttämään jopa 95% koko laskenta-ajasta. [8] Algoritmia saataisiin siis nopeutettua huomattavasti, jos testattavien polygonien määrää jokaista sädettä kohti saataisiin vähennettyä.

3 Avaruusjakopuut

3.1 Binary Space Partitioning

3.2 Bounding Volume Hierarchy

4 Renderoinnin optimoiminen avaruusjakopuiden avulla

4.1 Aliavaruuspuuta käyttävä rekursiivinen Ray-Tracing algoritmi

Viittaus [6], viittaus [4] ja viittaus [5] [2] [3] [1] [7]

Viitteet

- [1] Arthur Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, AFIPS '68 (Spring), pages 37–45, New York, NY, USA, 1968. ACM.
- [2] Martin Brownlow. *Game Programming Golden Rules*. Charles River Media / Cengage Learning, 2004.
- [3] Henry Fuchs, Zvi M. Kedem, and Bruce F. Naylor. On visible surface generation by a priori tree structures. *SIGGRAPH Comput. Graph.*, 14(3):124–133, July 1980.
- [4] John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley. *Computer graphics: principles and practice (3rd ed.)*. Addison-Wesley Professional, Boston, MA, USA, July 2013.
- [5] Steven J Janke. *Mathematical Structures for Computer Graphics*. Wiley, 2015.
- [6] Samuel Ranta-Eskola. Binary space partitioning trees and polygon removal in real time 3d rendering. Master's thesis, Uppsalan yliopisto, Uppsala, Ruotsi, 2001.

- [7] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
- [8] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, June 1980.