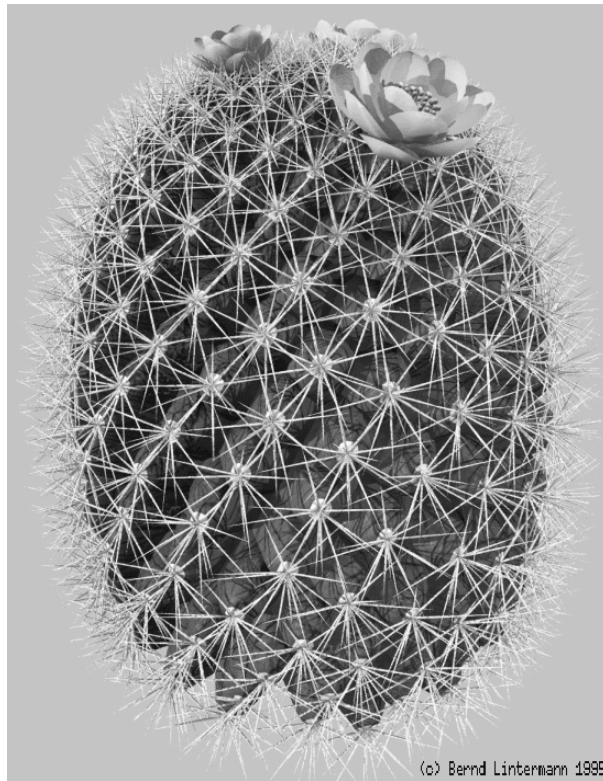


NORSIGD INFO

Nummer 1 2000



(c) Bernd Linbermann 1995

NORSK SAMARBEID INNEN GRAFISK DATABASEHANDLING

ISSN 0803-8317

Fotorealistisk grafikk gjennom bildebasert rendering

Peter Oel, Jens Riemschneider, Universitetet i Karlsruhe

Det har alltid vært et mål å lage datamaskingenererte bilder mest mulig fotorealistisk. Istedenfor å lage modeller som brukes i rendering-fasen, går bildebasert rendering (image-based rendering, IBR) en annen vei: Utgangspunktet er kameraopptak av objekter som brukes for å generere scener og bilder. Vi gir en oversikt over aktuelle metoder som brukes i dette unge fagfeltet.

Fotorealistiske datagrafikk-bilder lages vanligvis i en rendering-prosess, der det brukes 3D-scener som er bygd opp av geometriske objekter, f.eks. flater, kuler, triangler, terninger, etc. Både modellen og billedgenereringsalgoritmen må orientere seg mot forbildet: naturen selv. Med systemer som AutoCAD eller 3D Studio Max kan bare spesialister lage realistiske bilder. Hardwarekravene er dessuten veldig store. Dyre prosessorer når fort opp til sine begrensninger. Konsekvensen er ofte at den som har modellert sceneene bruker lange pauser for å vente på at et bilde blir ferdig prosessert.

Istedenfor å konstruere scener som består av mange detaljer kan det tas utgangspunkt i bilder fra en scanner eller et digitalt kamera. For å lage scener som er egnet som utgangspunkt for fotorealistiske bilder trengs det programvare som gjenkjenner og rekonstruerer objekter. Dessverre er den virkelige verdenen så kompleks at selv de raskeste datamaskinene ikke strekker til, f.eks. når millioner av tynne hår eller bladene til et tre skal modelleres. Ofte vil derfor resultatene bli altfor unøyaktige og utilstrekkelige.

Parametre som farge, glans, refleksjon, transparens og overflateegenskaper av objekter må beskrives for å oppnå et realistisk utseende. Dessverre er akkurat disse egenskapene til objektene forstyrrende i rekonstruksjonsprosessen. Å ekstrahere den geometriske modellen fra fotografier blir derfor sjelden brukt som modellingsmetode.

I 1980 fikk A. Lippman en idé om å lagre mange bilder av en scene på en videodisk. Movie-map-metoden velger det korrekte bilde for en gitt posisjon. Dessverre har denne metoden et høyt minneforbruk. Movie-map metoden viser et interessant aspekt: metoden bruker ikke geometri-informasjon for scenen, men kjenner bare til mange forskjellige kamera-opptak.

Image-based rendering (IBR) bruker en lignende teknikk, men istedenfor å lagre alle opptak av en scene på harddisken, beregnes nye opptak ut ifra relativt få original-opptak, uten å bruke objekt-gjenkjenning i bildene.

IBR er ennå et ungt og dynamisk forskningsområde. Blant teknikkene som brukes er *Panoramateknikk*, *Light field* og *View Morphing* de mest fremtredende. Noen forskningsgrupper viser også animasjoner basert på IBR, og det finnes allerede spill og presentasjonssystemer basert på IBR.

Billedgenerering basert på IBR består oftest av tre steg: Først leses basis-data inn, som foreligger i form av rekke fotografier fra ulike synsvinkler. Ut ifra dette bygges det egnede datastrukturer, som må kunne lagre mange detaljer på en egnet måte. I det siste steget beregnes det nye bilder ut ifra denne datastrukturen.

Stråler innen IBR

Innen IBR brukes det projeksjonsstråler for å projisere gjenstand på en 2D-flate. For IBR projiseres objekter ikke bare på plane flater, men også på bøyde overflater, f.eks. sylindre, kuler, terninger, mm. Når en lysstråle treffer en flate blir informasjonen om fargen til denne strålen lagret for dette punktet i flaten. Denne informasjonen brukes når det skal lages et nytt opptak fra en annen synsvinkel. Noen av strålene fra disse opptakene går gjennom projeksjonsflaten til det nye bildet, der man får fargeverdier for de enkelte billedpunktene. Foreligger det nok bilder for en scene, og dermed nok stråler, kan algoritmen beregne nok billedpunkter for det nye bildet.

En av hovedproblemstillingene er å lagre så mange stråler som mulig på minst mulig minneplass. IBR teknikkene bruker forskjellige metoder for å lagre disse strålene. Målet er å tilby minst mulig begrensninger i bildet med et minimum av minnebruk.

Det finnes begrensninger for noen av IBR metodene iht. posisjonen til tilskueren, og dens synsretning da det praktisk ikke er mulig å samle nok informasjon om alle stråler. Quicktime VR arbeider med en panoramateknikk som tillater forskjellige synsvinkler for en fast plass. Light field teknikken lagrer alle stråler som skjærer to

parallele flater, mens view morphing baseres på sammenhengen mellom stråler som skjærer det samme punktet i et bilde.

Panoramateknikk

I panoramateknikken kan opptakene bare tas fra ett sted, mens den tillater mange synsretninger. Som egnet lagringsprinsipp for billedinformasjonen brukes det et panoramabilde på 360° . Det finnes kameraer som lager slike panoramabilder i en prosess. Finnes det bare enkeltopptag tilgjengelig, blir disse projisert på en sylindrisk flate. Fotografiene skal helst ha overlapp for å få et kontinuerlig bilde. Algoritmen beregner nye bilder ved å ta utsnitt av panoramaet som projiseres på en flate.

Apples Quicktime VR er en av de første IBR applikasjonene. Programmet kan håndtere både enkeltbilder ved hjelp av den såkalte *stitcher* og spesiallagde panoramabilder.

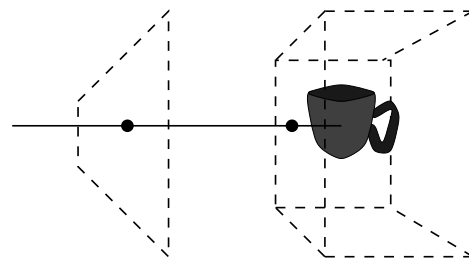
Et panorama blir hakket i småbiter (dicting) og komprimert omtrent som i en Quicktime film-fil. For å vise en scene dekomprimeres bare de delene som trengs, og en transformasjon til en plan projeksjon blir gjennomført (warping). Med en moderne PC kan disse opptakene beregnes i sanntid, og dermed kan utsnittet styres interaktivt. Quicktime VR kan lagre flere slike panoramabilder, som kan knyttes sammen slik at man skifter scene med et museklikk. Derimot er en kontinuerlig overgang mellom scener ikke mulig.

Med *Object Movie* opphever Apple denne begrensningen. For å lage en Object Movie legges det en virtuell kule rundt en gjenstand i scenen. Deretter tas det noen opptak av projeksjoner av det objektet på kulen som lagres på en Quicktime VR fil.

For en interaktiv presentasjon deles et vindu i forskjellige soner ved hjelp av et raster. Avhengig av sonen musepekeren befinner seg i, vises det rastelementet som er tilordnet dette bildet. I en utvidet versjon (absolute referenced object) kan det også tilordnes en (liten) film, f.eks. for å vise et levende voksllys fra forskjellige vinkler. En object movie som viser objekter fra den virkelige verdenen trenger mye minnekapasitet. For å få til en jevn avspilling må det foreligge mange opptak fra eksakte posisjoner, som er laget på forhånd. Nye opptak (som kan lages ved bruk av panoramateknikken) kan derimot ikke beregnes med *Object Movie*.

Light Field

Light Field teknikken tar et annet utgangspunkt: Fra en gjenstand fra virkeligheten tas det opptak fra flere retninger. Hvert opptak defineres som lysstråler som går fra objektet gjennom et projeksjonssenter og treffer billedflaten. Selve strålen bærer fargeegenskapene. Legges det en virtuell terning rundt gjenstanden, så treffer hver av disse strålene nøyaktig en av sidene til terningen, og i tillegg på en flate som ligger i en fast avstand til den trufne flaten til terningen. Begge skjæringspunkter samt farge info representerer det som kalles *Light Field* i en lysstråle. Skjæringspunktene kan lagres f.eks ved hjelp av 3D koordinater.



En light-field stråle som bærer fargeinformasjonen til et objekt defineres ved hjelp av to skjæringspunkter med flater.

For å redusere plassbehovet representeres dette ved hjelp av 2D koordinater i forhold til de kjente flatene. Dermed aksesseres et *Light Field* ved hjelp av et 4D-array. Fordelen består i besparelsen i antall parametre som også medfører en plassbesparelse.

For å vise et nytt opptak genererer algoritmen stråle-forespørsler slik som i raytracing-metoden. Hver stråle skjærer terningen og den tilhørende flaten, som definerer et punkt-par og dermed en bestemt 4D indeks. Selve informasjonen blir lest ut og fargen til et billedpunkt blir fastlagt. Vanligvis brukes rasterbilder på terningssidene med en oppløsning på 256×256 eller 128×128 . Dermed kommer den totale minnebruken opp i gigabyte-størrelse. For å bote på dette brukes det en komprimeringsmetode, som er en kombinasjon av Ziv-Lempel koding og vektorkvantisering.

Matrisen dekomprimeres allerede ved innlasting, fordi aksesstiden ellers ville blitt for stor. 4D feltet deles inn i blokker av naboindices, og komprimert per blokk. For hver stråle-forespørsel blir en slik blokk dekomprimert. Derfor behøver man ikke å dekomprimere hele light field, men bare de nødvendige delene.

På Internett finnes det noen eksempler tilgjengelig, som har en håndterbar størrelse på ca

2-6 MByte. Kvaliteten og fleksibiliteten til denne metoden vises spesielt når det foreligger komplekse belysningssituasjoner. Teknikken kan klare effekter som highlights, transparens og speil-effekter.

Flere bilder per sekund kan beregnes på dagens PCer. De vanlige grafikk-kortene kan teoretisk sett akselerere dette, men det finnes per idag ingen implementering tilgjengelig som utnytter dette.

Med *extended light fields* kan det genereres animasjoner. En overgangsfunksjon interpolerer mellom flere light fields slik man gjør i den såkalte 3D-morphing-teknikken.

View Morphing

Light Field metoden kan karakteriseres ved det store plassbehovet, samt det faktum at en høy komprimeringsrate kan oppnås. *View Morphing*-teknikken forsøker å utnytte redundans fra begynnelsen av uten å bruke komprimeringsmetoder.

Utgangspunktet er at kameraet beveges parallelt til billedflaten. Følger man et gitt punkt av et 3D-objekt på disse kameraopptakene, så beveger dens projeksjon seg også langs en linje. Avstanden mellom punktene er forskjellige, avhengig av overflaten til objektet.

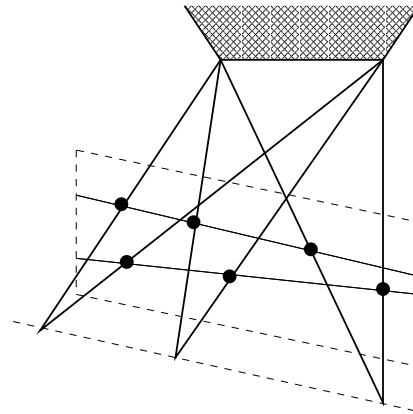
Et slikt punkt er ikke synlig på alle kameraopptak, fordi det kan være dekket til av andre deler av gjenstanden. For konvekse gjenstander kan dette imidlertid ikke skje. Det finnes også objekter der dette bare skjer i mindre grad (man tenke seg f.eks. en kaffekopp).

View morphing bruker denne egenskapen for å redusere antall opptak. Metoden lagrer to opptak og husker hvilke to billedpunkter i opptakene som tilsvarer det samme punkt på et objekt (korresponderende billedpunkter). På basis av dette genereres alle opptak som ligger mellom disse opprinnelige opptakene.

Matematikken bak view morphing er ganske enkel: Projeksjonssenteret til et nytt opptak ligger et sted på linjen mellom de billedpunktene på de to opptakene. View morphing beregner nye billedpunkter gjennom en lineær vektlegging av posisjonene.

Beregning av de korresponderende punktparene utgjør hoveddelen i metoden. Det finnes noen teknikker som er utviklet for noen spesielle situasjoner innen view morphing, men metodene kan utvides nærmest ubegrenset når det tas fler enn to kameraopptak. Når man beveger seg rundt en gjenstand og lagrer kameraposisjon og

bilde på egnede steder så kan man generere hele bevegelsen med bare få opptak. View Morphing har derfor mye mindre minnekrav enn light field, og metoden er enkelt og rask.



View morphing-teknikken utnytter at billedpunkter til et punkt av gjenstanden ligger på en linje når kameraposisjonene ligger på en linje.

View morphing har fått navnet sitt gjennom likheten med den velkjente morphing-metoden. Også der blir billedpunkter overført med lineær vektlegging når tiden går. View Morphing erstatter bruken av tid med bruken av kameraposisjoner.

Konklusjon

Verdenen som omgir oss er veldig kompleks. Realistisk utseende i datagenererte bilder blir idag bare laget av Raytracing og Radiosity metoder. De lange beregningstidene hindrer også i fremtiden dens bruk i omgivelser der sanntidsegenskaper trengs. Algoritmene kan til tider være rimelig kompliserte. IBR viser en utvei, da beregningstidene i mindre grad er avhengig av scene-kompleksiteten eller gjenstandene. Algoritmene settes sammen av enkle matematiske formler (ofte lineært) slik at beregningstider kan oppnås som tillater bruk av denne teknikken i sanntid. Noen spill utnytter disse metodene allerede. I et adventure-spill går man ikke lenger fra bilde til bilde, men fra panorama til panorama, f.eks. i *Black Dahlia* eller *Zork Nemesis*. Filmsekvenser forbinder panorama-sekvensene og forbedrer muligheten å orientere seg i spillet. Denne fremgangsmåten ligner veldig på fremgangsmåten i Quicktime VR.

Den største ulempen med IBR teknikkene er plassbehovet. Et komprimert light field har et minnebehov på flere MByte. Lagres scener som består av flere light fields er minnebehovet enormt.

Det finnes også hybridmetoder som både jobber med kameraopptak og er geometribaserte metoder. Ved Berkely University utviklet Paul Bebceric en metode som først bruker en grov geometri. Ved hjelp av IBR metoder blir så overflatestrukturen lagt til. Allerede med 10 til 20 bilder kan det genereres oppsiktsvekkende ani-

merte sekvenser. Slike metoder viser hvilket potensiale som ligger innen IBR. Fremtiden til datagrafikken ligger dermed ikke bare i å modellere triangler.

For den som vil vite mer om IBR metoder ligger det en side med mer informasjon på <http://i31www.ira.uka.de/~oel/ibmr-focus/>.

Oversikt over aktuelle IBR-teknikker

Teknikk	Funksjon	Posisjon tilskuer	Kommentarer
Delta Tree	Plan Plenoptic Modeling for objekter	fri posisjon utenfor en kule som omgir scenen, valgfri synsvinkel.	Kompakt og helst redundansfri lagring av data.
Plan Plenoptic Modeling	Tilbakeprojeksjon inn i ny billedplan, plane referansebilder.	fri posisjon med valgfri synsvinkel	forgjenger til Sylindrisk Plenoptic Modeling; problemer ved tildekte objekter.
Hybrid-metode	Blanding mellom IBR og geometri-basert rendering	alle	forbausende god kvalitet.
Light Field	Lagrer alle projeksjonsstråler som ligger i en terning	fri posisjon utenfor terningen som omgir scenen, valgfri synsvinkel	Programvare for konstruksjon og visualisering fritt tilgjengelig (også kildekode).
Lumigraph	lagrer alle projeksjonsstråler som ligger i en terning	fri posisjon utenfor terningen som omgir scenen, valgfri synsvinkel	Variant av Light Field, utviklet av Microsoft
Movie Map	Lagrer alle bilder, ingen interpolering	valgfri posisjon med valgfri synsvinkel, alle bilder er lagret.	ble utviklet som Digital Video Interactive (DVI) i begynnelsen av 80-årene.
Quicktime VR	Panoramabild med tilbakeprojeksjon inn i billedplan	fast posisjon med valgfri synsvinkel, dog synsvinkel er begrenset av sylinderhøyde.	første kommersielt produkt innen IBR.
Quicktime VR Object Movie	navigerbar film uten interpolering mellom bildene	Posisjon begrenset til en kule som ligger rundt objektet.	integrert i Quicktime VR.
View Morphing	lineær forskyvning av billedpunkter i flere referansebilder	fri posisjon med valgfri synsvinkel	enkel og rask metode
Sylindrisk Plenoptic Modeling	Tilbakeprojeksjon inn i ny billedplan fra flere panoramabilder	fri posisjon med valgfri synsvinkel, dog synsvinkel er begrenset av sylinderhøyde.	store problemer med tildekte objekter.

IBR på Web

Quicktime VR fra Apple er det mest kjente produktet som bruker IBR: www.apple.com/quicktime/qtvr/.

Light field rendering: www.graphics.stanford.edu/projects/lightfield/.

Volume Rendering med Extended Light Fields: www.ira.uka.de/~patrick/lightfields/.

View Morphing: www.cs.wisc.edu/~seitz/

interp/vmorph.html.

Arkitektur, generert med en hybrid-metode: www.cs.berkeley.edu/~debevec/Research/.

Plenoptisk modeling: www.cs.unc.edu/~mcmillan/penoptic.html.

Light-Field varianten Lumigraph vises på www.microsoft.com/SIGGRAPH96/96/Lumigraph.htm