HÖGSKOLAN
I GÄVLE

AKADEMIN FÖR TEKNIK OCH MILJÖ
Avdelningen för industriell utveckling, IT och samhällsbyggnad

Effektiva renderingstekniker av subsurface scattering med mental ray

Eric Jonasson

Juni 2013

Examensarbete, Grundnivå (kandidatexamen), 15hp
Datavetenskap
Creative Computer Graphics

Handledare: Ann-Sofie Östberg
Examinator: Torsten Jonsson

Effektiva renderingstekniker av subsurface scattering med mental ray

av

Eric Jonasson

Akademin för teknik och miljö
Högskolan i Gävle

S-801 76 Gävle, Sweden

Email:
ncg10ejn@student.hig.se

Abstrakt

Subsurface Scattering är nu för tiden väldigt populärt inom datorgrafik och används inom de flesta filmer och spel, men det är ingen enkel process för att uppnå ett bra resultat. Därmed testas och utvecklas subsurface scattering i verkligheten och inom Autodesk Maya med inbyggda renderingsmotorn mental ray. Subsurface scattering via mental ray kan lösas på flera olika sätt och nödvändig information om subsurface scattering shaders och andra noder förklaras för de metoder som känns till som lightmaps och photon maps. Det finnas även en tredje metod som fokuserar på passrendering av en lightmap shader. Slutsatsen är att metoderna kan rendera alla olika sorter av subsurface scattering där materialet inte har någon större betydelse och att det är kritiskt att förstå sig på dess attribut för att åstadkomma ett bra resultat.
Nyckelord: Subsurface Scattering, BSSRDF, Mental ray, Lightmap, Photon map, Passrendering

Innehåll

1 Introduktion 1
1.1 Syfte 1
1.2 Mål 1
1.3 Frågeställningar 1
1.4 Metod 1
1.5 Avgränsningar 1

2 Teoretisk bakgrund 2
2.1 BSSRDF ljusfunktion 2
2.2 Shaders 3
2.3 Mental ray 3
2.4 Mental rays misss shaders 3
2.5 misss_lightmap_write shader 3
2.6 misss_physical och photon tracing 4
2.7 Renderingspass med misss_fast_shader_x_passes och custom passes 4

3 Realisering 4
3.1 Metod 1 - Hud med lightmap 5
3.1.1 Ljusuppsättning och misss_fast_lmap_maya attribut 5
3.1.2 Misss_fast_simple_maya attribut 6
3.1.3 Utveckling av metod 1 8
3.2 Metod 2 - Stearinljus med photon map 10
3.2.1 Fotoner från ljuskällor och attribut 10
3.2.2 misss_physical attribut 11
3.2.3 Utveckling av metod 2 12
3.3 Metod 3 - Passrendering med misss_fast_skin_maya 14
3.3.1 Verktygsnoder i mental ray 14
3.3.2 Utveckling och förklarning av metod 3 15

4 Resultat 16
4.1 Rendering 1 – misss_fast_simple_maya 16
4.2 Rendering 2 – misss_physical 17
4.3 Rendering 3 – misss_fast_skin_maya passrendering 17

5 Diskussion 18
5.1 Subsurface scattering med lightmap 18
5.2 Subsurface scattering med photon map 19
5.3 Subsurface scattering passrendering med lightmap 20
5.4 Sammanfattning 20

6 Slutsatser 20

7 Förslag på fortsatt forskning 21

8 Referenser 22

1 Introduktion

Under de senaste åren så har subsurface scattering fått mycket uppmärksamhet inom datorgrafik. Det är nu det är störst konkurrens om vem som kan skapa mest realistiska arbeten och subsurface scattering är en stor del inom området. Subsurface scattering är en mekanism för ljustransport, där ljus penetrerar ytan av ett genomskinligt föremål. Ljuset sprids genom interaktion med materialet, och lämnar ytan vid en annan punkt. Ljuset penetrerar ytan och reflekteras ett antal gånger med oregelbundna vinklar inuti materialet och när det passerar tillbaka ut ur materialet vid en vinkel så kommer subsurface scattering effekten att skapas. Så allting som släpper igenom ljus och har någon sort av grumlig substans i sig kan skapa effekten, som t.ex. vax, mjuk plast, hud, mjölk, vatten, snö m.m. och det gäller även medium som inte är i fast form som t.ex. eld och rök.

1.1 Syfte

Syftet med arbetet är att studera olika shaders och att kunna använda dem på ett effektivt sätt. Genom att bedöma de olika metoderna av subsurface scattering så förmedlar uppsatsen kunskaper till studenter och professionella. De kommer kunna avgöra vilka metoder som anses vara bäst för dem.

1.2 Mål

Målet med uppsatsen är att få en bättre förståelse hur mental ray fungerar med subsurface scattering inom Autodesk Maya. Därav används shaders i olika metoder på ett effektivt sätt som åstadkommer realistiska resultat. Resultatet av metoderna visas genom renderade bilder. Resultaten jämförs sedan att så fördelar och nackdelar kan visas.

1.3 Frågeställningar

Frågor som besvaras under arbetets gång.

- *Vilken metod är bäst lämpad att använda för subsurface scattering med mental ray?*
- *På vilka sätt kan man åstadkomma subsurface scattering med mental ray?*
- *Vad är skillnaderna mellan lightmap och photon map? Vilka fördelar och nackdelar finns det?*

1.4 Metod

Huvudsakligen utvecklas mental rays misss(mental images subsurface scattering) shaders genom experimentering, sedan görs en komparativ studie mellan metoderna. De metoder som är mest lämpade för detta projekt tas fram via studier. Den teoretiska bakgrunden ger utökad kunskap om metoderna och dessa metoder fungerar som en ungefärlig utgångspunkt som förbereder läsaren för de mest komplexa subsurface scattering processer inom mental ray.

1.5 Avgränsningar

Det här arbetet håller störst fokus på hur subsurface scattering behandlas med hjälp av mental ray renderingsmotor inom Autodesk Maya 2013. Därmed så vidareutvecklas misss shaders och användas på ett så effektivt sätt som möjligt. Anledningen till misss shaders är för de är specifikt utvecklade för subsurface scattering. Det finns även

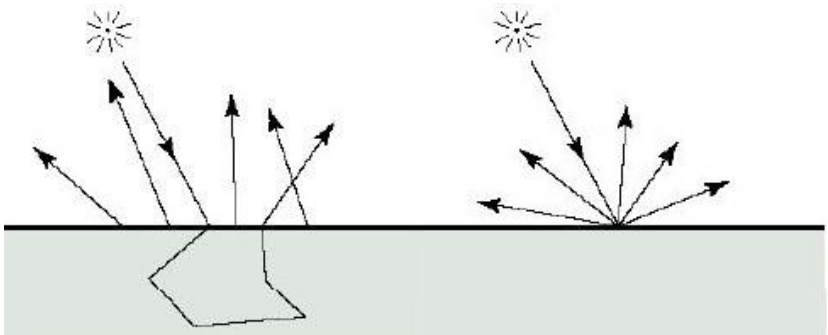
andra shaders inom mental ray som kan åkstadkomma effekten som t.ex. dgs_material och dielectric_material, men de är inte utvecklade för subsurface scattering. Utöver det så tas ljusuppsättning upp och hur subsurface scattering attribut tar hand om ljuset. Vilket är kritiskt i förståelsen av hur dessa subsurface scattering metoder fungerar. Varje scen har samma typ av belysning, så ingen scen får bättre resultat på grund av belysningen.

2 Teoretisk bakgrund

Denna bakgrund innehåller generella funktionsprinciper hur subsurface scattering fungerar. Den innehåller information om vad mental ray är och hur subsurface scattering effekten kan lösas inom det, som misss shaders, lightmap, photon maps, noder och olika sätt att göra passrenderingar med misss shaders. Vidareutveckling utav de metoder som nämns i teoretiska bakgrunden utövas i metoden.

2.1 BSSRDF ljusfunktion

Ljuset antingen absorberas eller sprids när den stöter på en yta och de beräkningar som man har för att generera belysningen av ljus kommer bli väldigt tungt för datorn. Därmed är det inte lämpligt att använda sig av en fullt utvecklad modell för att beräkna ljus så kan man använda sig av BSSRDF metoden istället [1]. BSSRDF (Bidirectional surface scattering reflectance distribution function) är skapad av H. W. Jensen och funktionen beskriver hur ljus transporteras under ytan(subsurface) mellan två strålar som träffar en yta [2]. Ljuset studsar även runt under ytan vilket betyder att det kommer studsa ut igen, se figur 1. BSSRDF är en komplex funktion av BRDF (Bidirectional reflectance distribution function). BRDF anger hur mycket av det ljus som infaller från en riktning ger ifrån sig i en annan riktning och det fungerar på de flesta material som inte är genomskinliga, se figur 1 [3].



Figur 1. BSSRDF till vänster och BRDF till höger av H. W. Jensen [2].

"The most advanced and accurate, yet computationally feasible model for calculating subsurface scattering is the BSSRDF method introduced by Jensen (2001a)." [1, p. 5]

BSSRDF's funktion har två inslag av hur ljus sprider sig i olika sorters av material. Sätten hur ljus kan sprida sig är single och multiple scattering. Single scattering är ljuset som går in i ett material som sedan studsar av en partikel och sedan så studsar ljuset direkt ut igen från materialet i en annan riktning, t.ex. starkt ljus mot glas. Multiple scattering är ljuset som kommer in i materialet som studsar runt på flera olika partiklar inuti och sedan åker ut ur materialet igen. Därmed så har multiple scattering en tendens att sprida ut sig över ett större område [1].

2.2 Shaders

Shaders inom 3D är ett litet program eller en uppsättning av algoritmer som avgör hur ytan av ett 3D objekt ska renderas, och hur ljuset interagerar med objektet i ett 3D program som t.ex. Maya. Vertices, färger, synvinklar och andra värden beräknas i programmet som sedan skickas till renderingsmotorn [4] [5].

2.3 Mental ray

Mental ray är ett fullt utrustad ray tracing program som har förmågan att skapa fotorealistiska bilder med hjälp av avancerade beräkningar. Mental ray är ett insticksprogram i Maya, 3ds Max och Softimage XSL. Det är en renderingsmotor som ger ett antal förbättringar utöver vanliga renderingsmotorer, bland annat kaustiska effekter (reflektion av ljus från en genomskinlig eller reflekterande yta), Indirect Illumination (sekundärt ljus) och volymetriska effekter (t.ex. vätskor och subsurface scattering)och många andra funktioner [6].

2.4 Mental rays misss shaders

Äldre versioner av Autodesk Maya's mental ray bidrar ytterst lite för subsurface scattering shaders. Maya 2008 och äldre versioner så var man tvungen att göra allt från början, med fixa egna shaders som att lägga till noder och shader grupper, och sedan koppla samman allt manuellt. Version Maya 2009 och över gör de arbetet åt dig. Däremot så är det bara grunden för subsurface scattering. För att uppnå realistiska effekter så måste man lägga till flera noder [7]. Maya har även färdiga subsurface scattering shaders man kan använda sig utav och vidare utveckla. Listan nedan är shaders från Maya 2009.

- misss_call_shader
- misss_fast_shader
- misss_fast_shader_x
- misss_fast_shader_x_passes
- misss_fast_simple_maya
- misss_fast_skin_maya
- misss_physical
- misss_set_normal
- misss_skin_specular

Alla misss shaders inom Maya använder sig utav samma grundläggande teknik, förutom misss_physical. Några shaders är även kombinationer av andra shaders [7]. Som t.ex. så kan man använda sig utav misss_skin_specular tillsammans med misss_fast_shader om du vill ha mer kontroll över specular attributen. Därmed utökas möjligheterna för att åstadkomma ett bättre resultat.

2.5 misss_lightmap_write shader

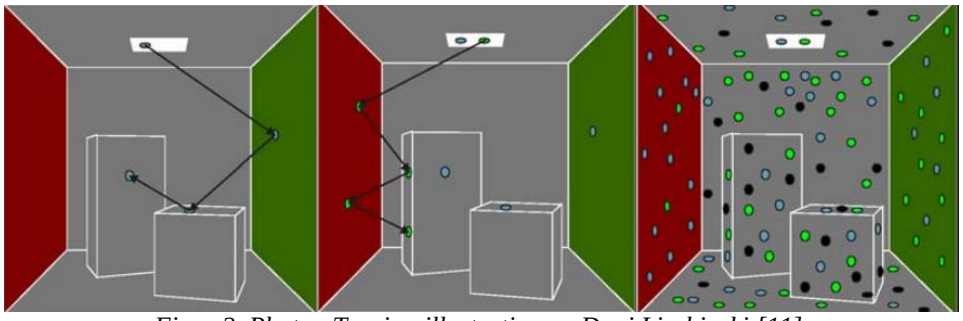
Lightmaps har under en lång tid använts för att öka realismen inom belysning och det är en relativt simpel metod att använda sig utav. Lightmaps används ofta i realtids då man kombinerar en lågupplöst mapp med högupplöst textur för att skapa en övertygande effekt. Däremot så måste en högupplöst lightmap användas om man vill ha skarpa skuggor [8].

Mental ray renderar subsurface scattering genom att generera lightmaps. Lightmap shaders tittar på vart varje yta är ifrån och tillbaka till kamerans perspektiv och sedan lagrar det totala ljusets(diffuse) information över en yta. Shaders bestämmer även tjockleken av ytan i beroende på kamerans position. Mappen används sedan i

t.ex. misss_fast_shader för att bestämma ytans spridning via raytracing. Detta innebär att lightmaps är viktigt för de flesta subsurface scattering effekter. Subsurface scattering kan inte uppstå utan en lightmap på grund av den viktiga informationen om ljus, djup och informationen om ytor [9] [6].

2.6 misss_physical och photon tracing

Till skillnad från alla andra misss shaders fungerar misss_physical på ett annorlunda sätt. Istället för att använda sig utav en lightmap som tar hand om ljuset så används fotoner. Vanligtvis så skapas fotoner av alla sorters av ljuskällor och ljuskällan skickar även ut lite energi som fotoner tar med sig. Energin i det här fallet är det som lyser upp ett rum t.ex. [5]. En photon map skapas med hjälp av photon tracing genom fotoner som åker in i ett genomskinligt material. De fotoner som inte kommer in i materialet blir hanterade som vanligt. Den största skillnaden till skillnad från ray tracing är att fotoner lagras vid konkat med ett material och när en foton lagras(se figur2), så skapas en ny foton från ljuskällan [10].



Figur 2. Photon Tracing illustration av Dani Lischinski [11].

2.7 Renderingspass med misss_fast_shader_x_passes och custom passes

Renderingspass renderar ut specifik information från en renderad bild. Renderingspass kan rendera ut *Refraction*, *Diffuse Color*, *Ambient Occlusion* och *Specular Highlights*, m.m. Syftet med renderings pass är att man har full kontroll över slutresultatet och så slipper man rendera om varje gång man vill ändra på något. Shaders som heter något med _passes har en inbyggd render pass support, därmed så kan man använda misss_fast_shader_x_passes för att få extra kontroll över sitt slutresultat. Det är även möjligt att göra egna pass i ett shader nätverk, så man kan ha mer kontroll över den information man samlar. Detta görs genom att använda en writeToColorBuffer nod. En writeToColorBuffer nod är en nod som kan användas inom shader nätverket, för att samla åt sig RGBA data för att sedan lagras i en frame buffer (custom color pass) [12].

3 Realisering

I detta kapitel beskrivs det praktiska arbetet som utförs inom Autodesk Maya och med hjälp av renderingsmotorn mental ray. Arbetet innehåller tre olika metoder hur subsurface scattering kan utföras med shaders och sedan utförs en komparativ studie. Eftersom effekten kan åstadkommas på alla transparenta material, så har varje ny metod ha ett nytt material att efterlikna. Den första metoden illustrerar människans hud, och det uppnås med hjälp av en misss_fast_simple_maya shader som använder sig utav lightmaps.

I den andra metoden används en misss_physical shader som kräver fotoner från ljuskällor inom Maya för att fungera och i denna metod så kommer ett brinnande stearinljus demonstreras.

I den tredje och sista metoden så används misss_fast_skin_maya som också använder sig av lightmaps, därmed är passrendering med hjälp av mental rays noder vara i fokus istället.

3.1 Metod 1 - Hud med lightmap

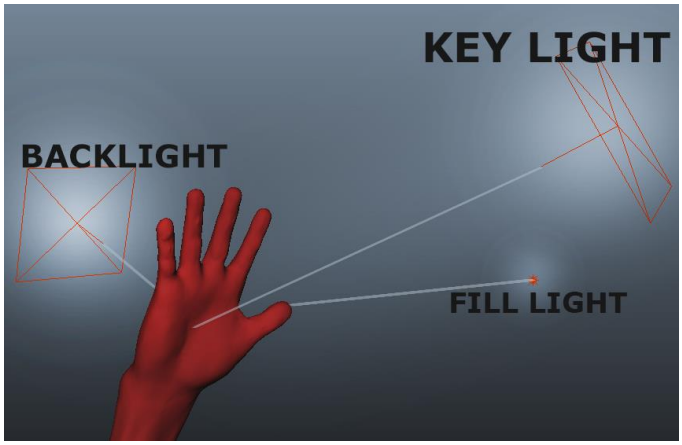
I första metoden så försöktes en realistisk effekt av subsurface scattering återskapas på en hand med hjälp av misss_fast_simple_maya som använder sig utav lightmaps och tanken är att ett starkt backlight ska demonstrera den största effekten av subsurface scattering. Eftersom människans hud har en stor grad av genomskinlighet och fingrarna liten volym utöver resten av handen, så har fingrarna störst effekt. För att förstärka resultatet så applicerades en normal map och cavity map från en 3D modell med högt antal av polygoner, för att få mer detaljer på handen.

Figur 3. Hand med låg polygon antal, normal map, cavity map och utan subsurface scattering shader.

3.1.1 Ljusuppsättning och misss_fast_lmap_maya attribut

Vanligtvis vid en rendering av en scen så krävs det vissa belysnings tekniker för att kunna åstadkomma ett realistiskt resultat. För denna effekt så används three-point lightning ljusuppsättning i alla scener.

I denna teknik (se figur 4) används tre lampor. Det första ljuset kallas för key light. Det är oftast det starkaste ljuset och detta ljus sätter belysningen av scenen. Det andra ljuset kallas för fill light, och de hjälper att fylla i skuggorna från key light, så det inte blir för skarpa kanter. Det sista ljuset kallas för backlight, och används för att skapa en kontur, utöver de andra ljuskällorna.



Figur 4. Three-point light ljusuppsättning i Maya.

De ljuskällorna som används i denna scen är två stycken area lights, som används som key light och backlight. Där backlight är placerat precis bakom objektet och key light är placerat framför objektet och lite högre upp än de övriga ljuskällorna. Den tredje ljuskällan är en point light som används som fill light och ljuset placerat vid sidan av objektet med ett svagt ljus.

I misss_fast_lmap_maya noden så fanns det några hjälpande attribut som man är tvungen att förstå sig på om man vill ha full kontroll över sin shader. Noden är direkt kopplad till misss_fast_simple_maya, därmed så kommer ändringar i noden påverka slutresultatet.

Lightmap Sample

- Include Indirect Lightning
- Diffuse Gamma Curve

I Lightmap Sample listan så finns det två attribut som kan vara hjälpsamma. Första attributen *Include Indirect Lightning* är det ljus som inte kommer direkt från en ljuskälla, utan de ljusstrålar som studsar från andra ytor och tillbaka till objektet, och det är menat att ge en fotorealistisk känsla. I denna scen så har inte indirect lightning någon effekt, eftersom det inte finns några objekt för ljusstrålarna att studsa på.

Diffuse Gamma Curve har funktionen att ta hand om den diffusa delen på sin lightmap, och den kommer inte påverka diffusen på sin shader. Om man har ett lågt värde kommer de ge uttrycket av ett platt objekt med mindre djup, och högre värde ger känslan av ett tjockare objekt, dvs. mer intressant spridning.

Lightmap Write

- Scatter Bias

Under Lightmap Write fliken så finns *Scatter Bias*, och där kan ens lightmap antingen fokusera ljuset mer åt framsidan eller baksidan på sin shader. Men bör hållas på default värde så länge som möjligt, eftersom det blir mindre spridning på motsatt sida.

3.1.2 Misss_fast_simple_maya attribut

Dessa attribut ingår i misss_fast_simple_maya shader och de alla är relevanta till effekten som skapades i den här metoden. De attribut som inte tas upp påverkar inte slutresultatet eller så ingår de i ett annat kapitel.

Unscattered Diffuse Layer

- Ambient
- Overall Color
- Diffuse Color/Weight

Syftet med denna flik är kontrollen över det som inte sprider sig inuti objektet. Där *Ambient* har kontrollen över hur ljust eller mörkt ytan ska vara, men bör hållas så mörkt som möjligt eftersom den kommer matta ut andra färger genom att addera värdet på *Diffuse Color*. *Overall Color* är en multiplicerare för *Diffuse Color* och de färger som anges under subsurface scattering layer.

Diffuse Color är den generella färgen på din shader som påverkas av *Ambient* och *Overall Color*. *Diffuse Weight* är hur mycket ljus som släpps igenom, därav ett högt värde så släpps mindre ljus igenom, så har man värdet 1.000 då default är 0.000, så kommer ditt objekt ha samma färg som din D*iffuse Color*.

Subsurface Scattering Layer

- Front SSS Color
- Front SSS Weight
- Front SSS Radius
- Back SSS Color
- Back SSS Weight
- Back SSS Radius
- Back SSS Depth

I Subsurface Scattering Layer har man mest kontroll över sin subsurface scattering effekt. *Front SSS Color* är baserad utifrån kamerans vy, och den färgen som väljs kommer alltid vara närmast kameran. Front color är också en multiplicerare till *Back SSS Color*. För motsatt effekt så finns *Back SSS Color* som också är baserad på kamerans vy, däremot så lägger sig färgen bakom front color och den färgen bör vara ljusare än front color för att förstärka konturen från sin backlight.

Front SSS Weight är den spridning som sker på framsidan av objektet. Det är menat att det ljus som kommer framifrån ska åka in en kort bit i objektet för sedan att studsa ut igen i en annan riktning. Detta är baserat vart ljuset kommer ifrån, så det krävs en ljuskälla framför objektet för att skapa en bra effekt. Därav så finns *Back SSS Weight*, som har samma funktion som front weight förutom att de är baserat på de ljusstrålar som kommer bakifrån. Men back weight släpper också igenom ljusstrålar helt till skillnad från front weight.

Front SSS Radius är en funktion som är baserat på Maya unit avstånd. Med front radius så har man kontroll över hur mycket ljus man kan släppa in i objektet, men som också är beroende av front weight. Har man ett värde på 0.000 så kommer inget ljus släppas igenom och ett värde på 1.000 kommer släppa igenom ljus upp till 1 Maya unit i radius. *Back SSS Radius* har samma funktion som front radius, men med *Back SSS Depth* kontrollerar man hur djupt ljusstrålarna går in från baksidan av objektet, och värdet bör vara lägre än tjockleken av objektet. Om depth värdet är satt till 0.000 kommer den automatiskt sättas till back radius värdet.

Algorithm control

- Scale Conversion
- Falloff
- Screen Composite

Med *Scale Conversion* under Algorithm control kan man säga åt sin shader att behandla objektet som att det är större än vad det egentligen är. Med ett värde på 5.000 skulle objektet behandlas som att det vore fem gånger större. Därmed alla attribut som är beroende av skala som front radius, back radius och depth kopplade tillsammans till S*cale Conversion*.

Falloff sätter avtagningsvärdet längst med scatter radius. Med låga värden får man en mjuk avtagning och med höga värden en skarp avtagning. Men det orsakar också att spridningen blir kortare och därmed måste man kompensera med att öka djupet på spridningen.

Screen Composite hjälper till att sammansätta alla lager genom att blanda lager med screen blend mode, vilket orsakar att färgerna blir ljusare och mattar ut varandra och gör det behagligt för ett mänskligt öga att titta på. Genom att inaktivera attributen så adderas lagren istället, som kommer orsaka starkare färger.

Lightmap

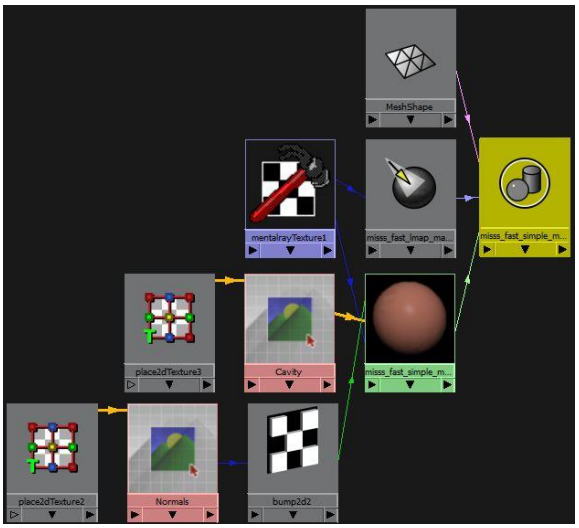
- Samples

Detta attribut är en kvalitets inställning för det antal samples som tas från ens lightmap vid rendering. Anledningen till att ens subsurface scattering shader blir grynig är för att default värdet i *Samples* är låg och bör därför höjas.

3.1.3 Utveckling av metod 1

På grund av alla subsurface scattering shaders, så finns det ett stort val av vilka som kan användas, därmed behövdes en undersökning om vad som verkade lämpligast för mänsklig hud. Det finns en shader vid namnet misss_fast_skin_maya, som var specifikt för hud och verkar ha onödiga attribut som man ska återskapa andra material, därav användes misss_fast_simple_maya eftersom den har många komplexa attribut, och har förmågan att skapa rätt effekt till alla sorters av material.

Efter att ha applicerat sin shader på objektet, nollställdes alla värden så man kan begripa vad som händer under arbetets gång. Till att börja med för att få handen se någorlunda bra ut behövdes en normal map som lades till via bump shader attribut. Sedan i 2d Bump Attributes måste man ändra till *Tangent Space Normals* så den skickar ut normalerna från sin normal map. Därefter applicerades en cavity map i O*verall Color*, eftersom mappen är i gråskala. Sedan fixades ljusuppsättningen (se figur 4) och med R*aytrace Shadows* aktiverat på alla lampor, därefter var allting klart för att börja med subsurface scattering effekten.



Figur 5. Slutgiltiga nätverket i hypsershade.

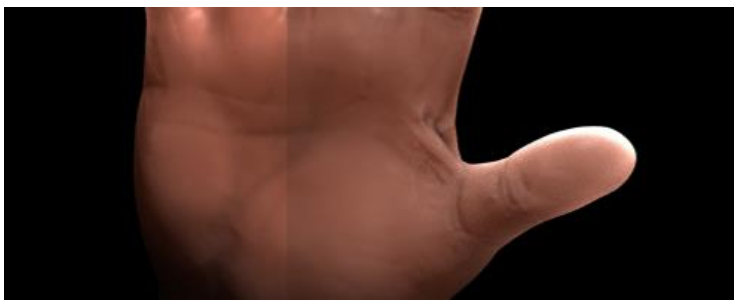
I figur 5 visas alla kopplade noder och shaders, som tillslut går till en shading group. Syftet med en shading group är att rendera ut den slutgiltiga geometrin och det behövs alltid en shading group som håller ihop nätverket.

Efter ha testa och experimenterat kom jag fram till värdena som visas i figur 7. Där *Ambient* värdet är svart för att inte matta ut andra färger från diffuse, front och back color. *Diffuse Color* har en liten nyans av orange färg i sig och väldigt ljust för att ha mer kontroll över front och back color. För front och back color efterliknades färgerna från verkligheten. Har man samma ljusuppsättning i verkligheten och håller handen i centrum så kommer man få känslan av starkare färger på framsidan av handen eftersom det är mindre ljus och motsatt effekt på baksidan. Därför är front color mörkare och har mer färg och back color ljusare och mindre färg. I *Shininess* användes ett lågt värde eftersom att människohud är relativt grovt och behöver ingen specifik highlight. I *Specular Colo*r användes en mörk röd färg för en "varmare" känsla.

Alla weight attribut är en multiplicerare till varje diffus färg som *Diffuse Color*, *Front SSS Color* och *Back SSS Color*. Därmed är det petigt att få en övertygande subsurface scattering effekt. Det som visas på framsidan av objektet är på grund av diffuse och front weight och en bra kombination av dem är 0.300 på D*iffuse Weight* och 0.700 på *Front SSS Weight*. Back weight har inte lika stor effekt av det som syns på framsidan av objektet och eftersom målet med back scattering är att det ska lysa igenom så användes ett högre värde.

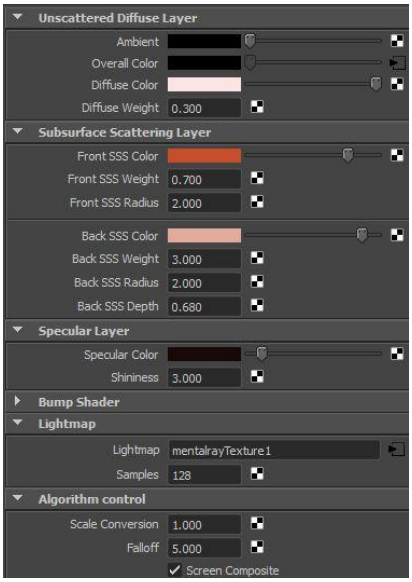
Med radius och depth är det viktigt att ta hänsyn till hur stort objektet är i Maya units och handen motsvarar ungefär 2 maya units i tjocklek. Därmed är värdena 2.000 för både front radius och back radius. *Back SSS Depth* har ett lågt värde eftersom värdet bestämmer hur långt ljusstrålarna åker in i objektet bakifrån. Strålarna kan åka in 0.700 maya units in i objektet. Med för högt värde åker strålarna igenom och gör att objektet blir vitt.

Sedan i Algorithm control lämnades S*cale Conversion* på default värde, eftersom scenen byggdes upp för en specifik modell. *Falloff* var relativt enkelt att förstå, efter ökat värde tog det längre tid innan effekten avtog, därmed ökades värdet till 5.000 för att lysa upp objektet mer. Med *Screen Composite* finns två valmöjligheter, aktivera och inaktivera. I detta fall så lämnas *Screen Composite* aktiverad, för att slippa sammansätta lager efter rendering. Med att lämna den inaktiverad kommer alla subsurface scattering lager adderas på varandra och orsaka att mindre spridning sker (se figur 6).



Figur 6. Till vänster är Screen Composite inaktiverad och aktiverad till höger.

Efter all experimentering med attribut blev resultatet som till höger i figur 6, därefter var det bara att få bort grynigheten inför slutresultatet. För att få bort det så behövde lightmap samples och anti-aliasing ett högre värde. anti-aliasing försöker minska det genom att skicka mer än en ljusstråle in till varje pixel och ta ut medelvärdet av resultatet och bestämma pixelns värde.

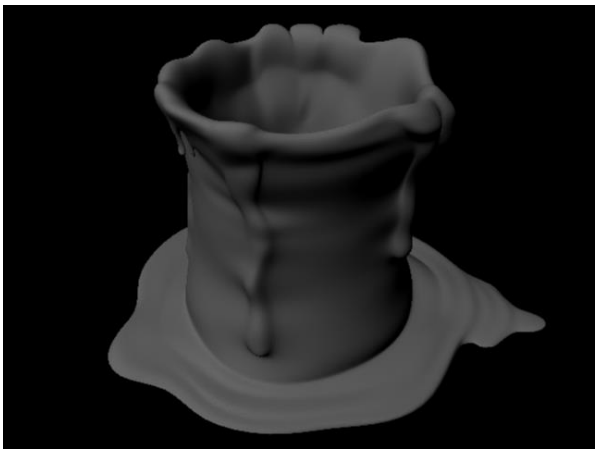


Figur 7. Slutgiltiga värden för handen.

3.2 Metod 2 - Stearinljus med photon map

I denna metod var syftet att återskapa en realistisk subsurface scattering effekt av ett stearinljus. Three-point light och en extra ljuskälla som motsvarar elden i stearinljuset. Subsurface scattering effekten kommer inte komma från alla ljuskällor. Det är elden som kommer illustrera den största delen av effekten, även backlight kommer förstärka effekten. Modellen som effekten kommer utövas på är skapad av Digital Tutors (se figur 8). Inga maps kommer tilläggas på objektet, eftersom stearinljus består av en färg.

Effekten illustreras när ljusstrålar skickar ut fotoner från ljuskällor, som sedan åker in i objektet och sprider. Därefter skapas en photon map, som används inom mental ray för att rendera global illumination och kaustik. Fotoner kan studsa runt i scenen, reflekteras och absorberas. Varje gång en foton träffar ytan sparas värdet av fotonen i en photon map. Fotoner inuti objektet sprider sig i två olika lager. Det första lagret är det lager man har kontroll över, då fotoner studsar i den riktning man vill att de ska studsa i. Sedan när fotoner avtar så börjar fotonerna studsa runt utan kontroll, vilket motsvarar det andra lagret [13]. Eftersom fotoner beter sig som fotoner gör i verkligheten, blir det genast en komplex metod att använda sig utav och det tar längre tid att rendera än användning utav lightmaps.



Figur 8. Stearinljus från Digital Tutors.

3.2.1 Fotoner från ljuskällor och attribut

Minst en ljuskälla måste ha *Emit Photons* aktiverad inom caustic and global illumination för att kunna rendera subsurface scattering med caustics eller global illumination. Egentligen så är caustics en typ av global illumination men de delas upp inom mental ray för Maya för bättre kontroll. Caustics är de ljus som orsakas av specularity, så i detta fall kommer inte materialet på stearinljuset producera mer caustics eftersom det är grumligt [14], men subsurface scattering effekten kommer vara densamma.

Caustic and Global Illumination

- Emit Photons
- Photon Intensity
- Exponent
- Caustic Photons

I mental ray inställningar måste *Caustic Photons* aktiveras, för misss_physical är beroende utav de kaustika fotonerna. Sedan kan *Emit Photons* aktiveras på varje ljuskälla som ska kunna skicka ut fotoner.

Photon Intensity är styrkan på C*austic Photons* och ett högt värde kommer inte orsaka mer spridning. *Exponent* representerar fotonenergins avtagning och ett värde på 2 motsvarar verklighetens avtagning av energin [14]. Med ett lågt värde tar det lång tid för energin att avta och ett högt värde kort tid.

Caustic Photons är det antalet fotoner ljuskällan skickar ut. Default värdet är 10,000 och det är ett lågt värde eftersom syftet är desamma som i verkligheten och bör ökas drastiskt inför slutresultatet.

3.2.2 misss_physical attribut

Egenskaper hos Algorithm control, kontrollerar kvalitén och den djupa informationen av renderingen. Misss_physicals shader är uppdelad i tre delar: single scattering, multiple scattering, och diffusion. Beroende på optiska egenskaperna på materialet, så kan varje del variera. En viktig faktor inom Algorithm control är att hantera det yttersta lagret, för att kunna bestämma det genomsnittliga avståndet en foton kommer färdas i materialet utan att spridas eller absorberas och är det motsvarande värdet av *Absorption* och *Scattering coefficients*.

Optimal properties

- Material
- Transmission
- Index of Refraction
- Absorption & Scattering coefficients
- Scale Conversion
- Scattering Anisotropy
- Approximate Diffusion, Single och Multiple scattering

Material har samma effekt som *Diffuse Color* i light misss fast shaders, där effekten är additiv och är den generella färgen på din shader. För att hantera färgen som subsurface scattering effekten ska man använda *Transmission*.

Index of Refraction finns för att ljus skall kunna in i ett material för att sedan studsa runt och studsa ut igen. Värdet varierar i beroende på vilket material som ska återskapas. Eftersom stearin är en typ av vax, så är refraktions värde 1.400 enligt Malvern Instruments Ltd [15].

Absorption och S*cattering coefficients* har nästan samma funktion, där absorption har känslan att stoppa ljus att gå igenom materialet och scattering har känslan att sprida vidare ljuset inuti materialet. Därmed bör absorption vara lågt och scattering högre, eftersom ljus sprider sig lätt i vax. De sänder ut värden i RGB, därmed ökar det kontrollen över färgen på sin shader.

Scale Conversion är baserat på millimeter och därmed är det viktigt att hålla koll hur stort objektet är i Maya. Om objektet är ungefär 10cm i Maya, bör värdet vara runt 100.

Scattering Anisotropy är graden av spridningens vinkel, negativt värde är baksidan, positivt värde är för spridning framåt och med neutralt värde sprids det åt alla håll. Värdet är beroende av materialet, om materialet har tendensen att sprida ljus framåt så ska man ha ett positivt värde.

Approx Diffusion, Single och *Multiple scattring* attribut är tre viktiga delar som behövs för att subsurface scattering ska uppstå. *Approx Diffusion* finns för att förstärka fotoner i innersta lagret med hjälp av de kaustiska ljuset runt om objektet. *Single scattering* är det fokuserade ljuset och highlight på objektet. *Multiple scattering* är den ackurata spridningen via photon tracing. De borde vara aktiverade men kan inaktiveras för experimentering.

Algorithm control

- Depth
- Max Photons
- Max Radius

Depth avgör hur mycket spridning det sker i det yttersta kontrollerbara lagret, dvs. ju högre värde desto djupare blir det yttersta lagret. Eftersom väggen på stearinljuset är ungefär 10mm tjockt, så bör värdet vara något liknande.

Max Photons är det totala numret av fotoner som varje kaustik foton kollar på från ljuskällorna och med *Max Radius* är den maximala radien av en sfär i millimeter, vid en sample punk som samlar in fotoner. Kan experimenteras mycket med i beroende hur mycket spridning man vill se.

Extra Attributes

- Mode

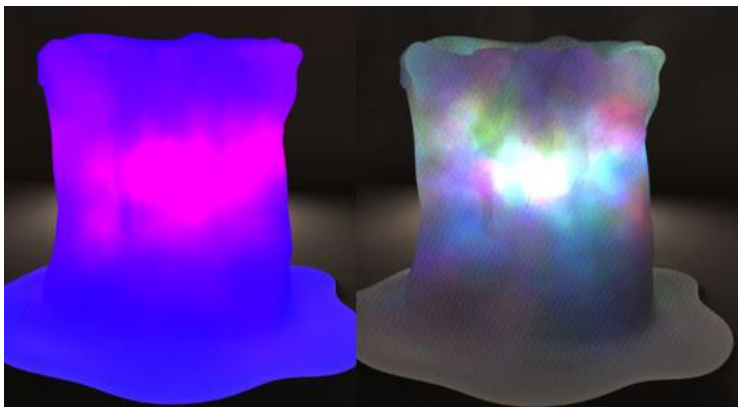
Mode hänvisar till vilket ljus sin shader ska ta emot fotoner ifrån. Därmed måste värdet vara 0 så alla ljuskällor i scenen kan ha effekt på objektet.

3.2.3 Utveckling av metod 2

Misss_physical shaders simulerar subsurface scattering på ett korrekt sätt, att försöka simulera verkligheten genom samma mekanism som händer i den naturliga världen. Dess mål är att en realistisk simulering utav subsurface scattering till skillnad från fast shaders som använder sig utav lightmaps och inte är fysiskt korrekt.

På grund av arbetets syfte och brist på tid, så användes en färdig modell från Digital Tutors. På objektet användes det bara en misss_physical shader, inga extra texturer eller shaders för att förbättra resultatet. Som ljuskällor användes det tre area lights för att lysa upp scenen och en point light som eld. Anledningen till area lights istället för point lights är för att area lights skickar fotoner åt vald riktning, och point lights skickar åt alla riktningar, därmed fungerar det bäst som eld. Inom mental ray kan fotoner simuleras antingen med hjälp av global Illumination eller caustics, men i denna scen används caustics. Det beror på global illumination som kräver att fotoner studsar minst på två ytor innan de träffar objektet.

Först lades en physical shader till, sedan justerades ljuskällor så de sänder ut fotoner. När alla attribut hade default inställningar, såg renderingen ut som i figur 9. Med default är *Material* och *Transmission* i gråskala, så blir det i RGB eftersom fotonerna sänds ut med RGB-värden. Därmed ändrades de till andra färger, så effekten blir tydligare (se figur 9).



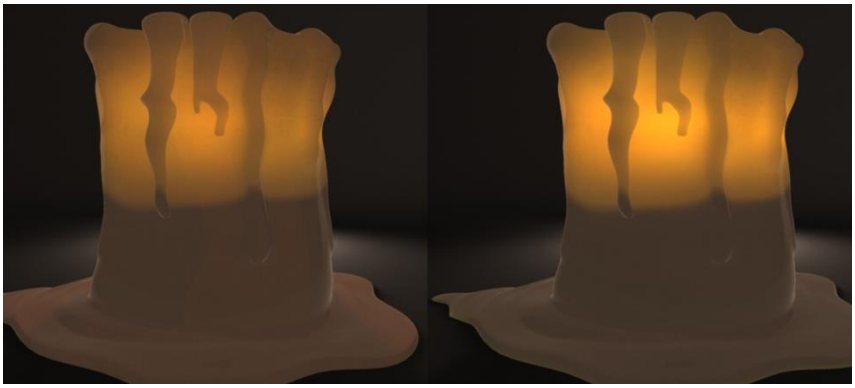
Figur 9. Färger till vänster och svartvitt till höger.

Därefter ökades foton antalet, eftersom spridningen var fläckvis. På ljuskällorna i caustic and global illumination så är default värdet för antal fotoner 10,000, och de ändrades till 500,000 för att inte öka renderingstiden drastiskt. Därav försvann en stor del av effekten och de fixades genom att öka M*ax Photons* till 8,000 i physical shader, så fler fotoner kollas upp.

Sedan sätts värdet på I*ndex of Refraction* till 1,4 eftersom de nämns i Malvern Instruments Ltd refractive index guide [15]. Men efter dessa inställningar så spred sig fotonerna för lite och därmed ökades *Max Radius* till 200, så man får känslan av eldlågan lyser upp hela övre delen av stearinljuset. Stearinljuset är modellerat som att det har brunnit halvägs ner och därmed så bör det vara en kontur vid botten i stearinljuset. För att lösa problemet, användes S*cale Conversion*. Eftersom objektet är ungefär 9 cm högt i Maya så sattes värdet till 90.

Sedan för att få utökad kontroll över spridningen ändrades djupet på det yttersta lagret med hjälp av depth attributen. Värdet är 10 eftersom väggen på stearinljuset ungefär är 10mm och med lägre värden får man en känsla av ett svagt ljus. Stearinljus är ett väldigt tranculent material som har de lätt för att sprida vidare ljus och ljuset mattar även ut färgen på materialet. För att kontrollera om det ska sprida sig framåt eller bakåt så använder man sig utav S*cattering Anisotropy*, och ett värde runt 0.500 gav en övertygande effekt av hur ljus sprider sig i vax.

Absorption och S*cattering coefficient* är komplexa attribut, där absorption är mätningen av ljuset som passerar ett grumligt medium och scattering är mätningen spridning av ljus som passerar materialet. Men absorption måste ha ett relativt lågt värde och scattering ett högt, för en bra spridning och de båda är beroende utav vilka färger som nämns i *Material* och *Transmission*. Därför gjorde jag två olika test där första testet bestod av normaliserade värden och en angiven färg i *Transmission* och de andra testet olika värden som experimenterades fram.



Figur 10. Normaliserade värden till vänster och olika värden till höger [13].

Tabell 1. De normaliserade och onormaliserade värden som används i figur 10 [13].

Normaliserade värden	Onormaliserade värden
Transmission RGB: 240, 45, 5	Transmission RGB: 36, 36, 36
Absorption coeff: 0.2, 0.2, 0.2	Absorption coeff: 0.061, 0.097, 1.450
Scattering coeff: 2.0, 2.0, 2.0	Scattering coeff: 0.180, 0.070, 0.030

Eftersom värdena i *Absorption* och *Scattering coeff* attribut är så pass känsliga, tar det lång tid att experimentera fram. Efter experimentering och misslyckande, så användes värden från Paulo Bertos dokument [13]. Till skillnad från Berto så behålls färgen i M*aterial* för att stödja färgerna från *Absorption* och *Scattering coeffcient*. *Approx Diffusion*, *Single* och *Multiple scattering* är aktiverade som default och de lämnas aktiverade. Eftersom hela metoden bygger på dessa attribut för att härma verkligheten.

I caustic and global illumination fliken på ljuskällan så *är Photon Color* vit, som i default på alla ljus och eftersom ljuskällan sänder ut fotoner så har I*ntensity* under point light attributes ingen effekt på fotonerna. Istället så finns P*hoton Intensity* som har värdet 4,000, som sänktes från 8,000 eftersom ljuskällan är så nära stearinljuset. *Exponent* behåller default värdet 2.000 eftersom, det är ungefär hur ljus avtar i verkligheten.

3.3 Metod 3 - Passrendering med misss_fast_skin_maya

I denna metod är passrendering i fokus och detta görs via misss_fast_skin_maya och eftersom det är en misss fast shader, så kommer lightmaps användas för att rendera effekten. Problemet är att misss_fast_shader_x_passes och andra mental ray material är begränsade till vilka pass som renderas ut, så måste man tvinga passen att renderas med hjälp av ett fungerande nätverk bland noder och shaders. Därav kommer diffuse, front, middle, back och specular passen renderas, för dem finns inuti misss_fast_skin_maya shader. Attribut inom misss_fast_skin_maya kommer inte förklaras utan de noder som krävs för att kunna rendera ut olika pass.

Som de andra metoderna kommer three-point light användas som ljus och subsurface scattering effekten kommer illustreras på vätska i ett glas. Både vätskan och glaset är från Digital Tutors och inga tillägg av maps finns på objekten. Glaset har en mia_material shader med hög refraction så ljusstrålar sprider sig igenom.

3.3.1 Verktygsnoder i mental ray

Den första tanken som kom upp i början av utvecklingen var hur alla lager ska delas upp, och vilka noder som behövs. Maya har ett stort antal noder som stödjer passrendering på olika sätt, men efter lite studering och experimentering så hittades två noder som är till stor hjälp.

Maya förser med verktygsnoder i create render node i hypershade, så man kan lägga till extra effekter i scenen. Till exempel kan man använda verktygen för att konvertera 2D eller 3D texturer till bump maps/normal maps, öka/minska kontrasten, och som i detta fall använda blanda färger med add/substract, multiply/divide, screen, osv.

- WriteToColorBuffer
- mib_color_mix

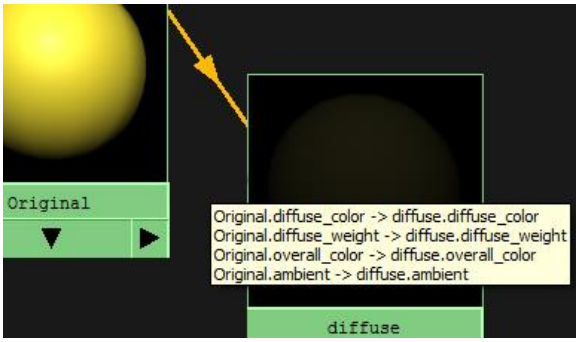
För att passrendera måste man dela upp den önskade informationen från en renderad bild till flera bilder och i denna scen kommer det finnas fem pass. Men för att göra de inom hypershader så används mental rays WriteToColorBuffer, som har syftet att fånga data i input color med RGBA värden, för sedan skriva det till en given frame buffer.

Verktygsnoden mib_color_mix accepterar en bas färg upp till 8 ingångar, men det exakta antalet bestäms av num parametern. De olika attribut noden använder är color, weight scalar och blend modes. De blend modes som finns är blend, mix, add, bounded add, multiply, bounded multiply och replace alpha. Men i detta fall kommer bara add att användas, där add är föregående resultat adderat med nuvarande färg som är multiplicerat med weight. Med weight kan man t.ex. lägga till weight maps ifall man måste förstärka subsurface scattering effekten vid önskade ställen.

3.3.2 Utveckling och förklarning av metod 3

Det har ingen betydelse vilken misss shader man använder och anledningen till att misss_fast_skin_maya används för det finns tre scattering attribut istället för två, därmed mer kontroll i slutändan.

Misss fast skin appliceras som vanligt på objektet, vätskan i detta fall och värden i materialet är inte relevant för tillfället. Efter ett fungerande nätverk så adderades fem misss_fast_skin_maya shaders som skall behandlas som de fem pass. Sedan kopplas dem alla ihop med original materialet genom connection editor i Maya och det är viktigt att bara koppla samman de värden som påverkar passet. Till exempel alla värden från original diffuse kopplas till diffuse attribut i passet (se figur 9). Sedan skall alla passen kopplas till samma MentalRayTexture som originalet använder sig utav, annars används inte samma lightmap. Efter det så kommer värden som ändras i original materialet ändras i respektive pass. I tidigare metoder så har *Screen Composite* lämnats aktiverad, men i detta fall ska *Screen Composite* inaktiveras på varje misss_fast_skin_maya eftersom bilderna sätts samman efter rendering. Övriga attribut i Algorithm control ska även vara samma för varje pass, *Scale Conversion* har värdet 1.000 eftersom allt är i rätt skala i scenen och *Falloff* 2.000.



Figur 11. Hur attribut kopplas mellan original materialet och passen.

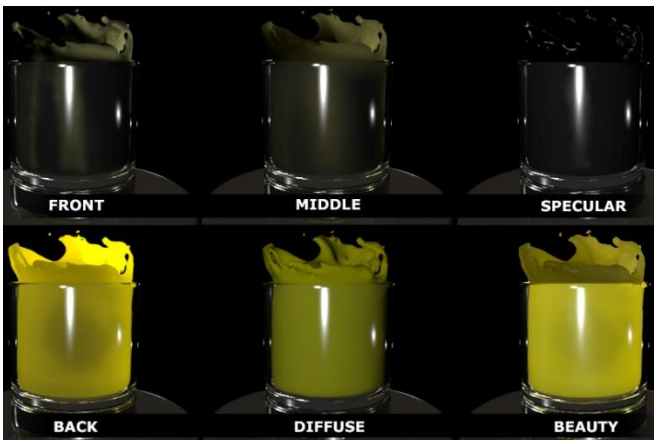
För att rendera ut passen kan man använda sig utav framebuffers. Framebuffer är mental rays sätt att göra passrendering. När man använder egna renderingspass, kan man använda writeToColorBuffer i ens nätverk som sedan skriver data till en custom color renderpass (framebuffer). Anledningen till användning utav passrendering istället för renderlayers är för att spara tid och undvika flera lager i renderlayers. Det finns även writeToDepthBuffer, writeToVectorBuffer och writeToLabelBuffer, men i detta falla används writeToColorBuffer eftersom RGBA värden skickas till custom color passes. OutValue från varje pass måste vara kopplade till varsin writeToColorBuffer via connection editor, till exempel *diffuse.outValue -> diffuse_Buffer.color*. Eftersom varje writeToColorBuffer kräver en custom color pass, så skapas de inom mental rays render settings under fliken passes. Innan custom color passen skapas måste ett renderlayer skapas med de objekten som skall vara med i scenen så color passen kan associeras med renderingslagret. Sedan när varje writeToColorBuffer har ett varsitt pass så kan passen kopplas till rätt buffer via custom color pass i writeToColorBuffer.

När man passrenderar så finns det alltid ett beauty pass, och det behövs i detta fall med. Beauty pass är all den fullfärgade renderingen av materialet, och ett beauty pass visar oftast inte några reflektioner, highlights, skuggning, spridning, osv. de visas i de separata passen. För att återskapa ett beauty pass inom nätverket så används mib_color_mix som är en nod som har 8 inputs som kan ta RGBA värden. Därmed så kopplas outValue från alla misss_fast_skin_maya materialen (passen) in i mib_color_mix, t.ex. *diffuse.outValue -> mib_color_mix1.color_0.* Därefter får alla inputs blend mode add, för att få med alla färger till beauty passet.

För att få ut beauty passet från mib_color_mix, måste noden kopplas till en shader, eftersom mib_color_mix är en nod. Vilken shader som används är inte

relevant, därmed så används en lambert. För att ett material skall fungera måste en shading group finnas . En shading group är ett shader nätverk som har ett material (t.ex. lambert) kopplad till den, som sedans kopplas till objektets mesh, därav visas materialet på objektet.

För bästa resultat så bör bilder renderas ut i HDRI(high dynamic range image). För att producera färg som verkar vara korrekt för det mänskliga ögat, måste färgkorrigering tilläggas. Därför kan man använda Mayas linear färgkorrigering innan rendering så beräkningar kan göras med rätt färgvärden och sedan färgkorrigera den resulterande bilden igen innan den visas. För att rendera ut i HDRI används Industrial Light and Magics filformat OpenEXR. För att rendera ut korrekt måste några inställningar fixas. I mental ray inställningar ska *RGBA (Float) 4x32* Bit under framebuffer och under color management ska *Enable Color Management* aktiveras. Sedan i mentalRayTexture skall *Linear sRGB* i color profile väljas. För att kunna se renderade resultatet måste *Linear sRGB* väljas i Image Color Profile och Display Color Profile väljas i render view Display > Color Management.



Figur 12. Alla passen från metod 3.

4 Resultat

I detta kapitel presenteras resultaten av varje metod som experimenterats fram under projektets gång. Varje scen startades från grunden och inga förinställda renderings inställningar eller shaders användes.

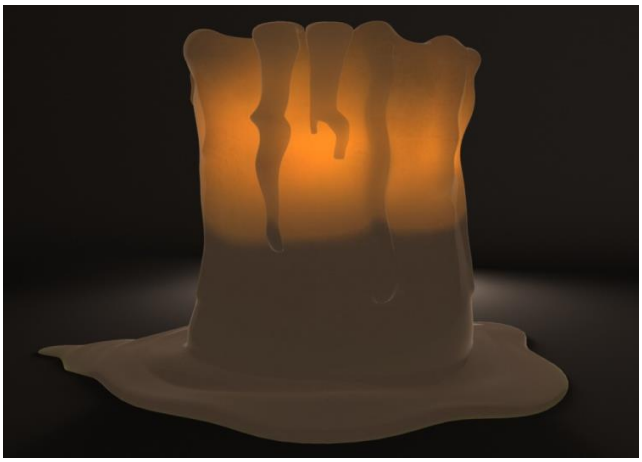
4.1 Rendering 1 – misss_fast_simple_maya

I figur 13 visas slutresultatet av metod 1 där subsurface scattering används via lightmap och renderingstiden var ungefär 1 minut.

Figur 13. Rendering av metod 1.

4.2 Rendering 2 – misss_physical

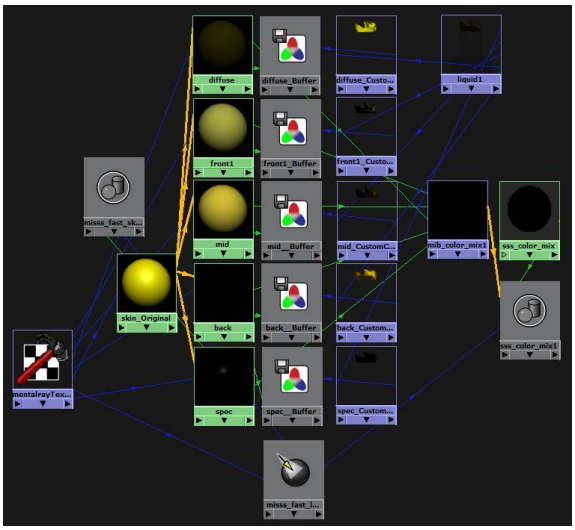
I metod 2 renderades subsurface scattering med hjälp av fotoner och slutresultatet visas i figur 14. Renderingstiden var ungefär 2 minuter.



Figur 14. Rendering av metod 2.

4.3 Rendering 3 – misss_fast_skin_maya passrendering

I figur 15 så visas det färdiga nätverket för att kunna passrendera ut viktig information från en misss shader. I figur 12 visas alla pass som renderades ut och renderingstiden är cirka 8 minuter per pass.



Figur 15. Shading nätverk för passrendering.

Alla passen lades samman i Adobe Photoshop, med extra färgkorrigering och kontrastskillnader. Det slutgiltiga resultatet visas i figur 16, som illustrerar tjock vätska så effekten syns bra.

Figur 16. Alla lager tillsammans med färgkorrigering.

5 Diskussion

Att skapa subsurface scattering med mental ray kan göras på många olika sätt och under experimentering utav dessa metoder, uppstod både positiva och negativa aspekter, som i detta avsnitt kommer presenteras.

5.1 Subsurface scattering med lightmap

Misss_fast shaders tillhandahålls som ett alternativ. Dessa shaders försöker inte simulera verkligheten. Istället är deras mål att göra ett rimligt, visuellt tilltalande, enkel att använda och med tillräckligt bra kvalité. Ett utav målen i misss_fast shaders är att tillhandahålla ett sätt att göra mänsklig hud och därmed de är mest lämpade för grund spridning. Det är mest eftersom spridningen är det inte riktigt volymetrisk och

mest lämpade för genomskinlighet av relativt tunna föremål som löv, mjuk plast, vax, vätskor, osv. För riktiga volymetriska resultat rekommenderas misss_physical.

Till skillnad från misss_physical är misss_fast shaders väldigt lätta att använda sig utav, och dess attribut är lättförståeliga och därmed är det också relativt enkelt att bygga upp scenen på ett korrekt sätt med ljus och skala. Denna shader fungerar för alla material, men som innan så är det egentligen "fake" och det är svårt att få den volymetriska känslan som misss_physical ger.

Renderingstiden för denna metod var relativt kort, därmed är användning av lightmaps inte så krävande. Därför rekommenderas misss_fast shaders när man är begränsad till lågt minne och lite tid. Utöver vad misss fast prestationsförmåga så hittades det inga nackdelar förutom om man använder flera objekt så kan inte objekten skriva till samma lightmap. Det kommer bli korrupt inom mental ray, där ofullständig eller felaktig data kan skapas. Alla attribut fungerar som de ska och de är enkla att förstå sig på. Om man är ny inom detta område bör man använda sig utav misss_fast shaders först och förstå sig på de attribut som finns innan man testar misss_physical som är mer komplex.

5.2 Subsurface scattering med photon map

Detta är användbart för scener som sker inomhus eller där fotoner kan studsa på väggar. En photon map skapar oftast inte tillräckligt bra resultat från direkt ljus, som i denna scen. Men det kan användas som grov approximation till belysningen i scenen för att påskynda beräkningen av global illumination och caustics. För bättre resultat och mer hanterligt skulle mer ytor behövas som fotoner kan studsa på.

En fördel med användning av fotoner inom mental ray är för tekniken efterliknar verkligheten, därmed bör de flesta material vara passande för misss_physical. Men misss_physical är en komplex shader att använda, därmed kan det vara svårt att uppnå vissa effekter, som till exempel hud med starkt backlight. Som visas i figur 11, där highlights skapas av ljuset bakifrån, och för att skapa något liknande med fotoner krävs många små justeringar eller användning utav en weight map som förstärker effekten. Därmed rekommenderas material som har tendensen att sprida vidare fotoner som till exempel vax, mjölk och transparenta stenar. Det underlättar även om materialet är densamma överallt, eftersom det är svårt att kontrollera hur mycket som ska spridas när väl fotonerna är inuti materialet.

En nackdel med misss_physical, är att lagring av fotoner kan strula beroende på objektets mesh. Objekten borde ha minst en ingång och utgångs yta för att kunna lagra fotoner korrekt. Till exempel ett golv i ett rum kan orsaka lagringsfel i foton mappen.

Det finns inga andra misss shaders som använder photon maps som kan förstärka resultatet av misss_physical och det fungerar bara till Polymesh och NURBS, därmed så fungerar det inte till SubDivision ytor.

Den nackdelen som hittades bland de attribut som finns är att *Absorption* och *Scattering coefficients* som tar emot fler värden än vad som visas i fältet. I fältet visar den bara tre nummer efter komma tecknet, men den kan ta fler och de värden som inte syns kan ha påverkan på slutresultatet.

Renderingstiden för detta projekt var ungefär 2 minuter, vilket är nästan dubbla renderingstiden utav metod 1. Med samma renderingsinställningar, samma storlek av maps, lika mycket sampling, osv., vilket bevisar att fotoner är mer krävande. Men resultaten från båda metoderna är ungefär densamma. Däremot skulle det vara svårt att åstadkomma samma resultat som metod 1 med photon map, fast det skulle vara simpelt att åstadkomma liknande resultat som metod 2 med lightmap.

Det visar att denna metod är den mest komplexa subsurface scattering metoden mental ray har att bidra med, därmed bör denna metod bara användas av erfarna användare. Det är också mest troligt att tekniker som denna används oftast inom produktion för långfilmer eftersom det är verklighetstroget. Det är kritiskt att veta hur

alla attribut fungerar inom misss_physical shader, eftersom alla är beroende utav varandra så kan en liten förändring ändra slutresultatet drastiskt. Det är också viktigt att ha kunskaper om hur fotoner fungerar i beroende på scenens skala och uppsättning av ljus.

5.3 Subsurface scattering passrendering med lightmap

Utveckling av ett komplext shader nätverk för att kunna rendera ut pass från en shader går att lösa på flera olika sätt. Men denna metod visar hur man gör det på ett effektivt sätt. Ett mindre effektivt sätt kan vara ifall man gav värden på de "passet" man vill rendera ut och sparar bilden, sedan byter man till nästa "pass". Men på det sättet sparas inte värden och det blir problematiskt om man måste ändra t.ex. weight eller radius värden efter rendering.

Varje pass tog ungefär 8 minuter att rendera ut, men det är på grund av att de renderas ut i HDRI. Under experimentering med noder så upptäcktes även att writeToColorBuffers tog upp mycket minne och gjorde att Maya blev långsamt, t.ex. att det tog 1-2 sekunder för Maya att reagera att man har dragit i en slider. Men det fixades genom att skriva *renderThumbnailUpdate 0* i script editor. Denna metod eller metod 2 bör användas i produktion för bästa resultat, när man har bra utrustning. Även fast man inte använder misss_physical som är fysiskt korrekt, så kan man åstadkomma samma resultat i slutändan.

En nackdel som uppstod med metoden och som hindrar effektiviteten är att det inte går att rendera ut alla passen vid samma rendering, som det vanligtvis görs med passrendering. Därmed måste önskat pass appliceras på objektet innan rendering.

5.4 Sammanfattning

Målet som förväntades fullbordades nästan. Personligen förvånades det över hur stor effekt subsurface scattering har och vad man kan åstadkomma med det. Det är många inställningar och attribut som är beroende av varandra och det krävs en bra struktur över sin shader för att åstadkomma ett bra slutresultat.

Det var några problem som uppstod i metod 2 och 3. Eftersom det är komplexa metoder så blev slutresultaten inte som förväntat, där det behövs mer tid och kunskaper för att åstadkomma ett bra resultat. Resultatet av metod 1 blev som förväntat, då attribut är relativt simpla att förstå sig på.

Några tänkbara lösningar i metod 2 skulle kunna vara annorlunda värden inom *Absorption* och *Scattering coefficients* och samtidigt att använda sig utav en weight map som skapar mer spridning så det blir ljusare längst upp på stearinljuset. Det skulle troligtvis hjälpa om man formade ljuskällan så den har samma form som eldslågan, för att få bättre spridning på fotoner.

Tänkbara lösningar för metod 3 skulle vara att ändra radius och depth värden på back passet och radius på middle, bara för att låta ljus sprida sig längre in i materialet för att få det mer transparent. I beroende på vilket material man vill illustrera så kan man byta misss_fast shader till något med mindre pass än misss_fast_skin_maya. Det går även att utveckla metoden ytterligare, som t.ex. förstärka color bleed effekten med att göra RGB värden till varsina pass.

6 Slutsatser

I detta kapitel kommer frågeställningar som tas upp i kapitel 1 svaras på.

Vilken metod är bäst lämpad att använda för subsurface scattering med mental ray?

Denna fråga har många svar, eftersom alla subsurface scattering metoder som mental ray erbjuder är både positiva och negativa i beroende på vilken uppgift som skall lösas. Användning utav lightmaps utan passrendering är bra för mindre seriösa projekt. Passrendering med lightmaps och photon maps bör användas för seriösa projekt.

På vilka sätt kan man åstadkomma subsurface scattering med mental ray?

Utöver de shaders som använder sig utav BRDF och BSSRDF, så lurar mental ray det mänskliga ögat för att illustrera subsurface scattering effekten med hjälp av lightmaps. Det bevisar också att det är möjligt att göra "fake" subsurface scattering med shaders som inte är inom misss kategorin, som t.ex. lambert.

Vad är skillnaderna mellan lightmap och photon map? Vilka fördelar och nackdelar finns det?

Photon map skickar fotoner från ljuskällor och lightmap skapas i beroende av kamerans perspektiv. De båda har för och nackdelar. Lightmaps fördelar är att det alltid kommer finnas ljus i scenen så länge det finns källor som sänder ljus, det är lätt hanterligt och det är sällan lightmap missar data från ljus. Till skillnad från lightmaps så är photon maps mer noggrant och ljus blir mer realistiskt, men nackdelen med photon maps är att den är känslig och det kan bli korrupt när den lagrar fotoner. Det blir även mer komplicerat, om det är flera ljuskällor i samma scen och det är svårt att få rätt antal fotoner i beroende med misss_physicals inställningar.

Därmed har det betydelse hur scenen är uppbyggd med. Om din scen är något komplex då är det lätt hänt att för lite fotoner når kamera. Men lightmap kommer att skjuta ljus precis där du behöver dem. Därmed tenderar lightmaps att fungera bättre för t.ex. animering.

7 Förslag på fortsatt forskning

Till fortsatt forskning så bör mental ray vidareutveckla sin förmåga att passrendera. Alla lager i sin shader bör finnas som pass i mental rays rendersettings, för att slippa tvinga ut dem genom ett shader nätverk. Utöver att se till att användarvänligheten blir bättre så effektiviteten ökas, så kan inte dessa shaders utvecklas så mycket mer. Eftersom subsurface scattering ligger i stadie då man knappt kan åstadkomma bättre resultat, allting beror på användaren själv, speciellt för fysiskt korrekta shaders.

8 Referenser

[1] K. Koutajoki, ”TKK,” [Online]. Available: http://www.tml.tkk.fi/Opinnot/Tik-111.500/2002/paperit/kalle_koutajoki.pdf. [Använd 9 3 2013].

[2] H. W. Jensen, ”A Practical Model for Subsurface Light Transport,” [Online]. Available: http://graphics.stanford.edu/~henrik/papers/bssrdf/bssrdf.pdf. [Använd 9 3 2013].

[3] F. E. Nicodemus, J. C. Richmond och J. J. Hsia, i *Geometric considerations and nomenclature for reflectance*, National Bureau of Standards (US), 1977, p. 3.

[4] B. T. Phong, ”Illumination for Computer Generated Pictures,” Communications of the ACM CACM Homepage archive, [Online]. Available: http://www.cs.northwestern.edu/~ago820/cs395/Papers/Phong_1975.pdf. [Använd 18 4 2013].

[5] A. Södergren, ”En jämförande studie mellan Mental Ray och V-Ray,” 2012. [Online]. Available: http://hig.diva-portal.org/smash/get/diva2:535749/FULLTEXT01. [Använd 15 3 2013].

[6] B. Livny, ”Mental Ray for Maya, 3ds Max and XSI: A 3D Artist's Guide to Rendering,” i *Subsurface Scattering*, Wildey Publishing. Inc, 2008, pp. 633 - 713.

[7] E. Keller, ”Mastering Maya 2013,” i *Subsurface Scattering*, Autodesk Maya Press, 2013, pp. 645-651.

[8] J. Rasmusson, ”Texture Compression of Light Maps using Smooth Profile,” [Online]. Available: http://fileadmin.cs.lth.se/graphics/research/papers/2010/lmc/lightMapComp.pdf. [Använd 10 3 2013].

[9] J. Mclnnes, ”Subsurface Scattering,” 18 5 2012. [Online]. Available: http://jordanmcinnes.com/category/renderinglighting/. [Använd 9 3 2013].

[10] P. Peers, ”A Compact Factored Representation of Hetrogeneous Subsurface Scattering,” 2006. [Online]. Available: http://people.csail.mit.edu/wojciech/CompactBSSRDF/index.html. [Använd 15 3 2013].

[11] D. Lischinski, ”Photon Maps,” [Online]. Available: http://www.cs.huji.ac.il/~danix/advanced/PhotonMaps.pdf. [Använd 3 5 2013].

[12] Autodesk, ”Render Pass Concepts and Techniques,” 2010. [Online]. Available: http://images.autodesk.com/apac_korea_main/files/maya_render_pass_concepts_and_techniques_whitepaper_us.pdf. [Använd 15 3 2013].

[13] P. Berto, ”mental ray physical subsurface scattering tutorials,” mental images GmbH, Berlin, Germany., 2004. [Online]. Available: http://www.lamrug.org/resources/doc/sss-physical-tutorial.pdf. [Använd 1 5 2013].

[14] L. Lanier, ”Using HDR Images with Global Illumination,” i *Advanved Maya Texturing and Lightning*, Autodesk Maya Press, 2006, p. 375.

[15] MAN 0079, ”Sample dispersion & refractive index guide,” Malvern Instruments Ltd, 1997. [Online]. Available: http://www.atomikateknik.com/pdf/Sample%20Dispersion%20&%20RI%20Guide.pdf. [Använd 29 4 2013].