Hur man ger datorn en miniräknare

Niclas Thuning och Leo Bärring

Mars 2016

För att vi ska kunna njuta av bättre och bättre grafik i datorspel krävs det allt snabbare beräkningsmetoder. Det är en ständig kamp för ingenjörer att tillgodose nya snabbare lösningar. En lösning är att utveckla helt nya algoritmer.

Precis som när man ställer upp en multiplikation med papper och penna, kan många beräkningar beskrivas med enkla datorinstruktioner. Behöver man räkna snabbare än vad man klarar av att skriva, kan man ta hjälp av en miniräknare. I datorernas värld är motsvarigheten speciella kretsar som är konstruerade for att utföra vissa specifika beräkningar. Precis som att en avancerad miniräknare är dyrare än en som bara behandlar de vanligaste räknesätten, så gör den här typen av kretsar att systemet de byggs in i blir dyrare. Därför är det viktigt att de används så mycket att det upphäver kostnaden, t.ex. genom att de är snabbare, eller drar mindre energi. Ofta talar man om då om signalprocessorer, som används i specifika sammanhang.

Man kan tänka sig en avancerad datorrenderad scen i ett spel, eller en film, simuleringar som kan innehålla massvis med reflekterade ljusstrålar. Dessa måste dessutom beräknas på nytt om ett objekt ändras så att ljuset reflekteras annorlunda. I sådana fall kan det vara en god idé att flytta över vissa beräkningar från att beskrivas med en sekvens av huvudprocessorns mer generella funktioner, till en specifik komponent som kan göra beräkningen i ett enda svep.

Ta t.ex. den inversa kvadratroten, d.v.s. $1/\sqrt{x}$, en funktion som är väldigt användbar vid beräkningar av hur ljus reflekteras, eller objekt studsar när de träffar en plan yta.

Det finns olika algoritmer att basera den här typen av kretsar på, och vilken algoritm som är lämplig varierar bland annat med vilken beräkning som ska utföras. Att avgöra vilken algoritm som är lämplig för en viss ekvation kan ibland kräva att kretsarna simuleras i olika utföranden.

Två algoritmer som här ställts mot varandra är *Newton-Raphsons metod*, upkallad efter Isaac Newton och Joseph Raphson, som härrör från slutet av sextonhundratalet, samt den nya *Harmonized Parabolic Synthesis*, som tagits fram av Erik Hertz och Peter Nilsson. Båda algoritmerna används här till att beräkna $1/\sqrt{x}$.

Undersökningen mellan dessa två algoritmer har visat sig få olika utfall beroende på vilka egenskaper man söker. *Harmonized Parabolic Synthesis* visar sig i de flesta fall vara bättre på att ge en mindre chip-area, högre beräkningshastighet och bättre spridning av felet. (Datorberäkningar av den här typen är sällan fullständigt exakta, så ofta finns det krav på hur beräkningsfelet ska se ut rent statistiskt sett.) *Newton-Raphsons metod* visade sig ha lägst energiförbrukning som största fördel.

Beroende på applikation blir valet antigen *Harmonized Parabolic Synthesis* eller *Newton-Raphsons metod*. För applikationer som använder batterier rekommenderas *Newton-Raphsons metod* för dess energiegenskaper, medan en snabb och/eller exaktare krets kan kräva *Harmonized Parabolic Synthesis*.

Undersökningen skedde i två steg, i den första togs själva beskrivningen för att få ett bra beräkningsresultat fram. I det andra steget simulerades kretsens fysiska egenskaper.