

Kandidatarbete
Datavetenskap
Januari 2008

Prestandajämförelse mellan shadow mapping och shadow volumes i Direct3D 10

Eddie Willman & Aron Nieminen

Avdelningen för
Interaktion och systemdesign
Blekinge Tekniska Högskola
Box 520
372 25 Ronneby

Denna avhandling är inlämnad till avdelningen för interaktion och systemdesign vid Blekinge Tekniska Högskola som en del av kraven för kandidatexamen inom datorvetenskap. Arbetet motsvarar 10 veckors heltidsarbete.

Kontaktinformation:

Författare:

Eddie Willman

E-post: eddiewillman@gmail.com

Aron Nieminen

E-post: aronnieminen@gmail.com

Handledare:

Bo Helgeson

Avdelningen för interaktion och systemdesign

Avdelningen för
Interaktion och systemdesign
Blekinge Tekniska Högskola
Box 520
372 25 Ronneby

Internet: <http://www.bth.se/tek/ais>
Telefon: 0457 38 58 00
Fax: 0457 271 25

Abstract

Shadows are important for the human perception of the world. In video games and other interactive 3D applications it is important to ease the understanding of the scene and realistic environments are often desired. This, and the fact that shadow rendering is complicated and computation intensive, makes shadows an important subject in real time graphics. In this thesis, we investigate the relation between the performances, measured in render time, of the two most popular methods in real time graphics; shadow mapping and shadow volumes. The research is limited to one method of our choice in each group. The introduction of Direct3D 10 opens up for potential optimizations, most importantly through the use of geometry shaders. We implement the algorithms with and without using new functionality provided in Direct3D 10 to answer the question if the performance can be improved with the use of new features. The results show that shadow mapping is superior except for scenes with very few polygons. Variables such as the light incidence angle or the sizes of the objects only have a minor effect on the performance. The results also show that using geometry shaders will for both algorithms considerably reduce performance.

Keywords: *real-time rendering, shadow algorithms, performance comparison, shadow mapping, shadow volume*

Sammanfattning

Skuggor är centrala för hur människan uppfattar världen. Inom datorspel och andra interaktiva 3D-applikationer är det viktigt att underlätta förståelsen av scenen och det finns även ofta en stark vilja att skapa realistiska miljöer. Detta samt att skuggsättning är en komplicerad och prestandakrävande operation gör det till ett viktigt område inom realtidsgrafik. Vi undersöker i denna avhandling relationen mellan prestandan (renderingstiden) för de två populära metoderna för skuggsättning inom realtidsgrafik, shadow mapping och shadow volumes. Undersökningen avgränsas till två utvalda varianter av de ursprungliga algoritmerna. I och med lanseringen av den nya versionen av Direct3D, version 10, öppnas möjligheten för optimeringar av algoritmerna genom användandet av framförallt geometry shaders. På grund av detta implementerar vi algoritmerna både med och utan den nya funktionalitet som introduceras i Direct3D 10 för att besvara om det är möjligt att förbättra prestandan för algoritmerna i Direct3D 10. Resultatet visar tydligt shadow mapping-algorithmens överlägsenhet för annat än scener med mycket få polygoner. Variabler så som ljusets infallsvinkel och objekts storlek har liten inverkan på det slutgiltiga resultatet för när vilken algoritm har bättre prestanda. Resultatet visar även tydligt att användandet av geometry shaders ger en mycket kraftig försämring av prestandan för båda algoritmerna.

Nyckelord: *realtidsrendering, skuggor, algoritmer, jämförelse, prestanda, shadow mapping, shadow volumes*

Innehållsförteckning

| | |
|---|----|
| Tack till | 1 |
| Inledning | 2 |
| Syfte | 3 |
| Metod..... | 4 |
| Avgränsning | 4 |
| Teoretisk bakgrund | 5 |
| Skuggor i realtidsgrafik | 5 |
| Shadow volumes..... | 6 |
| Shadow mapping | 9 |
| Parallell Split Shadow Maps..... | 12 |
| Relaterade arbeten | 13 |
| Testet..... | 13 |
| Undersökta variabler | 13 |
| Antal polygoner i scenen | 14 |
| Renderingsytans upplösning..... | 14 |
| Ljusets infallsvinkel..... | 14 |
| Objektens storlek i scenen | 14 |
| Shadow map-upplösning | 14 |
| Antal frustumdelar för shadow mapping-algoritmen | 14 |
| Utformning | 15 |
| Testspekifikation | 15 |
| Implementation..... | 16 |
| Testapplikationen..... | 16 |
| Shadow volumes..... | 16 |
| Shadow mapping | 17 |
| Resultat | 17 |
| Jämförelse mellan volumes 9 och volumes 10..... | 18 |
| Test 1.1 | 18 |
| Test 1.2 | 20 |
| Jämförelse mellan maps 9 och maps 10 | 22 |
| Test 2.1 | 22 |
| Test 2.2 | 23 |
| Test 2.3 | 23 |
| Jämförelse mellan shadow mapping och shadow volumes | 25 |
| Skuggkvalitet för shadow mapping | 25 |
| Jämförelse mellan maps 9 och volumes 9 | 27 |
| Diskussion..... | 30 |
| Slutsats | 31 |
| Referenser | 32 |

Tack till

Ett stort tack till vår handledare Bo Helgeson för hans goda råd, entusiasm och positiva inställning till arbetet.

Vi vill också tacka Oskari Nyman för värdefulla samtal kring skuggor och Direct3D 10 samt för att ha låtit oss ta del av hans shadow mapping-implementation.

Ett tack går även till Stefan Petersson för att ha delat med sig av sina lärdomar från sitt kandidatarbete och för hans tips på vad som var viktigt att tänka på för vårt arbete.

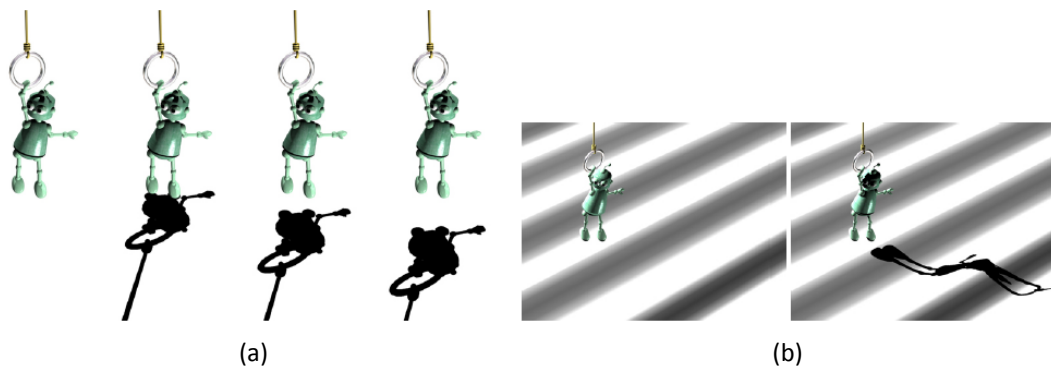
Tillslut vill vi även tacka Blekinge Tekniska Högskola som bistod med teknisk utrustning för testerna.

Inledning

Skuggor är centrala för hur människan uppfattar världen. Vid representation av tredimensionella scener på tvådimensionella ytor försvinner information som ges av stereoseende. Intrycket av tredimensionalitet i tvådimensionella bilder har länge varit förstått av konstnärer. Da Vinci är först med att göra en genomgående analys av skuggor och beskrev hur de kan användas för att förtydliga tredimensionaliteten i statiska bilder [Da Vinci 1770].

Lambert [Lambert 1759] och nu senare bland annat Knill et al. [Knill et al. 1997] beskriver matematiken bakom skuggor.

Skuggor hjälper förståelsen av en scen genom att tillföra information (se figur 1). Specifikt om positioneringen och storleken [Wanger 1992; Mamassian et al. 1998; Hubona et al. 1992; Kersten et al. 1994; Kersten et al. 1997] samt geometrin [Mamassian et al. 1998] för det skuggkastande objektet, och även information om geometrin för ytan som skuggan faller på [Mamassian et al. 1998].



Figur 1. Skuggor är viktiga för förståelsen av bilder. Utifrån den första bilden i figur 3(a) kan vi inte säga något om objektets position relativt till golvplanet. I de tre följande bilderna kan vi utläsa objektets positionering med hjälp av skuggan. I figur 1(b) hjälper skuggan oss att förstå geometrin för ytan som skuggan faller på. Bilden är hämtad från Hasenfratz et al.

Hubona et al. undersöker hur skuggor hjälper försökspersoner att positionera och skala om objekt så att de passar in i en geometrisk figur och har samma storlek som andra objekt i scenen [Hubona et al. 1999]. De undersöker hur resultatet förändras beroende på om scenen presenteras i stereo eller mono, om skuggor visas, antal ljus och bakgrundens geometri.

[Wanger et al. 1992] undersöker hur olika visuella ledtrådar, bland annat skuggning, hjälper försökspersoner att manipulera ett objekt så att det matchar ett annat objekts position, rotation och storlek. Wanger [Wanger 1992] undersöker i en annan studie hur den spatiala relationen mellan objekt uppfattas beroende på skuggans skärpa.

Experimenten pekar på vikten av skuggor i datorgenererade bilder för att underlätta förståelsen av scenen. Utöver detta ökar skuggor även realismen för renderade bilder.

Inom datorspel och andra interaktiva 3D-applikationer är det viktigt att underlätta förståelsen av scenen och det finns även ofta en stark vilja att skapa realistiska miljöer. Detta tillsammans med den snabba utvecklingen av hårdvara dedikerad till grafikrendering har lett till att utvecklingen av metoder för generering av skuggor i realtid har gått fort de senaste åren.

De huvudsakliga metoderna för skuggsättning i realtid är shadow mapping och shadow volumes. Det finns varianter av båda metoderna som genererar hårda skuggor och mer prestandakrävande varianter som genererar mjuka skuggor. Shadow volumes presenterades först av Crow [Crow 1977] och skapar utifrån scenens geometri volymer som representerar områden i skugga. Om en punkt är inuti en skuggvolym är den i skugga, i annat fall är den belyst. Shadow mapping presenterades först av Williams [Williams 1978]. Metoden skuggsätter scenen genom att spara en bild från ljusets perspektiv vilken senare kan användas för att bestämma om en punkt som ses av kameran även ses av ljuset, vilket betyder att den är belyst.

Shadow mapping är den populärare metoden av de båda. En viktig anledning till detta är att shadow mapping helt och hållet kan implementeras i hårdvara vilket gör den snabb och avlastar processorn. Detta är inte möjligt för den grundläggande shadow volumes-algoritmen. I och med introduktionen av Direct3D 10 och Shader Model 4 som innehåller geometry shaders kan detta dock komma att ändras. Geometry shaders gör det möjligt att implementera shadow volumes i hårdvara vilket har potential att minska dess renderingstid.

Syfte

Detta arbete ämnar undersöka relationen mellan prestandan, i form av renderingstid, för shadow mapping- och shadow volumes-metoden. Hur ny funktionalitet i Direct3D 10, framförallt geometry shaders, påverkar prestandan kommer även att undersökas. Prestandan kommer ställas i förhållande till hur många polygoner scenen består av och flera andra variabler. Ingen genomgående prestandajämförelse mellan de båda metoderna har tidigare presenterats. Arbetet syftar till att fylla detta tomrum och fungera som ett underlag vid val av skuggsättningsmetod.

Den huvudsakliga frågan är i vilka lägen shadow mapping-metoden är snabbare än shadow volumes-metoden och tvärt om. Innan denna besvaras kommer frågan om det är möjligt att förbättra prestandan för shadow mapping- eller shadow volumes-metoden med ny funktionalitet i Direct3D 10 att behandlas.

Metod

Spel och andra interaktiva 3D-applikationer har mycket varierande scener. För att jämförelsen ska vara värdefull krävs att den visar hur prestandan för de båda metoderna förändras beroende på scenens komposition.

För att erhålla den information som krävdes för att besvara frågeställningarna behövde en testapplikation utvecklas. Utan data från testerna kunde inte någon slutsats dras. Systemet är alltför komplext för att det ska vara möjligt att beräkna prestandan matematiskt. För att undvika att resultatet av testerna påverkades av andra faktorer än skugggenereringen behövde testapplikationen vara så enkel som möjligt.

Två versioner av båda algoritmerna implementerades således. Den första är implementerad så nära som möjligt hur algoritmen implementeras i Direct3D 9, och den andra använder sig fullt ut av ny funktionalitet i Direct3D 10. Genom att först jämföra dessa två versioner med varandra ämnas frågan om det är möjligt att förbättra prestandan för metoden besvaras. För att kunna jämföra shadow mapping- och shadow volumes-algoritmerna med varandra behöver vi först bestämma vilken eller vilka konfigurationer av shadow mapping-algoritmen som genererar ett likvärdigt resultat som shadow volumes-algoritmen. Vi bestämmer detta genom att göra en kvalitativ jämförelse av skuggkvaliteten för de olika konfigurationerna för att besvara frågan i vilka lägen som den ena metoden är bättre än den andra.

Avgränsning

Arbetet begränsas till att jämföra algoritmerna *Parallell Split Shadow Maps* [Zhang et al. 2006] och Everitts och Kilgards robusta shadow volumes-algoritm [Everitts & Kilgard 2002] implementerad helt i hårdvara enligt tekniken presenterad i Microsofts shadow volumes-exempel [Microsoft 2004]. Dessa två algoritmer har valts för att de är robusta, fungerar i alla situationer och används mycket i interaktiva 3D-applikationer, men även för att begränsa arbetets omfattning.

Alla tester körs på samma mjuk- och hårdvarukonfiguration under Direct3D 10. Genom att implementera båda versionerna av algoritmerna i Direct3D 10 och köra dem på en plattform utesluts flera faktorer som vi inte kan kontrollera vilka skulle kunna komma att påverkar resultatet. Exempelvis kan faktorer så som hur Direct3D 9 implementeras på ett Direct3D 10-kort och arkitekturella skillnader mellan grafikkort från olika tillverkare ignoreras. Detta förenklar analysen av testresultaten och begränsar arbetets omfattning avsevärt.

Arbetet tar inte upp mjuka skuggor. Vi anser att en jämförelse mellan shadow mapping- och shadow volumes-metoderna för hårda skuggor är av större intresse då de kan anses vara mer generella och därigenom kan resultatet stå till grund för vidare studier även för mjuka skuggor.

Undersökningen behandlar endast scener med ett ljus. Att applicera metoderna på scener med flera ljus är triviale och tillför inte något av intresse.

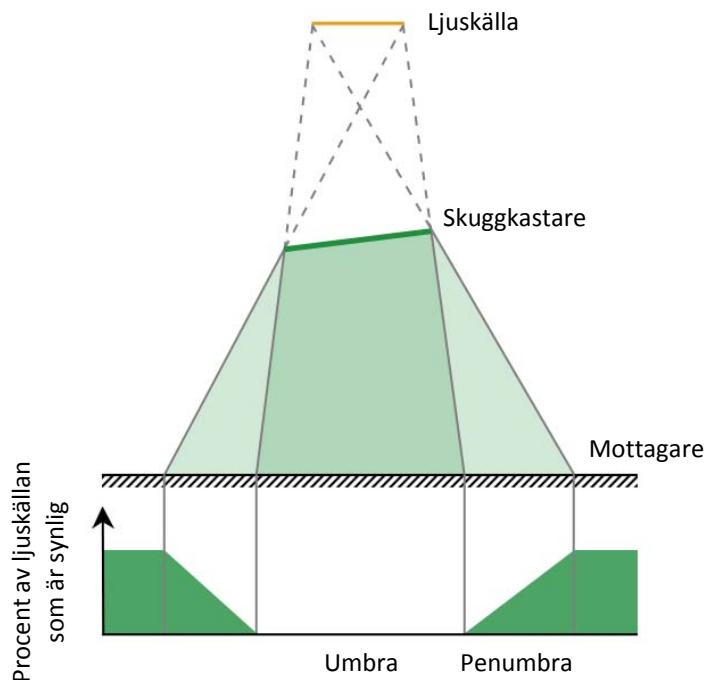
Rapporten förutsätter att läsaren har grundläggande kunskaper inom 3D-programmering. För ett standardverk inom realtidsrendering hänvisar vi till Real-Time Rendering [Haines & Möller 2002].

Teoretisk bakgrund

Skuggor i realtidsgrafik

Alla objekt i en scen som är belysta av en ljuskälla kallas mottagare. Umbra (latin för skugga) kallas områden i scenen som inte nås direkt av ljus från specifik ljuskälla. Området kan dock vara belyst av andra ljuskällor. Ett objekt som kastar en skugga kallas för en skuggkastare. En skuggkastare kan även vara en mottagare. Detta fall kallas för självskuggning.

En ljuskälla kan antingen vara en punkt, area eller volym. Om en ljuskälla är en area eller volym kommer delar av scenen att vara delvis belysta, alltså en viss del av ljuset når direkt till området. Områden i scenen som är delvis belysta kallas penumbra. Umbra och penumbra utgör tillsammans skuggan i scenen (se figur 2). I verkligheten är de flesta skuggor mjuka då få perfekta punktljus existerar.



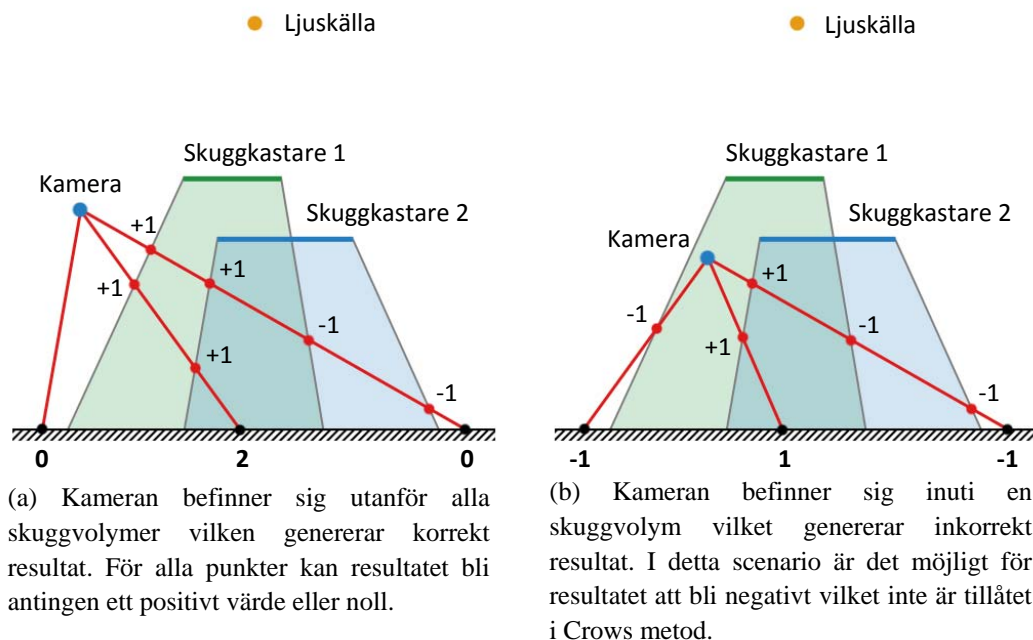
Figur 2. Ljuskällan är i figuren en yta vilket gör att penumbra-regioner uppstår. Bilden är hämtad från Hasenfratz et al. och är modifierad.

Shadow volumes

Shadow volumes presenterades först av Crow [Crow 1977]. Utifrån scenens geometri skapas volymer som representerar områden i skugga. Volymerna byggs genom att algoritmen först identifierar siluetterna för objekten i scenen sett från ljusets position. Därefter dras varje punkt i siluetten ut längst med vektorn mellan ljuset och punkten mot oändligheten. Den nya geometrin skapar en volym. Alla punkter i volymen är i skugga och alla punkter utanför volymen är belysta.

Vid rendering av bilden undersöks för varje pixel om punkten är i skugga eller inte. Detta görs genom att för varje framåtvänd polygon i volymen ökas räknaren, och för varje bortåtvänd polygon i volymen minskas räknaren. Om räknaren har ett värde större än noll är punkten i skugga, om värdet är noll är punkten belyst (se figur 3a). Värdet som är mindre än noll är inte hanterade. Metoden förutsätter att kameran är utanför alla skuggvolym i scenen för att undvika negativa värden och därigenom felaktigt resultat. Crow inser vikten av att fastställa om kameran befinner sig i en skuggvolym (se figur 3b).

Shadow volumes-metoden har fördelen att den skapar pixelperfekta skuggor. Detta beror på att den tar fram information om skuggor i object space, vilket betyder att informationen är tillgänglig för varje individuellt fragment i screen space. Den klarar även punktljus utan extra hantering.

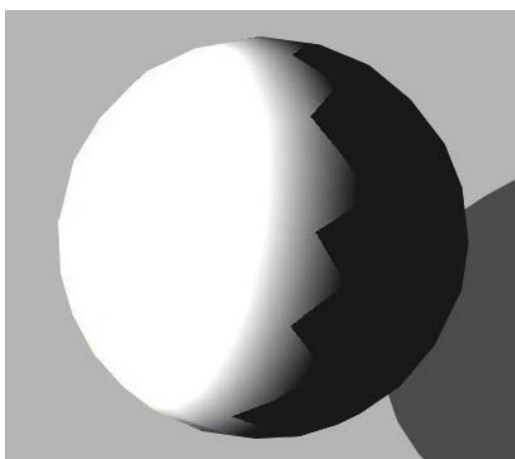


Figur 3. Crows metod producerar korrekt resultat så länge som kameran inte befinner sig inuti någon skuggvolym. Bilden är hämtad från Hasenfratz et al. och är modifierad.

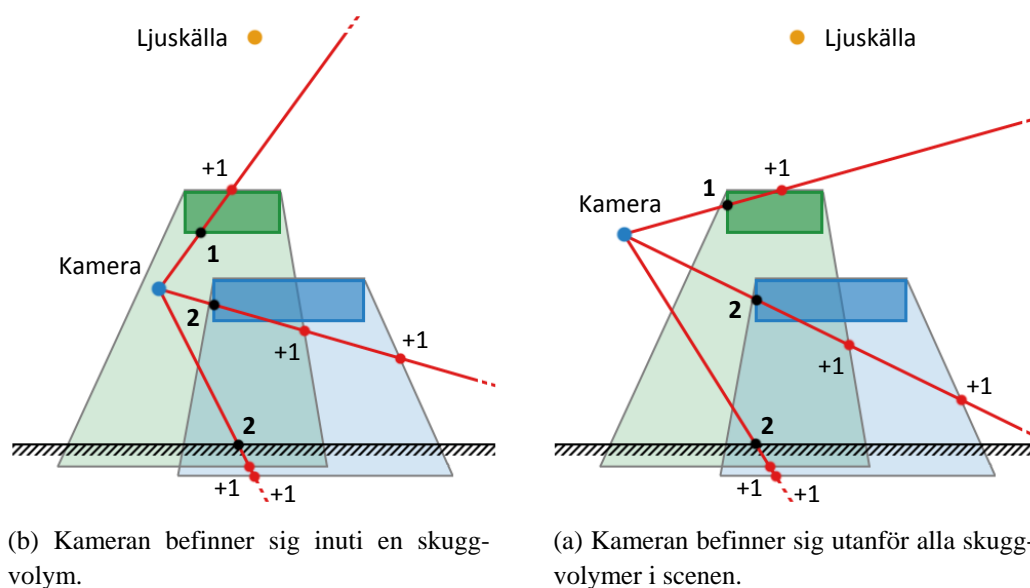
Att algoritmen jobbar i object space medför att dess renderingstid beror på antalet polygoner i scenen. Shadow volumes-metoden har två stora problem. Det största problemet är silhuett-detekteringen som är mycket beräkningsintensiv. Det andra problemet är att stora mängder fillrate konsumeras när skuggvolymerna renderas.

Ytterligare ett problem är att algoritmen utan modifiering av scenens geometri [Microsoft 2004] eller användande av texturer för att lagra data [Brabec & Seidel 2003] endast går att implementera helt och hållet i hårdvara som har stöd för geometry shaders. Skuggorna som algoritmen producerar har hårda kanter. Artefakter träder ibland fram vid silhuetten (se figur 4). Detta beror på att ljusberäkningar använder interpolerade vertexnormaler eller en normal map för att bestämma ljusstyrkan i varje punk. Shadow volumes-metoden däremot använder polygonens normal vilket resulterar i att en polygon är antingen helt i skugga eller helt belyst vilket kan ge felaktigt resultat.

Heidmann beskriver hur stencil-bufferten kan användas för shadow volumes [Heidmann 1991]. Heidmanns lösning är en flerstegsalgoritm. Det första steget är att tömma color- depth- och stencil-buffertarna. Därefter renderas scenen med ambient- och emissive-ljus. I det tredje steget renderas skuggvolymerna till stencil-bufferten. Stencil-bufferten ökas när djuptestet lyckas för framåtvända polygoner och minskas för bakåtvända polygoner. Efter det tredje steget innehåller stencil-bufferten för varje fragment ett positivt värde om punkten är i skugga och noll om punkten är belyst. Som tidigare beskrevs kommer negativa värden endast att uppkomma om kameran befinner sig inuti en skuggvolym. I detta fall producerar algoritmen felaktigt resultat. Denna variant av shadow volumes-algoritmen kallas för depth-pass, eller Z-pass.



Figur 4. Artefakter uppstår vid silhuetten på grund av att ljussättningen och skuggsättningen beräknas på olika sätt. Bilden är hämtad från Microsoft och är modifierad.



Figur 6. Z-fail producerar korrekt resultat både när kameran befinner sig i en skuggvolym och utanför, förutsatt att skuggvolymen är tillsluten i båda ändarna.

Ett ytterligare problem uppstår när far clip plane klipper skuggvolymen. Everitt och Kilgard [Everitt & Kilgard 2003] löser båda dessa problem genom att sätta far clip plane vid oändligheten och dra ut skuggvolymerna till oändligheten. På ny hårdvara är det möjligt att stänga av depth culling vilket gör det onödigt att sätta far clip plane vid oändligheten. I motsats till Z-pass klarar Z-fail situationer där near clip plane klipper skuggvolymen.

Brabec och Seidel [Brabec & Seidel 2003] presenterar en metod för att snabba upp shadow volumes-algoritmen genom att flytta över silhuett-detektionen från CPU:n till vertex shadern.

Lloyd et al. [Lloyd et al. 2004] visar hur fillrate-konsumtionen kan minskas genom att ta bort onödiga skuggkastande objekt och klippa bort onödig geometri i skuggvolymerna. Tekniken ger stora minskningar av fillrate-konsumtionen i scener där skuggvolymerna sträcks ut över stora och tomma områden.

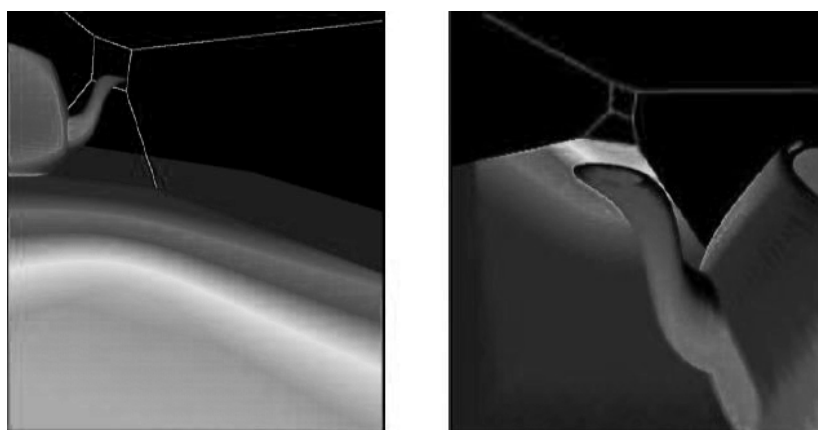
Shadow mapping

Shadow mapping presenterades först av Williams [Williams 1978]. Metoden utnyttjar faktumet att allt som är synligt från en kamera placerad vid ljuset är belyst och allt annat ligger i skugga.

Algoritmen är uppdelad i två steg. Scenen beräknas först från ljusets position och vy med Z-buffertest aktiverat. Djupet för varje pixel som klarar testet skrivs till en buffert. Resultatet efter första steget är en så kallad shadow map, som kan ses som en textur, vilken innehåller djupet till det närmsta fragmentet för varje texel. I det andra steget renderas scenen från kameran. För varje pixel som

beräknas transformeras dess position till ljusets koordinatsystem. Det transformerade djupet (Z-värdet) jämförs med respektive sparade värde i shadow mapen. Ett mindre värde i shadow mapen innebär att det finns en pixel som är mellan den aktuella pixeln och ljuset. Lika värde betyder att den aktuella pixeln är närmst. Om pixeln är närmst är den synlig för ljuskällan och därmed belyst, annars är den i skugga.

En viktig fördel med shadow mapping är att den helt och hållet går att implementera i hårdvara. Själva algoritmen är även oberoende av scenens vertexantal vilket gör den lämplig för komplexa scener. Utöver detta är den grundläggande algoritmen, kallad uniform- eller standard shadow mapping, lätt att implementera. Shadow mapping-metoden har ett inbyggt kvalitetsproblem i form av ojämna kanter - så kallad aliasing. Detta beror på att algoritmen arbetar i image space - detaljrikedom begränsas naturligt av den genererade shadow mapens upplösning. De två typerna av aliasing som uppstår är perspective samt projection aliasing och beror båda på att fler än en av skärmens pixlar är mappad till samma shadow map-textel. I scener där kameran och ljuset är riktade mot varandra försvåras problemet då högst detaljrikedom krävs nära kameran, men shadow mapen är som mest detaljerad nära ljuset. Situationen kallas dueling frusta (se figur 7). Aliasing-problemet är generellt störst i djupa scener där shadow mapen måste täcka ett stort område. Även om den uppenbara lösningen på att reducera algoritmens aliasing-problem är att öka upplösningen på shadow mapen så är detta tillvägagångssätt i många fall ej praktiskt genomförbart då större scener skulle kräva alltför stora texturer. Många vidareutvecklingar av den grundläggande algoritmen har uppkommit för att adressera dessa kvalitetsproblem och de flesta av dessa varianter är baserade på antingen warping eller uppdelning av frustumet.



Figur 7. Perspective aliasing och dueling frusta. Det ljusa området nära kameran (vänster) är tilldelat en låg andel av shadow map-upplösningen (höger). Problemet med aliasing är som störst när kameran och ljuset är riktade i motsatt riktning. Bilden är hämtad från nVidia och är modifierad.

Shadow mapping är även mindre lämpligt för punktljus som sprider ljuset i alla riktningar. Dessa fall löses ofta genom att använda sex separata spotlight-ljus riktade åt var sitt håll längs axlarna X,Y respektive Z.

Warping innebär att man förändrar ljusets vymatris inför shadow map-genereringen för att öka antal shadow map-textlar där det behövs, det vill säga nära kameran. Perspective Shadow Maps (PSM) [Stamminger 2002] är ett sådant warping-schema där man först transformerar scenen enligt kamerans projektmatrix. Detta gör att objekt nära kameran är större och lika stora objekt långt borta mindre. Sedan genereras en vanlig shadow map på den redan transformerade scenen. Resultatet blir en högre shadow map-detaljitet för objekt nära kameran. Det finns flera variationer som använder sig av warping, däribland Light Space Perspective Shadow Maps (LiSPSM) [Wimmer et al. 2004] och Trapezoidal Shadow Maps [Martin & Tan, 2004]. På grund av det lilla extra arbetet att ändra matriser är dessa metoder i princip likvärdiga med uniform shadow mapping räknat i renderingstid. En implementation som använder flera shadow maps med dynamiskt varierade upplösningar beskrivs i [Tadamura et al. 1999]. Algoritmen kräver flera pass både vid skugggenerering och utritning samt har ej hårdvarustöd på enheter där texturer med bredd och höjd som är tvåpotenser krävs.

Adaptive Shadow Maps (ASM) [Fernando et al. 2001] använder en hierarkisk struktur av shadow maps. Genom att generera celler med högre upplösning när det krävs, det vill säga vid skuggornas kanter, kan den effektiva upplösningen bli tillräckligt hög för att mappa varje kamerapixel till en enda shadow map-textel. Lefohn implementerar en väsentlig del av ASM-algoritmen på GPU:n men prestandan är fortfarande för låg för att användas i realtidsapplikationer [Lefohn et al. 2005].

Cascaded Shadow Maps (CSM) [Carmack 2004] och Parallell Split Shadow Maps (PSSM) [Zhang et al. 2006] bygger på att dela upp kamerafrustumet i flera delar parallellt med kamerans XY-plan där varje frustumdel tilldelas en egen shadow map.

Förutom ovan nämnda metoder så kan kvaliteten på skuggorna förbättras genom att begränsa ljusets frustum till det som är relevant för den renderade bilden (objekten i kamerans synfält samt eventuella skuggkastare utanför bild) för att maximera den utnyttjade shadow map-upplösningen [Brabec et al. 2002]. En sådan geometriapproximation konstrueras förslagsvis med bounding-volymer. Kvalitetsförbättringen beror på objektens placering och deras tänkta bounding-volym; en mindre bounding-volym ger högre kvalitet och en som redan överskrider ljusets vy gör ingen skillnad. Således blir det alltid en förbättring eller ingen skillnad i bildkvaliteten, aldrig en försämring.

Reeves [Reeves et al. 1987] visar en filtreringsmetod kallad Percentage Closer Filtering (PCF) för att minska aliasing. Metoden jämför varje pixel som ska renderas till skärmen med flera kringliggande värden i shadow mapen istället för ett enda värde som görs annars. PCF-filtrering är inbyggt i hårdvaran på nVidia-grafikkort och beräknas då utan förlust av prestanda.

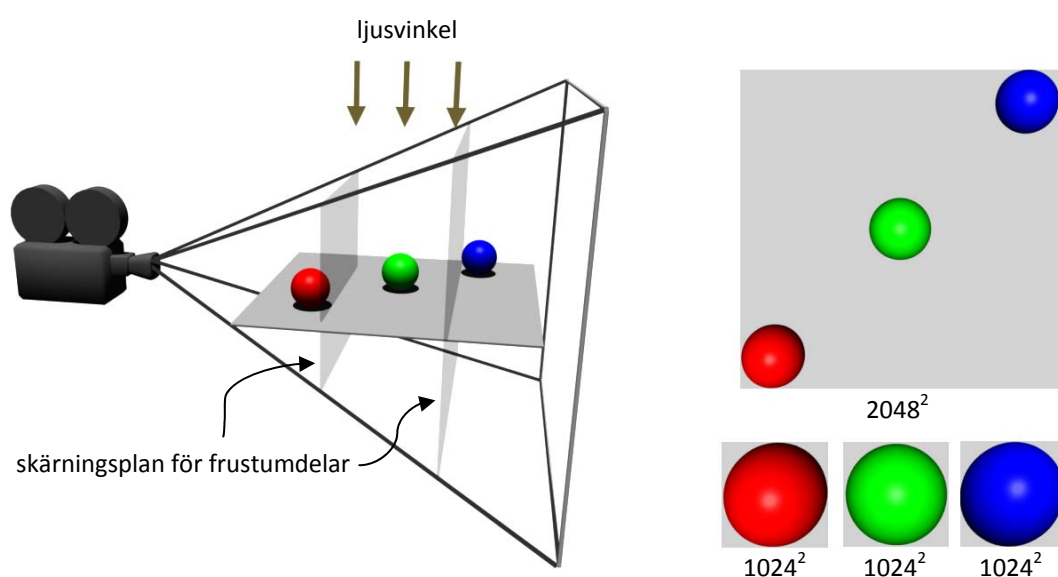
Parallell Split Shadow Maps

PSSM delar upp vyfrustumet med hjälp av flera plan parallella med kamerans XY-plan, ortogonala mot vyaxeln Z. Zhang (op cit) noterar flera påtagliga fördelar med tillvägagångssättet;

Då varje del är självständig i form av egen shadow map kan olika warping-scheman appliceras individuellt.

Upplösningen på en specifik shadow map behöver endast vara tillräcklig för objekten i den aktuella delen istället för varje objekt i scenen som är fallet med en enda shadow map.

Då geometrin i varje frustumdel kan approximeras separat finns möjlighet att sätta bounding-volymerna stramare (se figur 8). Tom rymd mellan objekt i olika delfrustum kan exkluderas.



Figur 8. Fokusering med hjälp av geometriapproximation ger högre effektiv upplösning på shadow mapen. Vid uppdelning av frustumet kan varje del fokuseras individuellt. Till höger visas shadow mapen för standard shadow mapping (ovan) samt PSSM med tre frustumdelar (nedan).

Relaterade arbeten

Även om utvecklingen för rendering av skuggor i realtid har gått snabbt de senaste åren har ingen studie presenterat en genomgående jämförelse av prestandan för olika algoritmer. Woo et al. [Woo et al. 1990] presenterar en sammanställning av de flesta tillgängliga skuggalgoritmer och tar upp deras för- och nackdelar men en jämförelse av renderingstider för de olika algoritmerna är utelämnad. Hasenfratz et al. [Hasenfratz et al. 2003] utökar Woos sammanställning med nya algoritmer för generering mjuka skuggor. Hasenfratz et al. nämner ungefärlig prestanda i bilder per sekund för de olika algoritmerna, men gör aldrig någon utförlig jämförelse.

När nya varianter av shadow mapping eller shadow volumes-metoderna presenteras så ingår alltid någon slags prestandamätning. Men prestandan mäts nästan aldrig mot några andra algoritmer än en eller två bra varianter av samma metod.

Testet

Målet med testerna är att undersöka tre separata frågor. Den första frågan handlar om hur prestandan för shadow volumes-algoritmen påverkas när den drar nytta av ny funktionalitet i Direct3D 10, framförallt geometry shaders. Således implementeras algoritmen i två versioner där den första, volumes 9, är implementerad så nära som möjligt hur den implementeras i Direct3D 9. Den andra versionen, volumes 10, utnyttjar alla nya möjligheter som erbjuds i Direct3D 10.

Den andra frågan behandlar hur shadow mapping-algoritmens prestanda påverkas när den utnyttjar ny funktionalitet i Direct3D 10. Även här implementeras algoritmen i två varianter. Den första, maps 9, är implementerad så nära som möjligt hur den implementeras i Direct3D 9. Den andra varianten, maps 10, är implementerad för att dra nytta av funktionalitet som erbjuds i Direct3D 10.

Den tredje och för arbetet centrala frågan gäller hur relationen mellan prestandan för shadow mapping och shadow volumes-algoritmerna ser ut. För att besvara denna fråga valdes den shadow volumes-implementation, volumes 9 eller 10, som presterade bäst och jämförs mot den shadow mapping-implementation, maps 9 eller 10, som presterade bäst.

Undersökta variabler

Testet undersökte följande variablers inverkan på prestandan.

Antal polygoner i scenen

Alla polygoner i scenen bearbetas av geometry shadern och alla vertexar i scenen bearbetas av vertex shadern. Eftersom algoritmerna utnyttjar antingen både vertex och geometry shader-steget eller endast vertex shader-steget är det intressant att mäta dessa shaders inverkan på prestandan. Om dessa shaders är beräkningsintensiva kommer en stark korrelation att uppträda mellan antal polygoner i scenen och prestandan. Det är dock viktigt att notera att en viss korrelation mellan antal polygoner i scenen och renderingstid kommer att uppstå eftersom geometrin måste hanteras på något sätt.

Renderingsytans upplösning

Renderingsytans upplösning är proportionell mot hur många fragment som skapas och därigenom hur många fragment som pixel shadern ska bearbeta. Om pixel shadern är beräkningsintensiv kommer en korrelation uppträda mellan renderingsytans upplösning och prestanda.

Ljusets infallsvinkel

Shadow volumes-algorithmens prestanda beror på hur stor del av renderingsytan som skuggvolymerna täcker. När ljusets infallsvinkel ökar blir skuggvolymerna längre. Dess inverkan på prestandan i relation till de andra variablerna är av intresse. Den resulterande skuggan genererad av shadow mapping-algoritmen utvecklar mer aliasing desto större ljusets infallsvinkel blir. Detta är viktigt att ta hänsyn till när skuggornas kvalitet jämförs.

Objektens storlek i scenen

När objekten i scenen ökar i storlek ökar även storleken på de skuggvolymerna som shadow volumes-algoritmen genererar. Eftersom shadow volumes-algorithmens prestanda beror på skuggvolymernas synliga yta är det intressant att mäta hur objektens storlek inverkar på dess prestanda.

Shadow map-upplösning

Upplösningen på de texturer som shadow mapping-algoritmen renderar skugginformation till inverkar på prestandan. Upplösningen är även viktigt för kvaliteten på den resulterande skuggan. Detta gör upplösningen intressant både för prestanda- och kvalitetsjämförelse.

Antal frustumdelar för shadow mapping-algoritmen

Den valda shadow mapping-algoritmen, Parallell Split Shadow Maps, delar upp scenen i ett antal delar och skapar en shadow map för varje del. Detta ökar kvaliteten för den resulterande skuggan, men kräver mer beräkningskraft för att dels dela upp världen och dels generera ytterligare shadow maps.

Utformning

Tre grupper av tester kördes. Den första gruppen behandlade skillnaden i prestanda mellan volumes 9 och volumes 10. Den andra gruppen behandlade skillnaden i prestanda mellan maps 9 och maps 10.

Den sista gruppen av tester jämförde prestandan mellan shadow mapping och shadow volumes. Då kvaliteten på den resulterande skuggan för shadow mapping-algoritmen skiljer sig mycket åt beroende på vilka inställningar som väljs för algoritmen är det intressant att se renderingstiden i förhållande till det visuella resultatet. På grund av detta genererades bilder av skuggorna som shadow mapping-algoritmen producerade vid alla inställningar.

Testspecifikation

Varje testkörning startades från ett av oss implementerat externt program i vilket vi kunde ställa in samtliga ovan nämnda variabler. Med hjälp av detta program kunde vi enkelt via en loop iterera över exempelvis objektantal och köra ett test för varje värde.

Varje sekund sparades värden för den aktuella sekunden till en lista och innan programmet avslutades så skrevs värdena till disk tillsammans med testparametrarna. Mätvärdena inhämtades under tio sekunder efter fem sekunders stabiliseringstid. Följande värden sparas:

- Antal renderade bildrutor sedan information sparades senast
- Tiden det tog att rendera dessa bildrutor
- Antal renderade trianglar under denna tid

Tiden mättes med QueryPerformanceCounter som är den högupplösta klockan i Windows-system.

Testresultaten som redovisas för en körning är medeltiden för en renderad bildruta beräknad på hela körningen. Antal trianglar är beräknat på det faktiska antalet trianglar i mesharna som laddas in. I volumes 9 ökas antalet trianglar efter inläsning och i både maps 9 och 10 dupliceras objekt om de skär planen som delar upp vyfrustumet.

Följande basinställningar användes då ej annat värde ges är:

- Renderingsytans upplösning: 1024x768
- Ljusvinkel: 30 graders infallsvinkel
- Shadow map-storlek: 1024x1024
- Antal frustumdelar för shadow map: 3

Implementation

Testapplikationen

Vid körning renderar testapplikationen en scen bestående av ett kvadratisk basplan och ett antal modeller. Varje modell ges en slumpmässig position inom ett av sju kvadratiska områden. Sex av de kvadratiska områdena slumpas ut över planets yta. Det sjunde området slumpas ut ett kort avstånd framför kameran. Applikationen ser till att mittpunkten på alla områden slumpas ut inom kamerans synfält men objekt kan slumpas ut delvis eller helt utanför synfältet om området skär vyfrustumet. Polygonantalen som redovisas tar hänsyn till detta och endast objekt som är delvis eller helt inom vyfrustumet räknas med. Kameran ser drygt halva basplanet och är placerad längst med en av basplanets kanter på en höjd något högre än höjden för en modell. De kvadratiska områdena har en bredd motsvarande en tolfedel av basplanets bredd. Modellerna har oskalade en bredd motsvarande en sjundedel av de kvadratiska områdenas bredd.

Förutom skuggsättning beräknas ljussättning för scenen med gouraud shading. Ingen texturering används på objekten. Alla variabler som testerna ämnar undersöka, inklusive vilken modell som används, kontrolleras utifrån programmet.

Shadow volumes

Everitt och Kilgards [Everitt & Kilgard 2002] robusta shadow volumes-teknik används för skugggenerering. Volumes 9 bygger på den helt hårdvaruaccelererade algoritmen som beskrivs i "Shadow Volume Sample" [Microsoft 2004]. Algoritmen undviker att hantera silhuett-detektionen och skuggvolymkonstruktionen på CPU:n. Eftersom det inte är möjligt att skapa vertexar på GPU:n i Direct3D 9 måste de vertexar som behövs för skuggvolymerna föras över till GPU:n innan dessa två steg utförs. Algoritmen som används i Microsofts exempel löser detta genom att alla meshar modifieras så att de innehåller de extra vertexar och polygoner som behövs för att det ska vara möjligt att dra ut alla möjliga silhuetter. Två extra polygoner behöver föras in vid varje polygonkant.

Volumes 10 består av Everitt och Kilgards algoritm med skillnaden att silhuett-detektionen och skuggvolymkonstruktionen är implementerad i geometry shader-steget. Geometry shadern skapar nya vertexar och polygoner vid silhuetten och drar ut dem för att skapa skuggvolymen.

Vid båda varianterna dras volymerna endast ut till något under basplanet i scenen. Detta är en enkel optimering som kan appliceras i alla applikationer.

Shadow mapping

När scenen slutligen skall renderas med ljussättning behövs tillgång till samtliga shadow maps. På grund av detta behövs en lista med samma längd som antalet frustumdelar innehållande shadow maps. Minnesåtgången ökar alltså linjärt med antal delar i både maps 9 och 10.

I maps 9 itereras frustumdelarna igenom i tur och ordning och vilka objekt i den aktuella delen som syns från ljuset bestäms. De bestämda objekten renderas från ljusets position och resultatet sparas på respektive plats i listan. Scenens rendering till djupbufferten sker således i n pass där n är antalet delar som frustumet är uppdelat i.

Maps 10 använder instancing i geometry shader-steget för att rendera objekten till djupbuffertarna i ett och samma pass. Ett objekt som sträcker sig från frustumdel i till j instansieras ($j - i + 1$) gånger och renderas till shadow map i, \dots, j i listan. Detta innebär att endast ett pass behövs för att skriva hela scenen till djupbuffertarna oberoende av antal frustumdelar.

Båda algoritmerna ritar ut den ljussatta scenen i ett pass. Maps 9 kräver alltså totalt $n+1$ pass och maps 10 $1+1$ pass. Färre pass kommer här på bekostnad av det extra geometry shader-steget i shader-programmet för maps 10.

Shadow map-implementationerna i detta arbete använder ej warping, dels för att behovet minskar drastiskt när man använder multipla shadow maps och dels för att dessa warping-tekniker ofta har specialfall som måste hanteras separat vilket gör dem komplicerade att få robusta.

Resultat

Resultaten av testerna presenteras i detta kapitel. Modellerna som användes var robot, drone och flygplan och är uppbyggda av 9200, 1700 och 600 polygoner respektive. Testerna kördes på följande system:

Intel Core2duo e6600 (2,4 GHz)
2048 MB DDR2 RAM
nVidia 8800GTX (768MB)

Windows Vista Enterprise 64bit
DirectX 10 SDK (April 2007)
nVidia Forceware 163.75

Även om processorn är tvåkärnig är testapplikationen enkeltrådig och utnyttjar en processorkärna.

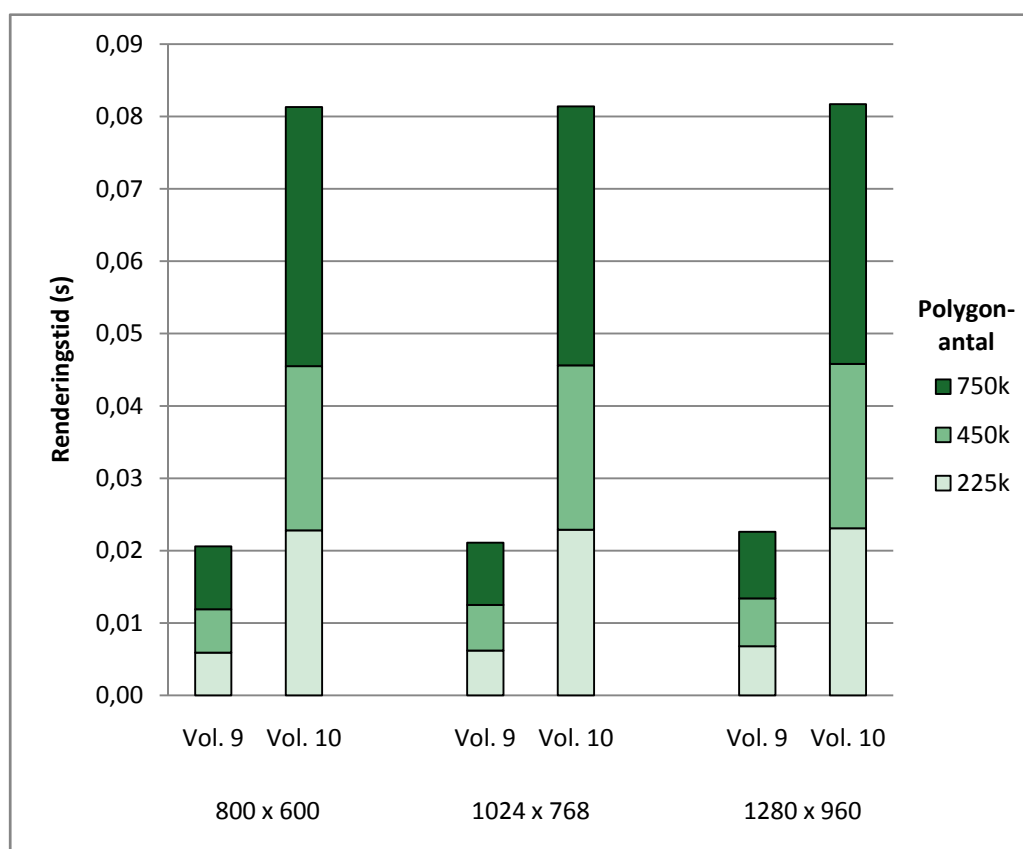
Jämförelse mellan volumes 9 och volumes 10

Följande tester undersöker prestandaskillnaden mellan volumes 9 och 10.

Test 1.1

Alla tre deltest i Test 1.1 undersöker prestandan då arbetet för pixel shadern intensifieras.

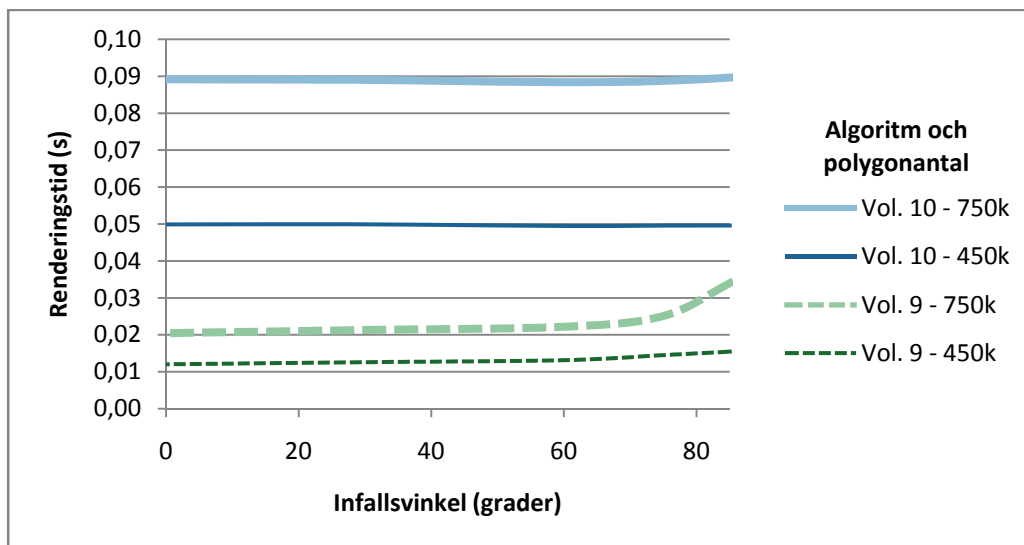
Det första deltestet undersöker hur renderingsytans upplösning påverkar prestandan (se figur 9). Tre olika upplösningar testas. Modellen som används är robot och för varje upplösning testas tre scener med olika polygonantal.



Figur 9. Resultat för test 1.1a

Resultatet visar att renderingstiden för volumes 10 är ungefär 3,5-4 gånger så lång som för volumes 9 i alla tre scener. Medan renderingstiden för volumes 9 ökar mellan 10 % och 15 % från lägsta till högsta upplösningen hamnar volumes 10 inom en procents marginal.

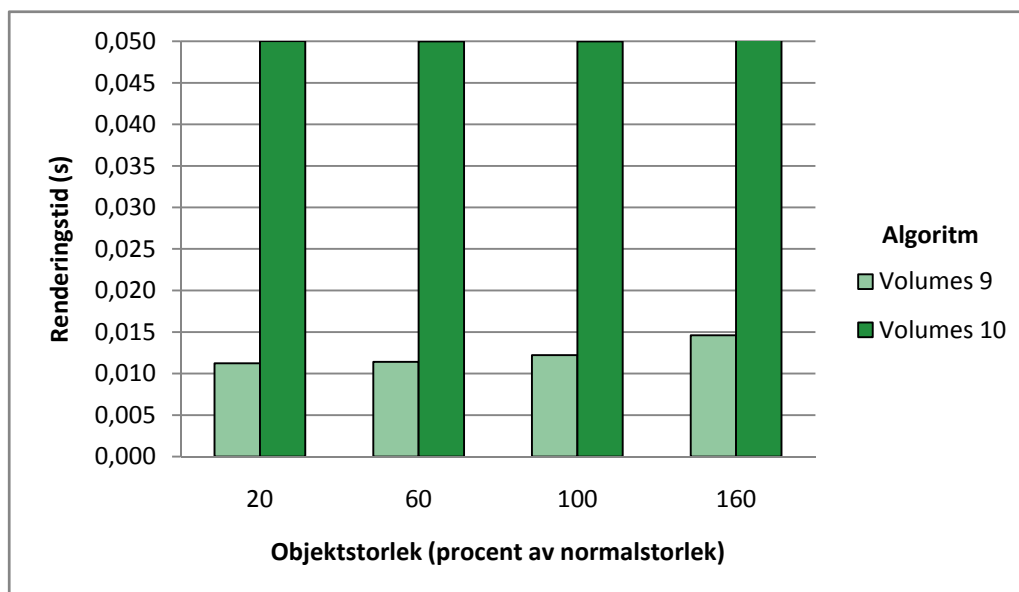
Det andra deltestet undersöker hur ljusets infallsvinkel påverkar prestandan (se figur 10). Fem olika infallsvinklar testas. Modellen som används är robot och det totala antalet polygoner i scenen är 450 000 samt 750 000.



Figur 10. Resultat för test 1.1b.

Ljuset kommer i testet rakt från sidan sett från kameran. En stor infallsvinkel förlänger skuggvolymerna avsevärt. För volumes 9 ges en 28 % längre renderingstid för 450 000 polygoner och 66 % längre renderingstid för 750 000 polygoner än motsvarande scen med 0 graders infallsvinkel. I volumes 10 uppmäts fortfarande ingen skillnad.

Det tredje deltestet undersöker hur objekts storlek påverkar prestandan (se figur 11). Fyra olika storlekar på objekten testas. Modellen som används är robot och det totala antalet polygoner i scenen är 450 000.

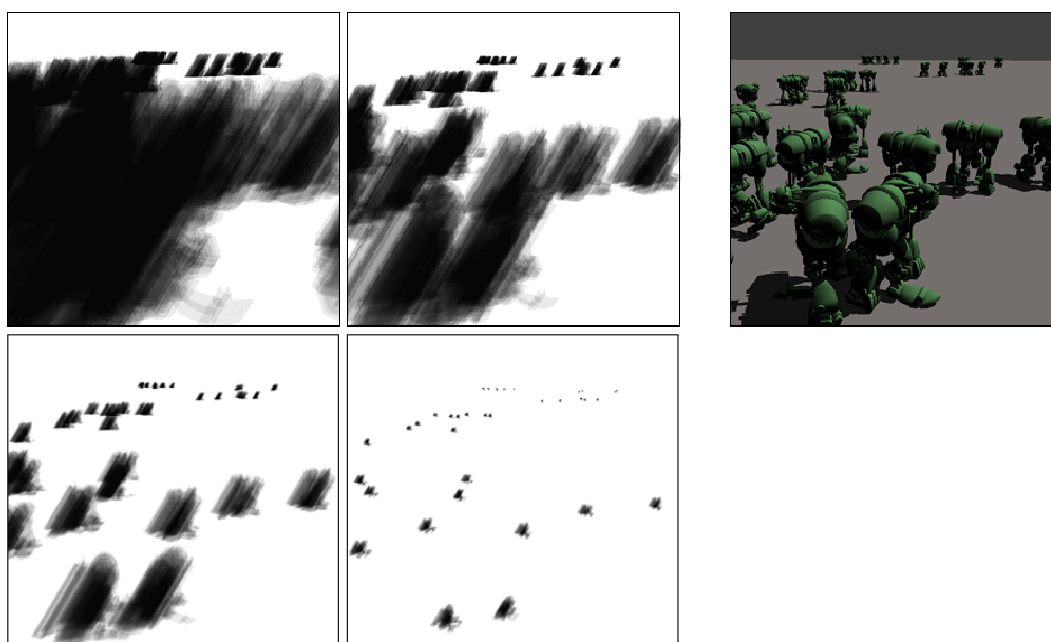


Figur 11. Resultat för test 1.1c.

Grafen visar återigen att volumes 10 är oberoende av fillrate-behovet och är begränsad av andra enheter än pixelprocessorerna. Volumes 9 påverkas ej vid uppförstoringen från 20 till 60 % av originalstorleken, men visar upp en försämring i renderingstid på runt 30 % vid ytterligare förstoring från 60 till 160 % av originalstorleken.

Tolkning av resultat

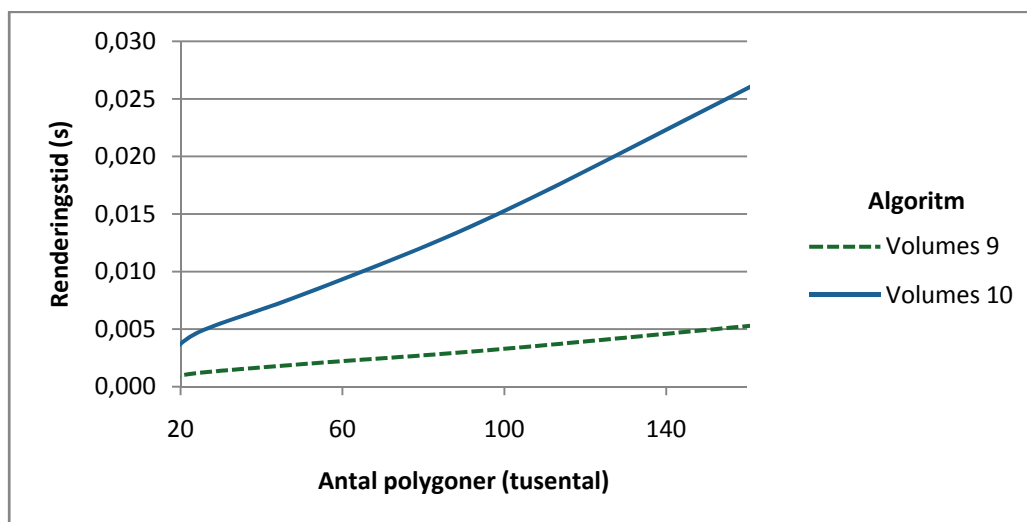
Volumes 10 som använder sig av geometry shaders berörs ej av något av de tre testen och är därmed ej begränsad av pixelenheternas prestanda. För objekt med nära 10 000 polygoner är volumes 9 nästintill oberoende av fillraten i scener då objektens skuggvolymer täcker upp en liten del av renderingsytan. Vid större ytor noteras pixelprocessorerna som flaskhalsar. Se figur 12 för visualiseringar från tredje deltestet.



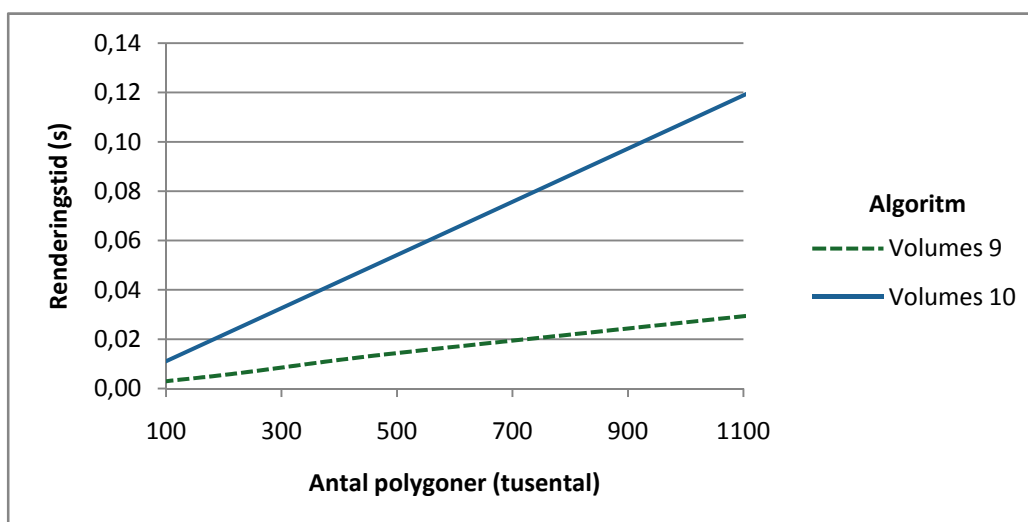
Figur 12. Skuggvolymernas komplexitet för de olika storlekarna är utritad. Desto mörkare en pixel är i bilderna, desto fler djuptester måste göras och desto fler kommer resultera i skrivning till stencil-bufferten. Till höger är en bild på den aktuella scenen.

Test 1.2

Det andra testet undersöker hur antal polygoner i scenen påverkar prestandan. Två körningar görs där den första mäter prestandan vid låga polygonantal (se figur 13) och den andra mäter prestandan vid medel till höga polygonantal (se figur 14). Vid båda körningarna testas fem olika polygonantal för scenen. Vid den första körningen används modellen drone och vid den andra modellen robot.



Figur 13. Resultat för test 1.2a.



Figur 14. Resultat för test 1.2b.

Grafernas linjäritet är förväntad. Fler objekt ger proportionellt fler vertexar att bearbeta för vertex shadern och fler trianglar för geometry shadern. Att grafen för volumes 9 och volumes 10 har olika lutning innebär att varje objekt i vårt fall ger en högre belastning för volumes 10 än för volumes 9. Kvoten mellan $\Delta t / \Delta(\text{obj})$ för volumes 10 och volumes 9 är i grafen med högre polygonantal cirka 4,0 och i grafen med lägre polygonantal är kvoten drygt 5,0. Varje extra objekt tar alltså 4-5 gånger så lång tid att processa för volumes 10 jämfört med volumes 9.

Tolkning av resultat

Den avsevärt sämre prestandan för volumes 10 kan inte annat än tillskrivas geometry shader-steget.

Jämförelse mellan maps 9 och maps 10

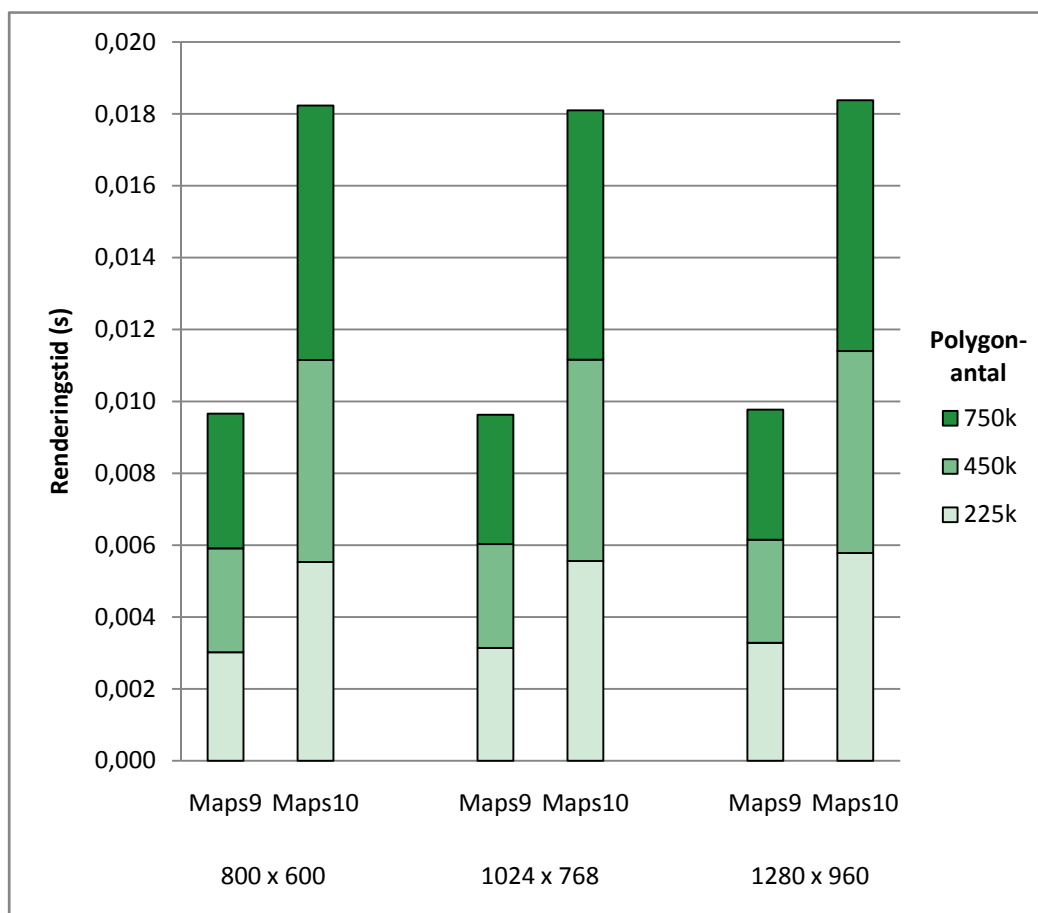
Test 2.1

Det första testet undersöker hur renderingsytans upplösning påverkar prestandan (se figur 15). Tre olika upplösningar testas. Modellen som används är robot och för varje upplösning testas tre scener med olika polygonantal.

Både maps 9 och maps 10 påverkas av större renderingsyta då fler jämförelser mot shadow mapen måste göras i utritningspasset. Den relativa skillnaden blir för båda mindre ju fler vertexar som måste bearbetas. Maps 9 förlorar mellan 1,1 % och 8,6 % i prestanda och maps 10 förlorar mellan 0,8 % och 4,5 %. De båda implementationernas absoluta tidsförluster är nästintill identiska.

Tolkning av resultat

Både maps 9 och maps 10 är något begränsade av pixel shadern i utritningspasset vilket ses av den ökade renderingstiden vid högre upplösning.



Figur 15. Resultat för test 2.1.

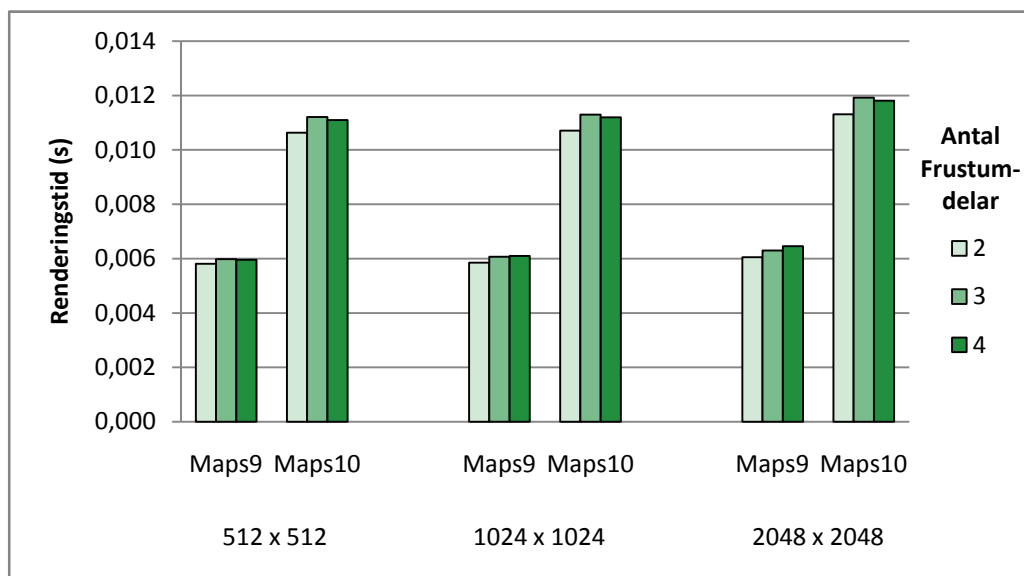
Test 2.2

Detta test undersöker hur renderingstiden påverkas av antal frustumdelar samt shadow mapens upplösning (se figur 16). Modellen är robot och polygonantalet är 450 000.

Testet visar att renderingstiden för shadow map-konfigurationerna ligger inom 12 % för maps 9. Exempelvis är shadow map-upplösningen 1024x1024 och 2 frustumdelar 10 % snabbare än 2048x2048 och 4 delar. Detta innebär att man i det förstnämnda fallet kan få 0,1s extra varje renderingssekund att lägga på andra beräkningar visavi det sista. Resultaten för maps 10 ligger för respektive upplösning inom en dryg procents marginal medan 2048x2048 är cirka 7 % långsammare än 512x512.

Tolkning av resultat

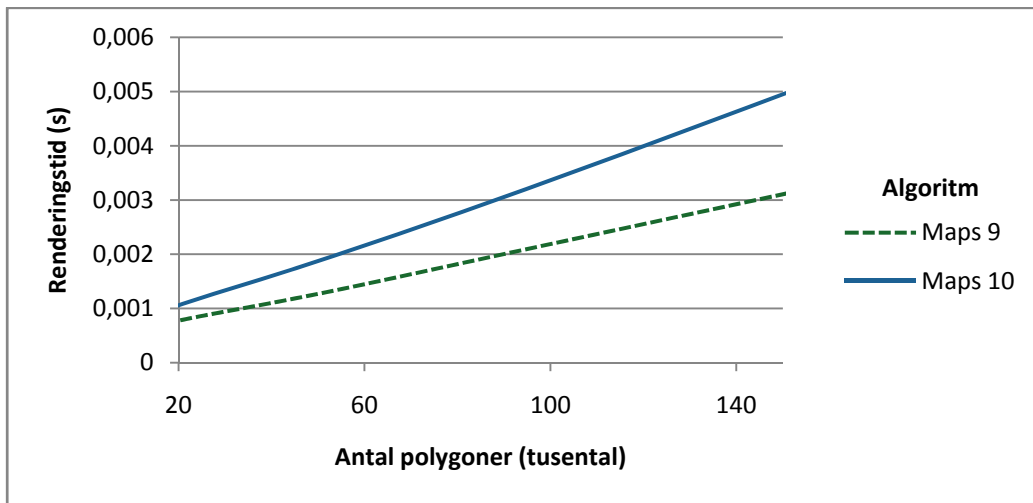
Testet visar tydligt att vertex shadern är vad som konsumerar den stora majoriteten av tillgänglig GPU-kraft. Det extra arbete som pixel shadern behöver utföra när upplösningen höjs är litet i jämförelse med arbetet som vertex shadern utför. Då en stor del av tiden används till vertexbearbetning skulle skillnaderna bli betydligt större i scener med färre polygoner.



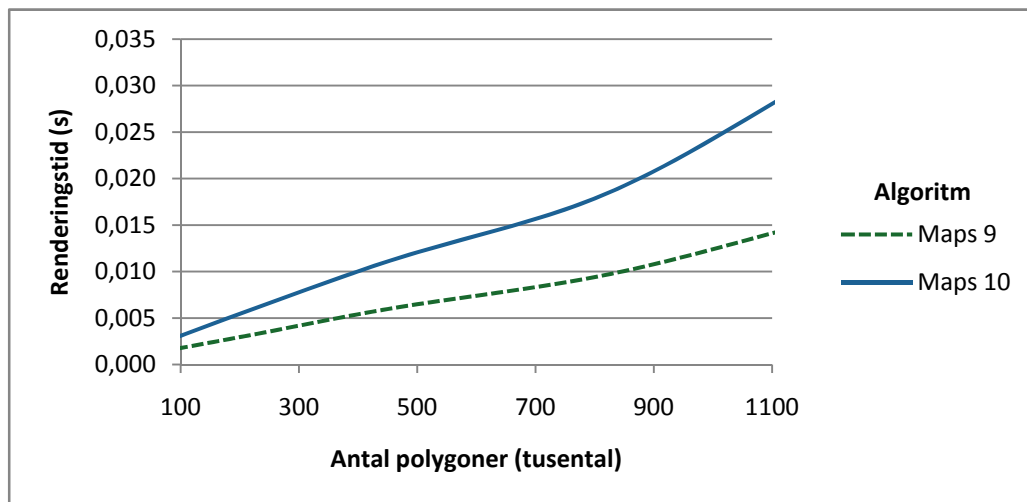
Figur 16. Resultat för test 2.2.

Test 2.3

Det tredje testet undersöker hur antal polygoner i scenen påverkar prestandan. Två körningar görs där den första mäter prestandan vid låga polygonantal (se figur 17) och den andra mäter prestandan vid medel till höga polygonantal (se figur 18). Vid båda körningarna testas fem olika polygonantal för scenen. Vid den första körningen används modellen drone och vid den andra modellen robot.



Figur 17. Resultat för test 2.3a.



Figur 18. Resultat för test 2.3b.

Maps 10 är i scenen med lågt polygonantal 60 % långsammare än maps 9. Motsvarande siffra i scenen med medel till högt polygonantal är cirka 80 %. Vi ser också att maps 9 använder 0,0030s för 140 000 polygoner i första scenen men endast 0,0080s, cirka 3 gånger längre tid, för en 5 gånger så stor mängd (700 000 polygoner) i den andra scenen.

Tolkning av resultat

Skillnaden mellan renderingstid för maps 9 och 10 i scenen med lågt polygonantal gentemot scenen med högt polygonantal kan förklaras med att maps 10 arbete per vertex är högre i och med att algoritmen utnyttjar geometry shadern. Detta leder till att vid scenen med högt polygonantal försämras prestandan för maps 10 mer än för maps 9 eftersom vertex shadern står för en större del av beräkningarna.

Jämförelse mellan shadow mapping och shadow volumes

Skuggkvalitet för shadow mapping

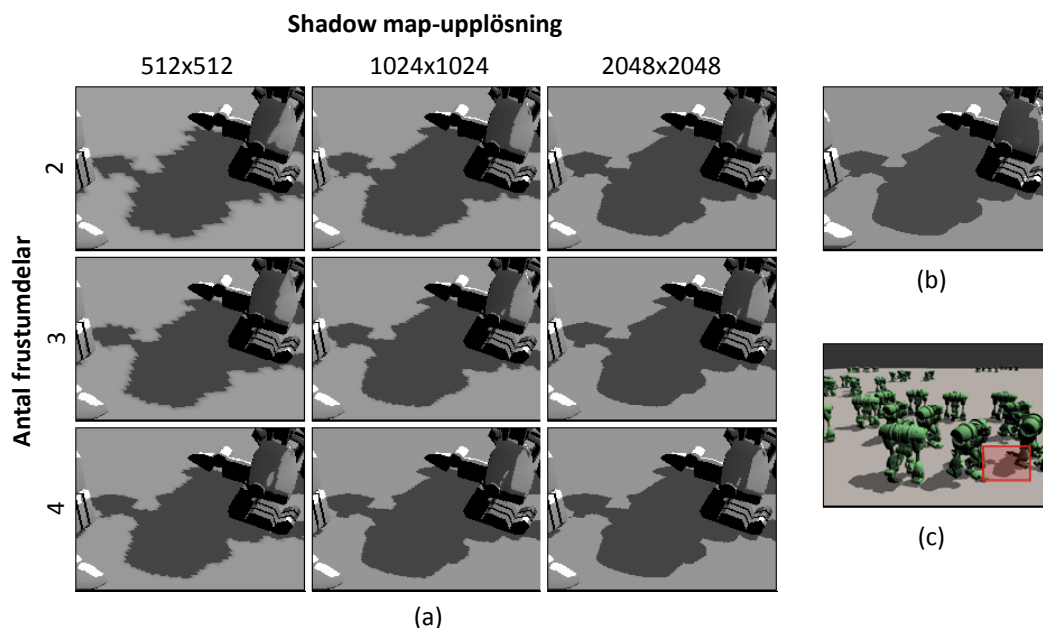
Följande två test ämnar illustrera skillnaden i kvalitet på skuggan genererad av shadow mapping-algoritmen i två olika situationer. Resultatet ligger även till grund för vilka shadow map-konfigurationer som inkluderas i den slutgiltiga jämförelsen mellan shadow mapping- och shadow volumes-algoritmerna.

Shadow mapping-konfigurationer anges i följande kapitel enligt formen (upplösning/antal frustumdelar).

Test 3.1

Det första testet syftar till att representera normalfallet i de flesta spel. Ljuset har infallsvinkeln 30 grader och scenen består av 50 objekt. Modellen som används är robot och scenen består av totalt 450 000 polygoner.

Se figur 19 för en visualisering de renderade skuggorna vid de olika konfigurationer. Alla tre inställningarna med 2048x2048 i shadow map-upplösning samt (1024/4) anser vi vara likvärdiga med shadow volumes för denna scen. Alla förutom (512/2), (512/3) genererar en av oss klart godkänd skugga.

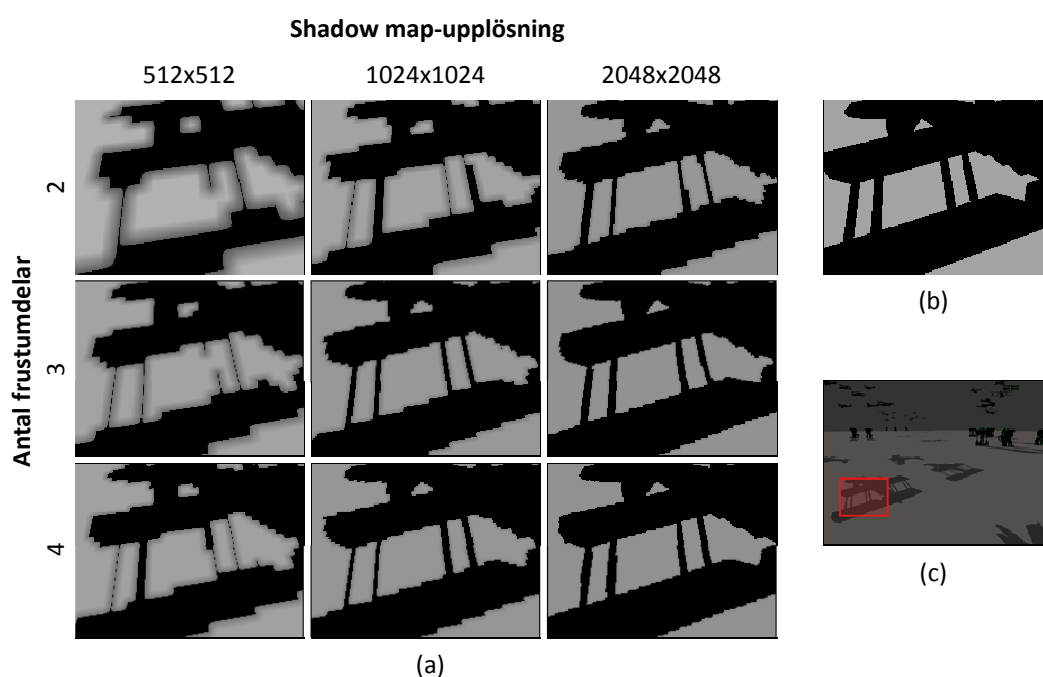


Figur 19. Nio olika kombinationer av inställningar för shadow map-upplösning och antal frustumdelar visas i figur 19(a). Figur 19(b) visar resultatet genererat av shadow volumes-algoritmen, vilket alltid är pixelperfekt. Bilden som används är endast en del av hela den renderade bilden. Figur 19(c) visar vilken del av den fullständiga bilden som används.

Test 3.2

Det andra testet syftar till att representera mer ovanliga situationer i spel där shadow mapping-algoritmen fungerar dåligt. Ljuset har infallsvinkeln 75 grader och scenen består av 65 objekt. Modellerna som används är robot och flygplan och scenen består av totalt 340 000 polygoner.

Se figur 20 för en visualisering de renderade skuggorna vid de olika konfigurationer. I denna scen har shadow maps stora problem med aliasing. Inga inställningar producerar lika bra resultat som shadow volumes, men (2048/3), (2048/4) kan anses vara fullgoda. Även (2048/2), (1024/4) och (1024/3) är klart acceptabla (se figur 20).



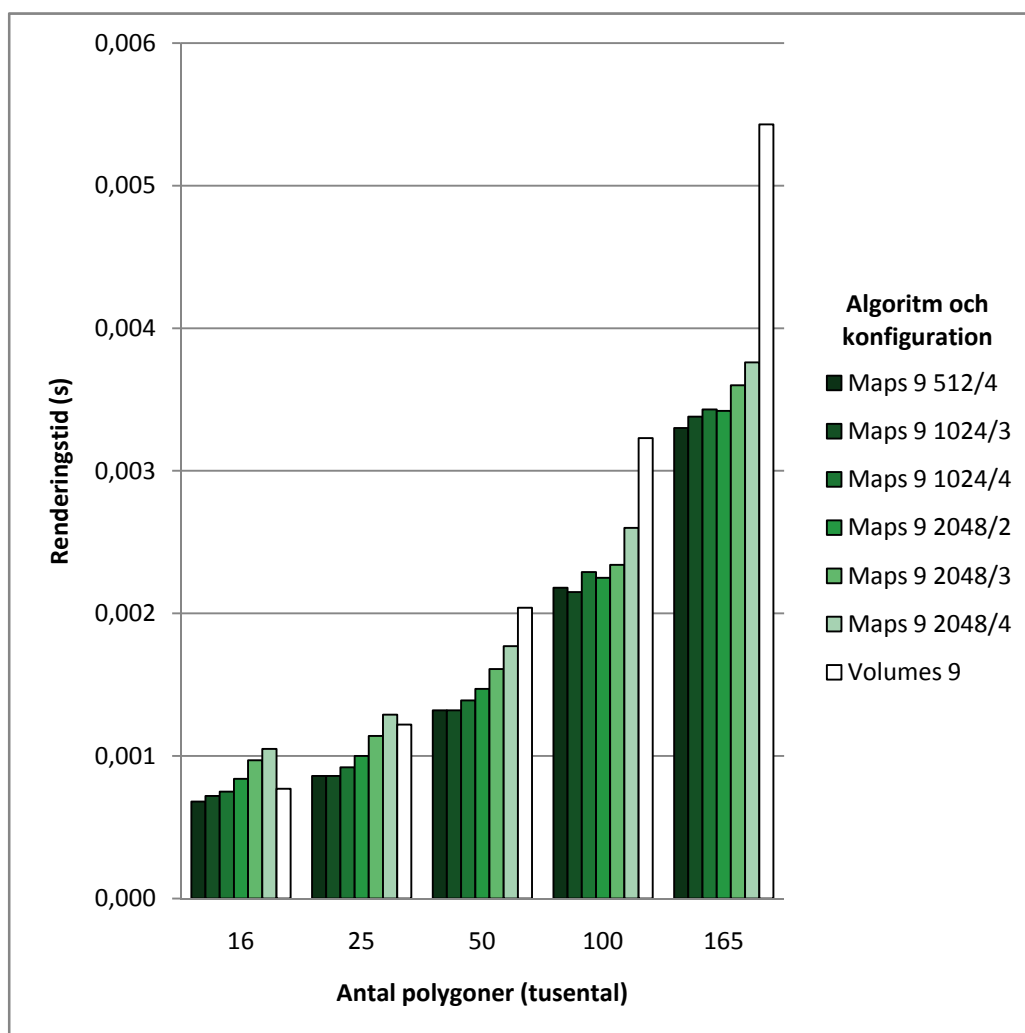
Figur 20. Nio olika kombinationer av inställningar för shadow map-upplösning och antal frustumdelar visas i figur 20(a). De något suddiga kanterna uppstår på grund av PCF-filtrering som vissa grafikkort från tillverkaren nVidia kan applicera utan prestandaförlust. Figur 20(b) visar resultatet genererat av shadow volumes-algoritmen, vilket alltid är pixelperfekt. Bilden som används är endast en del av hela den renderade bilden. Figur 20(c) visar vilken del av den fullständiga bilden som används.

Jämförelse mellan maps 9 och volumes 9

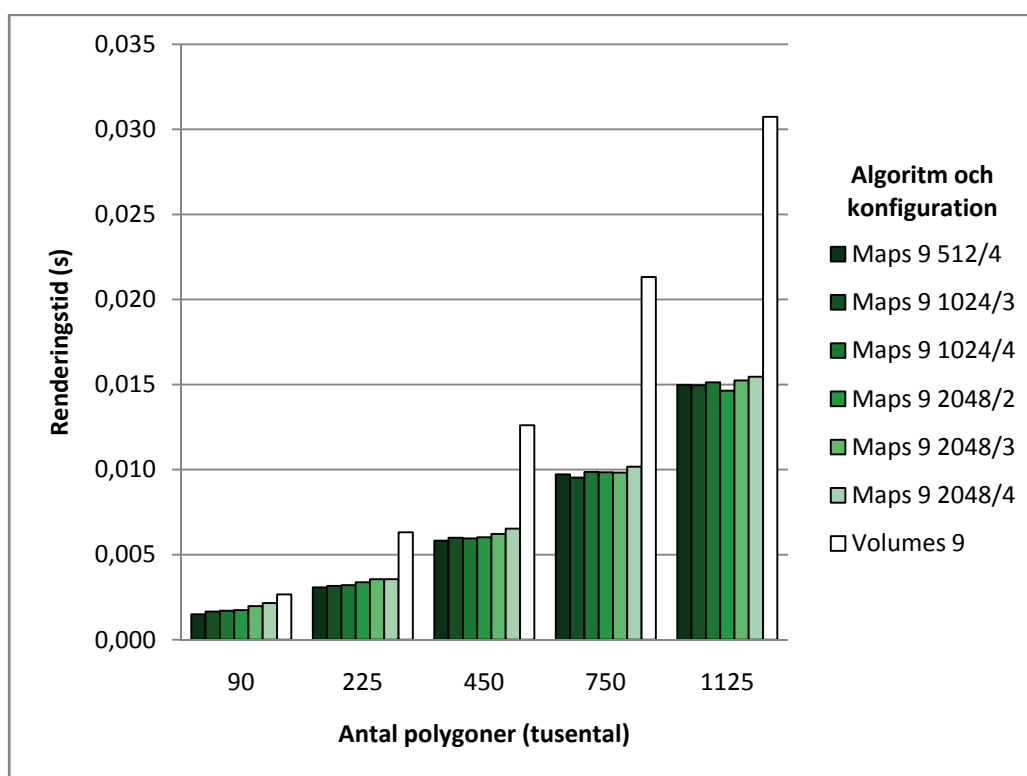
I följande tester jämförs shadow mapping- med shadow volumes-algoritmen. Utifrån föregående test har vi valt att jämföra maps 9 och volumes 9 då de i samtliga tester har visat upp bäst prestanda. Vi har valt att inkludera shadow mapping-konfigurationerna (512/4), (1024/3), (1024/4), (2048/2), (2048/3) och (2048/4).

Test 3.3

Det första testet undersöker hur antal polygoner i scenen påverkar prestandan. Två körningar görs där den första mäter prestandan vid låga polygonantal (se figur 21) och den andra mäter prestandan vid medel till höga polygonantal (se figur 22). Vid båda körningarna testas fem olika polygonantal för scenen. Vid den första körningen används modellen drone och vid den andra modellen robot.



Figur 21. Resultat för test 3.3a.



Figur 22. Resultat för test 3.3b.

Utifrån resultatet kan vi se att båda algoritmerna ökar linjärt, dock ökar volumes 9 snabbare än maps 9. Redan vid lägsta antalet polygoner har maps 9 (512/4) samma resultat som volumes 9. Vid 25 000 polygoner är det endast maps 9 (2048/4) som har en högre renderingstid än volumes 9. Därefter presterar alla maps 9-konfigurationer bättre än volumes 9. Vi ser även att i scenen med få polygoner presterar volumes 9 bättre än volumes 9 i scenen med medium till många polygoner. Ytterligare en intressant iakttagelse är att maps 9 (2048/2) presterar något bättre och är snabbare än (1024/4) i scenerna med fler polygoner.

Tolkning av resultat

Volumes 9 ger relativt sett bättre resultat i scenen med få polygoner än i scenen med medium till högt antal polygoner i jämförelse med shadow maps. Det beror dels på att det i scenen med få polygoner är fler pixelfragment per vertex vilket är till nackdel för maps 9 mer än för volumes 9 och dels på att i scenen med medium till högt polygonantal läggs större arbetsbörda på vertex shadern vilket är till större belastning för volumes 9.

Maps 9 (2048/2) gynnas i scenerna med fler polygoner eftersom det är mindre risk att objekt skärs av frustumgränserna när färre frustumdelar används. Vid scener med fler objekt är risken att detta händer större. När ett objekt skärs av en frustumgräns måste objektet dupliceras vilket gör att polygonantalet i scenen ökar.

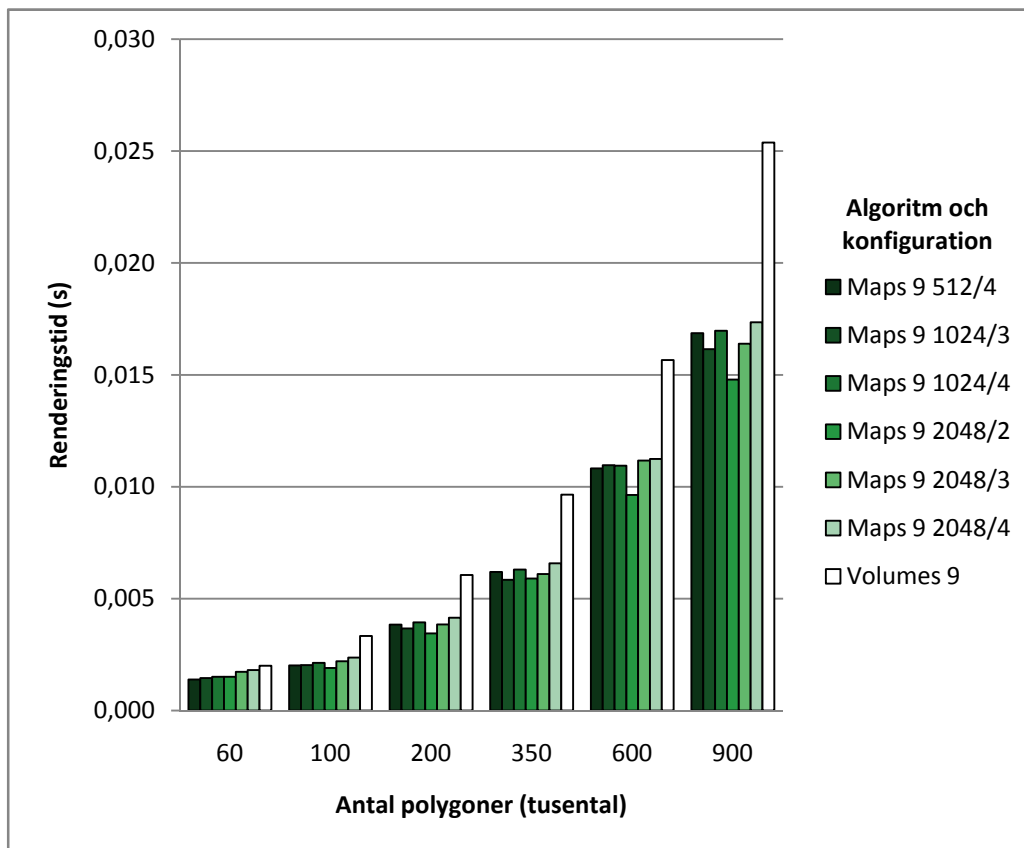
Test 3.4

Det andra testet undersöker hur ljusets infallsvinkel påverkar prestandan (se figur 23). Scenen som testas är den samma som för jämförelsen av skuggkvalitet för shadow mapping. Infallsvinkeln som testas är 75 grader och ljuset kommer framifrån. Modellerna som används är robot och flygplan. Det totala antalet polygoner i scenen är 450 000.

Här klarar sig volumes 9 relativt bra vilket är lite oväntat med tanke på att shadow volumes är långsammare vid låga ljusvinklar och längre skuggvolymer. Vi ser att shadow mapping-konfigurationer med få frustumdelar har en större fördel i detta test än det föregående i jämförelse med de konfigurationer med fler frustumdelar.

Tolkning av resultat

En förklaring är att ljusets frustum i shadow mapping måste täcka upp eventuella skuggkastare utanför kamerans vy. Vid större ljusvinklar kommer genereringen av shadow mappen alltså behöva inkludera objekt utanför den aktuella vyfrustumdelen. I praktiken betyder det att om ett objekt kastar skugga på objekt i flera frustumdelar så måste det renderas en extra gång för varje av dessa delar.



Figur 23. Resultat för test 3.4.

Diskussion

Arbetet fokuserade på prestandaskillnaden mellan shadow mapping- och shadow volumes-metoden genom de av oss utvalda algoritmerna, samt hur prestandan för de båda algoritmerna påverkas av användande av ny funktionalitet i Direct3D 10.

Det första testet visar att prestandan för volumes 10 påverkas nästan inte alls av ökande fillrate-krav. Det gör dock volumes 9. Det andra testet visar att det tar flera gånger längre tid för volumes 10 att processa vertexar än för volumes 9.

Det första testet i jämförelsen mellan maps 9 och 10 visar att storleken på renderingsytan gör liten skillnad för prestandan. Skillnaden mellan prestandaförsämringen när upplösningen ökas för maps 9 och 10 är inte heller speciellt stor, med något lägre siffror för maps 10. Det andra testet visar att prestandaförsämringen är liten när antalet frustumdelar eller shadow maps-upplösningen ökar. Det tredje testet visar att antalet polygoner i scenen påverkar prestandan för maps 10 i större utsträckning än maps 9.

Vid jämförelsen mellan maps 9 och volumes 9 visar det första testet att vid låga polygonantal är volumes 9 snabbare. Men redan från och med 25 000 polygoner har maps 9 bättre prestanda. Vid 450 000 polygoner uppvisar volumes 9 en renderingstid som är dubbelt så lång som den för maps 9. Det andra testet visar att ljusets infallsvinkel förbättrar resultatet för volumes 9 då dess renderingstid inte ökar lika snabbt i relation till renderingstiden för maps 9.

Genom alla tester observerades att implementationerna som använde sig av ny funktionalitet i Direct3D 10 (maps 10 och volumes 10) hade avsevärt sämre prestanda. Den väsentliga skillnaden mellan Direct3D 9 och 10 implementationerna är att Direct3D 10-implementationerna utnyttjar geometry shaders. Detta tyder på att geometry shadern på grund av antingen dåligt hårdvarustöd, dåliga drivrutiner eller en kombination av dem båda lider av dålig prestanda. Detta stöds ytterligare av faktumet att de exempelimplementationer vilka utnyttjar geometry shadern som är inkluderade i Direct3D 10-SDK:et lider av dålig prestanda.

Ett ungefärligt jämförelsevärde på skuggkvaliteten kan för shadow maps beräknas genom shadow map-upplösningen multiplicerat med antal frustumdelar. I en helt linjär fördelning stämmer värdet för upplösningen i Z-led men i X-led är självfallet shadow map-upplösningen begränsande då det är en enda shadow map i bredd oberoende av antal frustumdelar. En dubbelt så bred shadow map tar fyra gånger så mycket minne men är bara något långsammare (se figur 21 och 22) medan minnesåtgången är linjär mot antal shadow maps. Detta innebär att ett större antal frustumdelar är att föredra ur minnesperspektiv och högre upplösning på shadow mapen är att föredra ur prestandaperspektiv.

Undersökningens mål har uppnåtts. Frågor angående prestandan för geometry shaders har dock uppstått vilka är intressanta att undersöka vidare.

Slutsats

I detta arbete har vi undersökt relationen mellan prestandan för shadow mapping- och shadow volumes-metoden genom att jämföra renderingstiden för två av oss utvalda varianter. Undersökningen visar att för scener med annat än mycket lite geometri har shadow mapping-algoritmen bättre prestanda än shadow volumes-algoritmen. Shadow maps-upplösningen och antalet frustumdelar har liten inverkan på prestandan för shadow mapping-algoritmen. Vi kan inte se någon anledning i dagsläget att använda geometry shader-steget vid implementation av vare sig shadow mapping eller shadow volumes-algoritmen på grund av dess dåliga prestanda.

Det är dock osäkert hur länge det kommer att vara fördelaktigt att implementera algoritmerna utan att använda geometry shaders. Med ny hårdvara och med nya drivrutiner förutsätter vi att prestandan för geometry shaders kommer att förbättras. Vi föreslår därför att nya tester utförs på kommande generationer av Direct3D 10-hårdvara. Det kan även vara intressant att närmare undersöka mer specifikt vad som ligger bakom den stora prestandaförlusten för geometry shadern och vägar att kringgå problemet.

Referenser

- BATAGELO H., AND JUNIOR I. 1999. Real-Time Shadow Generation Using BSP Trees and Stencil Buffers. *Proceedings of SIBGRAPI Vol. 12*, pp. 93-102.
- BESTIMT, J., AND FREITAG, B. 1999. Real-Time Shadow Casting Using Shadow Volumes. Hämtad 22 januari, 2008, från http://www.gamasutra.com/features/19991115/bestimt_freitag_03.htm.
- BILODEAU, B, AND SONGY, M. 1999. Konferens för spelutvecklare sponsrad av Creative Labs, ej publicerade slides, Los Angeles, Maj 1999.
- BRABEC S., AND SEIDEL H.-P. 2003. Shadow Volumes on Programmable Graphics Hardware. *Proceedings of Eurographics 2003 Vol. 22*, pp. 433-440.
- BRABEC, S., ANNEN, T., AND SEIDEL, H-P. 2002. Practical Shadow Mapping. *Journal of Graphics Tools* 7, 4, pp. 9-18.
- CARMACK, J. 2000. Ej publicerad korrespondens. Hämtad 22 januari, 2008, från http://developer.nvidia.com/object/robust_shadow_volumes.html.
- CARMACK, J. 2004. John Carmacks keynote på QuakeCon 2004.
- CROW, F. C. 1977. Shadow Algorithms for Computer Graphics. *Computer Graphics (Proceedings of ACM SIGGRAPH 77)*, ACM, pp. 242-248.
- DA VINCI, L. 1970. The Notebooks of Leonardo da Vinci, Vol 1. Dover, New York.
- DIEFENBACH P. 1996. Multi-Pass Pipeline Rendering: Interaction and Realism through Hardware Provisions. PhD thesis, University of Pennsylvania.
- EVERITT, C., AND KILGARD, M. 2002. Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering. Hämtad 22 januari, 2008, från http://developer.nvidia.com/object/robust_shadow_volumes.html.
- FERNANDO, R., FERNANDEZ, S., BALA, K., AND GREENBERG, D. P. 2001. Adaptive Shadow Maps. *Proceedings of ACM SIGGRAPH 2001*, ACM Press/ACM SIGGRAPH, New York. E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, pp. 387-390.
- HAINES, E., AND MÖLLER, T. 2001. Real-Time Shadows. *Game Developers Conference, CMP*, pp. 335-352.
- HAINES, E., AND MÖLLER, T. 2002. Real-Time Rendering. A K Peters Ltd, 2nd edition.
- HASENFRATZ, J.-M., LAPIERRE, M., HOLZSCHUCH, N., AND SILLION, F. 2003. A Survey of Real-Time Soft Shadows Algorithms. *Computer Graphics Forum Vol. 22*, 4 (December), pp. 753-774.
- HEIDMANN, T. 1991. Real Shadows, Real Time. *Iris Universe*, 18 (November), pp. 23-31.
- HUBONA, G., WHEELER, P., SHIRAH, G., AND BRANDT, M. 1999. The Role of Object Shadows in Promoting 3D Visualization. *ACM Transactions on Computer-Human Interaction*, 6(3):pp. 214-242.
- KERSTEN, D., MAMASSIAN, P., AND KNILL, D. 1994. Moving Cast Shadows and the Perception of Relative Depth. *Technical Report no 6*, Max-Planck-Institut fuer biologische Kybernetik.
- KERSTEN, D., MAMASSIAN, P., AND KNILL, D. 1997. Moving Cast Shadows and the Perception of Relative Depth. *Perception* 26 , pp. 171-192.
- KILGARD, M. 2001. Robust Stencil Volumes. Presentation vid CEDEC 2001, Tokyo, 4:e september, 2001. Hämtad 22 januari, 2008, från http://developer.nvidia.com/object/cedec_stencil.html

- KNILL, D., MAMASSIAN, P., KERSTEN, D. 1997. Geometry of Shadows. *Journal of the Optical Society of America Vol. 14*, 12, pp. 3216–3232.
- LAMBERT, J. 1759. Die freye Perspektive. 1759.
- LLOYD B., WENDT J., GOVINDARAJU N., MANOCHA D. 2004. CC Shadow Volumes. *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering)*, Eurographics Association, pp. 197-205.
- MAMASSIAN, P., KNILL, D., AND KERSTEN, D. 1998. The Perception of Cast Shadows. *Trends in Cognitive Sciences Vol. 2*, 8, pp. 288–295.
- MARTIN, T., AND TAN, T.-S. 2004. Anti-Aliasing and Continuity With Trapezoidal Shadow Maps. *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering)*, Eurographics Association, pp. 153-160.
- MCCOOL M. 2000. Shadow Volume Reconstruction from Depth Maps. *ACM Transactions. on Graphics Vol. 19*, 1, pp. 1-26.
- MICROSOFT CORPORATION. 2004. ShadowVolume Sample. Hämtat den 22 januari, 2008, från. <http://msdn2.microsoft.com/en-us/library/bb147373.aspx>
- REEVES, W., SALESIN, D., AND COOK, R. 1987. Rendering Antialiased Shadows with Depth Maps. *Proceedings of SIGGRAPH '87*, ACM Press, pp. 283-291
- STAMMINGER, M. AND DRETTAKIS, G. 2002. Perspective Shadow Maps. *Proceedings of ACM SIGGRAPH 2002*, pp. 557-562.
- TADAMURA, K., QIN, X., JIAO, G., AND NAKAMAE, E. 1999. Rendering Optimal Solar Shadows Using Plural Sunlight Depth Buffers. *Computer Graphics International 1999*, pp. 166-173.
- WANGER, L. 1992. The Effect of Shadow Quality on the Perception of Spatial Relationships in Computer-Generated Imagery. *Computer Graphics (Interactive 3D Graphics 1992)*, Vol. 25, 2, pp. 39–42.
- WANGER, L., FERWERDA, J., GREENBERG D. 1992. Perceiving Spatial Relationships in Computer-Generated Images. *IEEE Computer Graphics and Applications Vol. 12*, 3, pp. 44-58.
- WILLIAMS, L. 1978. Casting Curved Shadows on Curved Surfaces. *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, ACM, pp. 270-274.
- WIMMER, M., SCHERZER, D., AND PURGATHOFER, W. 2004. Light Space Perspective Shadow Maps. *Proceedings of the Eurographics Symposium on Rendering*, Eurographics Association, pp. 143-152.
- WOO, A., POULIN, P., AND FOURNIER, A. 1990. A Survey of Shadow Algorithms. *IEEE Computer. Graphics and Applications Vol. 10*, 6, pp. 13–32.