

# Att rita rök snabbare på en dator

POPULÄRVETENSKAPLIG SAMMANFATTNING **Victor Strömberg**

Dagens datorgrafik strävar ständigt efter att skapa mer verklighetstroga bilder. För att kunna lägga in fler detaljer i bilden måste vi förbättra det sätt vi gör det på. Således testar vi ett alternativt sätt att rita upp rök i en dator snabbare.

## Inledning

För att på ett snabbare sätt kunna rita verklighetstroga bilder med grafikkortet i en dator så är det viktigt att förbättra, optimera, denna process. Man vill med andra ord förkorta den tid det tar att rita en bild. Desto mer man kan rita upp på kort tid, desto detaljrikare kan bilderna bli. Utan optimering så kan bildsekvenserna upplevas som sega och de tenderar att släpa efter.

Vid simulering av rök eller då man exempelvis skall visualisera bilder från en magnetröntgen använder man sig av volymer när man ritar bilderna i datorn. En volym kan anta olika former av tredimensionella kroppar. En enkel och vanlig form är en kub. I kuben finns data som representerar den volym man vill rita. Det är väldigt krävande att låta datorn rita volymer och därför är optimering av sådana processer av stort intresse.

## Ray marching

När man ritar volymer så använder man sig av en teknik som heter *ray marching*. Tänk dig att du tar en bild med en vanlig kamera och från ditt öga så skjuter du ut strålar. Varje stråle träffar och går igenom en punkt på fotot du tänker ta. Strålen fortsätter genom fotot och ut i verkligheten och träffar det du vill fotografera. Beroende på vad strålen träffar så blir denna punkt som strålen gått igenom färgad olika. Tänk dig nu att när vi skjuter en stråle så hoppar vi också fram på denna med små steg. För varje steg vi flyttar oss framåt på strålen så kollar vi om vi har träffat nått. Detta är konceptet för ray marching.

## Problem

När vi träffar vår volym så börjar vi ta ännu mindre steg genom hela volymen. För varje litet steg vi tar så hämtar vi information från volymen om hur röken ser ut just här för denna stråle. Finns det ingen information att hämta här, volymen är tom, så försöker vi ändå hämta den. Detta är slöseri med beräkningskraft och vi vill i stället försöka optimera detta så att vi inte behöver hämta information som ändå inte ger oss någonting.

## Lösning

För att möjliggöra en optimering byggs en trädstruktur, ett så kallat *octree*, upp av röken. Med hjälp av denna struktur kan vi dela in volymen i bitar. Dessa bitar kan då säga oss om det finns information här som vi kan använda. Är det tomt, så talar vårt octree om det för oss och hoppar då förbi allt detta tomrum direkt. Därmed kan vi spara beräkningskraft.

## Resultat

Resultaten visar att metoden har potential, men det sätt vi frågar vårt octree på omöjliggör en optimering. Detta beror på hårdvarubegränsningar i dagens grafikkort som grundar sig i dess oförmåga att förutspå och hantera komplexa förgreningar av kod. Octree:t ger dock ett mer noggrant värde av röken då vi varken hämtar för mycket eller för lite information om den i volymen. Detta resulterar i en mer visuellt korrekt återgiven rök.