



**THM**

TECHNISCHE HOCHSCHULE MITTELHESSEN

**CAMPUS  
FRIEDBERG**

**IEM**

Informationstechnik-  
Elektrotechnik-Mechatronik

# **Analyse unterschiedlicher Methoden zur Visualisierung von Subsurface Scattering in fotorealistischen Renderings**

Studiengang Medieninformatik

## **Masterarbeit**

vorgelegt von

**Natalie Göpfert**

geb. in London - England

durchgeführt an der  
Technischen Hochschule Mittelhessen, Friedberg

Referent der Arbeit: Prof. Dr. Cornelius Malerczyk  
Korreferent der Arbeit: Prof. Dr. Oliver Hein

Fachbereich

Informationstechnik-Elektrotechnik-Mechatronik IEM

Friedberg, 2016



*Für meine geliebte Großmutter, Inge Taubert. - I believe love is safe  
from the mayhem of the mind, because it lives in our hearts! -  
(Inspired by Still Alice)*



# Danksagung

An dieser Stelle möchte ich mich von ganzem Herzen bei den Menschen bedanken, die einen besonderen Beitrag zum Gelingen dieser Arbeit geleistet haben.

An erster Stelle danke ich meinem Freund, Hans Christian Arlt, dafür, dass er mir in der gesamten Zeit der Erstellung dieser Arbeit stets zur Seite gestanden und mich immer bestmöglich unterstützt und ermutigt hat.

Prof. Dr. Cornelius Malerczyk gilt ein besonderer Dank nicht nur für die Betreuung dieser Arbeit, sondern auch dafür, dass er meine Leidenschaft für die 3D-Computergrafik geweckt und mich sowohl während meines Bachelor- als auch meines Masterstudiums stets gefordert und gefördert hat.

Des Weiteren danke ich Prof. Dr. Oliver Hein herzlich für die Übernahme des Korreferates dieser Arbeit.

Meinen Eltern möchte ich von Herzen danken, dass sie mich immer unterstützen. Sie mir aber auch stets den nötigen Freiraum und die Unabhängigkeit gegeben haben, meinen eigenen Weg zu finden.

Anne und Joel Brage danke ich für viele aufmunternde, ermutigende und inspirierende Gespräche.

Hartmut und Johanna Arlt danke ich für das Korrekturlesen dieser Arbeit.

Daniela Dohrmehl danke ich dafür, dass sie immer da ist und außerdem für das Korrekturlesen dieser Arbeit. Meinem besten Freund Julian Langenhagen danke ich für ein offenes Ohr, wenn ich mal nicht weiter wusste. All meinen, nicht explizit genannten, lieben Freunden möchte ich für ihr großen Verständnis danken, dass sie in den letzten Monaten so wenig von mir gehört und gesehen haben.

Ein Dank gilt selbstverständlich auch all denen, die an der im Rahmen dieser Arbeit durchgeführten Evaluation teilgenommen haben.



# Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die eingereichte Masterarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Friedberg, März 2016

Natalie Göpfert



# Inhaltsverzeichnis

Danksagung	iii
Selbstständigkeitserklärung	v
Inhaltsverzeichnis	vii
Abbildungsverzeichnis	xiii
Listings	xix
Tabellenverzeichnis	xix
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problemstellung . . . . .	4
1.3 Zielsetzung . . . . .	8
1.4 Organisation der Arbeit . . . . .	9
1.5 Zusammenfassung der wichtigsten Ergebnisse . . . . .	10
<b>2 Stand der Technik</b>	<b>13</b>
2.1 Einleitung . . . . .	13
2.2 Digital Humans . . . . .	14
2.2.1 Kritische Betrachtungen des Uncanny Valley-Konzeptes . . . . .	17
2.2.2 Das Uncanny Valley-Konzept aus unterschiedlichen Blickwinkeln . . . . .	21
Psychologische Betrachtung des Uncanny Valley-Konzeptes . . . . .	21
Die Wahrnehmung des Uncanny Valley von Kindern . . . . .	22
Affen und das Uncanny Valley . . . . .	23
2.2.3 Digital Human League . . . . .	23
2.2.4 Realistic Human Faces Rendering - Matrix . . . . .	26
2.3 Stand der Technik in Bezug auf Subsurface Scattering . . . . .	27
2.3.1 Methoden zur Simulation von Subsurface Scattering . . . . .	28
2.3.2 Subsurface Scattering bei interaktiven beziehungsweise deformierbaren Objekten . . . . .	31
2.3.3 Subsurface Scattering in Bezug auf Multilayer-Materialien . . . . .	33

2.3.4	Reproduktion von Materialien mit Subsurface Scattering mittels eines 3D-Druckers . . . . .	35
2.3.5	Vergleich unterschiedlicher Rendersisteme in Bezug auf Subsurface Scattering . . . . .	36
2.4	Zusammenfassung . . . . .	39
<b>3</b>	<b>Grundlagen</b>	<b>41</b>
3.1	Einleitung . . . . .	41
3.2	Opazität und Transluzenz . . . . .	41
3.3	BRDF und BSSRDF . . . . .	42
3.3.1	BRDF . . . . .	42
3.3.2	BSSRDF . . . . .	44
3.3.3	Vorschlag eines erweiterten Visualisierungskonzeptes . . . . .	44
3.4	Der VRayMtl Shader . . . . .	46
3.4.1	Subsurface scattering Parameter . . . . .	48
	Translucency Color . . . . .	48
	Subdivs . . . . .	49
	Forward Backward Coefficient . . . . .	49
	Scatter bounces . . . . .	50
	Scatter coefficient . . . . .	51
	Thickness . . . . .	51
	Environment fog . . . . .	52
3.4.2	Andere Paramter des Shaders die den Subsurface Scattering-Effekt beeinflussen . . . . .	52
	Refraction Color . . . . .	52
	Refraction Amount . . . . .	53
	Refraction Glossiness . . . . .	54
	Refraction IOR . . . . .	54
	Fog Color . . . . .	54
	Fog multipilier . . . . .	55
	Fog bias . . . . .	56
3.5	Der VRayFastSSS2Mtl-Shader . . . . .	57
3.5.1	General Parameters . . . . .	58
	Scale . . . . .	59
	Index of Refraction . . . . .	60
3.5.2	Diffuse and SSS layers . . . . .	61
	Overall color . . . . .	61
	Diffuse color, Diffuse Amount und Subsurface Color . . . . .	62
	Scatter color . . . . .	63
	Scatter Radius . . . . .	64
	Phase function . . . . .	65
3.5.3	Specular layer . . . . .	66
	Specular color . . . . .	66
	Specular Amount . . . . .	66

	Glossiness . . . . .	66
	Specular Subdivisions . . . . .	67
3.5.4	Options . . . . .	67
	Single Scatter Mode . . . . .	67
	Single Scatter Subdivisions . . . . .	68
	Refraction Depth . . . . .	68
	Front Lighting and Back Lighting . . . . .	68
	Scatter GI . . . . .	69
	Prepass blur und Interpolation accuracy . . . . .	69
3.5.5	Multiple Scattering Options . . . . .	69
3.5.6	Bump and Normal Mapping . . . . .	69
3.6	Das V-Ray Skin Material . . . . .	70
3.6.1	General Parameters . . . . .	72
	Scale . . . . .	72
	Max SSS Amount . . . . .	73
	Max Reflection Amount . . . . .	73
	Opacity . . . . .	74
3.6.2	Diffuse . . . . .	75
	Diffuse Color . . . . .	75
	Diffuse Amount . . . . .	75
3.6.3	Shallow, Medium und Deep Scattering Layer . . . . .	76
3.6.4	Primary und Secondary Reflection . . . . .	79
	Color . . . . .	80
	Amount . . . . .	80
	Glossiness . . . . .	80
	Subdivs . . . . .	80
	Fresnel . . . . .	80
	Fresnel IOR . . . . .	80
3.6.5	Multiple Scattering Options . . . . .	81
3.6.6	Options . . . . .	81
	Trace Reflections . . . . .	81
	Max Reflection Depth . . . . .	81
	Reflection Cutoff . . . . .	81
3.6.7	Bump and Normal Mapping . . . . .	81
3.7	Zusammenfassung . . . . .	81
<b>4</b>	<b>Analyse des V-Ray Materials und des FastSSS2-Shaders für die Visualisierung von Subsurface Scattering</b>	<b>83</b>
4.1	Einleitung . . . . .	83
4.2	Exemplarische Implementierung einer Kerzen-Szene mit Subsurface Scattering	84
4.2.1	Modell und Szenenaufbau . . . . .	85
	Die Flamme . . . . .	87
4.2.2	Der FastSSS2-Shader . . . . .	88
4.2.3	Der V-Ray Material Shader . . . . .	91

4.2.4	Vergleich . . . . .	94
	Qualität der Ergebnisse . . . . .	94
	Handhabung der Shader . . . . .	95
	Renderzeiten . . . . .	95
4.3	Exemplarische Implementierung einer Marmordrachen-Szene mit Subsurface Scattering . . . . .	97
4.3.1	Modell und Szenenaufbau . . . . .	98
4.3.2	Der FastSSS2 Shader . . . . .	99
4.3.3	Der V-Ray Material Shader . . . . .	103
4.3.4	Vergleich . . . . .	105
	Qualität der Ergebnisse . . . . .	105
	Handhabung . . . . .	106
	Renderzeiten . . . . .	106
4.4	Zusammenfassung und Ergebnis . . . . .	108
<b>5</b>	<b>Analyse unterschiedlicher Methoden für die Visualisierung von Subsurface Scattering für das Rendering menschlicher Haut</b>	<b>111</b>
5.1	Einleitung . . . . .	111
5.2	Implementierung einer Szene mit einem menschlichen Gesicht mit Subsurface Scattering . . . . .	112
5.2.1	Modell und Szenenaufbau . . . . .	113
5.2.2	Der FastSSS2-Shader . . . . .	116
5.2.3	Das Skin Material . . . . .	118
5.2.4	Vergleich . . . . .	125
	Qualität der Ergebnisse . . . . .	125
	Handhabung der Shader . . . . .	127
	Renderzeiten . . . . .	128
5.3	Implementierung einer Szene mit einer menschlichen Hand mit Subsurface Scattering . . . . .	129
5.3.1	Modell und Szenenaufbau . . . . .	130
5.3.2	Der FastSSS2-Shader . . . . .	132
5.3.3	Das Skin Material . . . . .	133
5.3.4	Vergleich . . . . .	135
	Qualität der Ergebnisse . . . . .	135
	Handhabung der Shader . . . . .	135
	Renderzeiten . . . . .	136
5.4	Zusammenfassung und Ergebnis . . . . .	137
<b>6</b>	<b>Konzept und Entwicklung eines Plugins für Hautrendering</b>	<b>139</b>
6.1	Einleitung . . . . .	139
6.2	Konzept und Idee . . . . .	139
6.3	Maya Embedded Language-Skripte . . . . .	141
6.4	Installation . . . . .	142
6.5	Vorbereitung einer Szene . . . . .	143

6.6	Implementierung . . . . .	144
6.7	Aufbau des Skriptes . . . . .	146
6.8	Zusammenfassung . . . . .	150
<b>7</b>	<b>Evaluation</b>	<b>151</b>
7.1	Einleitung . . . . .	151
7.2	Konzept . . . . .	151
7.3	Statistische Auswertung . . . . .	157
7.3.1	Verwendete Software zur statistischen Analyse . . . . .	157
7.3.2	Erkennungsrate . . . . .	157
7.3.3	Gegenüberstellung V-RayMtl und FastSS2 . . . . .	160
7.3.4	Einfluss der CG-Vorkenntnisse . . . . .	164
7.3.5	Gegenüberstellung FastSS2 und Skin Material . . . . .	166
7.3.6	Test auf Fotorealismus . . . . .	170
7.3.7	Direkter Vergleich . . . . .	175
7.3.8	Drachen . . . . .	176
7.3.9	Kerzen . . . . .	177
7.3.10	Emily-Kopf . . . . .	178
7.3.11	Hand . . . . .	179
7.4	Zusammenfassung und Interpretation der Ergebnisse . . . . .	180
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>183</b>
8.1	Zusammenfassung . . . . .	183
8.2	Ausblick . . . . .	185
<b>A</b>		<b>187</b>
	<b>Literaturverzeichnis</b>	<b>197</b>



# Abbildungsverzeichnis

1.1	Beispiele für das Auftreten von Subsurface Scattering . . . . .	1
1.2	Grafik zur Darstellung der Reflektionen bei Subsurface Scattering . . . . .	2
1.3	Weitere Beispiele für Subsurface Scattering . . . . .	3
1.4	Vergleich von Foto gegen Rendering in der Produktwerbung . . . . .	4
1.5	Beispiel für Uncanny Valley-Verlauf . . . . .	6
1.6	Beispiele für das Uncanny Valley in Filmen . . . . .	7
1.7	Vergleich von einem Haut-Rendering mit und ohne SSS . . . . .	8
2.1	Breaking Bad Beispiel . . . . .	15
2.2	Graph zur Beschreibung des Uncanny Valley. . . . .	17
2.3	Hiroshi Ishiguro Experiment . . . . .	20
2.4	Emily 2.0 . . . . .	24
2.5	Kollage Digital Paul . . . . .	25
2.6	Realistic Human Face-Rendering im Kinofilm "Matrix" . . . . .	26
2.7	Gegenüberstellung direkter Reflektion und Subsurface Scattering . . . . .	28
2.8	Schaubild Dipole-Punktquellen-Approximation . . . . .	30
2.9	Gegenüberstellung von unterschiedlichen Algorithmen . . . . .	30
2.10	Versuch Aufbau und Messwerte von Jensen et al. . . . .	31
2.11	Abbildung des Workflow von Lensch et al. . . . .	31
2.12	Renderingvergleich von Hao et al. . . . .	32
2.13	Beispielrendering aus Mertens et al. . . . .	33
2.14	Vergleich des Dipole-Modells mit dem Multipole-Modell . . . . .	34
2.15	Skizze der Funktionsweise des Modells von Donner et al. . . . .	34
2.16	Aufbau zur Aufnahme multispektraler Fotografien . . . . .	35
2.17	Unterschiedliche Layer aus Donner et al. . . . .	35
2.18	Skizze des Verfahrens zur Reproduktion von SSS-Materialien mittels 3D-Drucker	36
2.19	Beispiel für die Zuordnung des CWSSIM . . . . .	37
2.20	Ergebnisse des Renderervergleichs . . . . .	38
3.1	Translucency vs Transparency . . . . .	42
3.2	Eigener Vorschlag zum erweiterten Visualisierungskonzept . . . . .	45
3.3	VRay Material Shader Attribute . . . . .	46
3.4	VRay Material Szenenaufbau . . . . .	47

3.5	VRay Material Vorbereitungen auf Subsurface Scattering . . . . .	48
3.6	VRay Material Translucency Color . . . . .	49
3.7	VRay Material Forward Backward Coeffizient . . . . .	50
3.8	VRay Material Scatter Bounces . . . . .	50
3.9	VRay Material Scatter Coefficient . . . . .	51
3.10	VRay Material Thickness . . . . .	52
3.11	VRay Material Refraction Color Graustufen . . . . .	53
3.12	VRay Material Refraction Color . . . . .	53
3.13	VRay Material Refraction Amount . . . . .	54
3.14	VRay Material Fog Color mit unterschiedlichen Graustufen . . . . .	55
3.15	VRay Material Fog Color mit unterschiedlichen Farbwerte . . . . .	55
3.16	VRay Material Fog Multiplier . . . . .	56
3.17	VRay Material Fog Bias . . . . .	56
3.18	VRayFastSSS2 Shader Attribute . . . . .	57
3.19	VRayFastSSS Szenen-Aufbau . . . . .	58
3.20	VRayFastSSS Übersicht aller Presets . . . . .	59
3.21	VRayFastSSS Scale . . . . .	60
3.22	VRayFastSSS IOR . . . . .	61
3.23	VRayFastSSS Overall Color . . . . .	62
3.24	VRayFastSSS Diffuse Amount . . . . .	63
3.25	VRayFastSSS Scatter Color . . . . .	64
3.26	VRayFastSSS Scatter Colort . . . . .	64
3.27	VRayFastSSS Scatter Radius . . . . .	65
3.28	VRayFastSSS Phase Function . . . . .	66
3.29	VRayFastSSS Glossiness . . . . .	67
3.30	Single Scatter Modes . . . . .	68
3.31	VRaySkin ShaderAttribute . . . . .	70
3.32	V-Ray Skin Material Szenenaufbau . . . . .	71
3.33	V-Ray Skin Material Szenenaufbau . . . . .	72
3.34	VRaySkin Scale . . . . .	72
3.35	VRaySkin Max SSS Amount . . . . .	73
3.36	VRaySkin Max Reflection Amount . . . . .	74
3.37	VRaySkin Oppacity . . . . .	75
3.38	VRaySkin Diffuse Color . . . . .	75
3.39	VRaySkin Diffuse Amount . . . . .	76
3.40	VRaySkin Shallow Scattering Radius . . . . .	77
3.41	VRaySkin Medium Scattering Radius . . . . .	77
3.42	VRaySkin Deep Scattering Radius . . . . .	78
3.43	VRaySkin Shallow Scattering Amount . . . . .	78
3.44	VRaySkin Medium Scattering Amount . . . . .	79
3.45	VRaySkin Deep Scattering Amount . . . . .	79
3.46	VRaySkin Primary Reflection Glossiness . . . . .	80
4.1	Kerzen Subsurface Scattering-Referenzen . . . . .	84

---

4.2	Kerzenmodell . . . . .	85
4.3	Szenenaufbau der Kerzenszene . . . . .	86
4.4	Positionierung der zusätzlichen Lichtquelle für den SSS-Effekt . . . . .	86
4.5	Maya Fluid Box . . . . .	87
4.6	Der finale Kerzen-Shader in der FastSS2-Variante auf dem Shader Template .	88
4.7	Attribute des FastSS2 Kerzen Shaders . . . . .	89
4.8	Ergebnis-Rendering der Kerze mit dem FastSS2 Shader . . . . .	90
4.9	Zusätzliche Lichtquelle für den Durchleuchte-Effekt beim VRay Material . . . .	91
4.10	Der finale Kerzen-Shader in der V-Ray Material-Variante auf dem Shader Template	91
4.11	Attribute des V-Ray Material Kerzen Shaders . . . . .	92
4.12	Ergebnis-Rendering der Kerze mit dem V-Ray material Shader . . . . .	93
4.13	Candle Compare . . . . .	94
4.14	Sample Rates für die Kerzenrenderings . . . . .	96
4.15	Marble Reference-Board . . . . .	97
4.16	Das Drachen-Modell . . . . .	98
4.17	Light-Set Up der Drachen Szene . . . . .	98
4.18	Marmor Textur im UV-Editor und Original . . . . .	99
4.19	Der finale Marmor-Shader in der FastSS2-Variante auf dem Shader Template	100
4.20	Marmor FastSS Shader Attribute . . . . .	101
4.21	FastSS2 Marmor Shader Netzwerk . . . . .	102
4.22	Rendering des Marmor-Drachens mit dem FastSS2-Shader . . . . .	102
4.23	Der finale Marmor-Shader in der VRay-Material-Variante auf dem Shader Template	103
4.24	Marmor VRay Material Shader Attribute . . . . .	104
4.25	Final Rendering des Marmor Drachens in der V-Ray-Material Variante . . . . .	105
4.26	Dragon Compare . . . . .	106
4.27	Vergleich der Sample Raten des Marmordrachen-Renderings . . . . .	108
5.1	Beispiele für den leuchtende Ohren in Serien . . . . .	112
5.2	Modell der Emily O Brien . . . . .	113
5.3	Die drei Haupttexturen des Emily-Models . . . . .	114
5.4	Die Displacement-Texturen des Emily-Models . . . . .	114
5.5	Die Augen-Texturen des Emily-Models . . . . .	114
5.6	Szenenaufbau Emily-Szene . . . . .	115
5.7	Lightsetup Emily-Szene . . . . .	115
5.8	Der finale Kopf-Shader in der FastSS2-Variante auf dem Shader Template . .	116
5.9	Attribute des Gesicht-Hautshaders in der FastSS2-Variante . . . . .	117
5.10	Emily Rendering Final FastSS2-Variante . . . . .	118
5.11	Proof of Concept Skin Material . . . . .	119
5.12	Der finale Kopf-Shader in der SkinMtl-Variante auf dem Shader Template . . .	120
5.13	Nachweis für die komplette Rotfärbung . . . . .	121
5.14	Maske und Textur für das glühende Ohr im Skin Material . . . . .	122
5.15	Raytraced Scatter Textures . . . . .	122
5.16	Attribute des Gesicht-Hautshaders in der Skin Material-Variante . . . . .	124
5.17	Emily Rendering Final Skin Material-Variante . . . . .	125

5.18	Emily Compare . . . . .	126
5.19	Vergleich Ohr Deuchleuchten Foto und Renderings . . . . .	127
5.20	Sample Rates Emily . . . . .	129
5.21	Handmodell . . . . .	130
5.22	Texturen für den Handshader . . . . .	131
5.23	Szenenaufbau der Handszene . . . . .	131
5.24	Attribute des Handshaders in der FastSSS2-Variante . . . . .	132
5.25	Finales Handrendering in der FastSSS2 Shader Variante . . . . .	133
5.26	Shadernetzwerk des Skin Materials für das Handmodell . . . . .	134
5.27	Maske und rot eingefärbte Textur für die Handtextur . . . . .	134
5.28	Vergleich der Renderings mit dem Skin Material mit und ohne Maske . . . . .	135
5.29	Hand Sample Rates . . . . .	137
6.1	Slider Interface . . . . .	140
6.2	Shadernetzwerk zur Einbindung des ColorCorrection-Knoten . . . . .	140
6.3	Skript-Editor . . . . .	142
6.4	Shelf-Button . . . . .	142
6.5	Szene vorbereiten - Textur einfärben . . . . .	143
6.6	Erstellen der Maske . . . . .	144
6.7	Anlegen des ColorCorrection-Knotens . . . . .	144
6.8	Shadernetzwerk nach Einbindung des ColorCorrection-Knotens . . . . .	145
6.9	Ergebnis Slider . . . . .	149
7.1	Weboberflaeche der demografischen Erhebung . . . . .	152
7.2	Weboberflaeche Fake oder Foto-Test . . . . .	153
7.3	Neue Hintergründe der Renderings für die Evaluation . . . . .	154
7.4	Kontrollbilder der Evaluation . . . . .	154
7.5	Bildsets für die Evaluation . . . . .	155
7.6	Auflösung des Fake oder Foto Tests . . . . .	156
7.7	Weboberflaeche Compare-Test . . . . .	157
7.8	Tabelle Gesamterkennungsrate . . . . .	158
7.9	Tabelle Erkennungsrate CG . . . . .	159
7.10	Tabelle Erkennungsrate Fotos . . . . .	159
7.11	Grafische Darstellung für die Erkennungsraten . . . . .	159
7.12	Kruskal-Wallis-Test für die Erkennungsraten . . . . .	160
7.13	Tabelle der Werte für den Vierfeldertest für die Marmoddrachen Renderings . . . . .	161
7.14	Visualisierung des Ergebnisses Fake oder Foto Tests für die Drachen-Renderings . . . . .	162
7.15	Ergebnis der Teststatistik für die Marmoddrachen-Renderings . . . . .	162
7.16	Tabelle der Werte für den Vierfeldertest für die Kerzen Renderings . . . . .	163
7.17	Visualisierung des Ergebnisses Fake oder Foto Tests für die Kerzen-Renderings . . . . .	163
7.18	Ergebnis der Teststatistik für die Kerzen Renderings . . . . .	164
7.19	Systat-Tabelle für das Merkmal CG-Vorkenntnisse für Marmoddrachen-Rendering . . . . .	165
7.20	Visualisierung der Ergebnisse der Tests bezüglich der Vorkenntnisse . . . . .	165
7.21	Systat Ergebnis CG Vorkenntnis-Test Marmodrache . . . . .	166

---

7.22	Systat-Datentabelle für das Emily-Rendering . . . . .	167
7.23	Visualisierung des Ergebnisses Fake oder Foto Tests für die Kopf-Renderings . .	167
7.24	Systat-Ergebnistabelle für das Emily-Rendering . . . . .	168
7.25	Systat-Datentabelle für das Hand-Rendering . . . . .	169
7.26	Visualisierung des Ergebnisses Fake oder Foto Tests für die Hand-Renderings .	169
7.27	Systat Ergebnistabelle für das Hand-Rendering . . . . .	170
7.28	Systat-Datentabelle für den Vergleich Drache und Pasta Foto . . . . .	171
7.29	Visualisierung des Ergebnisses Test auf Fotorealismus . . . . .	171
7.30	Systat-Ergebnis Vergleich Drache und Pasta Foto . . . . .	172
7.31	Systat-Datentabelle für den Vergleich Drache gegen drei reale Fotos . . . . .	173
7.32	Visualisierung des Ergebnisses Test auf Fotorealismus mit drei Fotos . . . . .	173
7.33	Systat-Ergebnis für den Mehrfelder-Test Drache gegen drei reale Fotos . . . . .	174
7.34	Übersicht der vier Vergleichstests . . . . .	175
7.35	Drachen Vergleichstest . . . . .	176
7.36	Kerze-Vergleichstest . . . . .	177
7.37	Kopf-Vergleichstest . . . . .	178
7.38	Hand-Vergleichstest . . . . .	179



# Listings

6.1	Funktion zum Starten des SSS-Interfaces . . . . .	146
6.2	Suche nach ColorCorrection-Knoten . . . . .	146
6.3	Aufruf der CreateGui-Methode mit Übergabe des CC-Variablen-Namens . .	147
6.4	Erzeugung des Interface . . . . .	147
6.5	Auslesen der aktuellen Werte für die Slider-Attribute . . . . .	148
6.6	Callback für Intensitäts-Slider . . . . .	148
6.7	Callback für Farbton-Slider . . . . .	149

# Tabellenverzeichnis

4.1	Vergleich der gemessenen Renderzeiten für die Wachs-Shader . . . . .	95
4.2	Vergleich der gemessenen Renderzeiten für die Marmor-Shader . . . . .	107
5.1	Vergleich der gemessenen Renderzeiten für die Kopf-Shader . . . . .	129
5.2	Vergleich der gemessenen Renderzeiten für die Hand-Shader . . . . .	136



# Kapitel 1

## Einleitung

### 1.1 Motivation



**Abbildung 1.1:** Unterschiedliche Beispiele für das Auftreten von *Subsurface Scattering*. Von links nach rechts sind eine brennende Kerze (<http://goo.gl/WW4x2L> Stand: 09. Oktober 2015), eine mit einer Lampe durchleuchtete menschliche Hand (<http://goo.gl/KJadUj> Stand: 09. Oktober 2015), ein lichtdurchflutetes Schweineohr (<http://goo.gl/KExNS0> Stand: 09. Oktober 2015) und eine Frauenfigur aus Wachs (<http://goo.gl/W2bmmy> Stand: 09. Oktober 2015) zu sehen.

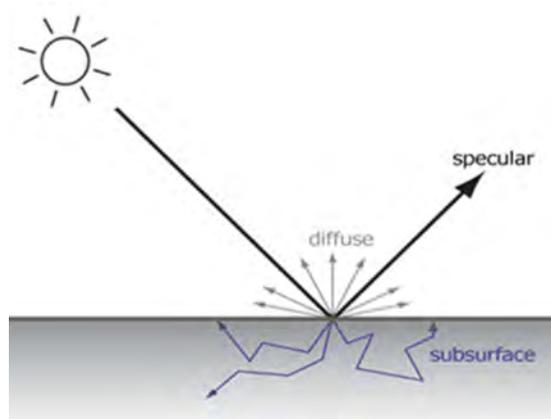
Am Computer generierte Bilder gehören in der heutigen Zeit zu unserem Alltag, sei es in Filmen, Spielen, in der Werbung oder auch in Bereichen wie der Medizin oder Architektur. Die Soft- und Hardware ist so weit entwickelt, dass in adäquater Zeit, Bilder mit einem so hohen Fotorealismus und physikalischer Plausibilität erzeugt werden können, dass selbst ein geschultes Auge sie oft kaum von realen Fotos unterscheiden kann. Oft glauben wir reale Aufnahmen zu sehen, sehen aber in Wirklichkeit vollständig oder zumindest teilweise am Computer generiertes Material. In Filmen oder in der Werbung liegen die Vorteile dessen klar auf der Hand, denn es können Aufnahmen produziert werden, deren Umsetzung so nicht möglich, zu gefährlich oder zu teuer gewesen wäre. Die Rede ist hier aber nicht nur von aufwendigen Spezialeffekten, wie wir sie alle zur Genüge aus den Action-Blockbustern im Kino kennen und bei denen sich gegebenenfalls schon zumindest erahnen lässt, dass das Gesehene vermutlich keine realen Aufnahmen sind, sondern auch von alltäglichen Gegenständen.

## 1. EINLEITUNG

---

Ein gutes Beispiel hierfür ist die Automobil-Werbung. Hier können nicht nur Kosten für aufwendige Fotoshootings und Filmaufnahmen gespart werden, sondern das Werbematerial kann mit Hilfe der Konstruktionsdaten auch bereits erstellt werden, bevor die eigentliche Produktion des entsprechenden Automobils wirklich abgeschlossen ist. In der Architektur können Gebäude visualisiert werden, bevor sie gebaut werden und in der Medizin können sowohl Trainingssoftware für die Ausbildung von Ärzten, als auch Visualisierungen für die Forschung und die bessere Veranschaulichung für Patienten am Computer generiert werden.

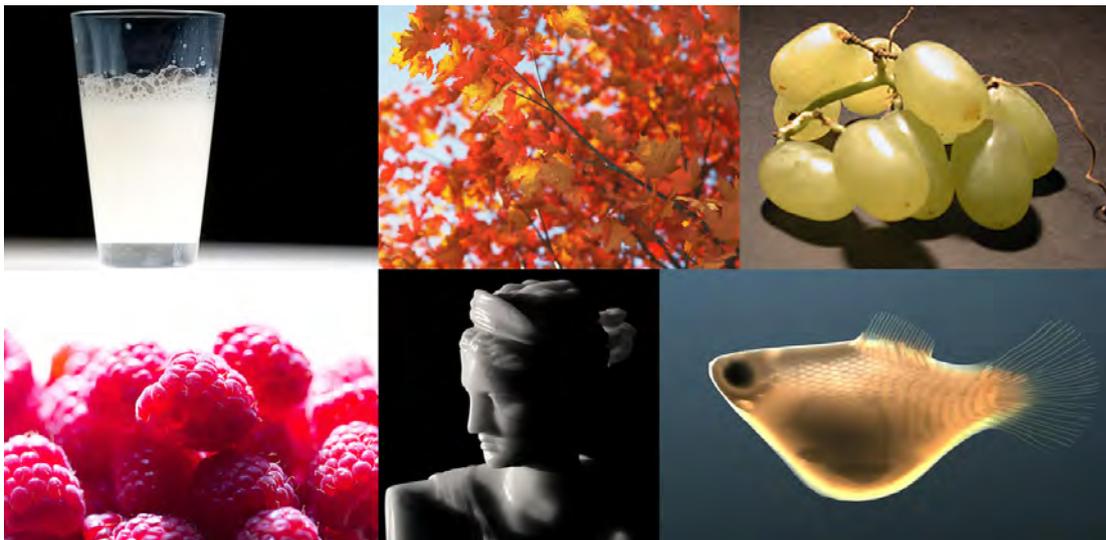
Trotz der enorm fortgeschrittenen Entwicklung im Gebiet der am Computer generierten Bilder, gibt es noch immer Bereiche in der Computergrafik, bei denen physikalisch korrekte oder auch nur plausible Darstellungen nicht trivial sind. Eines dieser Bereiche bildet das physikalische Phänomen des *Subsurface Scatterings*.



**Abbildung 1.2:** Visualisierung der Unterschiede im Verlauf der Reflektionsstrahlen bei diffuser Reflexion, spiegelnder Reflexion und *Subsurface Scattering* (<http://goo.gl/aWtKlf> Stand: 09. Oktober 2015).

*Subsurface Scattering*, zu Deutsch Volumenstreuung, ist eine jedermann bekannte Erscheinung, welche beispielsweise auftritt, wenn eine menschliche Hand vor eine Taschenlampe gehalten wird oder auch bei der Beobachtung einer brennenden Kerze zu sehen ist. Aber nicht nur dieser Effekt des "Durchleuchtens", sondern auch eine besondere Weichheit der Oberfläche, wie sie beispielsweise menschliche Haut oder Marmor aufweist, wird durch *Subsurface Scattering* erzeugt [JMLH01]. Das Phänomen ist auch als *Subsurface Light Transport* bekannt, weil es sich um eine Prozedur handelt, bei der Licht während seines Transportes in die Oberfläche eines transluzenten - also partiell lichtdurchlässigen - Materials eindringt und dort gestreut wird. Das bedeutet, dass das Licht unter der Oberfläche mit dem Material interagiert, was dazu führt, dass die Strahlen anstatt an der Oberfläche der Stelle, an der sie auftreffen, reflektiert zu werden, an einer möglicherweise ganz anderen Stelle des Objektes wieder austreten. Wichtig ist an dieser Stelle die Unterscheidung, dass transluzent lichtdurchlässig bedeutet, aber nicht mit Durchsichtigkeit gleichzusetzen ist. Dabei kann die Durchlässigkeit des Lichtes unterschiedliche Abstufungen haben, was bedeutet, dass auch

für Materialien, die auf den ersten Blick nicht wirklich lichtdurchlässig wirken, der *Subsurface Scattering*-Effekt dennoch eine große Bedeutung haben kann. Die physikalischen Reflexionsgesetze, wie sie bei spiegelnden Reflektionen gelten - Ausfallswinkel = Einfallswinkel - können hier nicht mehr angewendet werden. Da das Licht nach mehrfacher Reflektion innerhalb des Materials unter einem völlig anderen Winkel wieder austreten kann, ist das Verhalten des Lichtes unvorhersehbar. Eine Wahrscheinlichkeitsverteilung für die Austrittsstellen und Austrittswinkel des Lichtes lassen sich jedoch mit der BSSRDF, *Bidirectional Scattering Surface Reflectance Distribution Function* (zu Deutsch Bidirektionale Oberflächenstreuungs-Reflektanzverteilungsfunktion), beschreiben. Was es mit dieser Funktion auf sich hat, wird in Kapitel 3 dieser Arbeit genauer erörtert. Es gibt unterschiedliche Arten von Materialien, bei denen *Subsurface Scattering* auftritt. Beispielsweise tritt das Phänomen bei Haut auf, da hier mehrere unterschiedlich lichtdurchlässige Schichten übereinanderliegen. Bei Emulsionen, wie beispielsweise Milch, tritt der Effekt auf, weil sie aus einer Mischung von unterschiedlich stark lichtdurchlässigen Materialien bestehen, zwischen denen die Lichtstrahlen reflektiert werden. Weitere typische Beispiele, bei denen das Phänomen beobachtet werden kann, sind Marmor und Wachs.



**Abbildung 1.3:** Weitere Beispiele für das Auftreten von *Subsurface Scattering* bei Milch (<http://goo.gl/2pCTCo> Stand: 18. Dezember 2015), Blättern (<http://goo.gl/FcBb8g> Stand: 18. Dezember 2015), Trauben (<http://goo.gl/qqyc4s> Stand: 18. Dezember 2015), Himbeeren (<http://goo.gl/UMZryw> Stand: 18. Dezember 2015), Marmor (<http://goo.gl/UMZryw> Stand: 18. Dezember 2015) und Fischen (<http://goo.gl/aYaxMd> Stand: 18. Dezember 2015). Einige der Beispiele zeigen sehr gut, dass sich *Subsurface Scattering* nicht immer in dem damit verbundenen oft typischen Effekt des Durchleuchtens, sondern auch vor allem in einer besonderen Weichheit der Oberfläche zeigt, wie zum Beispiel bei den Trauben oder dem Marmor.

In der 3D-Computergrafik ist die Visualisierung dieses physikalischen Phänomens essentiell, vor allem wenn es um fotorealistic Renderings geht. Da ein Fehlen dieser Erscheinung an Stellen, an denen sie in der Realität auftreten würde, dem Betrachter, wenn vielleicht auch nicht sofort korrekt definierbar, direkt als Störung ins Auge fallen würde und das Bild als ein am Computer generiertes entlarvt werden würde.

### 1.2 Problemstellung

Das Rendern von möglichst fotorealisticen und physikalisch plausiblen Bildern ist seit langem ein großes Thema in der 3D-Computergrafik, sodass der Fortschritt für Produktionen dieser Art mittlerweile auf einem enorm hohen Level ist und täuschend echte Bilder erzeugt werden können. Dennoch ist dies ein bei weitem nicht ausgeschöpftes Forschungsgebiet und einige Bereiche der computergrafischen Visualisierung bedürfen noch immer einiger Entwicklung, wie im Folgenden deutlich wird.



**Abbildung 1.4:** Bilder einer Marketing-Kampagne der Produktreihe Gliss Kur von Schwarzkopf (<http://goo.gl/153yGn> Stand: 18. Dezember 2015). Beim rechten Bild handelt es sich um ein Foto, beim linken um ein am Computer generiertes Rendering. Dieser Vergleich zeigt auf beeindruckende Weise, wie verblüffend echt 3D-generiertes Bildmaterial in der heutigen Zeit produziert werden kann.

Wie bereits zuvor in diesem Kapitel beschrieben, ist das physikalische Phänomen des *Sub-surface Scatterings* ein für fotorealistic Umsetzungen essentieller Effekt. Um die Handhabung zu erleichtern und eine möglichst hohe physikalische Korrektheit zu gewährleisten, wird in der 3D-Computergrafik häufig versucht, die reale Welt in der Software so gut wie möglich nachzubilden. Renderer versuchen Licht, Kamera und Oberflächen mit physikalisch nachvollziehbaren Attributen zu definieren, die dann genutzt werden können, um in Realität gemessene Werte oder auch beispielsweise in physikalischen Tabellen nachschlagbare Werte einzutragen und dann auch das entsprechend plausible Ergebnis zu erhalten. Ein präde-

stiniertes Beispiel hierfür sind die Lichteinstellungen in *Chaosgroup's Renderer V-Ray*<sup>1</sup>, bei denen die Lichter so naturgetreu wie möglich eingestellt werden können sollen und dafür beispielsweise physikalische Einheiten wie Watt oder Lumen in der Software angegeben werden können, um die Lichtintensität zu bestimmen. Zum anderen zeigt auch die *Physical Camera*, ebenfalls aus dem *Renderer V-Ray*, dass mit möglichst wirklichkeitsnahen Einstellungen der Kamera in der 3D-Szene, wie zum Beispiel realistischen Werten für ISO, Brennweite und Belichtungszeiten, eine möglichst einfache Bedienung und physikalische Korrektheit erreicht werden sollen.

Beim *Subsurface Scattering* ist eine derart physikalisch korrekte Umsetzung in der 3D-Szene allerdings nicht möglich, da für die Entstehung dieses Effektes, wie bereits zuvor in diesem Kapitel beschrieben, die Zerstreuung der Lichtstrahlen innerhalb des Volumens eines Materials entscheidend ist. In 3D-Softwarepaketen wie *Autodesk Maya* werden die Materialeigenschaften den Objekten in der Szene aber lediglich als Shader der Objektoberfläche zugewiesen und somit besitzt das Material an sich kein Volumen, in dem das Licht gestreut werden könnte. Das bedeutet, dass die Erscheinung des *Subsurface Scatterings* in 3D-Szenen möglichst realistisch simuliert werden muss, um zumindest physikalisch plausible Ergebnisse zur erzielen.

Dennoch scheint auch für Umsetzungen bezüglich der Visualisierung von *Subsurface Scattering* *V-Ray* als *Renderer* die beste Wahl, da hier unterschiedliche Ansätze zur Realisierung des Effektes des *Subsurface Scatterings* zur Verfügung gestellt werden. Zum einen gibt es Möglichkeiten im Standard-*V-Ray*-Shader, dem *VRayMtl*, die Berechnung einer *Subsurface Scattering*-Simulation zu aktivieren und Einstellungen vorzunehmen. Zum anderen gibt es unterschiedliche speziell auf *Subsurface Scattering* abgestimmte Shader, den *FastSSS2*-Shader und einen speziellen *Skin*-Shader, der es ermöglicht, die unterschiedlichen Hautschichten menschlicher Haut zu simulieren. Auch das *Wikuhuman*-Projekt<sup>2</sup> der *Digital Human League*<sup>3</sup>, auf beide wird im Kapitel 2 genauer eingegangen, setzt für ihre Forschung bezüglich *Digital Humans* auf *V-Ray*.

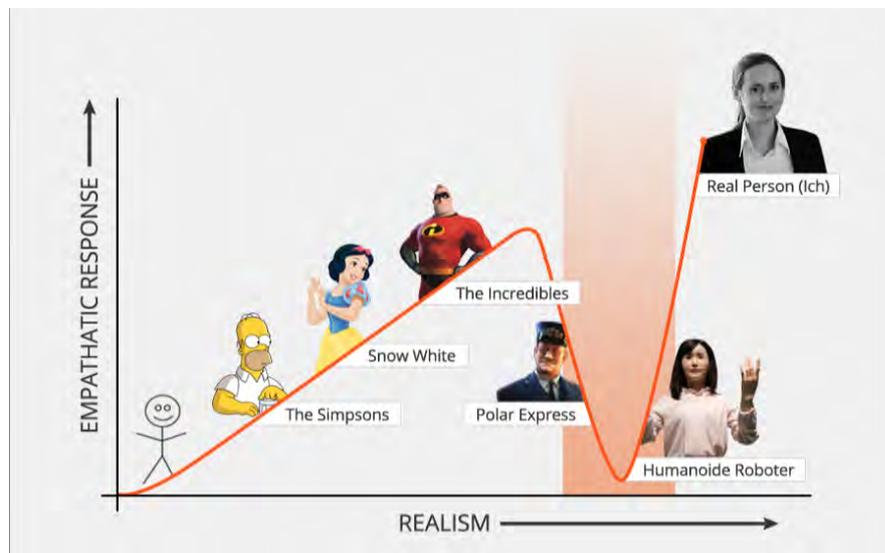
Während es schon schwierig ist überhaupt brauchbare Aufzeichnungen zu der richtigen Anwendung und Funktionsweise der einzelnen Shader zu finden, existieren bisher keinerlei Dokumentationen darüber, wie sich die unterschiedlichen Ansätze in ihrer Handhabung, in Renderzeiten und auch in der Qualität ihrer jeweiligen Ergebnisse, vor allem in Hinblick auf ihre physikalische Plausibilität, unterscheiden.

---

<sup>1</sup>[http://www.chaosgroup.com/en/2/vray\\_maya.html](http://www.chaosgroup.com/en/2/vray_maya.html) Stand: 08.Februar. 2016

<sup>2</sup><http://www.wikihuman.org> Stand: 08. Februar 2016

<sup>3</sup><https://labs.chaosgroup.com/index.php/category/digital-human-league/> Stand: 08. Februar 2016



**Abbildung 1.5:** Beispiel für den Verlauf des Akzeptanzabfalls und -anstiegs in Bezug auf das sogenannte "Uncanny Valley" (<http://goo.gl/oA9rVQ> Stand: 09. Oktober 2015). Die Kurve mit den verschiedenen Charakteren zeigt anschaulich, dass die Akzeptanz zunächst bei zunehmendem Realismus und größerer Annäherung an die Menschlichkeit ansteigt, dann aber ab einem bestimmten Level rapide abfällt und erst bei der Darstellung eines realen Menschen wieder ansteigt.

Eine der größten Herausforderungen in der Computergrafik bleibt bis heute außerdem die Darstellung von menschlichen Charakteren. Es ist ausgesprochen schwierig, wenn nicht bisher unmöglich, einen digitalen Menschen so darzustellen, dass er nicht ins sogenannte "Uncanny Valley" [MMK12] fällt. Das *Uncanny Valley*, zu Deutsch unheimliches Tal, beschreibt das Phänomen, dass die Akzeptanz des Betrachters ab einem bestimmten Grad der Menschlichkeit von Charakteren schlagartig abfällt, da der digitale Charakter zwar einem Menschen sehr ähnelt, aber nicht menschlich genug ist. Dies führt dazu, dass der Zuschauer den digitalen Menschen als eher unheimlich empfindet und die Akzeptanz daher sinkt. Ursprünglich kam der Begriff aus der Robotik und entstand mit der Entwicklung immer menschenähnlicheren Robotern. Heute wird das Uncanny Valley oft als Begründung genutzt, warum ein Animationsfilm beim Publikum schlechte Kritiken bekommt. Bekannte Beispiele aus der Filmbranche, bei denen die Charaktere sich im sogenannten unheimlichen Tal befanden und daher keine hohe Akzeptanz bei den Zuschauern erreichen konnten, sind (siehe auch Abbildung 1.5) Disneys "Eine Weihnachtsgeschichte"<sup>4</sup>, "Beowulf"<sup>5</sup> und Final Fantasy<sup>6</sup> und "Der Polarexpress"<sup>7</sup>, welcher von der Newsweek online in ihrem Artikel "The 10 scariest movies and why they creep you out"<sup>8</sup> als Beispiel aufgegriffen wurde, welches belegt, dass

<sup>4</sup><http://www.eine-weihnachtsgeschichte-derfilm.de/>

<sup>5</sup><http://www.warnerbros.de/beowulf/>

<sup>6</sup><http://www.final-fantasy-der-film.de/>

<sup>7</sup><http://www.warnerbros.de/movies/polarexpress/>

<sup>8</sup><http://goo.gl/KTefsg> Stand: 08. Februar 2016



**Abbildung 1.6:** Bekannte Beispiele für Charaktere aus Animationsfilmen, die im *Uncanny Valley* liegen. Das oberste Bild entstammt aus dem Film der Polarexpress (<http://goo.gl/5lgX7F> Stand: 09. Oktober 2015), das linke Bild aus dem Film Beowulf (<http://goo.gl/V5k6eh> Stand: 09. Oktober 2015), das zweite Bild rechts ist aus dem Disney Film eine Weihnachtsgeschichte (<http://goo.gl/tZNI1f> Stand: 09. Oktober 2015) und das rechte unterste Bild ist aus Final Fantasy (<http://goo.gl/7XbBWj> Stand: 09. Oktober 2015).

aufgrund des Uncanny Valleys eine zu realistische Umsetzung menschlicher Charaktere in Animationsfilmen Kindern Alpträume bereiten könnte. Erst ab einem extrem hohen Grad an Realismus, bei dem sich der digitale Charakter gar nicht mehr von der Aufnahme eines realen Menschen unterscheidet, steigt die Akzeptanz des Betrachters wieder an und das unheimliche Tal gilt als überwunden. Wie in Kapitel 2 detaillierter betrachtet wird, wird bis heute mit großem Eifer daran geforscht, was denn nun genau die Menschlichkeit in Bezug auf ihr Erscheinungsbild ausmacht und wie diese Akzeptanz des Betrachters gewonnen werden kann. Auch hier spielt *Subsurface Scattering* eine wichtige Rolle, da die menschliche Haut eines der Paradebeispiele für das Auftreten dieses Phänomens darstellt.



**Abbildung 1.7:** Ein Rendering menschlicher Haut links ohne und rechts mit *Subsurface Scattering* [JMLH01].

### 1.3 Zielsetzung

Ziel der hier vorliegenden Arbeit ist es, die neuesten Verfahren zur Visualisierung des physikalischen Phänomens des *Subsurface Scatterings* in der 3D-Software *Autodesk Maya* und mit dem Renderer *V-Ray* von der *Chaosgroup* zu analysieren. Dabei sollen die drei shaderbasierten Lösungen mit dem Standard-*V-Ray*-Material, dem *FastSSS2*-Shader und dem neuen *V-Ray-Skin*-Material zur Simulation des *Subsurface Scattering*-Effektes untersucht werden. Um die unterschiedlichen Verfahren an sich zu analysieren, sie aber auch gegeneinander Testen und miteinander vergleichen zu können, sollen unterschiedliche Szenen, jeweils mit zwei Varianten umgesetzt werden. Die Szenen werden so gewählt, dass unterschiedliche Materialien, bei denen *Subsurface Scattering* auftritt, herangezogen werden. So sollen die unterschiedlichen Verfahren in Bezug auf die Umsetzung von Marmor, Kerzenwachs, aber auch menschliche Haut angewendet werden. Die Verfahren sollen bezüglich ihrer Handhabung, der benötigten Renderzeiten und vor allem auch in Bezug auf die physikalische Plausibilität der jeweiligen Renderergebnisse untersucht werden. Der Fotorealismus der jeweiligen Ergebnisse soll außerdem anhand eines Vergleiches mit realen Bildern ermittelt werden und gegebenenfalls Lösungsvorschläge für eine Vereinfachung ermittelt und umgesetzt werden.

## 1.4 Organisation der Arbeit

Der Aufbau dieser Arbeit lässt sich in drei vorgeordnete Abschnitte unterteilen und gestaltet sich folgendermaßen: Die Kapitel 1-3 sollen das Thema an sich einleitend aufzeigen, den derzeitigen Stand der Technik präsentieren sowie die zum Verständnis der Arbeit wichtigen Grundlagen in Bezug auf die physikalischen Hintergründe des *Subsurface Scattering*-Effektes und die drei für die Umsetzung verwendeten Shader vermitteln. Die darauf folgenden Kapitel 4, 5 und 6 bilden den eigentlichen Kern, der hier vorliegenden Arbeit und befassen sich mit der Analyse von unterschiedlichen Methoden zur Visualisierung des *Subsurface Scatterings* in der 3D-Computergrafik. Den Schlussteil bilden die Kapitel 7 und 8 mit dem Evaluationskapitel sowie einer Zusammenfassung der gesamten Arbeit mit anschließendem Ausblick, aus daraus resultierenden möglichen weiteren Arbeiten.

Im Folgenden wird der Inhalt der einzelnen Kapitel noch einmal ausführlicher beleuchtet: In diesem ersten einführenden Kapitel wurde das Thema der Arbeit vorgestellt und dargelegt, dass die Visualisierung von *Subsurface Scattering* für Renderings, insbesondere auch im fotorealistischen Bereich, unabdingbar ist, obgleich eine physikalisch plausible Darstellung mit 3D-Software, vor allem aufgrund des nicht vorhandenen Volumens der Materialien nicht trivial ist. Daraus resultiert die Zielsetzung dieser Arbeit, unterschiedliche Verfahren zur Erzeugung eines *Subsurface Scattering*-Effektes zu analysieren.

Das zweite Kapitel dieser Arbeit soll sich zum einem mit dem aktuellen Stand der Technik im Bereich der sogenannten *Digital Humans* (zu Deutsch digitalen Menschen) befassen und beleuchten, womit sich die Forschung in Bezug dazu derzeit beschäftigt. Zum anderen wird in diesem Kapitel dargelegt, wie weit die Forschung bezüglich des physikalischen Phänomens des *Subsurface Scatterings* bisher gekommen ist. Dabei sollen die Arbeiten, welche die Visualisierung dessen in der Computergrafik im Fokus haben, präsentiert werden. Das dritte Kapitel erläutert die physikalischen Grundlagen sowie die Grundlagen der 3D-Computergrafik, welche in Bezug auf *Subsurface Scattering* für das Verständnis dieser Arbeit notwendig sind.

Das vierte Kapitel bildet zugleich das erste Methodik-Kapitel. In diesem sollen unterschiedliche Methoden, in Form des *V-Ray*-Standard-Materials und des auf *Subsurface Scattering* spezialisierten *FastSSS2*-Shaders, für die Visualisierung von *Subsurface Scattering* anhand von zwei unterschiedlichen Beispielszenarien für die Materialien Wachs und Marmor analysiert werden.

Das fünfte Kapitel und damit der zweite Abschnitt der Methodik setzt sich mit der Analyse zweier unterschiedlicher Methoden zur Visualisierung von *Subsurface Scattering* in Bezug auf das Rendering menschlicher Haut auseinander. Hierzu sollen zwei verschiedene Beispielszenarien, zum einen eine menschliche Hand und zum anderen ein kompletter menschlicher Kopf mit unterschiedlichen Ansätzen bezüglich des *Subsurface Scatterings* der Haut, zum einen mit dem *FastSSS2*-Shader und zum anderen mit dem neuen *V-Ray SkinMtl.*, gerendert werden.

Der abschließende Teil des methodischen Abschnittes beinhaltet mit Kapitel sechs die Programmierung einer Schnittstelle in Form eines Plugins zur Vereinfachung des Umgangs mit Texturen im Hinblick auf die Visualisierung des *Subsurface Scatterings* vor allem in Bezug auf das *Skin Material*.

Kapitel sieben beinhaltet die Auswertung einer Evaluation, welche der Analyse der unterschiedlichen Methoden neben den subjektiven Parametern und den Renderzeiten einen weiteren messbaren Parameter in Form einer statistischen Auswertung hinzufügt. Die Evaluation erfolgt bezüglich der Fragestellung, ob jeweils einer der beiden gegeneinander getesteten Shader näher an ein fotorealistisches Ergebnis herankommt.

Zum Abschluss folgt in Kapitel acht eine kurze Zusammenfassung der gesamten Arbeit sowie ein Ausblick auf mögliche Arbeiten, welche sich aus den Ergebnissen dieser Arbeit ergeben und sich ihr anschließen könnten.

### 1.5 Zusammenfassung der wichtigsten Ergebnisse

Der Einsatz von computergeneriertem Bildmaterial in der Film und Werbeindustrie steigt stetig an, sodass Produktionen dieser Art mittlerweile auf einem sehr hohen Level sind, vor allem auch, wenn es darum geht täuschend echte fotorealistische Bilder zu erzeugen. Trotz der enorm fortgeschrittenen Entwicklung, gibt es noch Bereiche in der Computergrafik, bei denen eine physikalisch korrekte Visualisierung nicht trivial ist. Eines davon ist das physikalische Phänomen des *Subsurface Scatterings*, welches transluzenten Materialien eine besonders weiche Oberfläche verleiht und in einigen Fällen, wenn das Objekt von hinten beleuchtet wird, auch einen Effekt des Durchleuchtens erzeugt.

Der marktführende Renderer bezüglich fotorealischem und physikalisch plausiblen Rendering ist *V-Ray* der Firma *Chaosgroup*. Für diesen Renderer werden drei unterschiedliche Shader für die Visualisierung von *Subsurface Scattering*-Effekten zur Verfügung gestellt. Diese drei Shader sind im Rahmen dieser Arbeit analysiert worden. Dafür sind prototypische Umsetzungen in Form von Renderings entstanden, die die Visualisierung von *Subsurface Scattering* an unterschiedlichen Materialien - Kerzenwachs, Marmor und menschlicher Haut - darstellen. Eine Kerzen-Szene sowie ein Marmordrache wurden dabei jeweils mit dem Standard-Shader, dem *V-Ray Material*, sowie dem speziell für *Subsurface Scattering*-Visualisierungen entwickelten *FastSSS2*-Shader umgesetzt. Ein Rendering eines menschlichen Kopfes sowie eine menschliche Hand wurden jeweils mit dem *FastSSS2*-Shader sowie mit dem neuen *Skin Material* realisiert. Dabei wurden die Shader jeweils bezüglich der Qualität ihrer Ergebnisse, ihrer Handhabung und der Renderzeiten analysiert. Hierbei hat sich herauskristallisiert, dass der *FastSSS2*-Shader jeweils durch bessere Qualität und Handhabung besticht, in den Renderzeiten jedoch deutlich hinter den anderen beiden Shadern zurückliegt.

Da das neue *Skin Material* aufgrund der Restriktionen der für den Nutzer steuerbaren Parameter gegenüber dem *FastSSS2*-Shader, besonders bezüglich Umsetzungen des *Subsurface*

*Scattering*-Effektes des Durchleuchtens schnell an seine Grenzen stößt, wurde ein texturbasierter Lösungsansatz für diese Problematik entwickelt und an einer zweiten Szene noch einmal in seiner Funktion verifiziert.

Aufgrund der Tatsache, dass dieser entwickelte Lösungsansatz über das Erstellen einer speziellen Maske und dem Einfärben einer Textur aufgrund vieler dafür nötiger Iterationen für die Feinanpassungen bis hin zum gewünschten Ergebnis mit einer Vielzahl an Wechseln zwischen der Bildbearbeitungs-Software und der 3D-Software verbunden ist, wurde ein *Plugin* konzipiert und prototypisch implementiert, welches es ermöglicht die Anpassungen der Textur über zwei *Slider*, für eine Effektverstärkung und eine Farbtonänderung, direkt in der *3D-Software* vorzunehmen.

Mit einer abschließenden Evaluation konnten, mit einer Reihe von durchgeführten Teststatistiken, die aus den prototypischen Umsetzungen gewonnenen Erkenntnisse, bezüglich des Merkmals der Qualität der Ergebnisse der jeweiligen Shader, statistisch nachgewiesen und verifiziert werden.



## Kapitel 2

# Stand der Technik

### 2.1 Einleitung

In diesem Kapitel soll der derzeitige Stand der Technik in Bezug auf die Visualisierung des physikalischen Phänomens *Subsurface Scattering* in der 3D-Computergrafik illustriert werden.

In der hier vorliegenden Arbeit wird dieses Kapitel noch einmal in zwei Sektionen unterteilt. Eines der vorherrschenden Anwendungsszenarien von *Subsurface Scattering*-Visualisierungen und bis heute eine der größten Herausforderungen, welche sich einem *3D-Artist* in der Computergrafik stellt, ist das Rendering realistisch erscheinender menschlicher Haut. Dabei spielt das Phänomen des *Uncanny Valley* eine wichtige Rolle. Wobei eine gelungene Visualisierung von physikalisch plausibel wirkendem *Subsurface Scattering*-Effekt dem Erscheinungsbild der gerenderten Haut einen deutlich gesteigerten Grad an Fotorealismus verleiht, was einen großen Schritt zur Überwindung des *Uncanny Valley* beitragen könnte. Damit sind das *Uncanny Valley* und *Subsurface Scattering* zwar unterschiedliche Themen und unabhängig voneinander zu betrachten, in Bezug auf die Erschaffung von glaubwürdigen *Digital Humans* sind sie aber eng miteinander verwoben.

Aus diesem Grund befasst sich zunächst der erste Abschnitt dieser Einheit mit dem aktuellen Forschungsstand im Gebiet der *Digital Humans* und das Phänomen des *Uncanny Valley* wird näher beleuchtet.

Der zweite Abschnitt dieses Kapitels setzt sich dann mit dem reinen *Subsurface Scattering* auseinander, die wichtigsten Arbeiten, welche sich mit dessen Visualisierung in der 3D-Computergrafik befassen, werden vorgestellt und somit der aktuelle Stand der Forschung zu diesem Themengebiet erörtert.

## 2.2 Digital Humans

*“ Computer generated characters have always been a challenge. Making them look and move like a believable organic beings has taken decades to perfect. Over the years, the tools and artist have evolved to face these challenges that are only limited to their imaginations. Yet there is still a challenge that remains hard to crack: Believable Digital Humans.”<sup>1</sup>*

So beschreibt das *Wikihuman*-Projekt, auf welches später in diesem Kapitel noch detaillierter eingegangen wird, auf ihrer Webseite die Herausforderung, welcher sie sich mit ihrem Projekt annehmen wollen. Weiter führen sie aus:

*“The Uncanny Valley has become such a powerful way to describe the negative reaction that people feel when something is ‘almost human,’ but not human enough.”<sup>2</sup>*

*Subsurface Scattering* und das *Uncanny Valley*-Phänomen sind für sich stehende Bereiche in der 3D-Computergrafik. Wer sich jedoch mit der Forschung auf dem Gebiet der *Digital Humans* beschäftigt, dem wird schnell bewusst, wie eng sie an dieser Stelle miteinander verwoben sind. Das *Uncanny Valley* kann nur überwunden werden, wenn die digitale Abbildung des Menschen so realistisch wie möglich wird und von einem realen Menschen eigentlich nicht mehr zu unterscheiden ist, was ohne eine möglichst physikalisch plausible Simulation des *Subsurface Scattering*-Effektes der Haut, der einen so enormen Anteil des menschlichen Erscheinungsbildes ausmacht, nicht gewährleistet werden kann. Wie auch die Macher von “The Matrix Reloaded” in der Arbeit “Realistic Human Face Rendering for ‘The Matrix Reloaded’” [BL03], auf die später in diesem Kapitel noch ausführlicher eingegangen wird, schnell erkannt haben.

*Subsurface Scattering* ist ein Effekt, der alle transluzenten Materialien betrifft und damit noch eine ganze Reihe weitere Anwendungsgebiete in der Computergrafik, neben den *Digital Humans* hat. Dennoch ist und bleibt das Rendering menschlicher Haut die Königsklasse des Renderings und eines der Hauptanwendungsgebiete für die Visualisierung des Lichttransportes unter die Oberfläche in der 3D-Computergrafik und für diese Umsetzungen ist es wichtig, das Zusammenspiel mit dem Phänomen des *Uncanny Valley* zu verstehen und zu verinnerlichen.

---

<sup>1</sup><http://www.wikihuman.org/index.php/about-us/> Stand: 25. Dezember 2015

<sup>2</sup><http://www.wikihuman.org/index.php/about-us/> Stand: 25. Dezember 2015



**Abbildung 2.1:** Gegenüberstellung eines Fotos (links) und eines gut umgesetzten 3D-Renderings in der Kategorie der *Digital Humans* (rechts), auf denen jeweils der Schauspieler Bryan Cranston in seiner Rolle als *Walter White* in der Serie *Breaking Bad* zu sehen ist. Der Vergleich zeigt, dass aufgrund des nicht ausreichenden *Subsurface Scattering*-Effektes (besonders am Ohr) schnell zu erkennen ist, bei welchem der beiden Bilder es sich um das reale Foto und bei welchem um das am Computer erzeugte Rendering handelt. Bildquelle: <http://goo.gl/ijPq1E> Stand: 27. Februar 2016

Mit dem Begriff *Uncanny Valley* wird der Effekt beschrieben, dass Menschen ein Gefühl von Unbehagen empfinden, wenn sie virtuelle Charaktere betrachten, die sehr menschlich wirken, aber eben noch nicht menschlich genug sind, um nicht doch als virtueller Charakter erkennbar zu sein.

Schon 1948 beschrieb Sigmund Freud in seinem Werk "Das Unheimliche" [fre] das Auftreten eines sehr unangenehmen Gefühls beim Anblick von Menschen mit künstlichen Prothesen. Nach Freuds Ansicht löst es ein extremes Unbehagen aus, in einem familiären, bekannten Kontext etwas Fremdartiges zu betrachten. Genau diese Reaktion machen sich Filmemacher besonders im Horror-Genre zu Nutzen und erzeugen so den Grusel-Effekt in ihren Werken. Ein klassisches Beispiel hierfür ist *Frankensteins Monster*. Hier wurde von James Whale im Jahre 1931 ein Wesen erschaffen, welches in einem Grenzbereich zwischen den Lebenden und den Toten lebt und somit zugleich Sympathie und Ekel hervorruft. [BGBC05]

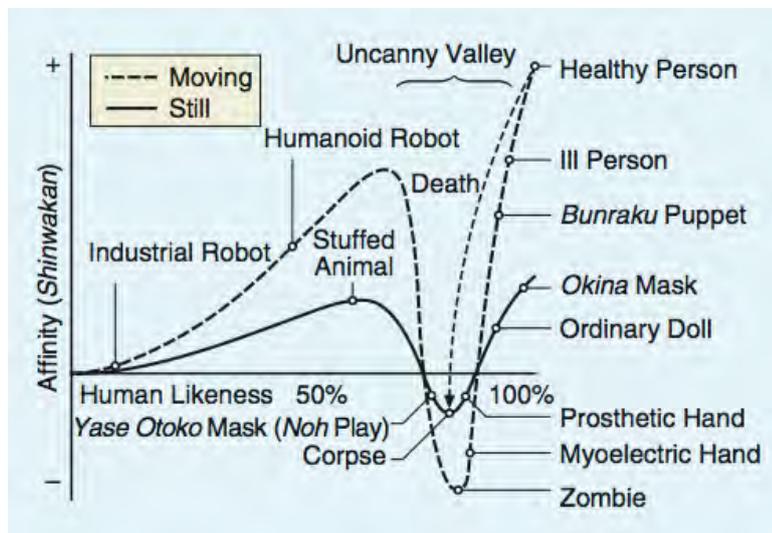
Im Jahre 1970 wurde die erste Erklärung für das Phänomen des *Uncanny Valley* und der dazugehörige Begriff durch den japanischen Professor der Robotik, Masahiro Mori, am Institut für Technologie in Tokyo eingeführt. In seiner Arbeit "Bukimi no tani" (Übersetzung aus dem Japanischen: "The Uncanny Valley") [MMK12] beschreibt Mori erstmals, dass Roboter, bei denen in ihrer Konstruktion versucht wird, eine dem Menschen zu ähnliche Erscheinung nachzubilden, eine gruselige und abschreckende Wirkung auf den Menschen haben und ein unbehagliches Gefühl auslösen, da die Annäherung an die Menschlichkeit nicht ausreichend vollendet werden kann.

Mori beschreibt das Auftreten des Phänomens wie folgt: Bei Beginn des Einsatzes von Robotern, vorrangig zum Ersatz von Fabrikarbeitern, wurde bei der Konstruktion hauptsächlich darauf geachtet, dass das Design ihre Funktion bestmöglich unterstützt. Die äußerere Erscheinung der Roboter war dabei gleichgültig und es gab keinerlei emotionale Verbindung. Wenn nicht sogar durch die Tatsache, dass der Einsatz dieser Maschinen eine Kürzung des Personals bedeutete, eine Abneigung ihnen gegenüber entstanden ist. Bei der Konstruktion von Spielzeugrobotern wurde schon vermehrt Wert auf das Design gelegt und so ähnelten diese Roboter der menschlichen Figur deutlich stärker. Sie hatten Arme, Beine und Gesichter, die in ihrer Anordnung dem Menschen nachempfunden waren und dadurch entstand eine Verbundenheit des Besitzers seinem Spielzeug gegenüber, wodurch die generelle Akzeptanz Robotern gegenüber stieg. Dass diese Akzeptanz dann aber wieder rapide absinkt, wenn die Erscheinung zu menschlich wird, erklärt Mori am Beispiel einer Handprothese. Je ähnlicher eine Prothese einer echten Hand wird, desto weniger fällt auf den ersten Blick auf, dass es sich überhaupt um eine Prothese handelt. Wenn wir aber jemandem die Hand geben und sich erst beim Handschlag herausstellt, dass die Hand unseres Gegenübers gar keine echte, sondern eine Nachbildung ist, erschrecken wir und empfinden negative Emotionen.

*“An example of a function that does not increase continuously is climbing a mountain the relation between the distance (x) traveled by a hiker toward the summit and the hiker’s altitude (y) owing to the intervening hills and valleys. I have noticed that, in climbing toward the goal of making robots appear like a human, our affinity for them increases until we come to a valley ..., which I call the uncanny valley.”*

Mori beschreibt das *Uncanny Valley* anhand einer Kurve, bei der die Akzeptanz des Androiden mit zunehmender Ähnlichkeit zum Menschen zunächst ansteigt und dann ab einem bestimmten Grad an Menschlichkeit schlagartig abfällt. Die Akzeptanz steigt erst dann wieder an, wenn es sich um das perfekte Abbild eines Menschen handeln würde. Der Bereich, der auf der Kurve zwischen dem ersten Punkt des Akzeptanzabfalls und der erneuten Akzeptanzgewinnung liegt, wird von Mori als *Uncanny Valley* oder unheimliches Tal bezeichnet. Ein weiterer Aspekt, auf den Mori in seiner Arbeit eingeht, ist der Einfluss von Bewegungen der Charaktere auf den Verlauf der *Uncanny Valley*-Kurve. Zunächst steigt die Akzeptanz eines Roboters der sich bewegt viel schneller an, als die einem stillen Objekt gegenüber. Aber durch Bewegungen, die für den Menschen unnatürlich wirken, fällt die Akzeptanzkurve dann ab dem Wendepunkt auch viel steiler und schneller sowie tiefer ins *Uncanny Valley* ab.

Mori empfiehlt, dass für die Konstruktion von Robotern der erste *Peak* der Kurve als Ziel angesehen wird und lieber ein nicht menschliches Design verfolgt wird, um eine größere Verbundenheit und Nähe und darüber eine höhere Akzeptanz dem Roboter gegenüber zu erreichen. Als Begründung für das *Uncanny Valley* sieht Mori eine Art Instinkt, der den Menschen vor Gefahren schützen soll und auf Grund dessen ein unheimliches Gefühl in uns ausgelöst wird.



**Abbildung 2.2:** Die Kurve zur Darstellung des Verlaufs des *Uncanny Valley* nach Moris Konzept [MMK12].

Unmittelbar in den Jahren nach der Veröffentlichung von Moris Arbeit in dem japanischen Journal "Energy" fand sie kaum Beachtung, wie im Vorwort des Herausgebers der autorisierten Übersetzung des Papers dargelegt wird [MMK12]. Allein 36.700 Einträge zum Suchbegriff "Uncanny Valley" bei Google Scholar<sup>3</sup> (Stand:22. November 2015) zeigen, dass in den letzten Jahren das Interesse an dem von Mori entworfenen Konzept rapide angestiegen ist und das nicht nur in der Robotik und der Mensch-Maschine-Interaktion, sondern auch in der Computergrafik. So entstanden in der Forschung der letzten Jahre viele Arbeiten, die untersuchen, ob und wie das *Uncanny Valley* überwunden werden kann und wo dessen biologische und soziale Wurzeln liegen, aber auch solche, die seine Wichtigkeit oder gar seine Existenz in Frage stellen. Einige dieser Arbeiten sollen im Folgenden vorgestellt werden.

### 2.2.1 Kritische Betrachtungen des Uncanny Valley-Konzeptes

In "The Uncanny Valley: Does it exist?" [BGBC05] werfen Harry Brenton et al. die Frage nach der Existenz dieses Phänomens auf. Sie beschreiben, dass obwohl der Begriff "Uncanny Valley" eine große Verwendung innerhalb der Computergrafik findet, keine hinreichende Recherche darüber vorliegt, ob dieser Effekt wirklich existiert. Mit ihrer Arbeit möchten sie ein Fundament für zukünftige Arbeiten bilden, indem sie Beispiele für das Auftreten des *Uncanny Valley*-Effektes in den verschiedenen Bereichen, in denen er eine Rolle spielt, aufzeigen und Beweise für dessen Existenz überprüfen.

<sup>3</sup><https://goo.gl/d5wVLx> Stand: 22. November 2015

David Hanson, ein Robotiker, verwirft das Konzept des *Uncanny Valley* als pseudowissenschaftlich und argumentiert, dass Designer von Robotern in ihrer Arbeit nicht auf Grund eines nicht wissenschaftlich nachgewiesenen Konzeptes eingeschränkt werden sollten [Dan03].

Laut Christoph Bartneck et al. [BKI09] spielte das Konzept des *Uncanny Valley* eine große Rolle in der Computergrafik, schon bevor es in der Robotertechnik zum aktuellen Thema wurde, was sie damit begründen, dass hochgradig realistische Renderings aufgrund der fortgeschrittenen Hardware und Software in diesem Bereich schon möglich waren, lange bevor die Robotertechnik einen solchen Grad an Realismus überhaupt erreichen konnte. Dies spiegelt sich in Arbeiten wie "Modeling the Impact of Shared Visual Information on Collaborative Reference" [GRK07], "The Uncanny Valley: Effect of Realism on the Impression of Artificial Human Faces", [SN07] "Introduction", "Capturing Puppets: Using the age old art of puppetry combined with motion capture to create a unique character animation approach" [Win06] und "Building Characters: Lessons Drawn from Virtual Environments" [VSS05] wider.

Gerade im Bereich der computergenerierten Animationen für Filme und Computerspiele gibt es daher zahlreiche Beispiele bei denen der Effekt des *Uncanny Valley* eine große Rolle spielt und *3D-Artists* auf Grund dessen in ihren *Character-Designs* Einschränkungen empfinden. So zitiert Lawrence Weschler in seiner Arbeit "Why is this man smiling? Digital Animators are closing in on the complex System that makes a face come alive" [Wes06], die Macher von "Shreck" <sup>4</sup>, die die äußere Erscheinung der Figur der Prinzessin Fiona nachträglich weniger menschlich gestalteten, da ihr zu real wirkendes Erscheinungsbild ein Unbehagen hervorrief. Auch der Animations-Direktor von "The Final Fantasy: the spirits within" <sup>5</sup> beschreibt ein groteskes Gefühl bei der Arbeit mit zu fotorealistischen virtuellen Charakteren.

Laut Brenton et al. [BGBC05] belegen diese und viele weitere Beispiele, dass eine Reaktion im Sinne des *Uncanny Valley* existiert, jedoch sei das noch keine Bestätigung für das Modell des *Uncanny Valley*, so wie es von Mori aufgestellt wurde. Ihrer Meinung nach müsste die Kurve, die das *Uncanny Valley* beschreibt, viel strikter durch gemessene, objektive Parameter bestimmt werden. Ein Problem bei der Forschung bezüglich des *Uncanny Valley* sei, dass Animatoren und *Character-Designer* scheinbar instinktiv davor zurückschrecken, Modelle zu verwenden, die Anzeichen des *Uncanny Valley-Effektes* aufweisen. So fährt die Computer-Animations-Firma *Pixar* <sup>6</sup> sogar sehr erfolgreich die Strategie sich in ihren Filmen häufig auf nicht menschliche Charaktere zu fokussieren. [BKI09]

Dennoch konnten Kristine Nowak und Frank Biocca in ihrer Arbeit "The Effect of the Agency and Anthropomorphism of Users' Sense of Telepresence, Copresence, and Social Presence in Virtual Environments" [NB03] aufzeigen, dass ansteigender Realismus zunächst

---

<sup>4</sup><http://dreamworks.wikia.com/wiki/Shrek> Stand: 07. Dezember 2015

<sup>5</sup><http://www.final-fantasy-der-film.de/> Stand: 07. Dezember 2015

<sup>6</sup><http://www.pixar.com/> Stand: 07. Dezember 2015

einen positiven Effekt auf die Akzeptanz beim Betrachter von Charakteren haben, sehr hoher Grad an Realismus aber eine Unheimlichkeit auslöst.

Mel Slater et al. haben in ihren Arbeiten "A Virtual Presence Counter" [SS00], "Presence and the Sixth Sense" [Sla02] und "Physiological Responses to Breaks in Presence: A Pilot Study" [SBS03] die Theorie aufgestellt, dass das *Uncanny Valley* auf einen sogenannten "Break in Presence" zurückzuführen ist. Das bedeutet, dass das menschliche Gehirn beim Betrachten von zu realen Charakteren zwischen zwei verschiedenen Hypothesen wechselt. Das Gehirn nimmt zuerst an, dass es sich bei dem Charakter um ein menschliches Wesen handelt und wechselt dann zu der Annahme, dass es sich nicht um ein menschliches Wesen handeln kann. Dieser Wechsel der Hypothesen, so glauben Slater et al., ist es, der das unheimliche Gefühl und das Unbehagen dem Charakter gegenüber auslöst. Weiterhin beschreiben Brenton et al. in ihrer Arbeit, dass ein weiteres Problem in Bezug auf den *Uncanny Valley-Effekt* ist, dass mit steigendem Realismus der äußeren Erscheinung der 3D-Charaktere auch die Erwartung an dessen menschliches Verhalten steigt. Die Qualität der Bewegungen und des Verhaltens von animierten Charakteren kommt aber an den Realismus ihres Erscheinungsbildes nicht heran, was ein Paradox in der Gesamtwahrnehmung erzeugt, was wiederum zu einem Unbehagen führt. Brenton et al. empfehlen daher beim Design und der Animation von virtuellen Charakteren ein möglichst gutes Gleichgewicht zwischen dem Realismus der verschiedenen bedeutungsvollen Aspekte, wie der äußeren Erscheinung, der Bewegung und auch der Interaktion zwischen den Charakteren und zu ihrer Umwelt zu suchen, um so das *Uncanny Valley* zu umgehen.

In "My Robotic Doppelgänger-A Critical Look at the Uncanny Valley" [BKI09] versuchen Christoph Bartneck et al. mit Hilfe eines Experimentes Moris Konzept des *Uncanny Valley* zu widerlegen.

Sie argumentieren, dass es zwar schon zahlreiche Versuche diesbezüglich gab, so zum Beispiel in Arbeiten wie "All Robots Are Not Created Equal: The Design and Perception of Humanoid Robot Heads" [DGF02], "The advisor robot: tracing people's mental model from a robot's physical attributes" [PK06], "Children and robots learning to play hide and seek" [TSP<sup>+</sup>06], "Subjective ratings of robot video clips for human likeness, familiarity, and eeriness: An exploration of the uncanny valley" [Mac06], um einige ihrer Beispiele zu nennen, diese Arbeiten aber alle einigen Einschränkungen unterliegen. Diese Einschränkungen liegen darin, dass für die Versuche meist keine echten Roboter, sondern nur Bilder oder Videos von diesen genutzt wurden, oder wenn sogar ein realer Roboter statt nur eines Abbildes genutzt wurde, sich auf einen einzelnen Roboter beschränkt wurde.

Um die genannten Einschränkungen vorheriger Experimente zu umgehen, haben Bartneck et al. ihr Experiment wie folgt aufgebaut. Sie haben sich Hiroshi Ishiguro und seinen Roboterdoppelgänger als "Exponate" des Versuchsaufbaus ausgewählt und sowohl den Menschen Hiroshi als auch zwei Varianten seines Roboters, einmal in möglichst realistischer Form und einmal in einer etwas abgespeckten, etwas weniger realistischen Form, in drei Räumen auf einem Stuhl platziert und ihnen Probanden gegenübergestellt, die nacheinander jeweils mit den drei unterschiedlichen Formen von Hiroshi interagieren und anschließend Fragebögen zu der Interaktion ausfüllen sollten. Dadurch, dass der Roboter ein Doppelgänger des Menschen



**Abbildung 2.3:** Aufbau der Versuchsreihe von Bartneck et al. Links ist ein Bild mit dem realen Menschen Hiroshi Ishiguro zu sehen und rechts eines mit seinem Roboter Doppelgänger. Bildquelle [BKI09]

ist, sollte verhindert werden, dass aufgrund von unterschiedlich attraktivem Empfinden der jeweiligen Charaktere ein Einfluss auf das Experiment entsteht.

Die Studie sollte zwei Hypothesen testen nämlich zum einen, ob hochgradig realistische Roboter von Menschen weniger gemocht werden als der reale Mensch und zum anderen, ob, wie Mori in seinem Konzept zum *Uncanny Valley* angibt, die Bewegung des Charakters einen Unterschied auf die Liebenswürdigkeit macht. Dafür wurden in das Experiment noch zwei unterschiedliche Varianten der Beweglichkeit des Roboters integriert. 16 Probanden saßen einem Roboter mit voller Bewegung und weitere 16 Probanden einem mit eingeschränkter Bewegung gegenüber. Die Ergebnisse des Versuchs ließen keine signifikanten Aussagen darüber zu, dass die Menschen, die den Test durchgeführt haben, den Roboter weniger liebenswürdig empfanden, als den Menschen und auch ein signifikanter Unterschied zwischen den beiden unterschiedlichen Bewegungsmodi konnte nicht nachgewiesen werden.

Damit sehen Bartneck et al. das Konzept des *Uncanny Valley*, so wie es Mori 1970 aufgestellt hat, als widerlegt an und sie regen dazu an, die Entwicklung nicht länger auf Grund dieses Konzeptes aufzuhalten. Dennoch gestehen sie auch ihrer Versuchsreihe einige Schwachstellen ein. So war die Interaktion der Probanden mit den Robotern sehr kurz und einfach gehalten und auch der Vergleich zwischen voller und eingeschränkter Bewegung entspricht nicht Moris Angaben von Robotern mit und komplett ohne Bewegung, die er in seinen Ausführungen miteinander verglichen hat. Bartneck et al. kommen letztendlich zu dem Schluss, dass Vermenschlichung und auch Liebenswürdigkeit multi-dimensionale Konstrukte sind, welche nicht mit einer zweidimensionalen Kurve abgebildet werden können.

Im Paper "More Human Than Human: Does The Uncanny Curve Really Matter" [ZPB13] haben sich Jakub Zlotowski et al. ebenfalls einer kritischen Betrachtung von Moris *Uncanny Valley*-Konzept angenommen. Sie nennen drei grundlegende Fragen, welche sie in Bezug auf die Erforschung, des auch ihrer Meinung nach noch nicht hinreichend wissenschaftlich belegten Konzeptes des *Uncanny Valley*, als essentiell betrachten. Zum einen das Terminologie-Problem, welches auch Bartneck et al. [BKI09] in ihrer Arbeit angesprochen

haben. Mori spricht in seiner originalen Fassung für die Beschriftung der Y-Achse des *Uncanny Valley*-Graphen von "Shinwankan". Dieser Begriff lässt sich aber nicht wörtlich ins Englische übersetzen, was dazu führt, dass in den unterschiedlichen Forschungsansätzen bezüglich des *Uncanny Valley* unterschiedliche Übersetzungen mit unterschiedlichen Bedeutungen für diesen Begriff eingesetzt werden. So zum Beispiel "familiarity" (zu Deutsch Vertrautheit) [Mac05], "likeability" (zu Deutsch Sympathie) [BKIH09], "affinity" (zu Deutsch Verbundenheit) [MMK12], "eeriness" (zu Deutsch Unheimlichkeit) [HM10], "empathy" (zu Deutsch Mitgefühl, Einfühlsamkeit) [Mis09a]. Aufgrund der Nutzung unterschiedlicher Begriffe lassen sich die Arbeiten schwer miteinander vergleichen, was dazu führt, dass auch sehr viel schwerer wirklich Schlüsse aus den Forschungsarbeiten gezogen werden können. Daher regen Zlotowski et al. dazu an, eine allgemeingültige Regelung für die Übersetzung dieses Begriffes zu finden, bevor weitere Zeit in nicht vergleichbare und gegebenenfalls sogar in eine falsche Richtung laufende Forschung gesteckt wird. Sie geben aber auch an, dass die Möglichkeit besteht, dass hier nicht nur ein Problem mit der Übersetzung besteht, sondern vielleicht schon der von Mori gewählte Begriff nicht treffend ist.

Das zweite grundlegende Problem beschreiben Zlotowski et al. damit, dass nicht bekannt ist, wo die Grenzen liegen, bis zu deren Grad eine Vermenschlichung notwendig ist, um die maximal mögliche Sympathie dem virtuellen Charakter gegenüber zu erzeugen. Um diese Grenzen zu erforschen, sollte ihrer Meinung nach das Gegenteil untersucht werden und damit die Aspekte, welche eine Entmenschlichung verursachen. Hierzu nennen Zlotowski et al. die Arbeit "Dehumanisation: An Integrative Review" von Nick Haslam [Has06], in welcher zwischen zwei Arten von Charakteristiken der Menschlichkeit unterschieden wird, den *Uniquely Human*-Attributen und den *Human nature*-Attributen. Der dritte und letzte entscheidende Aspekt für Zlotowski et al. liegt darin, dass der steile Anstieg der Kurve nach dem tiefsten Punkt des *Uncanny Valley* bis heute nie demonstriert werden konnte und ein unbekanntes Feld darstellt, weil die einzige Referenz, welche in der Forschung diesbezüglich verwendet wird, reale Menschen sind. Der Grad an Realismus, ab welchem ein virtueller Charakter menschlich genug wird, damit die Akzeptanz wieder ansteigt, ist bisher nicht erreicht worden. Daraus schließen Zlotowski et al., dass es sich um eine "Uncanny Curve" und nicht um ein "Uncanny Valley" handelt und vor allem auf die Erforschung der rechten Seite der Kurve mit Hilfe der Charakteristiken der Dehumanisierung ein Hauptaugenmerk gelegt werden sollte, um so im Umkehrschluss Erkenntnisse darüber zu erlangen, wie das *Uncanny Valley* überwunden werden könnte.

## 2.2.2 Das Uncanny Valley-Konzept aus unterschiedlichen Blickwinkeln

### Psychologische Betrachtung des Uncanny Valley-Konzeptes

Ein weiterer Aspekt des *Uncanny Valley*, dessen sich viele Forschungsarbeiten annehmen, sind dessen Hintergründe und damit die psychologische Sicht auf das Phänomen. Eine dieser Arbeiten ist die von Angela Tinwell "Applying Psychological Plausibility to the uncanny Valley Phenomenon" [Tin14]. Sie beschreibt, dass schon 1906 Jentsch in seinem Essay "On the psychology of the uncanny" [Jen97] das Unheimliche als einen Zustand geistiger Unstimmigkeit, wenn jemand nicht entscheiden kann, was echt oder unecht, lebendig oder tot

ist, erläutert. Sowohl MacDorman et al. in "The uncanny advantage of using androids in cognitive and social science research" [Mac05] als auch schon Mori in seinem originalen Paper zum *Uncanny Valley* 1970 [MMK12] legen dar, dass die Menschen von virtuellen Charakteren, die einem menschlichen Wesen sehr ähneln, an ihren eigenen Tod erinnert werden und dadurch eine negative emotionale Reaktion ausgelöst wird. In "The ambivalent power of the robot" [Kan09] erläutert M.Kang hingegen, dass die negativen Emotionen entstehen, weil die virtuellen Charaktere, das vom Menschen vordefinierte Weltbild nicht erfüllen können und damit als mögliche Bedrohung wahrgenommen werden. Eine weitere Theorie besagt, dass das fehlende Einfühlungsvermögen in die virtuellen Charaktere die Akzeptanz derer sinken lässt [Mis09b]. Dabei ist der Mangel an Emotionen, den virtuelle Charaktere nach außen transportieren können, ein wichtiges Hindernis, welches dazu führt, dass der Betrachter sich schwerer in diese Charaktere hineinversetzen kann. Sie wirken auf Grund dessen undurchschaubar und somit bedrohlich. In einer weiteren Arbeit beschäftigt sich Angela Tinwell mit der Frage "Is the Uncanny Valley a Universal or Individual Response?" [Tin15]. Dabei legt sie Argumente dar, warum darauf geschlossen werden könnte, dass es sich beim *Uncanny Valley* nicht um ein allgemein gültiges, universelles Phänomen handelt, sondern um einen individuellen Effekt, der abhängig vom sozialen Hintergrund, von psychischen Krankheiten, von Charaktereigenschaften und auch von der Religion sein kann.

### Die Wahrnehmung des Uncanny Valley von Kindern

In "Children's perception of uncanny human-like virtual characters" [TS14] setzen sich Angela Tinwell und Robin J.S.Sloan mit der Thematik auseinander, wie Kinder das *Uncanny Valley* wahrnehmen. Dazu haben sie ein Experiment durchgeführt, bei dem Grundschulern in jeweils drei Sitzungen innerhalb von drei Wochen zwölf Videoclips von teilweise echten Schauspielern und teilweise animierten Charakteren gezeigt wurden. Nach dem Anschauen der Videos, sollten dann einfache Fragebögen dazu ausgefüllt werden. Befragt wurden 67 Kinder zwischen fünf und sechs Jahren, davon 28 Mädchen und 39 Jungen. Das Experiment brachte die Ergebnisse, dass Kinder, gleichermaßen wie auch schon bei Erwachsenen beobachtet wurde, die animierten Charaktere als fremdartiger, unfreundlicher und weniger menschlich bewerteten als die Videoclips der realen Schauspieler. Auch Kinder reagierten negativ auf eine eingeschränkte Bewegung der oberen Gesichtshälfte der animierten Charaktere. Die vor dem Experiment aufgestellte Hypothese, dass Mädchen weniger sensitiv auf das *Uncanny Valley* reagieren würden als Jungen, konnte nicht bestätigt werden. Zwischen den Geschlechtern gab es keine signifikanten Unterschiede in der Wahrnehmung. Des Weiteren wird in dem Paper auf den Umstand hingewiesen, der ebenfalls mit dem Phänomen des *Uncanny Valley* in Zusammenhang gebracht wird, dass in Amerika der Kinder Animationsfilm "Mars needs Moms" <sup>7</sup>trotz deutlich höherem Budget von 150 Millionen Dollar gegenüber "The Incredibles" <sup>8</sup> mit einem niedrigeren Budget von 92 Millionen Dollar den Machern einen deutlich schlechteren Profit eingebracht hat, was darauf zurückgeführt wird, dass die Charaktere von *World Disney's* "Mars needs Moms" menschen-ähnlicher designed wurden und ins *Uncanny Valley* fallen, während bei "The Incredibles" die Charaktere sehr

<sup>7</sup> [movies.disney.com/mars-needs-moms](http://movies.disney.com/mars-needs-moms) Stand: 05. Dezember 2015

<sup>8</sup> <http://filme.disney.de/die-unglaublichen> Stand: 05. Dezember 2015

viel stilisierter und cartoonartiger und weniger menschlich gehalten wurden, was scheinbar dazu geführt hat, dass weniger Erwartungen an ein menschliches Verhalten hervorgerufen wurden und sich die Charaktere daher größerer Beliebtheit erfreuen.

### Affen und das Uncanny Valley

In "Monkey visual behavior falls into the uncanny valley" [SG09] beschäftigen sich Shawn A. Steckenfinger und Asif A. Ghazanfar mit der Frage, ob sich der evolutionäre Ursprung des *Uncanny Valley*-Phänomens anhand von Testreihen mit Affen belegen lässt. Für das Experiment wurden Affen unterschiedliche Bilder von Artgenossen gezeigt. Zum einen echte Fotos von anderen Affen, zum anderen sehr realistische Renderings von Affen und als drittes nicht realistische Renderings von Affen. Des Weiteren wurden alle Bilder jeweils mit einem gurrenden Gesichtsausdruck, einem schreienden Gesichtsausdruck und einem neutralen Gesichtsausdruck gezeigt und sowohl in statischer, wie auch in dynamischer Variante. Gemessen wurde, welche Bilder die Affen sich länger angeschaut haben. Es konnte nachgewiesen werden, dass ein signifikanter Unterschied besteht zwischen der Betrachtungsdauer bei den echten Bildern und den realistischen Renderings sowie zwischen der Betrachtungsdauer der realistischen Renderings und den unrealistischen Renderings. Jeweils die realistischen Renderings wurden kürzer betrachtet, zwischen den echten Bildern und den unrealistischen Renderings konnte kein signifikanter Unterschied nachgewiesen werden. Das bedeutet, dass auch Affen vom *Uncanny Valley*-Phänomen betroffen zu sein scheinen, da sie echte Bilder und auch unrealistische Renderings länger betrachten, als die realistischen Renderings ihrer Artgenossen. Die unterschiedlichen Gesichtsausdrücke haben keine Wirkung auf die Ergebnisse gezeigt. Die Affen haben generell etwas länger die dynamischen Varianten betrachtet als die statischen und der *Uncanny Valley*-Effekt war in den dynamischen Varianten etwas stärker zu erkennen.

### 2.2.3 Digital Human League

Im August 2014 hat sich eine Gruppe von führenden Persönlichkeiten der Computergrafikindustrie sowohl aus künstlerischer als auch aus wissenschaftlicher Motivation heraus zusammengeschlossen und die *Digital Human League*<sup>9</sup> gegründet, eine Gemeinschaft, die es sich zur Aufgabe gemacht hat, einen Versuch zu wagen, gemeinsam einen Weg zu finden das *Uncanny Valley* zu überwinden. Um ihr Wissen bezüglich der *Digital Humans* zu teilen und gemeinsam an weiteren Entwicklungen zu forschen, wurde ein *Open Source*-Projekt ins Leben gerufen, welches unter dem Namen "Wiki Human" bekannt ist. Chaos Group's Christopher Nichols, der zu den Gründern der *Digital Human League* gehört, beschreibt in einem Interview mit der "Computer Graphics World", dass er im *Wiki Human*-Projekt eine große Chance sieht, Probleme, an denen schon lange im kleinen immer wieder gearbeitet wurde, nun mit vereintem Wissen aus unterschiedlichen Disziplinen lösen zu können<sup>10</sup>.

---

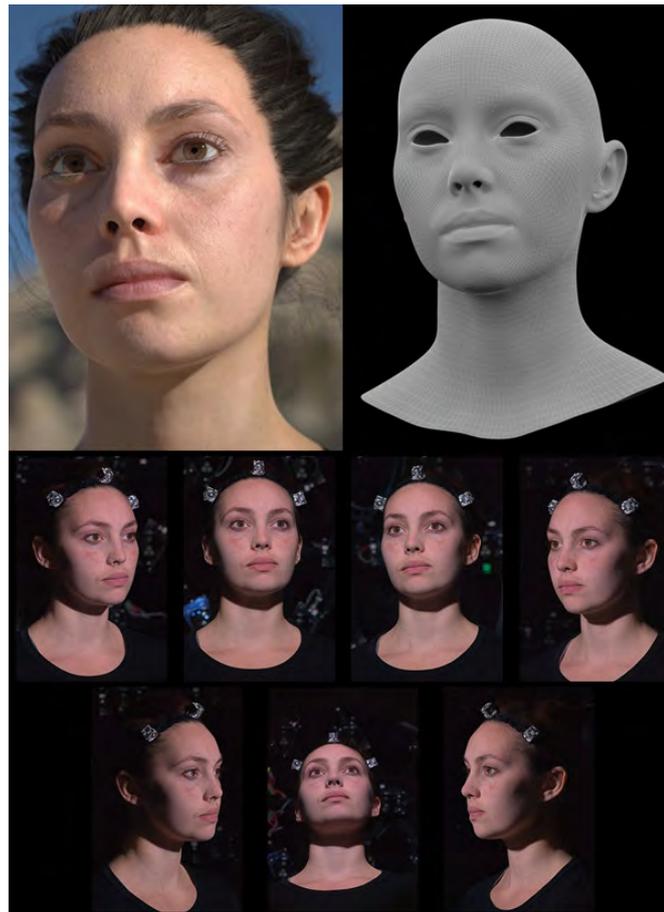
<sup>9</sup><https://labs.chaosgroup.com/index.php/category/digital-human-league/> Stand: 07. Dezember 2015

<sup>10</sup><http://www.cgw.com/Publications/CGW/2015/Volume-38-Issue-4-Jul-Aug-2015-/Almost-Human.aspx> Stand: 07. Dezember 2015

## 2. STAND DER TECHNIK

---

Im Augenblick liegt der Fokus der Arbeit der *Digital Human League* auf der Realisierung glaubwürdiger menschlicher Gesichter, die digital erstellt werden. Mit Hilfe neuester *Light Stage*-Technologie des *Institute for Creative Technologies* der Universität in Southern California und Unterstützung des Teams der *Digital Human League* konnte, mit der zweiten Version eines *High Resolution Scans* der Schauspielerin Emily O'Brien, der erste große Meilenstein im *Wiki Human*-Projekt errungen werden, so Nichol's weiter.<sup>11</sup>



**Abbildung 2.4:** Ein Rendering der digitalen Emily O'Brien (links) sowie ein Mesh ihres eingescannten Kopfes (rechts) und Aufnahmen während des Scanvorganges in der sogenannten *Lightstage* des *Institute for Creative Technologies* der Universität in Southern California (unten) (<http://goo.gl/8RDw20>, Stand: 07. Dezember 2015).

Den Grund an der Lösung des Problems bezüglich des *Uncanny Valley* mit Hochdruck zu arbeiten, sieht Nichols darin liegen, dass die Menschheit schon immer gerne sich selbst möglichst gut in der Kunst widerspiegeln und darstellen wollte und das nun eben auch für

<sup>11</sup><http://www.cgw.com/Publications/CGW/2015/Volume-38-Issue-4-Jul-Aug-2015-/Almost-Human.aspx> Stand: 07. Dezember 2015

die Computergrafik gilt. Außerdem ist dieses Hindernis eines, das schon eine ganze Zeit zu überwinden versucht wird und es würde eine große Erleichterung für viele bedeuten, würde dieses große Ziel endlich erreicht werden. Des Weiteren sieht er aus wissenschaftlicher Sicht die große Bereicherung in dieser Forschung darin, dass der Mensch unglaublich viel über sich selbst lernt, wenn er der Frage auf den Grund geht, was es ist, dass uns etwas als menschlich empfinden lässt.



**Abbildung 2.5:** Unterschiedliche Renderings des Scans von Paul Debevec sowie dessen 3D-Modell aus verschiedenen Perspektiven (unten). Paul Debevec ist *Chief Visual Officer* sowie *Research Professor* am USC im Fachbereich *Computer Science*. In seiner Doktorarbeit hat er sich mit der Technik des bildbasierten Modellierens beschäftigt. Später war er dann Vorreiter auf dem Gebiet des *Image based lightings*. Am *Institute for Creative Technologies* der Universität in *Southern California* war er Leiter in der Entwicklung der *Lightstages*, welche für zahlreiche Filme, wie auch später noch am Beispiel von *Matrix Reloaded* erläutert wird, verwendet wurden und mit welcher das Projekt der *Digital Emily* ermöglicht wurde (<http://ict.usc.edu/profile/paul-debevec/>, Stand: 07. Dezember 2015). Um das realistische Rendering einfacher zu machen, wurde auf seine Kopfbehaarung verzichtet (<http://goo.gl/8RDw20>, Stand: 07. Dezember 2015).

### 2.2.4 Realistic Human Faces Rendering - Matrix



**Abbildung 2.6:** Eines der *High Poly*-Modelle mit über 10.000 *Triangles*, rechts daneben ist dargestellt, wie von einem der Schauspieler mit 30 Kameras um seinen Kopf herum positioniert einige Aufnahmen für eine *Reflectionmap* gemacht wurden. In der zweiten Reihe das erste Bild zeigt diese *Reflection Map* als *UV-Map*, rechts daneben ist eine *2D Light Map* abgebildet. In Reihe drei sind zwei am computergenerierte Bilder (links und Mitte) zum Vergleich mit einem realen Foto des Schauspielers Keanu Reeves (rechts) zu sehen. Ganz unten auf der Abbildung ist ein vollständig virtuell erzeugtes Bild aus dem Film zu sehen. Die Abbildung ist entnommen aus [BL03].

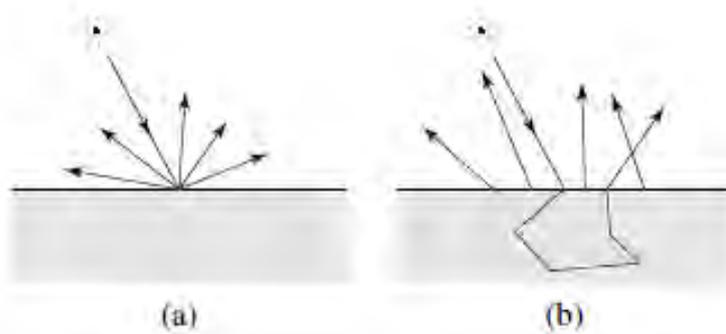
In "Realistic Human Faces Rendering for 'The Matrix Reloaded'" [BL03] beschreiben George Borshukov und J.P.Lewis, dass die absolut größte Herausforderung in der Computergrafik das realistische Rendering von menschlichen Gesichtern ist. Sie erläutern, dass aufgrund

der Tatsache, dass Menschen von Geburt an antrainiert wird, die menschlichen Gesichter in ihrer Umgebung genau zu beobachten und wahrzunehmen, unser Gehirn daher genaue Vorstellungen davon hat, wie jede Nuance eines menschlichen Gesichtes auszusehen hat. Das Rendering menschlicher Haut wird aus zwei Gründen besonders komplex zum einen dadurch, dass die Struktur extrem detailreich ist und zum anderen dadurch, dass Haut transluzent ist.

Für die Renderings in *Matrix Reloaded* wurden hochauflösende Scans der Schauspieler angefertigt, die, wie Borshukov et al. beschreiben, dessen Gesichter bis auf 100 Micrometer genau darstellen können. Dadurch erhielten diese Modelle aber extrem hohe Polygonzahlen von bis zu 10000 *Triangles* und mussten für die Produktion auf *Lowresolution*-Modelle von 5000 *Quads* reduziert werden. Die Details wurden dann über zuvor extrahierter *Lightmaps* wieder beim Rendering beziehungsweise übers *Shading* hinzugefügt. Um die entsprechenden *Reflectionmaps* für die Haut zu erzeugen, wurden 30 Kameras um den Kopf des Schauspielers herum platziert. Die Schauspieler wurden mit einer Vielzahl an Lichtern aus unterschiedlichen Richtungen beleuchtet und das Setup wurde einer sehr sorgfältigen *Color correction* unterzogen. Auf diese Weise konnte die Reflektionsverteilung anhand einer bildbasierten *Reflektion Map* berechnet werden, um so möglichst nah an die physikalische Korrektheit heranzukommen. Dass ohne die Simulation von *Subsurface Scattering* keine realistische Visualisierung menschlicher Haut möglich ist, wurde auch den Machern von *Matrix Reloaded* schnell klar. Zur Umsetzung der *Subsurface Scattering*-Simulation wurden unterschiedliche Ansätze kombiniert, um die Kosten gering zu halten und gleichzeitig gute Ergebnisse zu erzielen. So wurde eine *2D-Map* genutzt, um die vom Gesicht in Kamerarichtung reflektierte Beleuchtung zu speichern. Für den transluzenten Effekt an lichtdurchlässigeren Teile im Gesicht, wie zum Beispiel den Ohren, wurde traditionelles *Raytracing* verwendet. [BL03]

## 2.3 Stand der Technik in Bezug auf Subsurface Scattering

In unserer Umwelt sind wir stets von einer Vielzahl transluzenter also lichtdurchlässiger Materialien umgeben. Was dazu führt, dass eine akkurate Nachbildung der Streuung des Lichts in derartigen Materialien elementar für die Produktion fotorealistischer Renderings ist. Beispiele für lichtdurchlässige Materialien, wie sie zum Teil auch schon in Kapitel 1 aufgeführt wurden, reichen von Schnee über Milch, Marmor, Wachs, verschiedenen Pflanzen bis hin zu menschlicher Haut. Der Grad an Lichtdurchlässigkeit variiert dabei sehr stark, aber sie alle haben gemeinsam, dass das Licht nicht nur an der Oberfläche reflektiert wird, sondern auch unter die Oberfläche eindringt, wo es gestreut wird, um gegebenenfalls auch an einer anderen Stelle des Objektes wieder auszutreten. Dies wird als *Subsurface Scattering* bezeichnet. Durch dieses besondere Streuverhalten der Lichtstrahlen erhalten alle transluzenten Materialien ein sehr weiches Erscheinungsbild. Des Weiteren kann das Licht durch das Objekt durchscheinen, was vor allem dann auftritt, wenn transluzente Materialien von hinten beleuchtet werden.



**Abbildung 2.7:** Ein Schaubild zur Gegenüberstellung des Modells von direkter Reflektion und *Subsurface Scattering* [JMLH01].

In der 3D-Computergrafik wurde klassischerweise versucht, anhand einer *Lambertian* diffusen Reflektion eine Annäherung an das *Subsurface Scattering* zu erreichen [JB02]. Später wurde dieser Ansatz von Hanrahan und Krueger [HK93] in ihrer Arbeit "Reflection from layered surfaces due to subsurface scattering" aus dem Jahre 1993 verbessert, indem sie *Subsurface Scattering* in mehrschichtigen Oberflächen in Bezug auf die eindimensionale lineare Transporttheorie abgeleitet haben und so einen analytischen Term für *Single Scattering* erhielten, der es ermöglicht, gerichtete Effekte zu verrechnen. Des Weiteren haben sie eine Methode vorgeschlagen, bei welcher Photonen durch das Material hindurch verfolgt werden. Für ihr finales Modell haben sie dann aber doch eine *BRDF* (*Bidirectional Reflectance Distribution Function*) nach [NRH<sup>+</sup>77] verwendet, welche nur eine Annäherung an die *BSSRDF* (*Bidirectional Scattering Surface Reflectance Distribution Function*) ist (genauere Erläuterungen zu den beiden Funktionen finden sich in Kapitel 3). Das Problem an diesem Modell ist, dass die *Bidirectional Reflectance Distribution Function* immer davon ausgeht, dass das Licht das Material an genau derselben Stelle wieder verlässt, an der es auftritt und daher kann die Funktion nur zur Berechnung der Streuung an einem einzigen Punkt verwendet werden, nicht aber für die Simulation des Transportes von Licht unter die Materialoberfläche.

Es gibt eine Reihe von unterschiedlichen Ansätzen dazu, wie der Transport des Lichtes und die Streuung unter der Oberfläche simuliert werden können, die im Folgenden vorgestellt werden sollen.

### 2.3.1 Methoden zur Simulation von Subsurface Scattering

Dorsey et al [DEJ<sup>+</sup>06] haben in ihrer Arbeit "Modeling and Rendering of Weathered Stone" im Jahre 1999 anhand von *Photon Mapping Subsurface Scattering* simuliert.

Ebenfalls 1999 haben Jensen et al. in "Rendering of wet materials" [JLD99] mit Hilfe von *Path-Tracing Subsurface Scattering* für nasse Materialien simuliert.

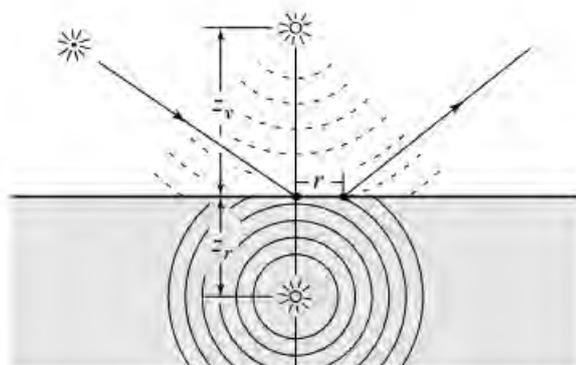
In "Monte Carlo Evaluation Of Non-Linear Scattering Equations For Subsurface Reflection" aus dem Jahr 2000 [PH00], beschreiben Pharr et al. ein *Framework* basierend auf einer mathematischen Gleichung, die anhand eines nichtlinearen Integrals das Streuverhalten eines Objektes oder eines Volumens beschreibt.

Stam et al. haben 2001 in ihrer Arbeit "An Illumination Model for a Skin Layer Bounded by Rough Surfaces" [Sta01] eine diskret geordnete Lösung der Strahlungstransportgleichung verwendet, um die multiple anisotrope Streuung für die Schichten menschlicher Haut zu visualisieren, welche durch zwei grobe Oberflächen begrenzt werden. Des Weiteren haben Stam et al. in ihrer Arbeit eine Ableitung einer bidirektionalen Übertragungsverteilungsfunktion (*Bidirectional Transmittance Distribution Function* kurz *BTDT*) gebildet, um das *BRDF*-Modell damit zu ergänzen.

Die genannten Verfahren sind alle in der Lage, *Subsurface Scattering*-Effekte zu erzeugen und Bilder mit dessen Visualisierung zu produzieren, aber sie sind alle sehr kostenintensiv in Bezug auf die Berechnung der Renderings und die Bildsynthese ist daher sehr langsam und benötigt enorme Rechenleistung, vor allem im Vergleich zu Renderings von Materialien die vollständig opaque sind. Besonders für hochgradig *scatternde* Materialien sind die genannten Verfahren daher eher ungeeignet.

Für stark *scatternde* Materialien schlägt Joe Stam 1995 in "Multiple scattering as a diffusion process" [Sta95] daher die Verwendung der "Diffusion Theory" vor. Er löst eine Streuungsgleichungs-Annäherung unter der Verwendung einer Mehrgittermethode und verwendet die Methode um Wolken mit multiplem *Scattering* zu rendern.

2001 gelang Jensen et al. mit ihrer Arbeit "A practical model for subsurface light transport" [JMLH01] eine Vereinfachung und vor allem deutlich Steigerung der Effizienz bei der Simulation von *Subsurface Scattering*. Sie erreichen eine Annäherung an ein *BSSRDF*-Modell mit Hilfe einer genauen Lösung für ein *Single Scattering* und einer *Dipole Point Source*-Approximation für das *Multiple Scattering*. Dabei wird ein eintreffender Lichtstrahl in zwei virtuelle Punktquellen konvertiert, von denen eine positive oberhalb der Oberfläche und eine negative unterhalb der Oberfläche liegt. Über den Abstand dieser beiden Punktquellen, welcher von vielen Faktoren, wie beispielsweise dem Streukoeffizienten abhängt, wird ein *Diffusion*-Profil gebildet. Dieses Verfahren geht von der Annahme aus, dass das Material planar, unendlich groß und dick und außerdem homogen ist.



**Abbildung 2.8:** Ein Schaubild zur Visualisierung der *Dipole Point Source-Approximation* die Jensen für seinen Algorithmus verwendet [JMLH01].

Der Algorithmus ist sehr schnell. Am Beispiel der Marmorstatue (siehe Abbildung 2.9) konnte die Renderzeit von 1250 Minuten unter Benutzung einer vollen *Monte Carlo-Simulation* auf 5 Minuten mit geringen Unterschieden in den visuellen Effekten reduziert werden. Damit haben sie die praktische Umsetzung des *Subsurface Scattering*-Effektes erst wirklich realisierbar gemacht. Und moderne Renderer wie V-Ray der Chaosgroup haben in ihren *Subsurface Scattering*-Shadern Algorithmen implementiert, die auf dem Verfahren von Jensen basieren.



**Abbildung 2.9:** Eine Gegenüberstellung von Renderings rechts nur mit *BRDF* gerendert, Renderzeit: 2 Minuten (aber kein *Subsurface Scattering*), in der Mitte das Rendering mit dem Approximations-Algorithmus von Jensen et al. Renderzeit 5 Minuten und rechts eine *Monte Carlo-Simulation*: Renderzeit 1250 Minuten [JMLH01].

Um ihr Modell verifizieren zu können, haben Jensen et al. mit Hilfe eines Versuchsaufbaus (siehe Abbildung 2.10) die Parameter realer transluzenter Gegenstände gemessen und mit ihren Ergebnissen verglichen. Im aktuellen V-Ray *FastSSS2-Shader* sind als *Presets* von der Chaosgroup bis auf drei Ausnahmen (*Chicken1*, *Chicken2* und *Apple*) exakt diese Objekte übernommen worden.

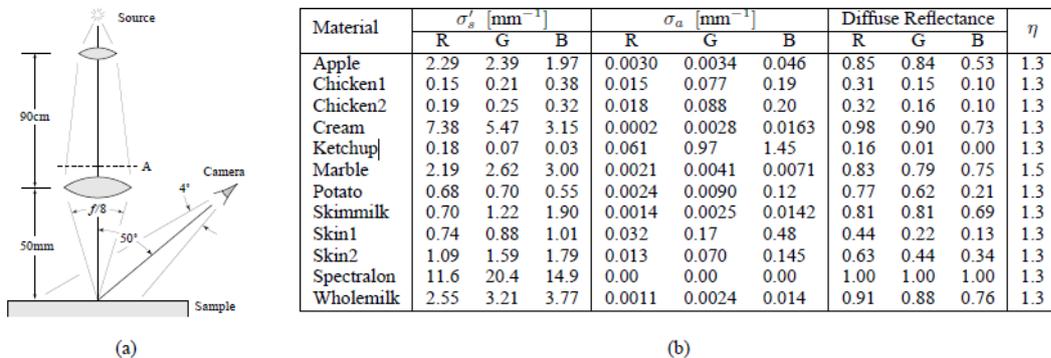


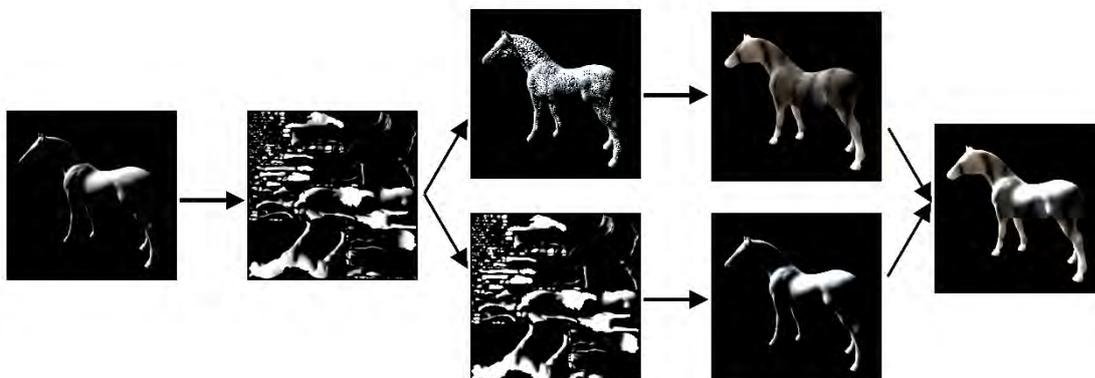
Figure 5: (a) Measurement apparatus, (b) measured parameters for several materials.

**Abbildung 2.10:** Eine Skizze des Versuchsaufbaus sowie die Tabelle mit den Messwerten, die Jensen et al. zur Verifizierung ihrer Arbeit gemacht haben [JMLH01].

Im Jahre 2002 haben Jensen und Buhler dann mit ihrer Arbeit "A rapid hierarchical rendering technique for translucent materials" eine Verbesserung der Renderzeiten auf bis zu 7 Sekunden pro Frame für komplexe Beleuchtungssituationen erreichen können, indem sie das *Single Scattering* ignoriert und sich ausschließlich auf die *Multiple Scattering Diffusion Approximation* konzentriert haben.

### 2.3.2 Subsurface Scattering bei interaktiven beziehungsweise deformierbaren Objekten

Ein weiterer Schwerpunkt in Bezug auf die Visualisierung des *Subsurface Scattering*-Effektes in der 3D-Computergrafik ist die Modellierung von *Subsurface Scattering* bei interaktiven oder auch deformierbaren Objekten. Eine Reihe von Arbeiten, welche sich mit diesem Schwerpunkt auseinandersetzen, werden in diesem Abschnitt behandelt.



**Abbildung 2.11:** Darstellung des *Workflows* des von Lensch et al. entwickelten Algorithmus [LGB<sup>+</sup>03].

Lensch et al. beschreiben in “Interactive Rendering of Translucent Objects” [LGB<sup>+</sup>03] eine Rendering-Methode, bei der die Beleuchtung und die Perspektive der Betrachtung interaktiv verändert werden können. Dafür werden in einem Vorverarbeitungsschritt für jeden Punkt auf der Oberfläche des Objektes, die Antworten von auftreffenden Lichtimpulsen berechnet und auf zwei unterschiedliche Weisen abgespeichert. Die Art der Speicherung erfolgt in Abhängigkeit davon, ob es sich um eine globale oder lokale Rückmeldung des Lichtimpulses handelt. Die lokalen Effekte werden als *Per-texel Filter Kernel* modelliert, die als *Texture Map* eingesetzt werden und die einfallende Beleuchtung repräsentieren. Die globale Resonanz wird als *Vertex zu Vertex-Durchsatz* für das *Mesh* des Objektes gespeichert. Während des Renderings des Bildes wird die Beleuchtungskarte des Objektes dann in Bezug auf die aktuelle Beleuchtungssituation berechnet und durch den vorberechneten Kernel gefiltert. Für das finale Ergebnis werden die globalen und lokalen Ergebnisse kombiniert.

In ihrer Arbeit “Interactive subsurface scattering for translucent meshes” [HBV03] stellen Hao et al. ein *Framework* vor, welches mit einem einfachen lokalen Beleuchtungsmodell *Subsurface Scattering*-Effekte erzielen kann. Sie begründen ihren Ansatz darauf, dass *Subsurface Scattering* auf Grund seines exponentiellen Abfalls einen relativ geringen Einfluss auf benachbarte Objekte hat, aufgrund dessen sie für die Erschaffung dieses Effektes lediglich eine Annäherung an das *BSSRDF*-Modell unter ausschließlicher Verwendung von lokaler Beleuchtung verwendet haben, womit sie die ihrer Meinung nach wichtigsten Features des *Subsurface Scattering*-Effektes erzeugen können. Ziel ihrer Arbeit ist es, die Visualisierung von *Subsurface Scattering* für interaktive Frameraten zu ermöglichen, sodass auch diese den zusätzlichen Fotorealismus für Renderings von transluzenten Materialien erhalten. Dafür wird in ihrem Verfahren zunächst ein Vorverarbeitungsschritt durchlaufen, welcher die Informationen über die benachbarten Vertex sammelt. Nach der Vorverarbeitungsstufe besteht die Prozedur aus zwei Phasen, welche zur Laufzeit abgewickelt werden. In der ersten Phase wird die Reflektion und die Übertragung des Lichtes auf die Oberfläche berechnet. In der zweiten Phase wird dann für das finale Ergebnis eine Art “bleeding” auf die benachbarten Flächen des Vertex für den *Scattering*-Effekt erzeugt. Mit der Entwicklung ihres Algorithmus haben sie es geschafft, die vorherige Berechnung von Jensen aus “A rapid hierarchical rendering technique for translucent materials” [JB02] deutlich zu beschleunigen, wie am Beispiel ihrer Gegenüberstellung der Renderings des *Utah Teapots* (siehe Abbildung 2.12) zu sehen ist. Im Gegensatz zu Jensen et al., die eine Renderzeit von 7,5 Sekunden pro Frame benötigten, betrug die Renderzeit mit dem Algorithmus von Hao et al. 7 Frames pro Sekunde.



**Abbildung 2.12:** Ein Vergleich dreier Renderings des *Utah Teapots*: einmal ohne *Subsurface Scattering* (links), einmal das Rendering aus der Arbeit von Jensen et al. (Mitte) und das Rendering von Hao et al. (rechts) [HBV03].

Mit "Efficient rendering of local subsurface scattering" [MKB<sup>+</sup>05] schlagen Mertens et al. 2005 einen Ansatz zur interaktiven Simulation von *Subsurface Scattering* vor, bei welchem im Gegensatz zu den vorherigen Algorithmen, keine aufwändigen Vorberechnungen stattfinden sollen. Jedes Frame soll von Grund auf neu berechnet werden, sodass für jedes Frame virtuell alles verändert werden kann. Ihr Ziel ist es, deformierbare transluzente Objekte zu rendern, deren Beleuchtungssituation und Betrachtungsperspektive von Frame zu Frame veränderbar sind. Wichtig ist dabei, dass dies in interaktiven Frameraten geschehen soll. Für ihr Modell machen sich auch Mertens et al. zu Nutze, dass *Subsurface Scattering* abhängig vom Material und der Skalierung des Objektes ein lokaler Effekt bleibt. Für ihren Algorithmus schreiben sie das im vorherigen Abschnitt in diesem Kapitel beschriebene Modell von Jensen et al. [JMLH01], welches nicht für interaktives Rendering verwendet werden kann, um, sodass es als ein Intergral im Bildraum anstatt als Intergral über das Objekt funktioniert. Dieses kombinieren sie dann mit einem *Sampling*-Verfahren der *BSSRDF*-Funktion. Das Verfahren kann sowohl als Software, wie auch auf Grafikkhardware implementiert werden und erreicht über die Grafikkhardware interaktive Bildraten für das Rendering von *Subsurface Scattering*-Effekten für dynamische Modelle.

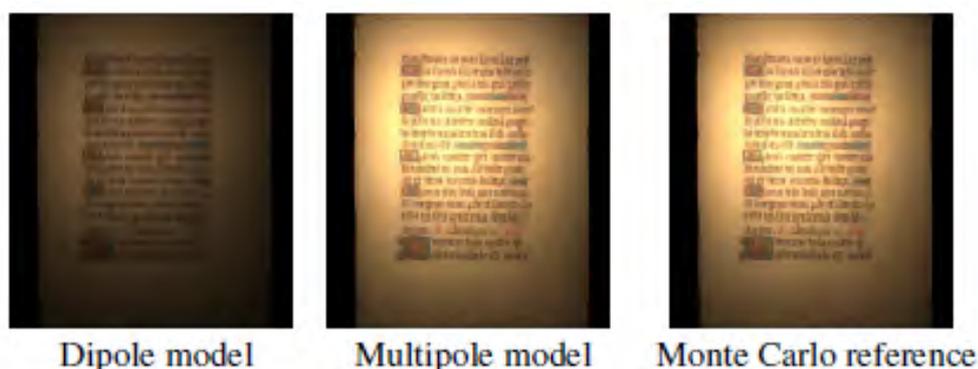


**Abbildung 2.13:** Ein Beispielrendering, welches mit dem interaktiven Verfahren von Mertens et al. produziert wurde. Es wurde auf Grafikkhardwarebasis gerendert mit einer Renderzeit von ca. 4-5 Frames pro Sekunde. [MKB<sup>+</sup>05]

### 2.3.3 Subsurface Scattering in Bezug auf Multilayer-Materialien

Abgesehen von interaktiven Anwendungen gibt es noch weitere Fälle in denen das im oberen Abschnitt dieses Kapitels beschriebene Modell von Jensen et al. aus dem Jahre 2001 [JMLH01] scheitert. Dünne Platten, bei denen Licht durchscheint sowie alle mehrschichtigen Materialien. Doch für genau diese Anwendungsfälle haben Donner und Jensen im Jahre 2005 in ihrer Arbeit "Light diffusion in multi-layered translucent materials" [DJ05] eine neue Technik entwickelt. Ihr Algorithmus basiert auf mehreren Dipolen, um die Streuung in dünnen Scheiben berechnen zu können. Um Fehlanpassungen in den Brechungsindizes an der Ober- und Unterseite der lichtdurchlässigen Scheiben zu berücksichtigen und um die Auswirkungen

von rauhen Oberflächen abbilden zu können, haben sie die *Multipol*-Theorie noch erweitern müssen. Um ihren Algorithmus auch auf mehrschichtige Materialien anwenden zu können haben sie die *Diffusion*-Profile der einzelnen Platten miteinander kombiniert und für das *Multiple Scattering* zwischen den Scheiben eine Variante der Kubelka-Munk-Theorie im Frequenzraum verwendet. Die Methode ist vor allem für Materialien, die besonders dünn oder eben mehrschichtig sind wie Papier, Farbe oder menschliche Haut geeignet.



**Abbildung 2.14:** Das Rendering eines Pergamentpapiers, welches von hinten beleuchtet wird. Verglichen wird Jensens ursprüngliches *Dipole Model* aus [JMLH01], das neue *Multipole Model* und das extrem kostenintensive *Monte Carlo*-Verfahren als Referenz [DJ05].

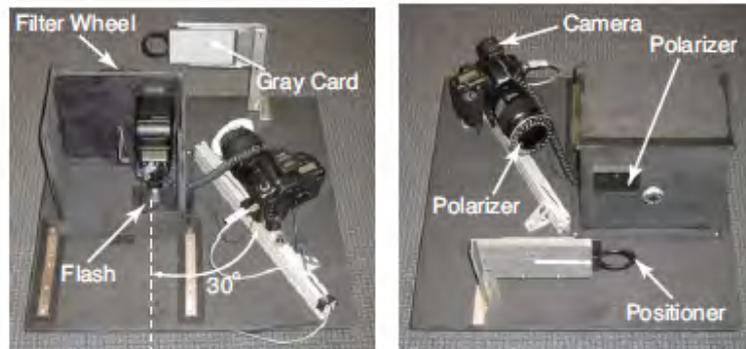
Donner et al. stellen in “Heterogeneous Reflectance Model for Acquiring and Rendering Human Skin” [DWD<sup>+</sup>08] aus dem Jahr 2008 ein geschichtetes, heterogenes spektrales Reflektionsmodell für das Rendering menschlicher Haut vor. Das Modell erfasst die Streuung des Lichtes zwischen den einzelnen Schichten, bei denen jede jeweils einen eigenen Satz an variierenden Absorptions- und Streuungsparametern haben kann. Um eine bessere Kontrolle und eine größere physikalische Korrektheit zu erhalten, fügen sie eine extrem dünne Absorptionsschicht zwischen die einzelnen Schichten in denen das *Scattering* auftritt (siehe Abbildung 2.15).



**Abbildung 2.15:** Eine Skizze des Modells, wie Licht in den multiplen Schichten gestreut wird und dabei die dünne Absorptionsschicht zwischen den Schichten durchdringt [DWD<sup>+</sup>08].

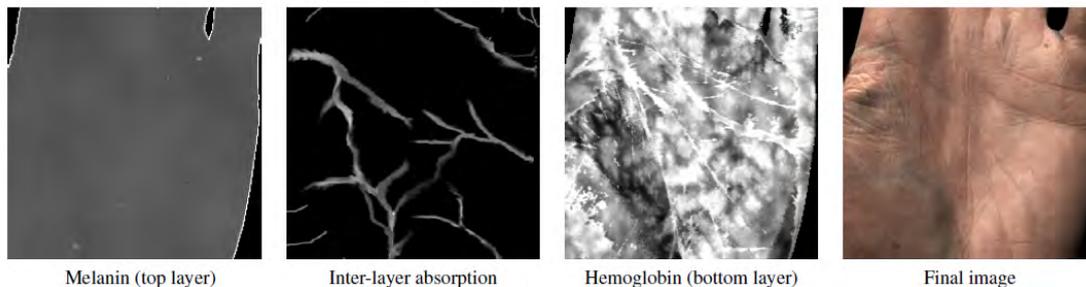
Die Parameter, die sie für ihr Model verwenden, erhalten sie über ein derzeit innovatives Verfahren, das mit der Produktion von multispektralen Aufnahmen beginnt (siehe Abbildung

2.16).



**Abbildung 2.16:** Aufbau zur Aufnahme der multispektralen Fotografien für die Parameterberechnungen [DWd<sup>+</sup>08].

Unter der Verwendung einer inversen Rendertechnik in Kombination mit bekannten chromomorphen Spektren suchen sie die optimalen Parameter für jedes Pixel des Hautstückes. Durch die Genauigkeit und Flexibilität ihres Modells schaffen sie es komplexe Hautstrukturen, wie Venen, Hautausschläge oder Sommersprossen oder auch Tattoos in visuellen Effekten wiederzugeben.

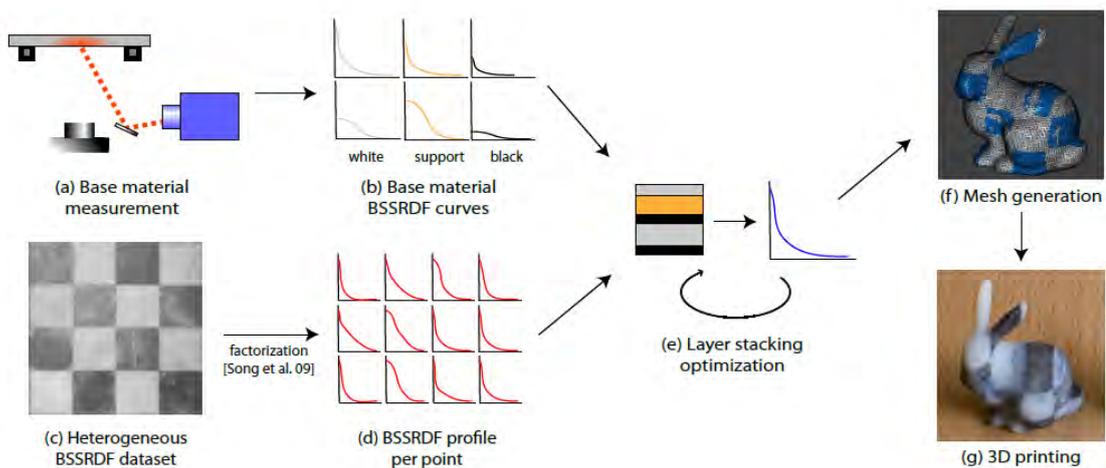


**Abbildung 2.17:** Handgezeichnete Parameterkarten der Streuungsschichten für Melanin und Hämoglobin sowie für die dünne Absorbtionsschicht zwischen den beiden *Layern*. Um die Details sichtbar zu machen, wurden sie stark vergrößert. Das letzte Bild (rechts) zeigt ein fertiges Rendering. [DWd<sup>+</sup>08]

### 2.3.4 Reproduktion von Materialien mit Subsurface Scattering mittels eines 3D-Druckers

In "Physical reproduction of materials with specified subsurface scattering" [HFM<sup>+</sup>10] untersuchen Hasan et al. eine komplette *Pipeline*, in der es darum geht Objekte mit *Subsurface Scattering*-Effekten mit einem 3D-Drucker zu produzieren. Dabei wollen sie sich der Herausforderung stellen nicht mehr länger nur Materialien drucken zu können, die vollständig *opaque* sind, sondern auch mehrschichtige transluzente Objekte mittels 3D-Druck zu realisie-

ren. Dafür sollen die nötigen Parameter zunächst gemessen, die Modelle dann am Computer erstellt und letztendlich als fertiges Produkt mit vorher bestimmtem Streuungsverhalten erzeugt werden. Das Verfahren beginnt mit der Messung der Streuungseigenschaften eines gegebenen Satzes an Basismaterialien. Dabei werden ihre radiale Reflektion und ihre Übertragungsprofile bestimmt. Sie beschreiben ein mathematisches Modell, das Vorhersagen über die Profile verschiedener Zusammensetzungen der Basismaterialien mit beliebiger Dicke trifft. Im umgekehrten Prozess kann dann zu einem gewünschten Reflexionsprofil, das entsprechend am besten passende Schichtverbundmaterial berechnet werden. Der Algorithmus des Modells durchsucht effizient alle Möglichkeiten an Kombinationen und sortiert die aus, die durch physikalische Begrenzungen nicht möglich sind. Mit ihrem Modell stellen sie sowohl homogene als auch heterogene Verbundwerkstoffe mit Hilfe eines Multimaterial-3D-Druckers her, die der Reproduktion der Streuungseigenschaften von komplexen Materialien sehr nahekommen. Für die Simulation des *Subsurface Scatterings* verwendet sie die ebenfalls in diesem Kapitel beschriebene *Multi-Layer-Material-Methode* von Donner und Jensen [DJ05] und sehen durch die Demonstration der Übereinstimmung von Simulation und realen Materialien diese Methode als durch sie bestätigt an.



**Abbildung 2.18:** Überblick über die entwickelte *Pipeline* von Hasan et al. zur Produktion von 3D-Druckerzeugnissen mit *Subsurface Scattering*-Effekten [HFM<sup>+</sup>10].

### 2.3.5 Vergleich unterschiedlicher Rendersysteme in Bezug auf Subsurface Scattering

In seiner Arbeit “Photorealistic physically based render engines: a comparative study” [PR12] hat Francisco Pérez Roig die Renderer:

- 3Delight Raytrace
- MentalRay
- V-Ray

- MaxwellRender
- Fryrender

in Bezug auf fotorealistisches Rendering miteinander verglichen. Ein Aspekt, den er dabei berücksichtigt, ist wie gut die einzelnen Renderer *Subsurface Scattering* realisieren können. Für die Vergleichsreihe wurden mit jedem Renderer jeweils 9 Renderings mit unterschiedlichen *Subdiv*-Einstellungen gemacht und jeweils die Renderzeiten und das *CW-SSIM* gemessen. *CW-SSIM* steht für *Complex Wavelet Structural Similarity Index*. Über diesen Index kann die Qualität des Bildes angegeben werden. Bei dem Verfahren wird der Qualitätsmaßstab zunächst über ein möglichst fehlerfreies "Originalbild" festgelegt. Dann vergleicht das Verfahren Pixelmuster der beiden Bilder über die Luminanz und den Kontrast und je nach Abweichungen zum Original wird ein Index zugewiesen.

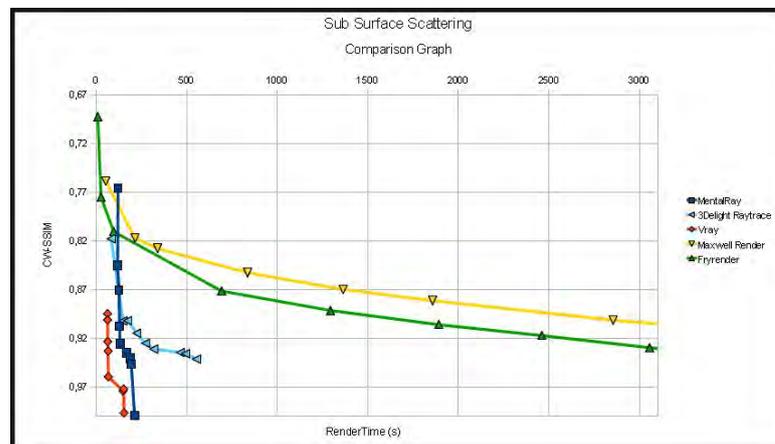


**Abbildung 2.19:** Darstellung wie Bilder mit unterschiedlichen Bildstörungen *Complex Wavelet Structural Similarity Index*-Werte (kurz *CW-SSIM*) zugeordnet wurden.[PR12]

Eine erste Unterteilung lässt sich zwischen *biased* und *unbiased* Renderersystemen machen. Es wird auch von voreingenommenen und nicht voreingenommenen Systemen gesprochen. *Unbiased* Renderverfahren berechnen das Bild mit einer schrittweisen Annäherung über einen *Monte Carlo*-Algorithmus. Das bedeutet, dass das Bild anfangs eine große Varianz hat und damit fehlerhaft ist, was sich in Form von Rauschen im Rendering widerspiegelt. Dieses

## 2. STAND DER TECHNIK

Rauschen wird durch weiterer Annäherung schrittweise minimiert und das Ergebnis perfektioniert. Die Annäherung an ein ideales Bild kann unendlich weitergeführt werden, daher muss entweder anhand einer zeitlichen Begrenzung eine Bedingung geschaffen werden, wann das Rendering gestoppt wird, oder das Rendering wird manuell beendet, sobald das Ergebnis zufriedenstellend ist. *Biased* Renderer verwenden geschwindigkeitsoptimierende Verfahren wie *Lightcache* und *Irradiance Map*, die vor dem eigentlichen Rendering des Bildes ablaufen, um die Beleuchtung zu berechnen. *Biased* Rendering bietet dem Nutzer Kompromisse zwischen Qualität und Renderzeiten, die er selber über das *Sampling* steuern kann. *MaxwellRender* und *Fryrender*, die beiden *unbiased* Systeme, benötigen eine deutlich höhere Renderzeit, um an gute Ergebnisse zu gelangen, wie die anderen Systeme und sind daher für die Produktion von Bildern mit *Subsurface Scattering* laut Francisco Pérez Roig, weniger geeignet wie die anderen.



**Abbildung 2.20:** Graph über die Ergebnisse der Messungen der fünf Rendersysteme. Der gelbe und der grüne Graph entsprechen den beiden *unbiased* Systemen und man erkennt, dass sie auf der X-Achse, welche die Renderzeit widerspiegelt, deutlich weiter gehen als die anderen drei Graphen, welche für die drei *biased* Systeme stehen [PR12].

Für die *biased* Rendersysteme bleiben nun also noch drei "Konkurrenten" übrig: *MentalRay*, *V-Ray* und *3Delight Raytrace*. *3Delight Raytrace* ist der langsamste der drei und fällt damit raus. *V-Ray* und *MentalRay* sind sich relativ ähnlich, wobei *V-Ray* noch einen etwas besseren Stand hat und nach einer Renderzeit von 154 Sekunden eine CW-SSIM-Rate von knapp 100 Prozent erreicht, welche *MentalRay* erst nach 252 Sekunden schafft. *V-Ray* ist also die beste Wahl, wenn es um Rendering von fotorealistischen *Subsurface Scattering*-Effekten geht.

Ein weiteres Entscheidungskriterium, das für *V-Ray* spricht, ist die Tatsache, dass die *Chaosgroup* seit der aktuellen Version *V-Ray 3.1* mit dem neu implementierten *Progressive*-Renderverfahren als derzeit einziger Renderer die Möglichkeit bietet, innerhalb des Renderers zwischen einem *biased* und einem *unbiased* Rendering-System zu wechseln.

## 2.4 Zusammenfassung

Nachdem *3D-Artists* auf der ganzen Welt jahrzehntlang an ihrem Handwerk gefeilt und ihre *Tools* perfektioniert haben, um glaubwürdige organische Wesen realisieren zu können, ist die Umsetzung von digitalen Menschen eine der größten Herausforderung in der 3D-Computergrafik. Einer der Hauptgründe dafür wird oft im *Uncanny Valley* gesehen, einem Konzept, dass, nachdem es erstmals 1970 von einem japanischen Robotiker namens Mori eingeführt wurde, vor allem in den letzten Jahrzehnten immer stärkere Aufmerksamkeit nicht nur in der Mensch-Maschine-Interaktion, sondern überwiegend auch in der Filmindustrie beziehungsweise der Computergrafik erfahren hat. Über die Theorie, die besagt, dass Roboter oder übertragen auf die Welt der Computergrafik digitale Charaktere, zunächst mit zunehmender Ähnlichkeit dem Menschen gegenüber eine ansteigende Akzeptanz erfahren, diese aber ab einem bestimmtem Grad schlagartig abfällt und erst wieder ansteigt, wenn der Android beziehungsweise der digitale Charakter nicht mehr von einem realen Menschen zu unterscheiden wäre, werden eine Menge Nachforschungen betrieben. Während einige mit kritischen Auseinandersetzungen versuchen, das Konzept zu be- oder widerlegen, arbeiten andere daran, die Wurzeln seines Ursprungs aufzudecken, oder einen Weg zu finden, das unheimliche Tal endlich zu überwinden. Mit der Digital Human League haben sich einige der besten herausragenden Persönlichkeiten aus Wissenschaft und Industrie zusammengeschlossen, um sich mit vereintem Wissen der Herausforderung zu stellen, glaubwürdige *Digital Humans* zu realisieren und das *Uncanny Valley* zu überwinden. Dafür haben sie das *Open Source*-Projekt *Wiki Human* ins Leben gerufen.

Bei der Umsetzung von möglichst realistischen digitalen Menschen spielt das Rendering der Haut eine übergeordnete Rolle. Dies ist aus zwei Gründen besonders komplex. Zum einen ist menschliche Haut enorm detailreich und zum anderen ist sie transluzent. Transluzente also lichtdurchlässige Materialien haben die Besonderheit, dass Licht nicht nur direkt von der Oberfläche reflektiert wird, sondern unter die Oberfläche und somit in das Material eindringt und dort gestreut wird, was der Substanz ein außergewöhnliches weiches Aussehen verleiht. Besonders, wenn die Objekte von hinten beleuchtet werden, führt das *Subsurface Scattering* auch zu einem Effekt des Durchleuchtens. Es ist nicht trivial, dieses Phänomen in der 3D-Computergrafik nachzubilden, da die normale BRD-Funktionen, die für die Berechnung von Reflektionen verwendet werden, nicht ausreichen, um den Transport des Lichtes unter die Oberfläche der Geometrie zu berechnen. Eine Reihe unterschiedliche Ansätze waren zwar erfolgreich, aber viel zu kostenintensiv in Bezug auf die Rechenleistung und die Renderzeiten. Der richtige Durchbruch gelang erst 2001 mit dem Verfahren von Jensen et al., die es in ihrer Arbeit "A practical model for subsurface light transport" mit einer Annäherung an eine *BSSRD*-Funktion geschafft haben ein Model zu entwickeln, dass es ermöglichte, gute Ergebnisse für *Subsurface Scattering*-Simulationen in adäquaten Renderzeiten zu erhalten. Ein Jahr später konnten Jensen und Buhler die Renderzeiten sogar noch einmal verbessern und das auch für komplexe Beleuchtungssituationen.

Im Laufe der Jahre gab es weitere Verbesserungen, die es ermöglichen, *Subsurface Scattering* auch interaktiv für deformierbare Objekte zu berechnen sowie Algorithmen für mehrschichtige und auch sehr dünne Materialien sind entwickelt worden.

Auch für die Reproduktion von Materialien mit *Subsurface Scattering* mit einem 3D-Drucker wurde eine *Pipeline* entwickelt, welche in diesem Kapitel vorgestellt wurde.

Da in der hier vorliegenden Arbeit analysiert werden soll, inwiefern sich die unterschiedlichen zu Verfügung stehenden Varianten der Umsetzung von *Subsurface Scattering* mit dem Renderer *V-Ray* der *Chaosgroup* in ihrer Handhabung und in ihren Ergebnissen unterscheiden und welche der Möglichkeiten gegebenenfalls die physikalisch plausibelsten Resultate liefert, werden, bevor anschließend mit der Methodik dieser Arbeit begonnen wird, im folgenden Grundlagen-Kapitel die drei unterschiedlichen Shader, welche in *V-Ray* zur Umsetzung von *Subsurface Scattering*-Effekten zur Verfügung stehen, im Einzelnen erklärt.

# Kapitel 3

## Grundlagen

### 3.1 Einleitung

Für das Verständnis dieser Arbeit ist die Kenntnis einiger Grundlagen von Nöten sowie zum einen die Grundzüge der Funktionsweisen der *Bidirectional Reflectance Distribution Function (BRDF)*, zu Deutsch Bidirektionale Reflektanzverteilungsfunktion, sowie der *Bidirectional Surface Scattering Reflectance Distribution Function (BSSRDF)* zu verstehen. Des Weiteren ist es unabdingbar den Aufbau der drei für *Subsurface Scattering*-Anwendungen geeigneten Shader des Renderers *V-Ray* der *Chaos Group* zu kennen, die Funktionen der wichtigsten Parameter nachvollziehen zu können sowie deren Zusammenspiel miteinander zu verstehen. Diese Grundlagen sollen daher in diesem Kapitel dargelegt und erörtert werden.

### 3.2 Opazität und Transluzenz

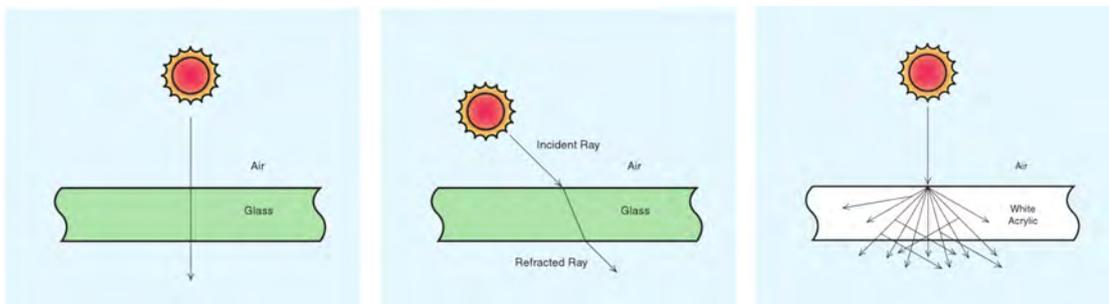
In Zusammenhang mit *Subsurface Scattering* ist immer wieder die Rede davon, dass der Effekt bei Materialien auftritt, welche transluzent sind. An dieser Stelle soll nun etwas genauer erläutert werden, was darunter zu verstehen ist. In der Physik wird von "Transmission" gesprochen, immer dann, wenn Wellen, in diesem Fall Lichtwellen, durch eine Materie hindurchdringen. Ein lichtdurchlässiges Material ist fähig einen bestimmten Anteil des auftretenden Lichtstroms oder Strahlungsflusses durch sich selbst hindurch zu lassen. Dieser Anteil der Lichtwellen wird als Transmissionsgrad bezeichnet und wird folgendermaßen berechnet:

$$T = \tau = \frac{I}{I_0}.$$

Dabei ist  $I_0$  die Wellenintensität vor Eindringen in das Material und  $I$  die Wellenintensität danach. Der Kehrwert des Transmissionsgrades ist die Opazität, also die Lichtundurchlässigkeit. Dieser Wert ist eine wichtige Angabe, beispielsweise für Papier oder Druckerfarbe, bestimmte Lacke oder Sonnenschutzkleidung. Für die Opazität lässt sich in der Optik über eine logarithmische Funktion die sogenannte optische Dichte oder Extinktion berechnen.

$$E_\lambda = -\log_{10} \frac{I}{I_0} = \log_{10} \frac{I_0}{I} = \log_{10} \frac{1}{\tau_\lambda} = \log_{10} O_\lambda$$

Die Opazität ist also das Maß für Lichtundurchlässigkeit, im Gegenteil zur Transluzenz der Lichtdurchlässigkeit. Hier ist allerdings noch einmal eine Unterscheidung zwischen Transluzenz und Transparenz wichtig. Transluzent sind teilweise lichtdurchlässige Materialien mit streuendem Verhalten und oft auch mehrschichtige Substanzen. Transparente Materialien haben ein geringes Streuverhalten und daher eine sehr geringe Trübung, sie sind klar und werden daher auch als durchsichtig bezeichnet, da meist deutlich zu erkennen ist, was hinter ihnen liegt, was bei transluzenten Materialien keineswegs der Fall sein muss. In diesem Zusammenhang wird auch von direkter Transmission und diffuser Transmission gesprochen, weil bei transparenten Materialien die Lichtstrahlen bei der Transmission direkt und grade durch das Material hindurch gehen und bei den nicht transparenten aber diffusen Materialien gestreut also diffus abgelenkt werden [HBF11].<sup>1</sup>



**Abbildung 3.1:** Gegenüberstellung direkter und diffuser Transmission von Licht durch ein transparentes (links) beziehungsweise transluzentes Material (rechts). Die mittlere Grafik zeigt die Refraktion, also die Ablenkung des Lichtes bei Übergang von einem Medium in ein anderes. Bildquelle: [HBF11]

## 3.3 BRDF und BSSRDF

### 3.3.1 BRDF

Die Interaktion der Oberfläche mit eintreffenden Lichtstrahlen fällt bei unterschiedlichen Materialeigenschaften trotz gleicher Beleuchtung verschieden aus. Bidirektionale Reflektanzverteilungsfunktionen (*Bidirectional Reflectance Distribution Functions*) sind eine Klasse von Funktionen mit denen diese unterschiedlichen Streuverhalten von Materialien beschrieben werden können. [Geb03] Erstmals eingeführt wurde sie 1965 von Fred Nicodemus [NRH<sup>+</sup>77]. Um eine Standardisierung der Darstellung und Berechnung der Reflektion zu erreichen wurde die Bidirektionale Reflektanzverteilungsfunktion 1977 vom *National Bureau of Standards* (USA) amtlich normiert definiert. Bei der Formel für die Berechnung handelt es sich um eine sechsdimensionale Funktion. In der Computergrafik wird jedoch meist von homogenen Materialien ausgegangen, wodurch die Funktion auf eine vierdimensionale vereinfacht werden kann. Mit dieser kann für eine gegebene Wellenlänge errechnet werden, wie

<sup>1</sup><http://www.analytik-news.de/Fachartikel/Volltext/fritsch27.pdf> Stand:10. Februar 2016

viel Licht von der Oberfläche reflektiert wird. Die Formel wird folgendermaßen dargestellt:

$$f(\Theta_{in}, \Phi_{in}, \Theta_{ref}, \Phi_{ref}) = f(L, V) \quad (3.1)$$

[Wat93] Die Variablen

$$\Theta_{in}$$

und

$$\Phi_{in}$$

beschreiben die Azimut- und Deklinationswinkel des Lichteinfalls. Die anderen beiden Variablen

$$\Theta_{ref}$$

und

$$\Phi_{ref}$$

stehen für Azimut- und Deklinationswinkel der Reflektionsrichtung. Damit hier jedoch eine Verkürzung erreicht werden kann, bezeichnet L die Richtung zur Beleuchtungsquelle und V die Richtung zum Beobachter auf die Oberfläche. Mit Hilfe der Irradianz und Radianz kann die Bidirektionale Reflektanzverteilungsfunktion wie folgt definiert werden:

$$fr(x, \omega_i, \omega_o) = \frac{1dLo(x, \omega_o)}{dE(x, \omega_i)} = \frac{dLo(x, \omega_o)}{Li(x, \omega_i) \cos \Theta_{id} \omega_i} \quad (3.2)$$

In der obigen Formel steht x für eine Position auf der Oberfläche von welcher das Licht reflektiert wird,  $\omega_i$  für die Richtung, aus der das Licht auf das Objekt trifft, und  $\omega_o$  für die Betrachtungsrichtung. L steht für die Radianz, auch Leuchtdichte genannt. Diese wird definiert über die Strahlungsflussdichte pro projizierte Einheitsfläche pro Einheitsraumwinkel. Sie gibt also an, wie viel Lichtenergie an einer Position auf der Objektoberfläche auftritt. Ein Raumwinkel ist analog zu einem ebenen Winkel im dreidimensionalen Raum, die Einheit wird als sr = steradian angegeben. Bei der Radianz wird zwischen einfallender und abgestrahlter Radianz unterschieden. Die Einheit der Radianz (Leuchtdichte) wird in  $\frac{Watt}{m^2 sr}$  angegeben. E ist die Intensität oder auch Beleuchtungsstärke oder Irradiance und ist definiert als Strahlungsflussdichte pro Raumwinkel. Die Einheit für die Beleuchtungsstärke ist  $\frac{Watt}{m^2}$ . Eine physikalisch korrekte Bidirektionale Reflektanzverteilungsfunktion muss zwei Bedingungen erfüllen:

1. die sogenannte "Helmholtz-Reziprozität"
2. die Energieerhaltung

Die Helmholtz-Reziprozität besagt, dass sich die Reflektion nicht verändern darf, wenn Einfallswinkel und Betrachtungswinkel getauscht werden. Energieerhaltung bedeutet, dass nicht mehr Energie durch Lichtstrahlen vom Material reflektiert werden kann, als eingetroffen ist. [Geb03].

#### 3.3.2 BSSRDF

Bei der *Bidirectional Scattering Surface Reflectance Distribution Function*, zu Deutsch Bidirektionale Oberflächenstreuungs-Reflektanzverteilungsfunktion, handelt es sich um eine Funktion, die im Gegensatz zur Bidirektionale Reflektanzverteilungsfunktion nicht nur die Reflektion des Lichtes an einem Punkt der Oberfläche berechnet, sondern auch den Transport des Lichtes unter die Oberfläche des Materials.

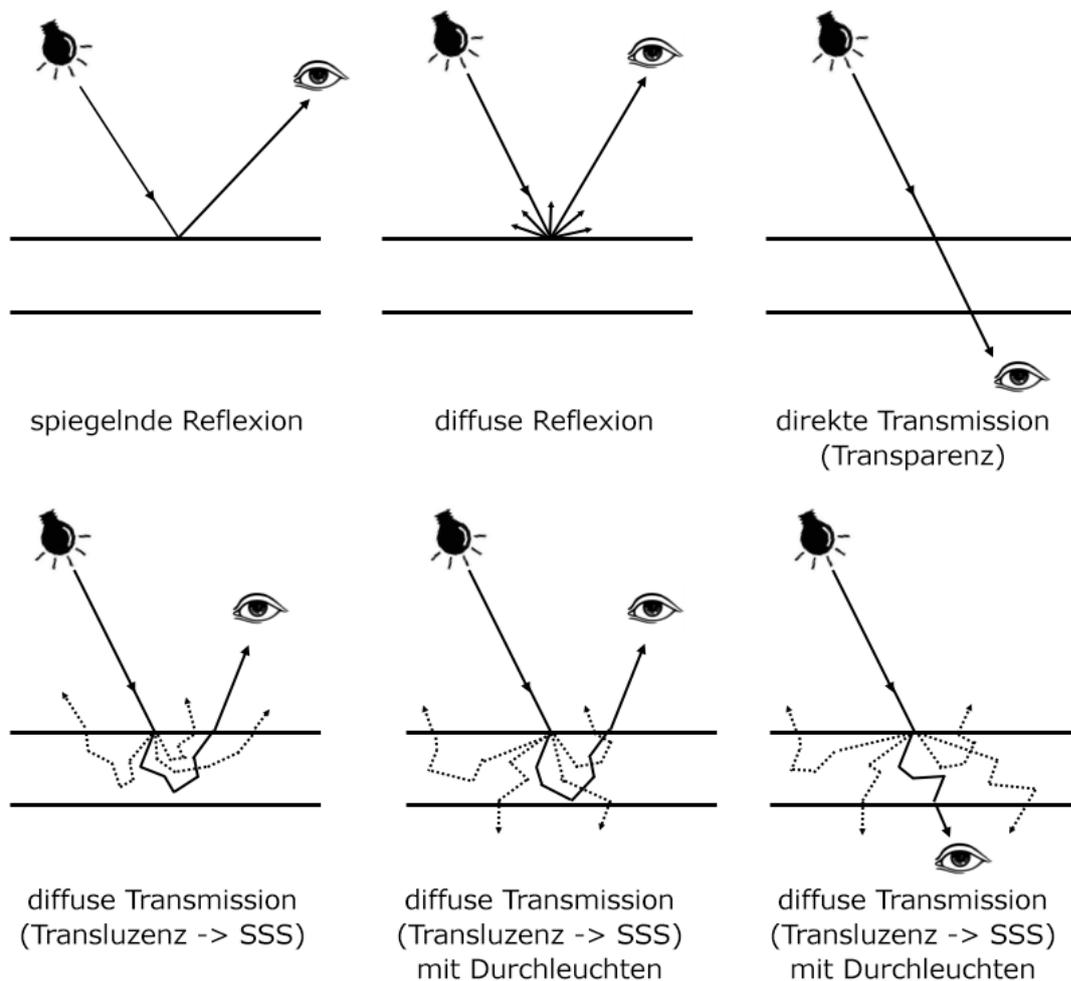
$$S(x_i, \omega_i, x_o, \omega_o) = \frac{dL(x_o, \omega_o)}{d\Phi(x_i, \omega_i)} \quad (3.3)$$

Das Licht kann unter der Oberfläche einfach oder mehrfach gestreut werden und an einer anderen Stelle des Objektes wieder austreten. Im Unterschied zur Bidirektionalen Reflektanzverteilungsfunktion werden die Einfalls- und Abstrahlungspunkte der Lichtstrahlen in der *Bidirectional Scattering Surface Reflectance Distribution Function* ebenfalls in die Berechnung miteinbezogen. Je nachdem, ob es sich, um einfache oder mehrfache Streuung handelt, erweitert das die Formel auf eine 8 bis 12-dimensionale Funktion. Für die Computergrafik ist die Implementierung dieser Funktionen, wie in Kapitel 2 bereits ausführlich erläutert wurde, zu komplex, daher werden lediglich Approximationen und vereinfachte Modelle der Funktion implementiert.

#### 3.3.3 Vorschlag eines erweiterten Visualisierungskonzeptes

Bei den auffindbaren Grafiken, die versuchen zu visualisieren, was bei *Subsurface Scattering* passiert und was den Transport unter die Oberfläche der Materials von einer herkömmlichen Reflektion unterscheidet, werden meist, wie auch in Abbildung 2.7 aus Kapitel 2, zwei Skizzen gegenübergestellt, die zwar für das grobe Verständnis der jeweiligen Vorgänge eine gute Basis darstellen, aber viele Fragen offen lassen. So wird für die Reflektion beispielsweise immer sowohl die direkte als auch die diffuse Reflektion dargestellt, für die Variante bei der das Licht in das Material eindringt, wird aber lediglich die Variante mit diffusem Verhalten des Lichtes visualisiert, was die Frage aufwirft, was im Falle von Transmission geschieht, wenn das Licht direkt durch das Material dringt. Auch bleibt die Frage unbeantwortet, in welchen Fällen nur die typische Weichheit des *Subsurface Scatterings* auftritt und wie die Skizze aussehen muss für die Sonderfälle, in denen auch der Effekt des Durchleuchtens erzeugt wird.

Daher hier der Ansatz einer erweiterten vollständigen Visualisierung:



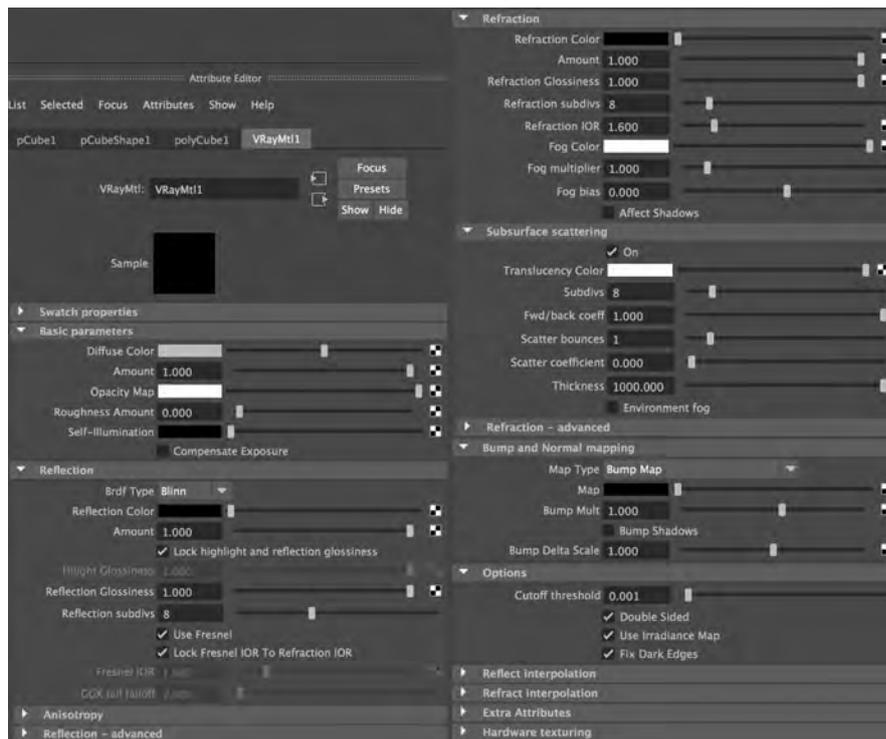
**Abbildung 3.2:** Eigener Vorschlag zum erweiterten Visualisierungskonzept.

Auf der ersten Grafik wird das Licht direkt reflektiert, dadurch entsteht eine Spiegelung oder spiegelnde Reflexion. Die zweite Grafik zeigt ebenfalls Licht, dass an der Oberfläche reflektiert wird, jedoch in diesem Fall diffus. Auf der dritten Grafik wird das Licht direkt ohne Ablenkung durch das Material transmittiert. Das Material ist damit transparent. Die vierte Grafik zeigt ebenfalls Licht, dass unter die Oberfläche des Materials eindringt, jedoch wird es hier diffus gestreut und tritt an einer anderen Stelle als es eingetreten ist, wieder aus. Eine solcher Materie wird als transluzent bezeichnet. Hier entsteht der typische *Subsurface Scattering*-Effekt, der dem Material den weichen Eindruck verleiht. Die fünfte Grafik zeigt ebenfalls diffuse Transmission, hier dringt das Licht allerdings tief genug in das Material ein, sodass auch Licht auf der anderen Seite des Objektes austritt. Wenn der Betrachter, wie in

der sechsten Grafik nun auf der entgegengesetzten Seite der Lichtquelle zum Material steht, so würde er den Effekt des Durchleuchtens wahrnehmen können, den *Subsurface Scattering* neben der typischen Weichheit erzeugen kann.

### 3.4 Der V-RayMtl Shader

Das *V-RayMtl* ist der Standard-Shader, der mit dem Renderer V-Ray verwendet wird. Mit diesem Shader lassen sich nahezu alle Materialien von diffusen über reflektierende bis hin zu durchsichtigen Substanzen, wie beispielsweise Glas nachbilden. Dieses Standard-Material bietet auch die Möglichkeit mit bestimmten Einstellungen einen *Subsurface Scattering*-Effekt zu visualisieren. Der grundsätzliche Aufbau und die Funktionsweise dieses Shaders und des Renderers V-Ray werden in dieser Arbeit als gegebenes Vorwissen vorausgesetzt, lediglich der Teil des Shaders, welcher speziell mit der Visualisierung von *Subsurface Scattering* verwoben ist, soll hier näher beleuchtet und die für die Realisierung dieses Effektes wichtigen Parameter sowie deren Zusammenspiel miteinander erörtert werden.

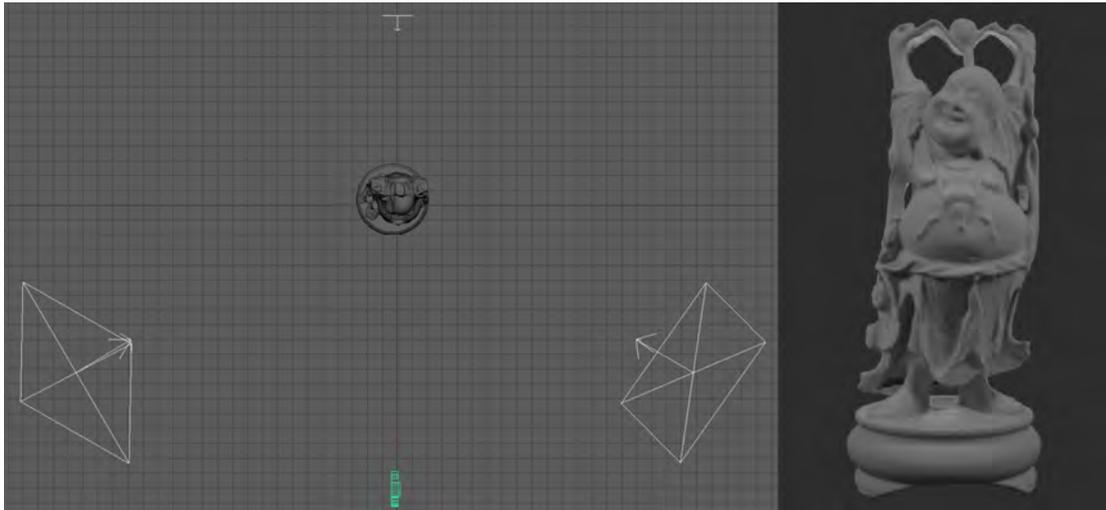


**Abbildung 3.3:** Die grundlegenden und für *Subsurface Scattering* wichtigen Attribute des *V-Ray Materials*.

Als Modell für die Renderings, um die Auswirkungen der einzelnen Parameter auf das Ergebnis der Visualisierung darzustellen, wird der *Stanford Happy Buddha*<sup>2</sup> gewählt, da es als Re-

<sup>2</sup><http://graphics.stanford.edu/data/3Dscanrep/> Stand: 26. Dezember 2015

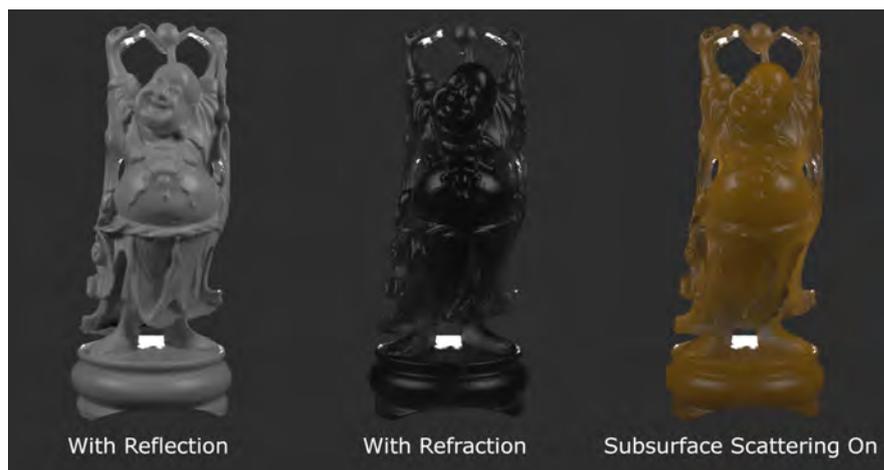
ferenzmodell einen ausreichenden Komplexitätsgrad hat, um Effekte gut erkennbar machen zu können, aber dabei gleichzeitig vergleichsweise niedrige Renderzeiten zu erzielen. *Shader-templates*, wie sie sonst oft verwendet werden, um Shadereinstellungen und Renderzeiten zu vergleichen, sind nicht sonderlich gut geeignet, um für den Vergleich von Parametern für das Rendering von *Subsurface Scattering*-Effekten herangezogen zu werden, da für die Visualisierung dessen ein gewisser Detailgrad von Nöten ist, den die zu einfachen Geometrien der *Shadertemplates* nicht gewährleisten können. Auch die unterschiedlichen Dicken der Materialien, die einen enormen Unterschied auf die Wirkung der Parametereinstellungen machen können, sind bei den Referenzmodellen der Stanford-Modelle besser gegeben. Für die Beleuchtung der Szene werden *V-Ray Rectangle Lights* gewählt, da diese aus dem Repertoire der V-Ray Lichter die größtmögliche Kontrolle bieten und nicht nur die Größe der Lichtquelle, sondern auch die Richtung, aus der das Licht ausgestrahlt wird, sehr genau gewählt werden kann. Beleuchtet wird die Szene mit drei Lichtquellen. Zwei davon strahlen das Objekt von vorne an und gewährleisten so eine Grundhelligkeit der Szene und Reflektionen auf dem Modell. Eine weitere Lichtquelle wird hinter dem Objekt platziert, um auch den *Subsurface Scattering*-Effekt des Durchleuchtens erzielen zu können und Parameter-Eigenschaften visualisieren zu können, die eine Beleuchtung des Modells von hinten benötigen.



**Abbildung 3.4:** Der Szenenaufbau für das Rendering der Referenzbilder zur Visualisierung der für *Subsurface Scattering* wichtigen Parametereigenschaften des V-Ray Materials (links) und ein Rendering des genutzten Referenzmodells (rechts).

Um mit dem Standard *V-Ray Material Subsurface Scattering*-Effekte erzielen zu können, sind einige Vorbereitungen von Nöten. Als erstes werden die per default im *V-Ray Material* ausgeschalteten Reflektionen für den Shader eingeschaltet. *Subsurface Scattering* tritt auch in der realen Welt nur bei lichtdurchlässigen Materialien auf, was bedeutet, dass, eigentlich logischer Weise, auch dem Shader zunächst eine gewisse Transluzenz gegeben werden muss, bevor beim Rendern ein Effekt des *Subsurface Scatterings* entstehen kann. Dies geschieht über die *Refraction*. Dabei kann die *Refraction Color* mit Graustufenwerten zwischen Schwarz

und Weiß oder der *Refraction Amount* auf einen Wert größer 0 bis 1 gestellt werden. Je nachdem, ob das Material nur auf teilweise oder auf vollständig transluzent eingestellt wird, wird dann noch ein diffuser Anteil und dementsprechend die diffuse Farbe dazu gerechnet oder nicht. Im nächsten Schritt muss für das Rendering von *Subsurface Scattering*-Effekten das Häkchen für "Subsurface Scattering on" im Bereich der *Subsurface Sattering*-Parameter des Shaders gesetzt werden.



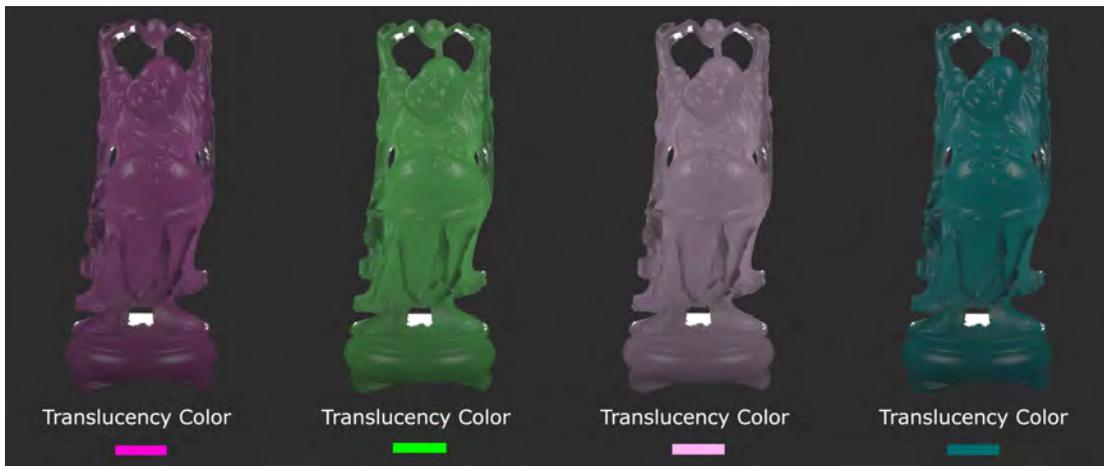
**Abbildung 3.5:** Das Referenzmodell in den drei unterschiedlichen Vorbereitungsstufen des Shaders für die Realisierung von *Subsurface Scattering* von links nach rechts. Zunächst wird die Reflektionen des Shaders aktiviert, im zweiten Schritt hat der Shader über das *Refraction*-Attribut die für das *Subsurface Scattering* notwendige Transluzenz bekommen (in diesem Fall eine vollständige Lichtdurchlässigkeit) und im dritten Schritt wird das *Subsurface Scattering* aktiviert.

#### 3.4.1 Subsurface scattering Parameter

Das *V-Ray Material* hat eine Reihe von Parametern, die speziell für das Rendering von *Subsurface Scattering*-Effekten angelegt wurden und solange ausgegraut sind, bis die Berechnung des *Subsurface Scatterings* durch Setzen des entsprechenden Häkchens für den Shader aktiviert wird. Auf diese Parameter soll im Folgenden näher eingegangen und beleuchtet werden, welche Auswirkungen die unterschiedlichen Einstellungen für diese Attribute im Einzelnen für das Rendering des Effektes haben.

##### Translucency Color

Mit dem *Translucency Color*-Parameter kann die Farbe festgelegt werden, die das Licht nach Durchdringen der Oberfläche des Materials annimmt und damit die Farbe des eigentlichen *Subsurface Scattering*-Effektes. Dieser Parameter ist stark abhängig von der *Fog Color* und kann durch dessen Wert beeinflusst werden. Für diesen Parameter kann jeder beliebige Farbwert oder auch Texturknoten verwendet werden. Je nach *Refraction*-Einstellungen wird dieser dann gegebenenfalls mit dem Wert der *Diffus Color* entsprechend anteilig vermischt.



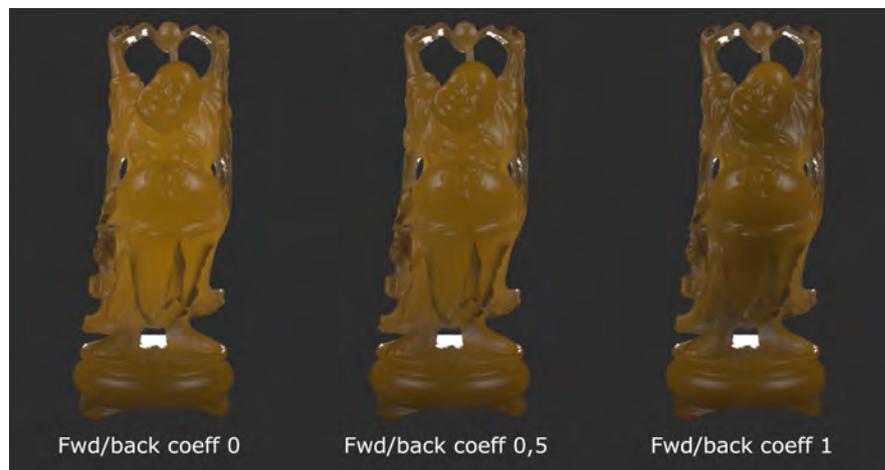
**Abbildung 3.6:** Die Auswirkungen von unterschiedlichen Farbwerten für die *Translucency Color* auf das Rendering des Shaders.

### Subdivs

Durch *Scattering* entsteht im Rendering von Materialien schnell Rauschen, dieses kann über eine Erhöhung der *Subdivs* minimiert werden. Dabei muss allerdings bedacht werden, dass eine Erhöhung der *Subdivs* auch immer eine Steigerung der Renderzeiten mit sich bringt. Daher sollte der Wert langsam gesteigert werden, um einen Wert zu treffen bei welchem möglichst minimales Rauschen bei gleichzeitig möglichst kurzen Renderzeiten gegeben sind.

### Forward Backward Coefficient

Mit dem Parameter für den *Forward Backward Coefficient* kann bestimmt werden in welche Richtung das *Subsurface Scattering* verläuft. Geregelt wird dies über einen Slider der Werte zwischen 1 und 0 annehmen kann, dabei bedeutet 0 reines *Forward Scattering* und 1 reines *Backward Scattering*. Alle Werte dazwischen bilden ein entsprechendes Mischverhältnis von beiden. Wichtig ist zu verstehen, dass *Backward Scattering* nicht bedeutet, dass Lichtquellen, die das Objekt von hinten beleuchten, besonders sichtbar werden, sondern dass das Gegenteil der Fall ist. Das Licht, welches in das Material eingedrungen ist, wird in die Richtung, aus der es gekommen ist, zurückgeworfen. Beim *Forward Scattering* dringt es weiter in die Richtung vor, in die es gestartet ist. Je nach Beleuchtungssituation und gewünschtem Effekt muss abgewogen werden, welche Einstellung für dieses Attribut die geeignetste ist. Oft erzielt jedoch ein mittlerer Wert um die 0,5 ein gutes Ergebnis, da dann *Forward- und Backward-Scattering* gleichermaßen berücksichtigt werden.



**Abbildung 3.7:** Darstellung der Auswirkung der drei unterschiedlichen Haupteinstellungen des *Forward Backward Coefficienten* auf die Visualisierung des *Subsurface Scattering*-Effektes.

#### Scatter bounces

Der *Scatter bounces*-Parameter gibt an, wie oft der Lichtstrahl nach dem Eintreten unter die Oberfläche innerhalb des Objektes reflektiert werden darf. Bei einem Wert von 0 geht das Licht grade durch das Objekt hindurch. Durch eine höhere Anzahl an *Bounces* wird der *Scattering*-Effekt verstärkt. Mit einer Erhöhung dieses Parameters und damit des *Scattering*-Effektes, wird aber auch das Rauschen erhöht, was durch eine Steigerung der *Subdivs* ausgeglichen werden kann.



**Abbildung 3.8:** Effekt einer Erhöhung der *Scatter Bounces* auf das Material.

### Scatter coefficient

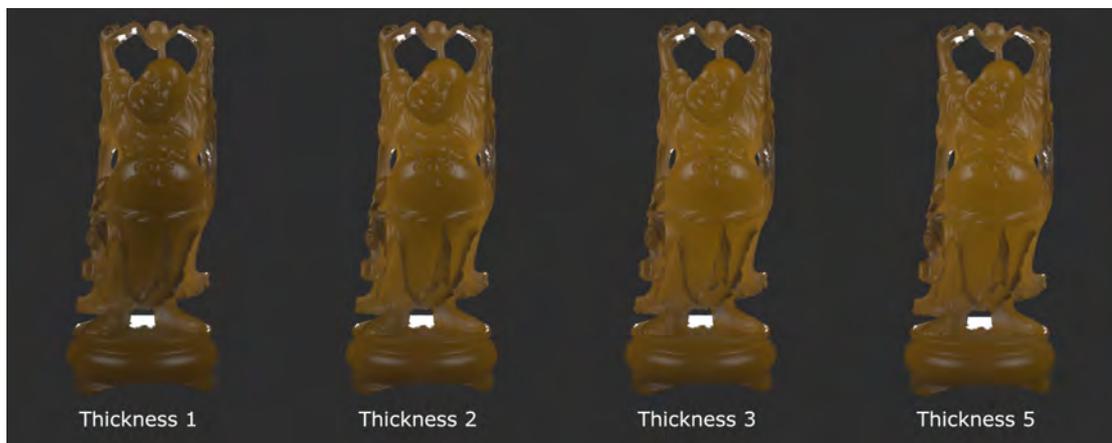
Der *Scatter coefficient* entscheidet darüber wie die Lichtstrahlen innerhalb des Objektes abgelenkt werden. Bei einem Wert von 0,0 wird das Licht in alle Richtungen gleichermaßen gestreut, bei einem Wert von 1,0 wird das Licht nicht abgelenkt und geht gradlinig durch das Objekt hindurch. Zwischenwerte liefern entsprechende Mischformen. Die Unterschiede im Ergebnis sind sehr subtil.



**Abbildung 3.9:** Die Auswirkungen des *Scatter coefficient* auf das Material.

### Thickness

Mit dem *Thickness*-Parameter kann die Dicke des Volumens simuliert und damit gesteuert werden, ob der *Subsurface Scattering*-Effekt das komplette Volumen oder nur Teile des Objektes bis zu einer bestimmtem Dicke betreffen soll. Dabei entspricht die Einheit der Szenenskalierung. Standardmäßig ist das Attribut auf einen Wert von 1000 eingestellt was sicherstellt, dass in den meisten Anwendungsfällen das gesamte Volumen vom *Subsurface Scattering* beeinflusst wird. Die in dieser Arbeit verwendete Referenz-Szene hat eine Dicke von 4,8 cm an der dicksten Stelle, deshalb kann bei einer Erhöhung des *Thickness*-Parameters von 3 auf 5 Einheiten nur noch ein geringer Unterschied erkannt werden (siehe Abbildung 3.10), welcher sich am Bauch des Buddhas, also an der dicksten Stelle des Modells entdecken lässt. Danach macht es keinen visuellen Unterschied mehr, ob der Wert auf 5 oder auf 1000 steht, ab dem Wert der vollen Dicke wird immer das komplette Volumen des Objektes *gescattered*.



**Abbildung 3.10:** Darstellung der Auswirkung der Erhöhung des *Thickness*-Parameters auf die Visualisierung des *Subsurface Scattering*-Effektes. Wenn der *Thickness*-Parameter die gleiche Dicke wie das Objekt erreicht hat, ändert sich danach bei weiterer Erhöhung des Wertes nichts mehr.

#### Environment fog

Wenn der *Environment fog* aktiviert ist, wird die direkte Beleuchtung für die Berechnung des *Subsurface Scattering*-Effektes verwendet. Die *Translucency color* spielt dann für den *Scatter*-Effekt keine Rolle mehr und wird nicht mehr berücksichtigt, sondern nur noch die Farbe, die die Lichtquellen ausstrahlen, mit denen das Objekt in der Szene beleuchtet wird.

#### 3.4.2 Andere Parameter des Shaders die den Subsurface Scattering-Effekt beeinflussen

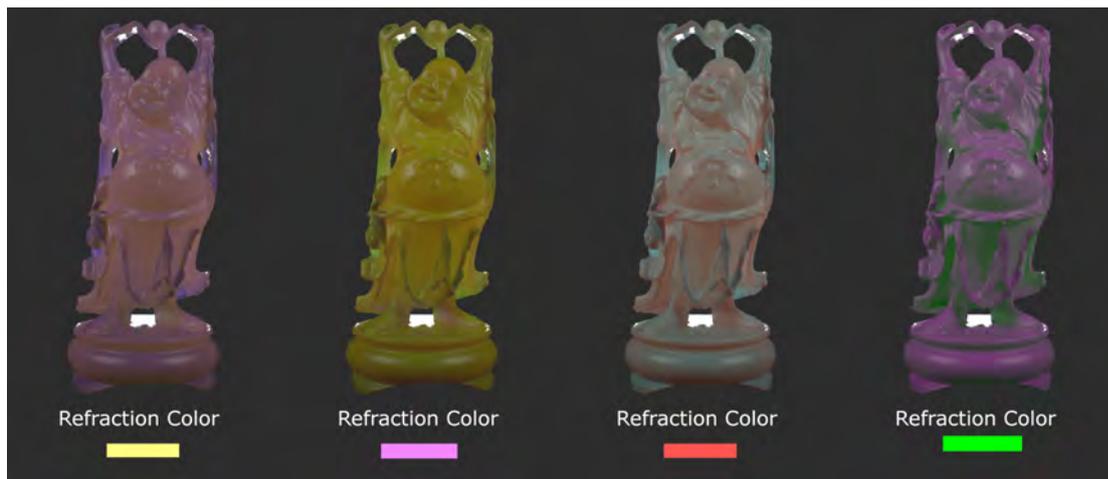
Neben den speziell für den *Subsurface Scattering*-Effekt angelegte Parametern des Shaders, gibt es noch eine Reihe weiterer Attribute im *V-Ray Material*, die einen großen Einfluss auf das Ergebnis des *Scatterings* haben, auf die im Folgenden eingegangen wird.

#### Refraction Color

Hier wird die Farbe des gebrochenen Lichtes bestimmt. Von Schwarz nach Weiß wird das Material immer lichtdurchlässiger. Nimmt man für den Farbwert der *Refraction Color* also einen Graustufenwert, hat dieser Parameter den gleichen Effekt wie auch der *Refraction Amount* (vergleiche die beiden Abbildungen 3.11 und 3.13). Im Fall der *Refraction Color* ist es aber auch möglich, einen beliebigen anderen Farbwert auszuwählen und damit völlig andere Effekte zu erzielen (siehe Abbildung 3.12).



**Abbildung 3.11:** Änderung des Shaderverhaltens, wenn der Farbwert für die *Refraction Color* mit unterschiedlichen Graustufen belegt wird.



**Abbildung 3.12:** Änderung des Shaderverhaltens, wenn der Farbwert der *Refraction Color* mit unterschiedlichen farbigen Werten belegt wird. Auffällig ist hier, dass die Farbe des *gescatterten* Lichtes eine Farbe annimmt, die gegensätzlich zu der gewählten Farbe auf der Hue-Skala liegt.

### Refraction Amount

Der *Refraction Amount* bestimmt wie transluzent das Material ist. Wenn das *V-Ray Material* für das Rendering von *Subsurface Scattering*-Effekten verwendet werden soll, muss das Material eine Transluzenz bekommen, da ohne eine Lichtdurchlässigkeit dieser Effekt logischerweise auch nicht entstehen kann. Über den *Refraction Amount* kann nun geregelt werden, ob der *Diffuse*-Anteil des Shaders komplett ausgeblendet werden und nur noch über

die Einstellungen für den *Subsurface*-Effekt gearbeitet werden soll, oder ob ein Mischverhältnis des diffusen Anteils mit dem *Subsurfes*-Anteil verwendet werden soll.



**Abbildung 3.13:** Änderung des Shaderverhaltens bei unterschiedlichen Werten für den *Refraction Amount*. Dieser Parameter erzeugt den gleichen Effekt wie auch eine Änderung der *Refraction Color* in unterschiedlichen Graustufen.

#### Refraction Glossiness

Die *Refraction Glossiness* bestimmt, wie scharf oder unscharf die Lichtbrechungen sind. Wenn bei der Verwendung des *V-Ray Materials* für die Realisierung von *Subsurface Scattering*-Effekten der *Refraction Amount* auf 1 und die *Refraction Color* auf weiß und somit mit einer vollständigen Lichtdurchlässigkeit gearbeitet wird, kann über die *Refraction Glossiness* die Transparenz des Materials beeinflusst werden.

#### Refraction IOR

Der *Refraction IOR* oder auch *Index of Refraction* bestimmt den Grad der Lichtablenkung innerhalb lichtdurchlässiger Objekte. Einfach ausgedrückt werden könnte es derart, dass der Brechungsindex die Härte der Oberfläche widerspiegelt. Je höher dieser Wert ist, desto verzerrter wird das Bild, weil das Licht die Oberfläche sehr viel langsamer durchdringt. Wenn der Wert für den IOR 1,0 beträgt, bedeutet das, dass das Licht gerade durch das Material hindurch geht, ohne eine Ablenkung zu erfahren.

#### Fog Color

Mit der *Fog Color* kann das Objekt mit Farbe gefüllt werden. Wichtig ist, dass für die *Fog Color* nicht weiß oder eine Farbe mit einem Helligkeitswert von 1 gewählt werden darf, weil das bedeutet, dass diese Farben keinen eigentlichen Schwarzanteil haben und dadurch kein Licht absorbieren. Ohne, dass das Licht absorbiert wird, kann der *Fog*-Effekt nicht gerendert werden. Da das *Subsurface Scattering* in diesem Shader eng mit der *Fog Color* verwoben

ist, kann kein *Subsurface Scattering*-Effekt gerendert werden, wenn die *Fog Color* auf Weiß oder einer Farbe ohne Schwarzanteil steht.



**Abbildung 3.14:** Visualisierung der Auswirkungen der unterschiedlichen Graustufen für die *Fog Color* auf den Effekt des *Subsurface Scatterings*.



**Abbildung 3.15:** Visualisierung der Auswirkungen der unterschiedlichen Farbwerte für die *Fog Color* auf den Effekt des *Subsurface Scatterings*.

### Fog multipliiert

Mit dem *Fog multiplier* kann die Dicke eines Objektes simuliert werden, indem der Eindruck vermittelt wird, dass weniger Licht durch das Material dringt. Der Parameter ist abhängig von der Größe des Objektes.



**Abbildung 3.16:** Visualisierung der Auswirkungen der unterschiedlichen Einstellungen für den Parameter des *Fog Multipliers* auf das *Subsurface Scattering*.

#### Fog bias

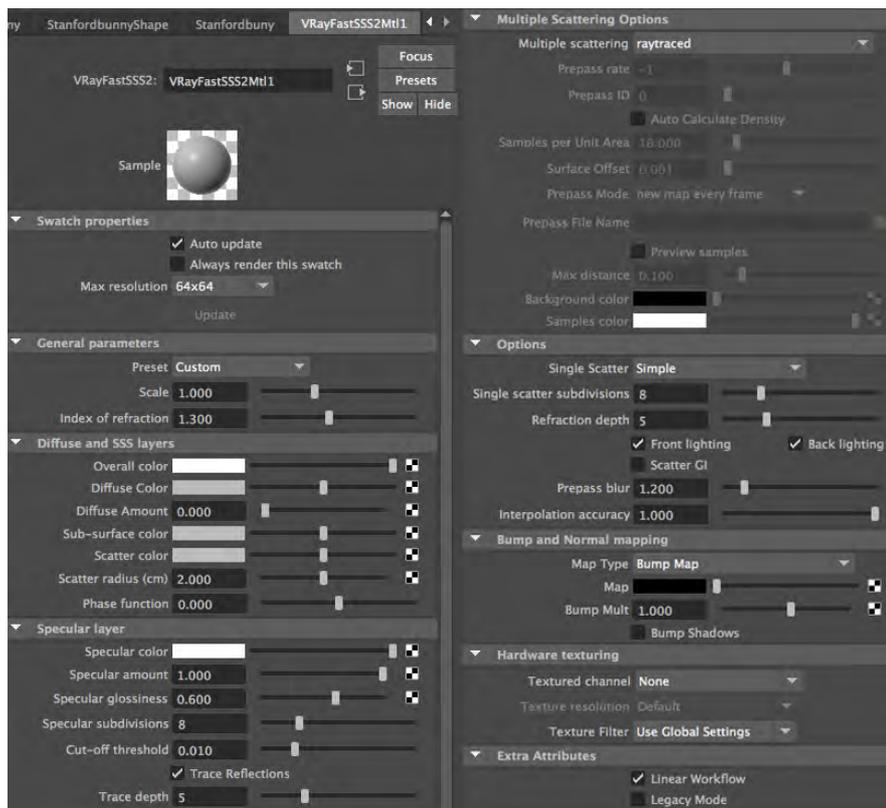


**Abbildung 3.17:** Visualisierung der Auswirkungen der unterschiedliche Einstellungen für das Attribut *Fog Bias* auf den Effekt des *Subsurface Scatterings*.

Der *Fog bias*-Parameter bestimmt das Verhältnis zwischen den dicken und den dünnen Teilen eines Objektes. Ein geringerer Wert bedeutet, dass das Licht nur durch die dünneren Teile des Objektes durch dringt. Je höher der Wert wird, desto mehr dringt das Licht auch durch die dickeren Teile des Objektes.

### 3.5 Der VRayFastSSS2Mtl-Shader

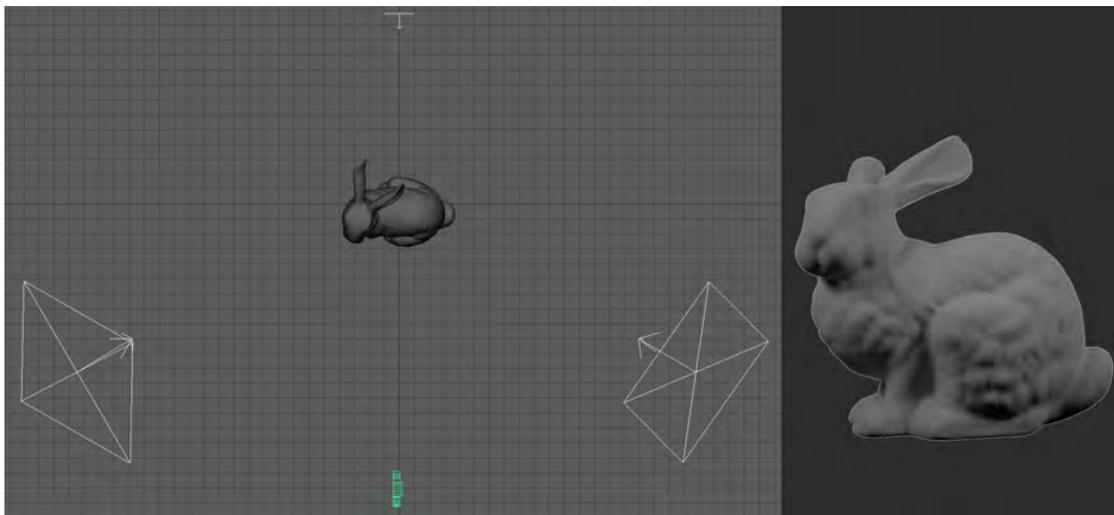
Das *VRayFastSSS2Mtl* ist einer von zwei Spezialshadern, den die Chaosgroup in ihrem Renderer V-Ray für die Visualisierung von *Subsurface Scattering*-Effekten zur Verfügung stellen. Im Folgenden sollen die wichtigsten Parameter dieses Shaders und deren Funktion sowie gegebenenfalls ihr Zusammenspiel untereinander für die Erzeugung des *Subsurface Scattering*-Effektes erläutert werden.



**Abbildung 3.18:** Alle Parameter des V-Ray FastSSS2Mtl-Shaders im Attribut Editor.

Als Referenzmodell für die Visualisierung der jeweiligen Auswirkungen der einzelnen Attribute des Shaders wird das Modell des *Stanford Bunny*<sup>3</sup> aus der Reihe der Stanford Referenzmodelle gewählt, da auch dieses Modell einen guten Kompromiss aus Komplexitätsgrad und dennoch geringen Renderzeiten offeriert, um die Effekte möglichst gut darstellen zu können. Der Szenenaufbau und die Beleuchtungssituation entspricht der im vorherigen Abschnitt für die Versuchsreihen zum *V-Ray Material* bereits erläuterten Vorgehensweise.

<sup>3</sup><http://graphics.stanford.edu/data/3Dscanrep/> Stand: 26. Dezember 2015



**Abbildung 3.19:** Aufbau der Szene, welche für die Referenzrenderings zur Visualisierung der einzelnen Parameter verwendet wurde. In der Mitte das Modell des *Stanford Bunny*-Referenzobjektes beleuchtet von drei *V-Ray Rectangle Lights*. Vorne sieht man die physikalische *V-Ray Kamera*, mit welcher die Renderings gemacht wurden. Rechts ist das Referenzmodell mit einem diffusen Shader abgebildet.

#### 3.5.1 General Parameters

In den *General Parameters* gibt es drei Einstellungen. Zunächst kann ausgewählt werden, ob als Grundlage des Shaderaufbaus eines der zehn vorgefertigten *Presets* dienen soll, oder ob ein eigener *Custom Shader* von Grund auf aufgebaut werden soll. Die *Presets* sind aus den Daten der Messungen der im Kapitel 2 vorgestellten Arbeit von Jensen et al. [JMLH01] entnommen worden. Wählen kann man zwischen den folgenden *Presets*:

- *Skin (brown), Skin (pink), Skin (yellow)*
- *Milk (skimmed), Milk (whole)*
- *Marble (white)*
- *Ketchup*
- *Cream*
- *Potato*
- *Spectralon*

wobei das *Spectralon* als eine Referenz für ein Material komplett ohne *Subsurface Scattering* dient.

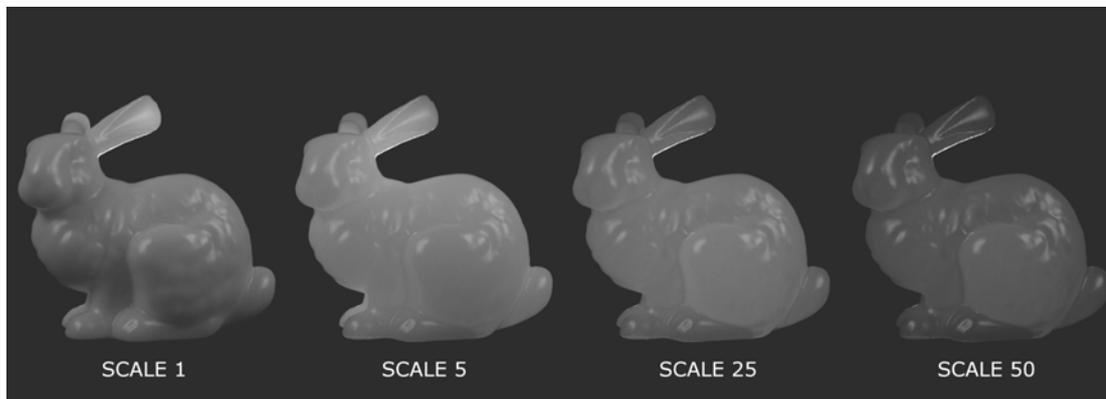


**Abbildung 3.20:** Übersicht aller zur Auswahl stehenden *Presets* des *V-Ray FastSSS2-Shader*s.

### Scale

Der Parameter *Scale* beeinflusst wie tief das Licht in das Material eindringt. Je größer der Wert für diesen Parameter gewählt wird, desto tiefer dringt das Licht ein und desto transluzenter erscheint auch das Material. Dabei arbeitet das Attribut abhängig von der Szenenskalierung. Im Grunde hat der *Scale*-Parameter die gleiche Funktion wie auch das noch folgende *Scatter Radius*-Attribut, aber er kann unabhängig von den ausgewählten *Presets*

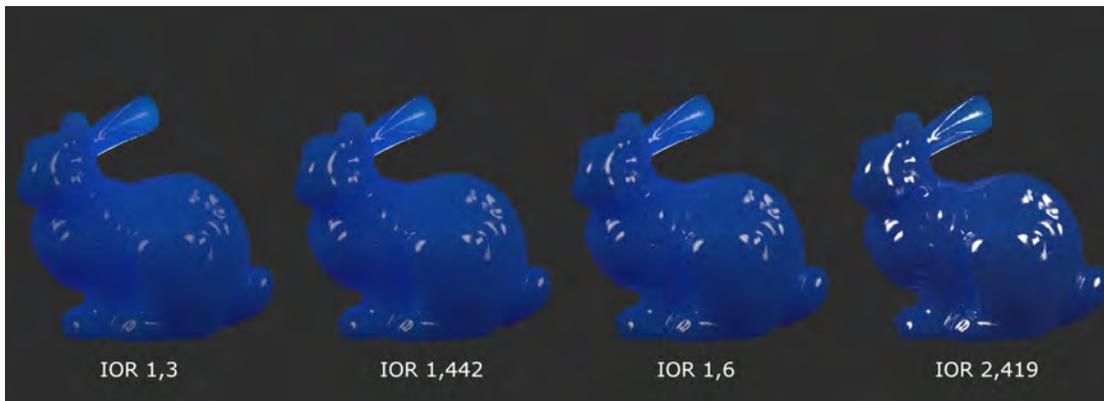
eingestellt werden und wird auch bei einem Wechsel des *Presets* nicht zurückgesetzt. Eine Erhöhung des Wertes für den *Scale*-Parameter ist der einfachste Weg, um den *Subsurface Scattering*-Effekt in einem Material zu verstärken.



**Abbildung 3.21:** Visualisierung wie die Transluzenz des Materials mit steigendem Wert für den Parameter *Scale* immer mehr zunimmt. Der Shader für die Versuchsreihe hat einen mittleren Grauton für die *Sub-surface color* (RGB Werte: 0,5/0,5/0,5) und einen dunklen Grauton für die *Scatter color* (RGB Werte: 0,244/0,244/0,244).

#### Index of Refraction

Der Parameter *IOR* steht für *Index of Refraction*, zu Deutsch Brechungsindex. Dieser Index kann für die unterschiedlichen realen Materialien nachgeschlagen werden. Er gibt Auskunft darüber, wie das Licht auf der Oberfläche gebrochen und wie es reflektiert wird. Die Einstellungen dieses Parameters entscheiden also über das Reflektionsverhalten des Materials. Die meisten Materialien mit hohem *Subsurface Scattering* haben einen *Index of Refraction* von 1,3, was sich auch in vielen der *Presets* für Haut, Milch etc. widerspiegelt.



**Abbildung 3.22:** Visualisierung der Auswirkungen der unterschiedlichen Einstellungen des Refraction IOR-Parameters auf die Reflektionen des Materials. Hier wurden mit 1,3 ein IOR für Eis, mit 1,442 ein IOR für Wachs mit 1,6 ein IOR für Jade und mit 2,419 ein IOR für Diamanten verwendet. Der Shader für die Versuchsreihe hat einen mittleren blauen Farbton für die *Sub-surface color* (RGB Werte: 0,108/0,216/0,784) und einen entsprechend dunkleren Blauton für die *Scatter color* (RGB Werte: 0,0/0,06/0,373).

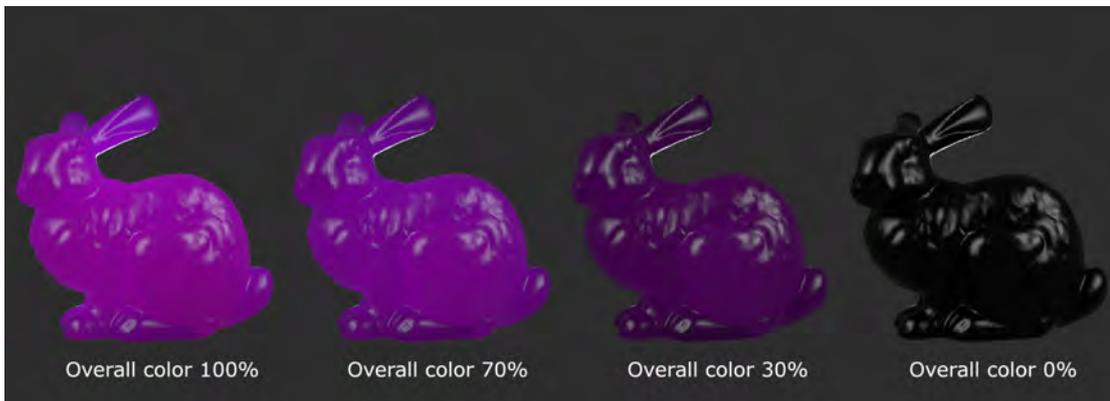
### 3.5.2 Diffuse and SSS layers

#### Overall color

Die *Overall color* regelt, wie der Name des Attributes schon treffend beschreibt, die allumfassende Farbe des Materials, dabei beeinflusst sie sowohl den diffusen, als auch den *Subsurface*-Anteil des Materials, also die Farbe, die das Material unterhalb der Oberfläche hat.

Meist ist es einfacher diesen Parameter in einem schwarz-weiß beziehungsweise Graustufenwert zu belassen und somit lediglich die Helligkeitsverteilung darüber zu regeln. Die Farbanteile für die diffusen und den *Subsurface*-Anteil lassen sich jeweils in ihren eigenen entsprechenden *Layers* anlegen.

In Fällen, in denen aber eine dritte Farbe benötigt wird, die keinen Einfluss auf den *Sub-surface Scattering*-Effekt an sich hat, die aber auch den diffusen Anteil im Material nicht zwangsweise erhöht, ist der *Overall color*-Kanal die richtige Wahl. Beispielsweise können hier *Scratch*- oder *Dust Maps* in den Farbkanal gelegt werden, die keinen Effekt auf das *Scattering* selbst haben sollen, aber dem Material mehr Realismus verleihen können.



**Abbildung 3.23:** Auswirkungen auf das Material, wenn die *Overall color* zur Helligkeitsregelung verwendet wird und Graustufenwerte von 100% Weiß bis 0% Weiß einnimmt. Dabei hat die *Sub-surface color* des Shaders einen relativ hellen lila Farbton (RGB Werte: 0,532/0,0/0,843) und eine dunkleren lila Farbton für die *Scatter color* (RGB Werte: 0,323/0,071/0,451).

#### **Diffuse color, Diffuse Amount und Subsurface Color**

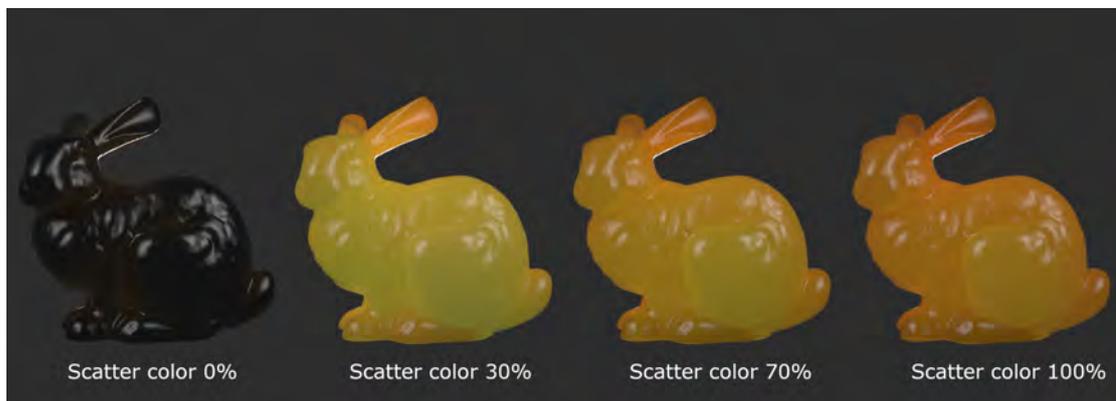
Das Prinzip mit den vielen unterschiedlichen Farblayern und Reglern wirkt auf den ersten Blick recht komplex, aber das Modell, welches sich hinter diesen drei Parametern verbirgt, ist eigentlich recht einfach. Die *Diffuse color* ist ein Farbwert, der von der Oberfläche aus in alle Richtungen gleichmäßig reflektiert wird, er ist opak und nicht transluzent. Das Modell lässt sich derart vorstellen, dass die *Diffuse color* die Farbe der äußeren Oberfläche des Materials widerspiegelt, während die *Sub-surface color*, wie der Name präzise wiedergibt die Farbe des Materials unterhalb der Oberfläche beschreibt. Der *Diffuse Amount* regelt, wie viel der diffusen Farbe mit der *Sub-surface color* vermischt wird, oder anders ausgedrückt, wie hoch der Anteil der *Sub-Surface color* ist, der durch den diffusen Farbanteil des Materials hindurch scheint. Wichtig ist also, dass mit dem *diffuse Amount*-Regler nicht nur die jeweiligen Farbanteile der finalen Mischung bestimmt werden, sondern auch, wie hoch der diffuse beziehungsweise im Kehrwert der transluzente Anteil des Materials ist. Dabei lässt sich das Verhältnis von einem *Amount* von 0 bis 1 regeln, wobei 0 den diffusen Anteil völlig ignoriert und nur noch mit der *Subsurface color* arbeitet und 1 einem vollkommen diffusen Shader entspricht. Die von V-Ray vorgegebenen *Presets* arbeiten alle ohne diffusen Anteil in den Materialien, zwar ist als *Diffuse color* der gleiche Farbwert wie auch als *Sub-surface color* eingetragen, der *Diffuse Amount* aber steht auf Null und wird daher nicht in der Berechnung für den Shader berücksichtigt. Wichtig zu wissen ist, dass auch bei einem *Diffuse Amount* von 1, der zwar das Durchleuchten der *Sub-surface color* verhindert und den transluzenten Effekt aus dem Material herausnimmt, dennoch durch die spezielle Berechnungsweise des Shaders der für *Subsurface Scattering* typische weiche Oberflächeneindruck im Rendering entsteht, wie auf Abbildung 3.24 zu sehen ist.



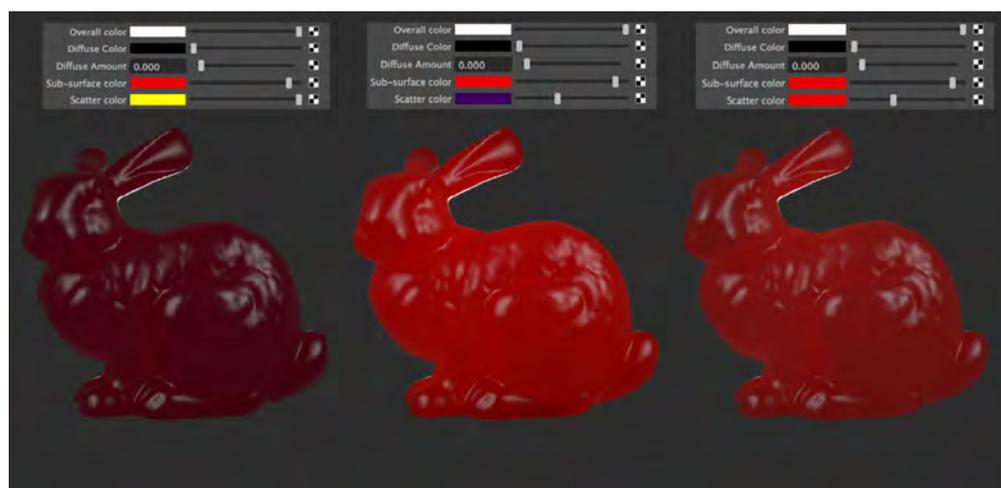
**Abbildung 3.24:** Auswirkungen der *Diffuse color*, wenn diese durch Regelung des *Diffuse Amounts* mit der *Sub-surface color* vermischt wird. Zu beachten ist, dass sich die Mischung der beiden Farbkanäle nicht nur auf den Farbwert an sich auswirkt, sondern auch auf die Transluzenz des Materials. Der Shader für die Versuchsreihe hat einen hellen blauen Farbton für die *Sub-surface color* (RGB Werte: 0,000/0,704/0,843) und einen entsprechend dunkleren Blauton für die *Scatter color* (RGB Werte: 0,071/0,226/0,451). Für die *Diffuse Color* wurde ein hellgrüner Farbton zugewiesen (RGB Werte: 0,882/0,821/0,017).

### Scatter color

Die *Scatter color* beeinflusst die Stärke des *Scattering*-Effektes. Dabei wird der Effekt von Schwarz nach Weiß verstärkt, je heller die Farbe wird und desto mehr transluzent wirkt dann das Material. Es ist also möglich, einfach einen Graustufenwert für diesen Parameter zu nehmen. Des Weiteren lassen sich oft gute Ergebnisse damit erzielen, einen Farbwert zu wählen, der dem der *Sub-surface color* nahe kommt. Wenn eine Einfärbung des Effektes gewünscht ist, kann auch eine völlig andere Farbe gewählt werden. Hier ist allerdings zu beachten, dass die Farbe der Einfärbung dann entgegengesetzt der Hue-Skala des gewählten Farbtons wirkt. Der Effekt ist sehr subtil, aber kann in einigen Szenen viel ausmachen. In manchen Fällen ist zu beobachten, dass Rauschen entsteht, wenn die *Scatter color* heller gewählt wird als die *Sub-surface color*. Dem kann aber durch eine Erhöhung der *Subdivisions* entgegengewirkt werden.



**Abbildung 3.25:** Visualisierung der Auswirkungen der unterschiedlichen Graustufenwerte für den Parameter der *Scatter color* auf den Shader. Gewählt wurden für die Visualisierung vier verschiedene Abstufungen von schwarz bis weiß für die *Scatter color*. Die *Sub-surface color* des Shaders hat für diese Versuchsreihe einen orangenen Farbton (RGB Werte: 0,886/0,524/0,067).

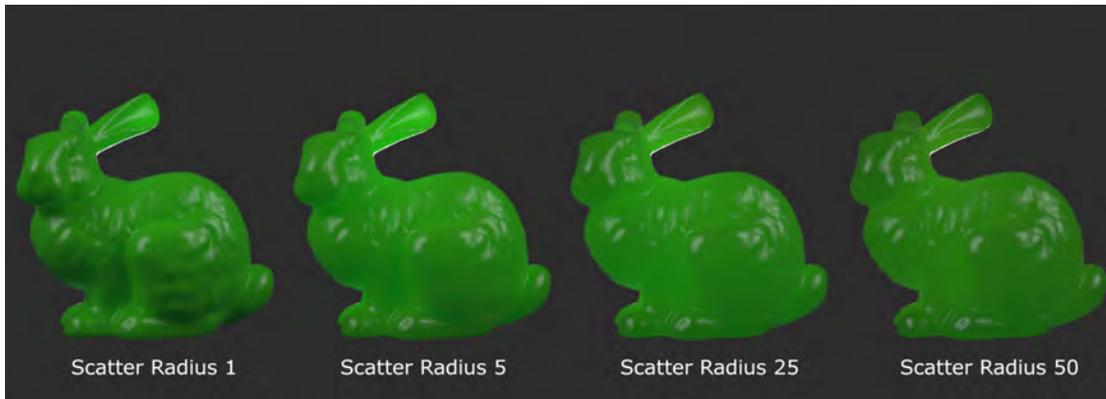


**Abbildung 3.26:** Auswirkungen von Farbwerten als *Scattercolor*. Dabei wurden in den ersten beiden Beispielen (links und Mitte) von der *Subsurface color* abweichende Farben verwendet, hier ist zu erkennen, dass der Effekt, der entsteht, der entgegengesetzten Farbe aus der Hue-Skala der jeweils gewählten Farbe für die *Scatter color* entspricht. Für das letzte Beispiel (rechts) entspricht die *Scatter color* der *Sub-surface color*. Um die Stärke des Effektes zu verändern, könnte jeweils noch mit der Helligkeit der Farbwerte gespielt werden.

#### Scatter Radius

Der *Scatter Radius* bestimmt wie viel Licht unter die Oberfläche des Materials eindringt. Der Parameter arbeitet wie auch der *Scale*-Parameter, der bereits zuvor schon beschrieben

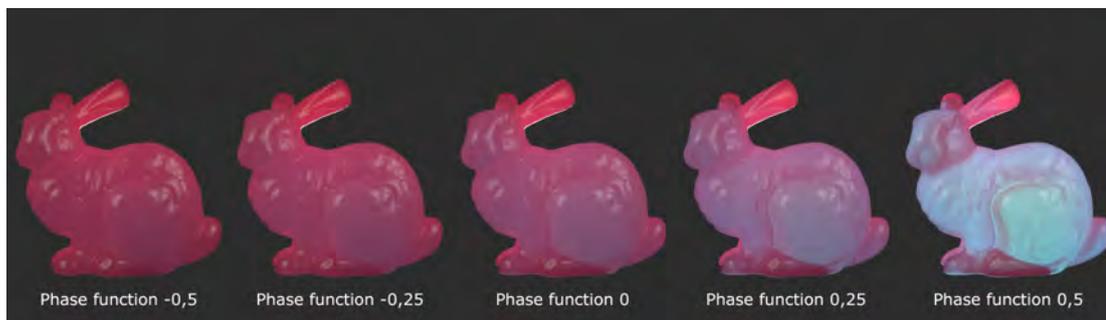
wurde. Je größer der Wert, der für dieses Attribut gewählt wird, desto tiefer dringt das Licht in das Material ein und desto größer ist der *Scattering*-Effekt. Das Material wird transluzenter und weniger diffus.



**Abbildung 3.27:** Visualisierung, wie mit zunehmendem Wert für den *Scatter Radius*-Parameter die Transluzenz des Materials zunimmt. Der Shader für diese Versuchsreihe hat eine hellgrüne *Sub-surface color* (RGB Werte: 0,108/0,784/0,149) und eine *Scatter color* in einem dunkleren grünen Farbton (RGB Werte:0,021/0,373/0,0).

### Phase function

Die Einstellungen des *Phase function*-Parameters entscheiden darüber, wie das Licht innerhalb des Materials *gescattert* wird. Das kann je nach Lichtsetzung sehr große Auswirkungen auf das Ergebnis haben. Einstellen lassen sich Werte von -1 bis +1. Bei einem Wert von 0 wird das Licht gleichmäßig in alle Richtungen *gescattert*. Dies wird als *isotropic scattering* bezeichnet. Bei positiven Werten kleiner 1 *scatters* das Licht überwiegend vorwärts, das bedeutet in die Richtung, aus der es kommt. Bei negativen Werten kleiner 1 *scatters* das Licht entgegengesetzt der Richtung, aus der es kommt. Meist führen negative Werte zu einem deutlich stärkeren transluzenten Effekt als positive Werte, was aber abhängig von der jeweiligen Ausleuchtung der Szenen ist.



**Abbildung 3.28:** Visualisierung der immensen Auswirkungen der *Phase function* auf das Material. Der Shader der Versuchsreihe hat eine *Sub-surface color* in einem Magentaton (RGB-Werte: 0,6/0,095/0,195) und eine *Scatter color* in einem hellen Grauton (RGB Werte: 0,7/0,7/0,7), ein diffuser Farbanteil wird nicht berechnet.

#### 3.5.3 Specular layer

In diesem Bereich der Shader-Einstellungen lassen sich alle Parameter finden, die die spiegelnden Eigenschaften des Shaders beschreiben. Dabei funktionieren die Attribute wie auch in einem einfachen *V-Ray Material*.

##### Specular color

Über die *Specular color* lässt sich eine Farbe für die Reflektionen bestimmen. Meist wird hier aber ein schwarz-weiß Wert gewählt, um lediglich eine Helligkeit für die Reflektionen zu definieren. Aber auch eine konkrete Farbwahl wäre bei Bedarf möglich.

##### Specular Amount

Der *Specular Amount* bestimmt wie stark das Material reflektiert, jedoch ist es wichtig zu wissen, dass ein *fresnel Falloff* für den *Specular Amount* implementiert wurde, der je nach Einstellung des *Index of Refraction* des Shaders automatisch interagiert.

##### Glossiness

Die *Glossiness* bestimmt, wie scharf oder weich gezeichnet die Reflektionen auf der Oberfläche des Materials zu sehen sind.



**Abbildung 3.29:** Visualisierung wie sich die Oberfläche der Materials mit zunehmendem Faktor für den Parameter der *Specular Glossiness* von Matt zu immer stärker glänzend verändert. Der Shader für diese Versuchsreihe hat eine gelbe *Sub-surface color* (RGB Werte: 0,784/0,746/0,108) und eine bräunliche *Scatter color* (RGB Werte: 0,373/0,275/0,0).

### Specular Subdivisions

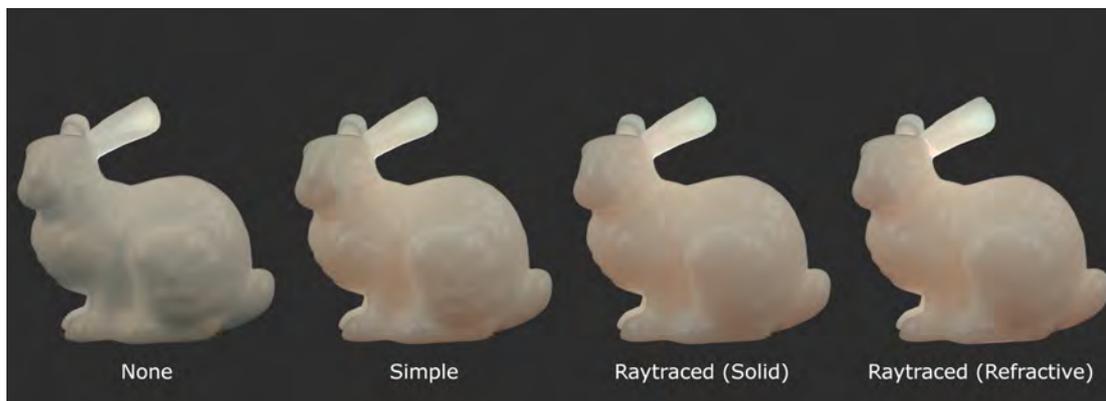
Mit dem Parameter für die *specular Subdivisions* lassen sich die *Samples* festlegen, mit denen die spiegelnden Reflektionen gerendert werden. Je höher die *Samples* eingestellt werden, desto genauer wird die Berechnung und desto weniger Rauschen erscheint auf dem Bild, aber desto höher sind auch die Renderzeiten.

### 3.5.4 Options

#### Single Scatter Mode

Über den *Single Scatter*-Modus kann zwischen vier unterschiedlichen Modi gewählt werden, die bestimmen, wie die *Single Scattering*-Komponente des *Subsurface Scattering*-Effektes berechnet wird. Gewählt werden kann zwischen den folgenden Modi.

- None
- Simple
- Raytraced (solid)
- Raytraced (refractive)



**Abbildung 3.30:** Auswirkung der unterschiedlichen Verfahren für das *Single Scattering* auf das Renderergebnis des Shaders. Der Shader für diese Versuchsreihe hat eine beige-farbene *Sub-surface color* (RGB Werte: 0,98/0,9/0,73) und eine rötliche *Scatter color* (RGB Werte: 0,751/0,233/0,127).

Ist für diesen Modus das Attribut "None" gewählt, so wird gar kein *Single Scattering* berechnet. Der Modus "Simple" ist sinnvoll für Materialien, die nur wenig Licht durch die Oberfläche hindurchlassen, da hier das *Scattering* durch ein Annäherungsverfahren über das Licht an der Oberfläche berechnet wird. Der Modus "Raytraced (solid)" ist geeignet für hochgradig transluzente Materialien. Durch dieses Verfahren wird das *Scattering* akkurat berechnet, indem das Volumen innerhalb des Objektes *gesampled* wird. Der Modus "Raytraced (refractive)" funktioniert im Prinzip genauso wie auch der Modus "Raytraced (solid)", zusätzlich werden aber auch die *Refraction Rays* verfolgt und berechnet. Das macht diesen Modus besonders auch für transparente Materialien wie Wasser oder Glas zur bevorzugten Wahl. Wichtig zu wissen ist, dass mit der Verwendung dieses Modus auch die Schatten transparent werden.

#### Single Scatter Subdivisions

Wenn einer der beiden Modi *Raytraced (solid)* oder *Raytraced (refractive)* aktiv ist, kann über den Parameter *Single Scatter Subdivisions* geregelt werden, mit welcher Anzahl an *Samples* die Berechnungen für die *Single Scatter* Komponente des *Subsurface Scattering*-Effektes durchlaufen werden sollen.

#### Refraction Depth

Wenn der Modus *Raytraced (refractive)* verwendet wird, kann über den Parameter der *Refraction Depth* bestimmt werden, wie tief die *Refraction Rays* in das Material eindringen.

#### Front Lighting and Back Lighting

Die Checkboxes für *Front und Back Lighting* aktivieren jeweils das *multiple Scattering* für Licht, das auf der gleichen (*Front Lighting*) oder der entgegengesetzten Seite (*Back*

*Lighting*) des Objektes, aus Perspektive der Kamera betrachtet, strahlt. Für Materialien, die sehr opaqu und wenig transluzent sind, kann ein Deaktivieren des *Back Lightings* die Renderzeiten minimieren.

### Scatter GI

Ist die *Scatter GI* aktiviert, wird die *Global Illumination* in die Berechnung der *Illumination Map* des *multiple Scattering* integriert. Andernfalls wird eine einfache diffuse Approximation nach der Berechnung des *Subsurface Scatterings* erstellt. Diese ist ungenauer, vor allem für hochgradig transluzente Materialien, kostet aber auch weniger Renderzeit.

### Prepass blur und Interpolation accuracy

Die beiden Attribute spielen nur dann eine Rolle, wenn als *Multiple Scattering*-Verfahren eine der im nächsten Abschnitt erläuterten *Map*-Verfahren, *Prepass-based illumination map* oder *Object-based illumination map*, verwendet werden. Da diese in dieser Arbeit aus Gründen, die ebenfalls im nächsten Abschnitt erläutert werden, nicht zum Einsatz kommen, wird hier auf diese Attribute nicht näher eingegangen.

### 3.5.5 Multiple Scattering Options

Es gibt drei Optionen, zwischen denen für das *Multiple Scattering* gewählt werden kann. Per Default ist *raytraced* als Verfahren ausgewählt und dies ist auch das einzige der drei Modi, welches das *Subsurface Scattering* über echtes *Raytracing* innerhalb des Volumens der Geometrie berechnet. Somit ist diese Prozedur die am physikalisch plausibelsten und genaueste Methode und da es in dieser Arbeit um fotorealistische und physikalisch plausible Effekte gehen soll, wird ausschließlich mit diesem Verfahren gearbeitet. Die anderen beiden zur Option stehenden Verfahrensweisen sind zum einem die *Prepass-based illumination map* und zum anderen die *Object-based illumination map*. Beide Verfahren verwenden *Maps* ähnlich der *Irradiance Map*, um damit *Subsurface Scattering*-Effekte zu approximieren. Sie unterscheiden sich lediglich in ihrem *Sampling*-Verfahren. Die beiden *Map*-Varianten sparen deutlich an Renderzeit, kommen aber in ihrer Qualität nicht an die *Raytracing*-Methode heran.

### 3.5.6 Bump and Normal Mapping

Auch für *Bump* und *Normal Maps* hat der Shader Attribute vorgesehen, um dadurch einen gesteigerten Grad an Realismus zu ermöglichen.

## 3.6 Das V-Ray Skin Material

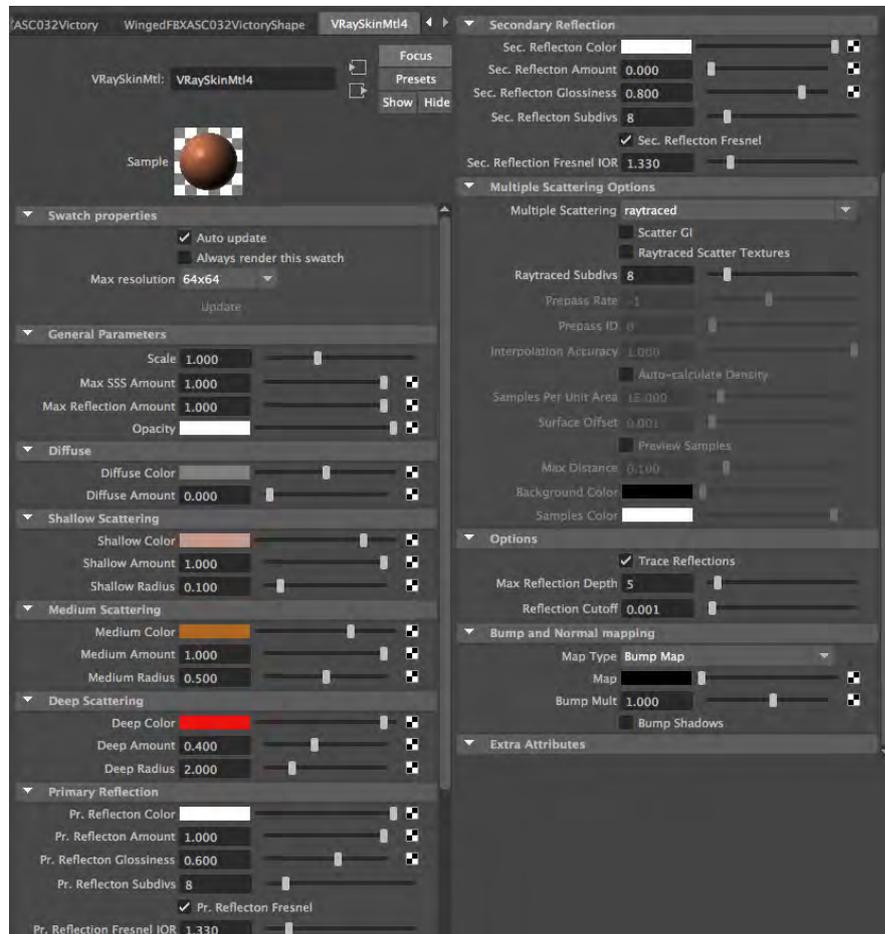
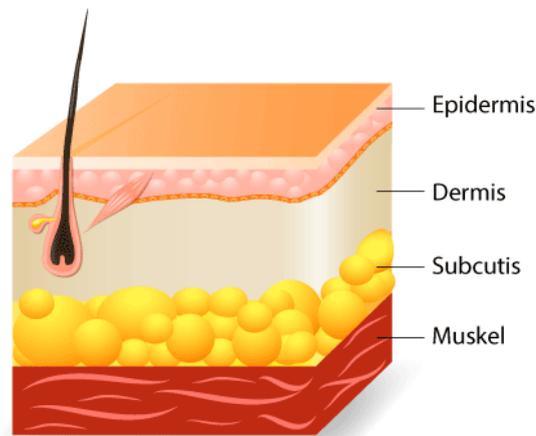


Abbildung 3.31: Alle Parameter des VRaySkinMtl.-Shaders im Attribut Editor.

Der zweite Spezialshader den die Chaosgroup, für Autodesk Maya für den Renderer V-Ray seit der aktuellen Version 3.1, zum Rendern von *Subsurface Scattering*-Effekten zur Verfügung stellt, ist das *VRaySkinMtl*. Wie der Name schon verrät, ist dieser Shader darauf ausgelegt, die Besonderheiten beim Rendering von Haut zu visualisieren. Im Gegensatz zum vorher beschriebenen *FastSSS2 Shader*, welcher auch *Presets* für das Lösen dieser Aufgabe bereithält, bietet das *VRaySkinMtl* durch drei *Scattering*-Schichten sowie zwei Reflektionsschichten, die den unterschiedlichen Schichten der Haut nachempfunden sind, um auch die Feinheiten menschlicher Haut darstellen zu können (siehe Abbildung 3.32). Dabei entspricht der *Shallow Scattering Layer* des Shaders der Epidermis oder Oberhaut und hat per default einen rosa Farbwert zugeordnet, der *Medium Scattering Layer* entspricht der Dermis oder Lederhaut und hat einen bräunlich/gelben Farbton und der *Deep Scattering Layer* entspricht der Muskelschicht und ihm ist ein roter Farbton zugewiesen.

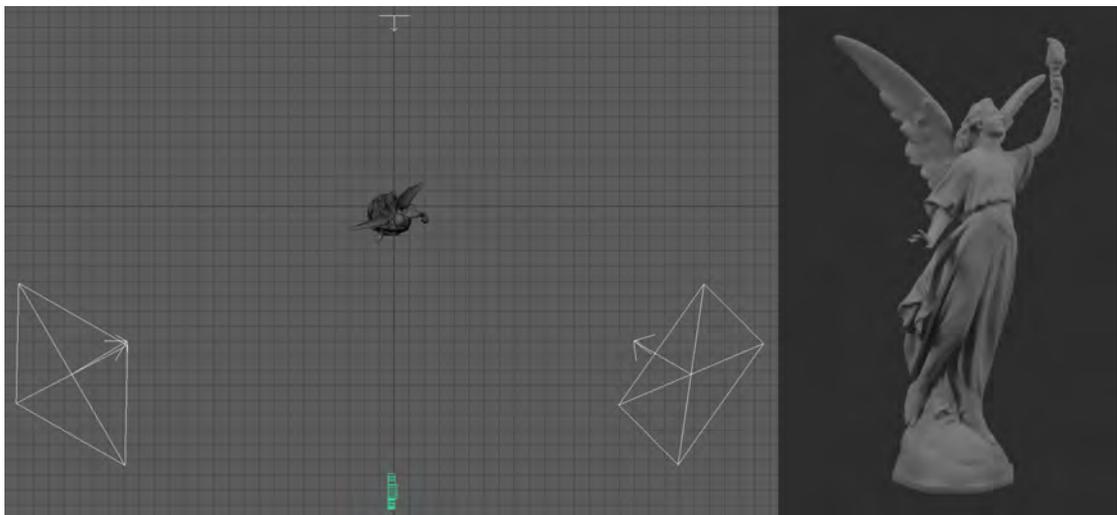


**Abbildung 3.32:** Modell zur Visualisierung der menschlichen Hautschichten (<http://goo.gl/VU00pr> Stand: 10. Februar 2016).

Als Referenzmodell für die Visualisierung der jeweiligen Auswirkungen der einzelnen Attribute des Shaders wird auch für die letzte Testreihe auf ein Modell aus der Reihe der Stanford Referenzmodelle zurückgegriffen. Diesmal wird das Modell des Engels *Lucy*<sup>4</sup> gewählt. Wie schon bei den anderen beiden Testreihen zuvor auch, wird bei der Auswahl des Referenzmodells darauf geachtet, dass das Modell einen guten Kompromiss aus Komplexitätsgrad und dennoch annehmbaren Renderzeiten offeriert, um die Effekte möglichst gut darstellen zu können. Sowohl der Szenenaufbau, als auch die Ausleuchtung der Testszene sind entsprechend der beiden anderen Versuchsreihen umgesetzt und wurden im Abschnitt über das V-Ray Material bereits ausführlicher beschrieben.

---

<sup>4</sup><http://graphics.stanford.edu/data/3Dscanrep/> Stand: 26. Dezember 2015



**Abbildung 3.33:** Szenenaufbau für das Rendering der Referenzbilder zur Visualisierung der Parametereigenschaften des *V-Ray Skin Materials* (links) und ein Rendering des genutzten Referenzmodells (rechts).

#### 3.6.1 General Parameters

##### Scale

Der *Scale*-Parameter bestimmt, wie auch beim *FastSSS2*-Shader, wie tief das Licht in das Objekt eindringt und beeinflusst damit die Stärke des *Subsurface Scattering*-Effektes. Anders als bei dem zuvor beschriebenen Shader dient der *Scale*-Parameter im *Skin Material* aber als kombinierter *Multiplier* für die drei Radius-Attribute des *Shallow*, *Medium* und *Deep Layers* des Shaders.



**Abbildung 3.34:** Visualisierung, wie sich mit Erhöhung des *Scale*-Parameters der *Subsurface Scattering*-Effekt verstärkt, weil das Licht tiefer unter die Oberfläche des Materials eindringt.

### Max SSS Amount

Der *Max SSS Amount* bietet eine kombinierte Begrenzung für das *Subsurface Scattering* der drei *Layer Shallow*, *Medium* und *Deep*. Wenn die Summe der drei Schichten den *Max SSS Amount* überschreitet werden die Mengen normalisiert, sodass sie in ihrer Summe wieder dem angegebenen Wert für den *Max SSS Amount* entsprechen.



**Abbildung 3.35:** Visualisierung, wie sich mit Erhöhung des *Max SSS Amounts* die Grenze für das sichtbare *Subsurface Scattering* erhöht und damit ein stärkerer Effekt sichtbar ist. Wie stark der sichtbare Effekt ist, ist aber abhängig von den Einstellungen der drei *Scattering Layer Shallow*-, *Medium*- und *Deep*- *Scattering*. Der *Max SSS Amount* legt nur eine obere Grenze für die Summe der drei *Scatter Amounts* fest.

### Max Reflection Amount

Mit dem *Max Reflection Amount* kann ein gemeinsames Maximum an Reflektion, kombiniert für die beiden *Reflection Layer* angegeben werden. Wird durch die Summe beider, *Layer* das angegebene Limit überschritten, so werden die einzelnen *Amounts* der *Primary Reflection* und der *Secondary Reflection* normiert, sodass die Summe wieder dem angegebenen maximalen Wert entspricht und die Begrenzung eingehalten wird.



**Abbildung 3.36:** Visualisierung, wie das Attribut *Max Reflection Amount* sich auf das Rendering-Ergebnis auswirkt. Bei einem *Amount* von 0 hat Material keine Reflektion mehr. Ansonsten greift dieser Parameter nur als Begrenzung ein, wenn die Summe der beiden *Amounts* des *Primary Reflection layers* und des *Secondary Reflection Layers* den Wert des *Max Reflection Amounts* überschreiten.

#### Opacity

Die *Opacity* zu Deutsch Opazität bestimmt, wie zuvor in diesem Kapitel beschrieben wurde, das Maß der Lichtundurchlässigkeit des Materials und damit wie opaq der Shader wird. In diesem Fall ist die Implementierung leider etwas verwirrend gestaltet, da der Regler nicht wie es für einen *Subsurface Scattering*-Shader Sinn machen würde, diffuse Transmission und damit den Grad der Transluzenz regelt, sondern direkte Transmission und damit Transparenz. Das bedeutet, dass die Transluzenz, die für den Effekt des *Subsurface Scatterings* elementar ist, für den Shader voreingestellt und nicht regelbar ist. Über den *Opacity*-Regler lässt sich eine zusätzliche Transparenz des Materials einstellen. Dabei bedeutet schwarz vollständige Transparenz und weiß vollständige Opazität. Auf das *Subsurface Scattering* hat dieser Parameter aber keinerlei Auswirkungen und auch für ein mit 100% weiß also vollständig lichtundurchlässiges Material wird automatisch ein *Subsurface Scattering* berechnet. Hinzu kommt, dass menschliche Haut keine Transparenz aufweist, und somit der Anwendungsfall für einen solchen Parameter nicht ganz einleuchtend ist.



**Abbildung 3.37:** Visualisierung, wie die *Oppacity* das Material transparenter macht, je kleiner der Wert wird.

### 3.6.2 Diffuse

#### Diffuse Color

Mit dem Attribut für die *Diffuse color* kann ein beliebiger Farbwert für den diffusen Farbanteil des Materials zugewiesen werden. Auch kann über diesen Parameter eine Helligkeitsverteilung vorgenommen werden, wenn statt eines Farbwertes ein Graustufenwert eingesetzt wird.



**Abbildung 3.38:** Visualisierung, wie sich das Material verändert, wenn die *Diffuse color* mit unterschiedlichen Graustufenwerten belegt wird.

#### Diffuse Amount

Das Attribut *Diffuse Amount* regelt wie hoch der diffuse Anteil des Materials sein soll. Gemischt wird dann der angegebene Farbwert der *Diffuse color* entsprechend des angegebenen

*Amounts* mit den drei Schichten für den *Subsurface Scattering*-Effekt *Shallow Scattering*, *Medium Scattering* und *Deep Scattering*.



**Abbildung 3.39:** Visualisierung, wie sich das Rendering verändert, wenn über eine Steigerung des *Diffuse Amounts* das Mischungsverhältnis von diffusem und *Subsurface Scattering*-Anteil geändert wird.

#### 3.6.3 Shallow, Medium und Deep Scattering Layer

Die drei *Scattering Layer Shallow*, *Medium* und *Deep* wurden, wie in der Einleitung dieses Unterkapitels bereits erläutert, den unterschiedlichen Hautschichten der menschlichen Haut nachempfunden. Für jeden dieser Layer gibt es einen Farbkanal, ein Radiusattribut sowie einen *Amount*. Dem *Shadow Layer* ist per default ein rosa Farbwert zugeordnet, dem *Medium Layer* ein gelblich brauner Farbton und dem *Deep Layer* ein roter Farbwert. Diese Farbwerte entsprechen in etwa denen der jeweiligen *Counterparts* im Hautschichten Modell der menschlichen Haut (siehe Abbildung 3.32). Die Farbwerte können alle nach eigenem Belieben geändert oder durch Texturen ersetzt werden. Mit dem Radius-Parameter kann ein Radius für das *Scattering* der jeweiligen Ebene bestimmt werden. Per Default-Einstellung hat der *Shallow Scattering Layer* einen Radius von 0,1, der *Medium Scattering Layer* einen Radius von 0,5 und der *Deep Scattering Layer* einen Radius von 2. Auch dieser Wert ist beliebig anpassbar.



**Abbildung 3.40:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den Radius des *Shallow Scattering Layers*.



**Abbildung 3.41:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den Radius des *Medium Scattering Layers*.



**Abbildung 3.42:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den Radius des *Deep Scattering Layers*.

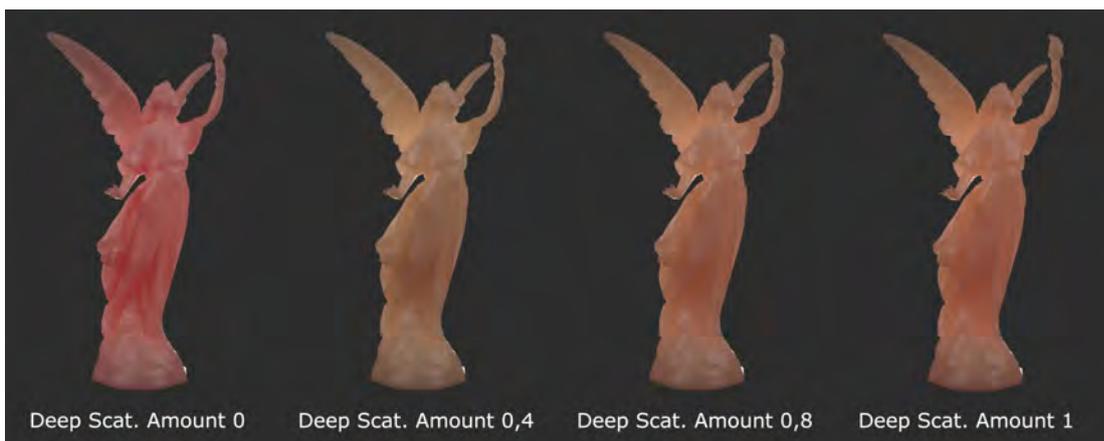
Das *Amount*-Attribut darf nicht mit der Funktion eines üblichen *Amount*-Parameters verwechselt werden, denn es steuert nicht die Stärke des *Scattering*-Effektes für den einzelnen *Layer*, sondern regelt den Anteil der entsprechenden *Scattering*-Ebene im Bezug zu den anderen beiden *Scattering Layern*. Sollten die drei *Amounts* in ihrer Summe den gewählten und unter den General Parameters eingetragenen *Max SSS Amount* überschreiten, werden die einzelnen Werte der *Amounts* automatisch vom System normiert, damit der Gesamtwert wieder dem zuvor festgelegten *Max SSS Amount* entspricht, und diese Begrenzung eingehalten wird. Per Default-Einstellung haben *Shallow* und *Medium Scattering Layer* einen *Amount* von 1 und der *Deep Scattering Layer* einen *Amount* von 0,4.



**Abbildung 3.43:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den *Amount* des *Shallow Scattering Layers*.



**Abbildung 3.44:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den *Amount* des *Medium Scattering Layers*.



**Abbildung 3.45:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den *Amount* des *Deep Scattering Layers*.

### 3.6.4 Primary und Secondary Reflection

Zusätzlich zu den drei *Scatter*-Ebenen hat das *Skin Material* zwei weitere Ebenen, welche für die Reflektionen verantwortlich sind. Die erste dieser Ebenen ist die *Primary Reflection*-Ebene, die zweite die *Secondary Reflection*-Ebene. Die Chaosgroup macht keine Angaben wofür genau die beiden Reflektionsschichten implementiert wurden. Die Vermutung liegt aber nahe, dass damit die Möglichkeit zur Verfügung gestellt werden soll unterschiedliche Reflektionsstärken beispielsweise zum Einen für die generelle Reflektion der Haut über die *Primary Reflection* und zum Anderen für die schweißbedeckte Stirn über die Verwendung einer *Reflectionmap* für die *Secondary Reflection* steuern zu können. Beide *Reflection-Layer* haben die folgenden Attribute:

#### **Color**

Mit der *Color* kann die Farbe der Reflektionen festgelegt werden, meist wird hier jedoch weiß oder ein Graustufenwert gewählt, um durch die Farbwahl lediglich die Helligkeit der Reflektionen zu beeinflussen. In Spezialfällen kann auf Wunsch aber auch jede beliebige andere Farbe gewählt werden.

#### **Amount**

Der *Amount* bestimmt wie stark das Material reflektieren soll.

#### **Glossiness**

Die *Glossiness* bestimmt die Schärfe der Reflektionen und damit, wie matt oder glänzend das Material ist.

#### **Subdivs**

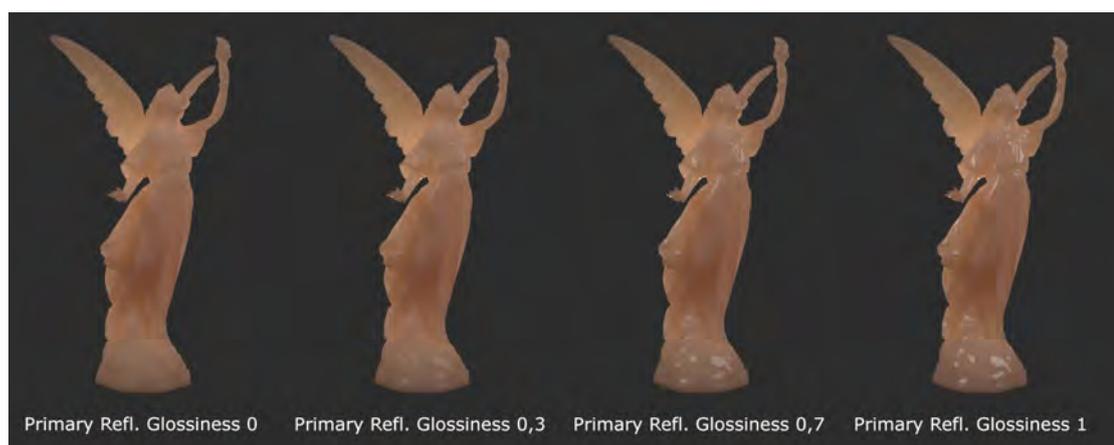
Die *Subdivs* bestimmen wie hoch die Reflektionen gesampled werden und damit, wie gut die Qualität, beziehungsweise wie hoch oder niedrig das Rauschverhalten an den Stellen des Materials, an denen die Reflektionen auftreten, ist.

#### **Fresnel**

Mit diesem Parameter kann der *Fresnel*-Effekt für die Reflektionen aktiviert werden.

#### **Fresnel IOR**

Ist der *Fresnel*-Effekt für die Berechnung der Reflektionen aktiviert, kann hier der Brechungsindex für die Berechnung festgelegt werden.



**Abbildung 3.46:** Die Veränderungen des Shaders mit unterschiedlichen Einstellungen für den *Amount* der *Glossiness* des *Primary Reflection Layers*.

### 3.6.5 Multiple Scattering Options

Wie auch schon beim *FastSSS2*-Shader gibt es auch beim *VRaySkin*-Shader drei Optionen, zwischen denen für das *Multiple Scattering* gewählt werden kann. Per Default ist *Raytraced* als Verfahren ausgewählt und dies ist auch der einzige der drei Modi, der das *Subsurface Scattering* über echtes *Raytracing* innerhalb des Volumens der Geometrie berechnet. Somit ist diese Prozedur die am physikalisch plausibelsten und genaueste Methode, und da es in dieser Arbeit um fotorealistische und physikalisch plausible Effekte gehen soll wird ausschließlich mit diesem Verfahren gearbeitet. Die anderen beiden zur Option stehenden Verfahrensweisen sind zum einen die *Prepass-based illumination map* und zum anderen die *Object-based illumination map*. Beide Verfahren verwenden *Maps* ähnlich der *irradiance map*, um damit *Subsurface Scattering*-Effekte zu approximieren. Sie unterscheiden sich lediglich in ihrem *Sampling*-Verfahren. Die beiden *Map*-Varianten sparen deutlich an Renderzeit, kommen aber in ihrer Qualität nicht an die *Raytracing*-Methode heran.

### 3.6.6 Options

#### Trace Reflections

Die Berechnung des *Tracings* der Reflektionen, die über die beiden zuvor beschriebenen *Reflection Layer* gesteuert werden, kann mit diesem Attribut aktiviert und deaktiviert werden. Shader, bei denen die Reflektionen keine entscheidende Rolle spielen, können durch die Deaktivierung im Rendering deutlich beschleunigt werden, da dann nur noch Highlights berechnet werden.

#### Max Reflection Depth

Mit diesem Parameter kann eine maximale Tiefe für die Reflektionen der *Primary* und der *Secondary Reflections* festgelegt werden.

#### Reflection Cutoff

Der *Reflection Cutoff*-Parameter erlaubt es Reflektionen, die unterhalb eines bestimmten *Cutoff* Wertes liegen, nicht zu berechnen.

### 3.6.7 Bump and Normal Mapping

Auch das *VRaySkinMtl.* stellt für die Berechnung von Tiefeneindrücken über Texturen einen Knoten für *Bump* und *Normal Maps* bereit.

## 3.7 Zusammenfassung

Zum weiteren Verständnis dieser Arbeit und des *Subsurface Scattering*-Effektes an sich, ist die Kenntnis einiger Grundlagen von Nöten. Zum einen ist es von Bedeutung, sich bewusst zu werden, dass unter Transmission in diesem Zusammenhang das Durchdringen von Lichtwellen durch ein Material hindurch zu verstehen ist. Wichtig ist hier die Unterscheidung

zwischen direkter Transmission und diffuser Transmission, die im ersten Fall zu Transparenz also Durchsichtigkeit des Materials führt, im zweiten Fall zu Transluzenz also Lichtdurchlässigkeit ohne Durchsichtigkeit. Transluzente Materialien sind wiederum die, bei denen der Subsurface Scattering-Effekt auftritt. Berechnen lässt sich die Lichtdurchlässigkeit mit ihrem Kehrwert, also der Lichtundurchlässigkeit oder auch Opazität.

Verschiedene Materialien haben bei gleicher Beleuchtung ein unterschiedliches Reflektionsverhalten. Um das Streuverhalten unterschiedlicher Materialien beschreiben zu können, gibt es eine Klasse von Funktionen die *Bidirectional Reflectance Distribution Functions*. Diese Funktionen sind lediglich für die Berechnung von Reflektionen an der Oberfläche von Objekten ausgelegt. Um zusätzlich auch das Eindringen von Licht unter die Oberfläche und das Austreten an einer anderen Stelle des Objektes bei den Berechnungen zu berücksichtigen, und somit *Subsurface Scattering* berechnen zu können, muss eine andere Art von Funktion, die *Bidirectional Scattering Surface Distribution Function*, verwendet werden. Da diese Formeln zur Implementation für die 3D-Computergrafik zu komplex sind, werden lediglich Approximationen an diese Funktionen implementiert.

Die Chaos Group stellt in seiner aktuellen Version 3.1 des Renderers V-Ray für Maya, dem aktuellen Vorreiter auf dem Gebiet der physikalisch plausiblen Renderer, drei Shader zur Verfügung mit denen die Realisierung von *Subsurface Scattering*-Visualisierungen ermöglicht werden. Das *V-Ray Material*, der Standard V-Ray Shader, bietet die Möglichkeit mit einigen Einstellungen *Subsurface Scattering* umzusetzen. Außerdem gibt es zwei auf *Subsurface Scattering* spezialisierte Materialien, den *FastSSS2 Shader* und das neue *Skin Material*. Alle drei Shader haben eine Vielzahl an Attributen und Parametern, die zahlreichen Einstellungsmöglichkeiten zur Kontrolle des gewünschten Effektes bieten.

In den folgenden Kapiteln werden diese drei Shader für prototypische Umsetzungen herangezogen, um anhand dessen die Ergebnisse, die sich mit ihnen erzielen lassen, sowie ihre Handhabung und die benötigten Renderzeiten zu analysieren.

## Kapitel 4

# Analyse des V-Ray Materials und des FastSSS2-Shaders für die Visualisierung von Subsurface Scattering

### 4.1 Einleitung

Das hier vorliegende Kapitel ist das erste von zwei Kapiteln, welches sich mit prototypischen praktischen Umsetzungen auseinandersetzt, die dazu herangezogen werden sollen, zu analysieren, welche Methoden zur Visualisierung von *Subsurface Scattering* die besten Ergebnisse liefert. Im Rahmen dieser Arbeit werden alle prototypischen Implementierungen mit Autodesk Maya 2016<sup>1</sup> und V-Ray 3.1 von der Chaosgroup<sup>2</sup> realisiert.

In diesem ersten methodischen Kapitel wird die Hypothese aufgestellt, dass der auf *Subsurface Scattering*-Visualisierungen spezialisierte *FastSSS2*-Shader bei der Umsetzung von Renderings mit *Subsurface Scattering*-Effekten bessere Resultate, sowohl in Bezug auf den Fotorealismus der Ergebnisse, als auch hinsichtlich der Renderzeiten, erzielt, als das Standard *V-Ray Material*, welches *Subsurface Scattering* lediglich als eine zusätzliche Option seinem Funktionsumfang hinzugefügt hat.

Aufgrund der Tatsache, dass die *Chaosgroup* mit dem *FastSSS2*-Shader nachträglich einen Spezialshader für *Subsurface Scattering*-Visualisierungen für das Rendering mit *V-Ray* als Ergänzung zu ihrem Standard *V-Ray Material* auf den Markt gebracht hat, liegt die Vermutung nahe, dass dieser spezialisierte Shader deutliche Vorteile dem Standard Shader gegenüber haben muss, welche sich in schnelleren Renderzeiten, einer einfacheren Handhabung oder einer besseren Qualität der Ergebnisse widerspiegeln. Um diese Hypothese zu belegen und die Unterschiede in den Funktionsweisen der beiden Materialien herauszuarbeiten, sol-

---

<sup>1</sup><http://www.autodesk.de/products/maya/overview>

<sup>2</sup><http://docs.chaosgroup.com/category/vray>

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---

len exemplarisch an zwei Anwendungsbeispielen die beiden Materialien Wachs und Marmor jeweils mit dem *Standard V-Ray Material* und dem *FastSSS2-Shader* gerendert und analysiert werden, welche Variante sowohl subjektiv als auch objektiv betrachtet, die besseren Ergebnisse liefert. Die subjektiven Parameter sind dabei die Handhabung und der eigene visuelle Eindruck des Ergebnisses im Vergleich mit Fotos. Als objektive Parameter dienen zum einen die Renderzeiten und zum anderen der Versuch die subjektiven Parameter mit einer statistischen Auswertung, die im Kapitel 7 folgt, objektiv bewertbar zu machen. Ziel ist es zu analysieren, welcher der beiden Shader besser für die Umsetzung von fotorealistischen Renderings mit *Subsurface Scattering*-Effekten geeignet ist. Dazu werden zwei exemplarische Beispielszenen realisiert, deren Materialien möglichst unterschiedlich sein sollen, um die Shader in einer möglichst großen Vielfalt ihrer Möglichkeiten zu testen. Daher wird für die prototypischen Umsetzungen zum einen das Material Marmor ausgewählt, welches ein typisches anteilig transluzentes Material ist, das sehr schön die Weichheit, welche *Subsurface Scattering* der Oberfläche von Objekten verleiht, repräsentiert. Zum anderen wird Kerzenwachs als zweites Anwendungsbeispiel bestimmt, da dieser zusätzlich zum weichen Effekt auch noch den speziellen Effekt des Durchleuchtens, den einige transluzente Objekte durch *Subsurface Scattering* erhalten, beinhaltet. Ein weiterer Grund für die Wahl dieser beiden Materialien ist, dass für Marmor im *FastSSS2-Shader* ein *Preset* zur Verfügung steht, für Kerzenwachs hingegen nicht. Auch hier wird der Analyse der Materialien dadurch eine weitere Vertiefung hinzugefügt, da sowohl die Verwendung eines bereitgestellten *Preset*s als auch ein ohne *Preset*, von Grund auf, aufgebauter *FastSSS2-Shader* mit der Verwendung des *V-Ray Materials* verglichen wird.

#### 4.2 Exemplarische Implementierung einer Kerzen-Szene mit Subsurface Scattering

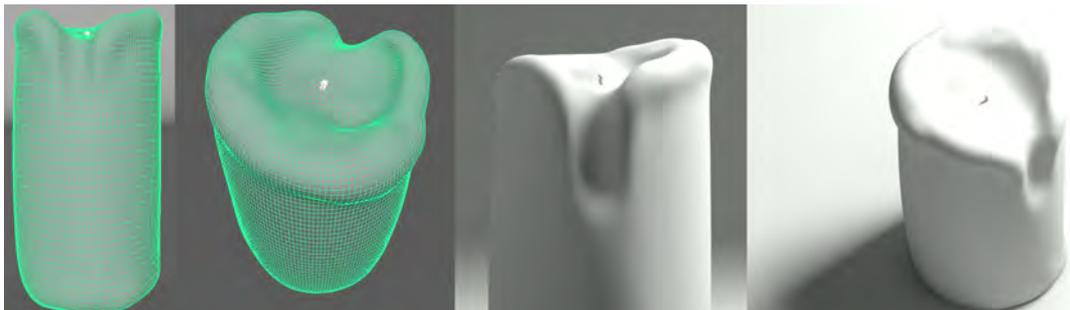


**Abbildung 4.1:** Eine Reihe von realen Aufnahmen von brennenden Kerzen mit *Subsurface Scattering*-Effekt, welche als Referenzen dienen, um ein möglichst fotorealistisches Rendering zu erzeugen. Bildquellen von links nach rechts: <http://goo.gl/Mm2pto> Stand: 26. Januar 2016, <http://goo.gl/5fDQKC> Stand: 26. Januar 2016, <http://goo.gl/BM90ul> Stand: 26. Januar 2016, <http://goo.gl/M8IGgE> Stand: 26. Januar 2016.

Eine brennende Kerze ist ein Paradebeispiel für *Subsurface Scattering*, bei dessen Visualisierung es nicht nur auf den Effekt der besonderen Weichheit der Oberflächenstruktur des Wachses, wie er für Materialien bei denen *Subsurface Scattering* auftritt, typisch ist, ankommt, sondern auch auf das etwas speziellere Durchleuchten des Lichtes. Aus diesem Grund soll die Umsetzung einer solchen Szene in diesem Kapitel prototypisch jeweils mit dem *V-Ray Material* und dem *V-Ray FastSSS2-Shader* durchgeführt und die beiden Materialien so hinsichtlich ihrer Handhabung, Renderzeiten und Ergebnisse miteinander verglichen werden.

### 4.2.1 Modell und Szenenaufbau

Das Modell der Kerze ist ein freies Modell von TurboSquid Aphrodite3d<sup>3</sup> und besteht aus 12800 Polygonen. Das ursprüngliche Modell hat noch einen Kerzenständer sowie eine ausmodellerte Flamme, diese werden aber entfernt, da hier nur das reine KerzenwachsmodeLL verwendet werden soll. Die Flamme wird durch eine deutlich realistischere *Fluid-Simulation* visualisiert, wie später in diesem Kapitel noch näher erläutert wird. Auch der Docht wird neu modelliert und ausgetauscht, da der originale Docht aus dem Modell zu dünn und zu gerade ist und daher nicht den Anforderungen für diese Szene entspricht.



**Abbildung 4.2:** Das Mesh des verwendeten Modells der Kerze aus zwei Perspektiven in *Autodesk Maya*. Das Mesh besteht aus 12800 Polygonen. Dazu ein Ambient Occlusion Rendering, welches dafür verwendet wird das Model auf Fehler im Mesh zu überprüfen. Schwarze Flächen auf der Oberfläche des Renderings würden auf Fehler im Modell hinweisen.

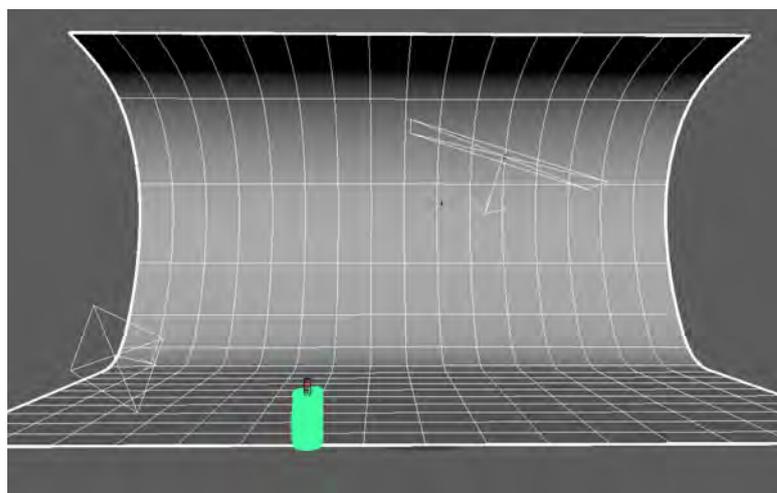
Um den Fokus auf das *Subsurface Scattering* zu legen und auch beim Vergleich der beiden Shader für die objektiven Parameter, wie den Renderzeiten, möglichst gute Bedingungen zu schaffen, wird für die Szene ein einfaches Studio-Setup gewählt und in *Autodesk Maya* aufgebaut. Dazu wird das Kerzenmodell in eine Hohlkehle gestellt, welche mit einem diffusen, neutralen grau einen nicht ablenkenden Hintergrund schafft, welcher gegebenenfalls in *Photoshop* für Nachbearbeitungen auch schnell ausgetauscht werden kann. Durch seine diffusen, nicht reflektierenden Eigenschaften hat er auch keine Auswirkungen auf den *Subsurface Scattering*-Effekt der Shader. Als Grundbeleuchtung der Szene wird mit zwei *V-Ray Rectangle Lights* gearbeitet. Einem Hauptlicht von oben rechts mit einer Intensität von 0,025 Watt und einem Fülllicht von vorne links mit einer Intensität von 0,003 Watt. Für die

<sup>3</sup><http://www.turbosquid.com/3d-models/3d-candle-light-model/519987>

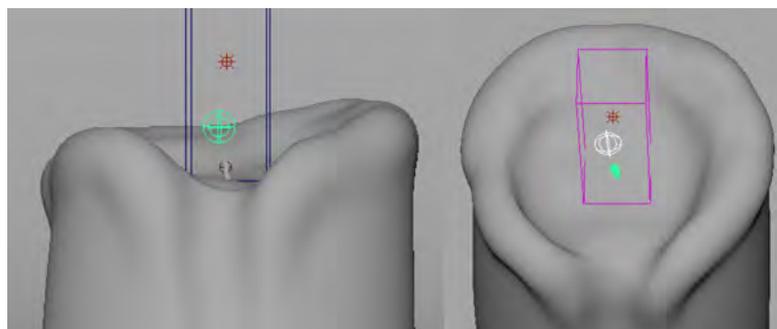
#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---

Aufnahmen wird eine physikalische *V-Ray* Kamera mit einer Blende von 2,8, einem ISO von 200 und einer *Shutterspeed* von 250 angelegt. Für die Umsetzung des erweiterten *Subsurface Scattering*-Effektes, bei dem das Licht durch das Objekt durchleuchten soll, wird zusätzlich zum *Basic-Lightset* eine weitere Lichtquelle benötigt. Diese wird, um den Szenenaufbau so physikalisch plausibel wie möglich zu gestalten, auf Höhe des Dochtes positioniert, so dass die Lichtquelle in etwa die gleiche Lage aufweist, wie bei einer realen Kerze die Flamme, welche in der Realität das Licht ausstrahlt, welches unter der Oberfläche gestreut wird (siehe Abbildung 4.4).



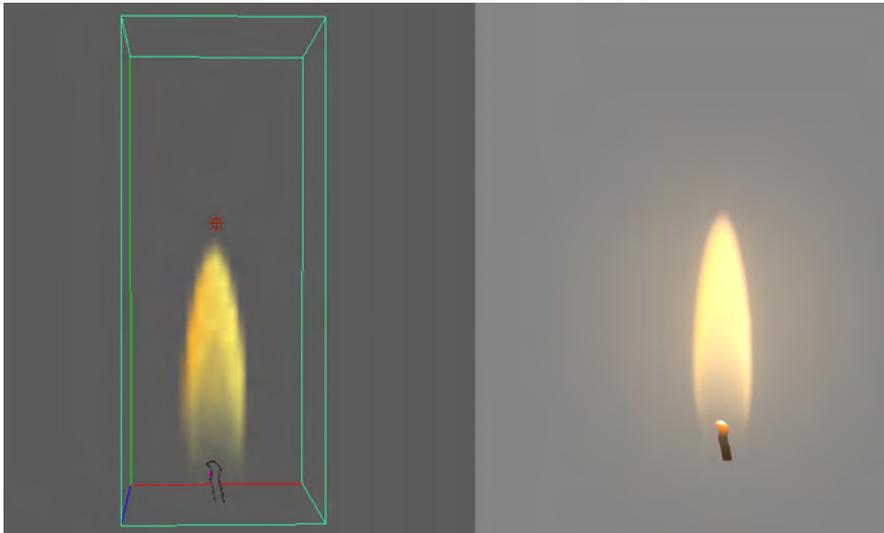
**Abbildung 4.3:** Der Aufbau der Szene. Das Modell der Kerze befindet sich in einer Hohlkehle und wird von einem Haupt und einem Fülllicht bestehend aus zwei *V-Ray Rectangle Lights* beleuchtet.



**Abbildung 4.4:** Die Positionierung der zusätzlichen Lichtquelle, welche für die Erzeugung des Effektes des Durchleuchtens durch *Subsurface Scattering* notwendig ist. Um die größtmögliche physikalische Plausibilität zu gewährleisten, ist sie in der Nähe des Dochtes, also dort, wo auch die reale Flamme die Lichtquelle darstellt, positioniert.

### Die Flamme

Um auch für die Flamme der Kerze ein möglichst realistisches und physikalisch plausibles Ergebnis zu erhalten, wird sie mithilfe von *Maya Fluids* erzeugt. Dafür wird eine *Fluid Box* erstellt und der Docht der Kerze als *Emitter*-Objekt festgelegt. Mit einer Reihe von Einstellungen des *Emitters* und des *Fluids* wird das *Fluid* in die Form und Farbe einer realistisch wirkenden Kerzenflamme gebracht und mit *V-Ray* gerendert. Da sich *Fluid*-Simulationen erheblich negativ auf Renderzeiten auswirken und dies für die Szenen mit dem eigentlichen *Subsurface Scattering* vermieden werden soll, wird die Flamme in einer separaten Szene erstellt, einzeln gerendert und die beiden Renderings der Kerze und der Flamme dann anschließend in *Photoshop* zusammengesetzt. Auf diese Weise kann auch aufwendiges *Lightlinking* umgangen werden, das sonst von Nöten gewesen wäre, um zu verhindern, dass Lichtquellen der Flamme das *Subsurface Scattering* beeinflussen, oder Lichtquellen zur Ausleuchtung des Wachsmodelles das Erscheinungsbild der Flamme verändern.



**Abbildung 4.5:** Die *Fluid*-Simulation der Kerzenflamme in der *Maya*-Szene (links) und das Rendering der Flamme (rechts).

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---

##### 4.2.2 Der FastSSS2-Shader



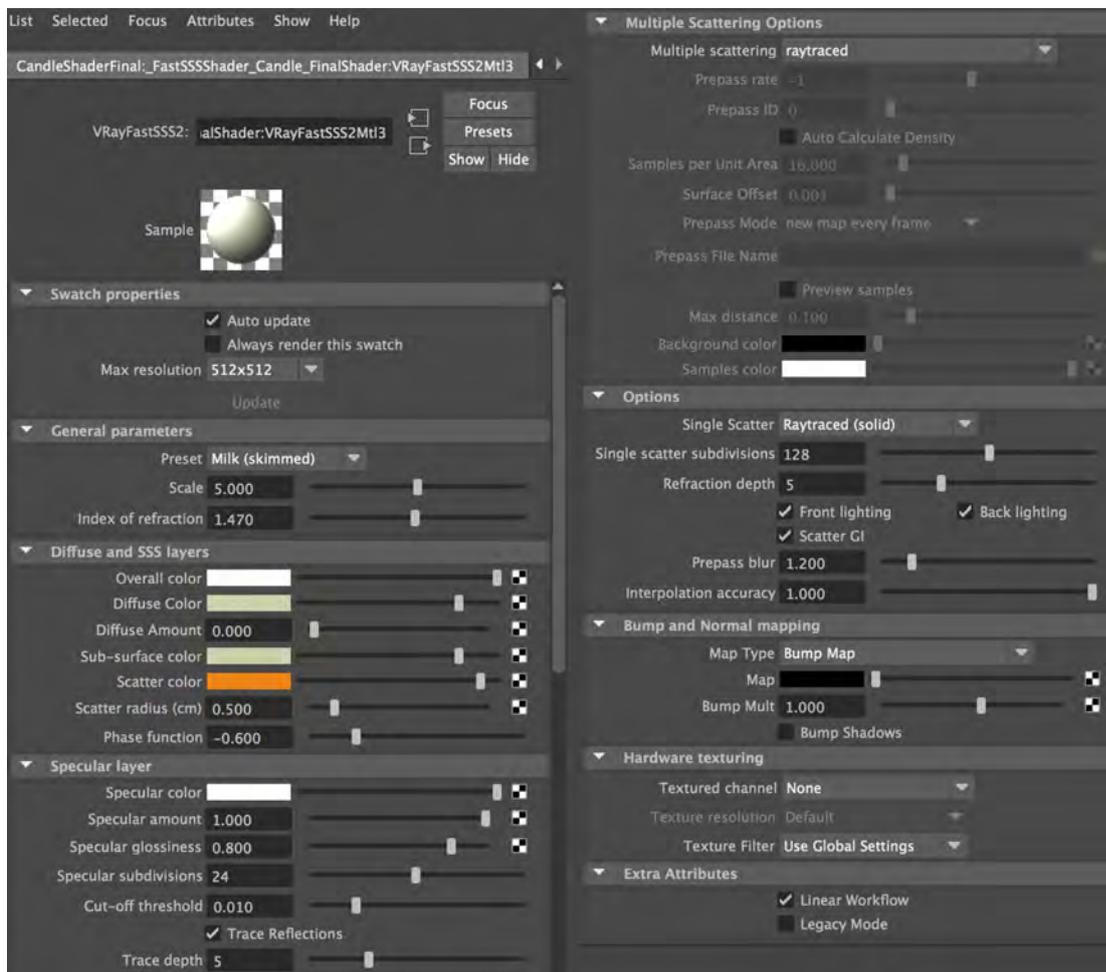
**Abbildung 4.6:** Eine Visualisierung des finalen Wachs-Shaders in der *FastSSS2*-Shader-Variante auf einem *Shader Template*.

Für Wachs gibt es kein vorgefertigtes *Preset*, welches die *Chaos Group* für den *FastSSS2*-Shader zur Verfügung stellt, daher muss der Shader entweder über den *Custom*-Modus aufgebaut werden oder ein anderes *Preset* abgewandelt werden. Hier wird als Grundlage für den Aufbau der *FastSSS2*-Variante des Wachs-shaders das *Preset* "milk (skimmed)" gewählt, da die Farbgebung der gewünschten Farbe für das Wachs schon recht nahe kommt.

Für den *Index of Refraction* wird ein Wert von 1,47 gewählt. Dieser Wert wird für Öle angegeben. Da Wachs in seiner Beschaffenheit der von Öl entspricht, wird hier der dafür angegebene IOR verwendet.

Als *Scale* wird ein Wert von 5 bestimmt, da für den Effekt des Durchleuchtens der brennenden Kerze, das Licht relativ tief in das Material eindringen soll. Um zu Beginn verschiedene *Presets* testen zu können, ohne den Wert immer wieder zu verlieren, wird das *Scale*-Attribut in den *General Parameters* und nicht der *Scatter Radius* für diese Einstellung herangezogen. Dieser ist mit 0,5 daher vergleichsweise sehr klein gehalten.

## 4.2. Exemplarische Implementierung einer Kerzen-Szene mit Subsurface Scattering



**Abbildung 4.7:** Die Attribute des finalen Wachs-Shaders in der *FastSSS2*-Variante.

Die *Overall Color* dient hier lediglich für die Helligkeitsverteilung und wird auf weiß gestellt, um die maximale Helligkeit für den Shader zu gewährleisten.

Mit einem *Diffuse Amount* von 0 wird festgelegt, dass der Shader keinen diffusen Anteil besitzt und damit auch der Farbanteil vollständig über die *Sub-surface color* bestimmt wird. Die *Sub-surface color* gibt der Kerze mit einem cremefarbenen Ton ihre Farbe und mit einem dunkleren ocker-/orange-farbenen Ton für die *Scatter color* wird das eindringende Licht eingefärbt.

Da ein recht starker Durchleucht-Effekt gewünscht ist, wird für die *Phasefunction* ein negativer Wert von -0,6 bestimmt. Der negative Wert sorgt dafür, dass die Lichtstrahlen entgegen der Richtung *gescattert* werden, aus der sie kommen.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---

Da Kerzenwachs relativ klare Reflektionen aufweist, wird eine *Reflection Glossiness* von 0,8 gewählt, der den Reflektionen nur eine geringe Unschärfe verleiht und die *Specular Subdivisions* werden auf 24 erhöht, um das Rauschen in den Reflektionen zu minimieren. Da es in dieser Arbeit um physikalische Plausibilität und Fotorealismus geht, fällt die Auswahl für das *Multiple Scattering*-Verfahren ganz eindeutig auf "raytraced", da es die einzige der zur Auswahl stehenden Varianten ist, welche das *Multiple Scattering* physikalisch plausibel berechnet und nicht anhand einer *Map* nur nachbildet.

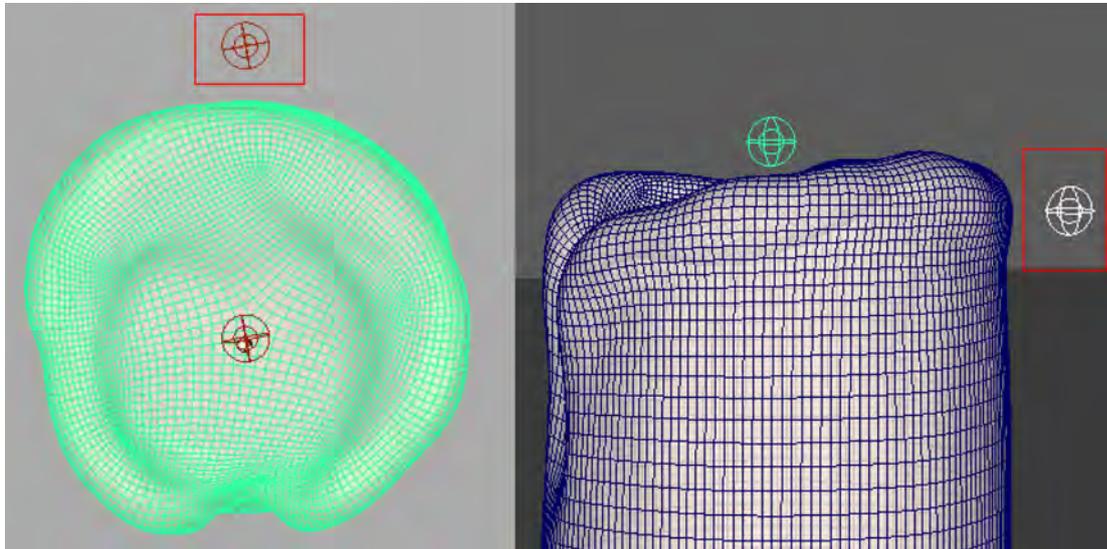
Als *single Scattering-Option* wird für den Wachs-Shader *Raytraced (solid)* gewählt, da dieses das Verfahren der Wahl ist, wenn es um hochgradig *scatternde* Materialien geht, welche nicht transparent sind. Und genau dieses Kriterium soll mit dem Kerzenmodell erfüllt werden, eine hochgradig *scatterndes* Material, bei dem viele Lichtstrahlen unter die Oberfläche eindringen und dort gestreut werden, welches aber nicht wie Glas oder Wasser transparent ist. Daher ist hier die zusätzliche Verfolgung der *Refraction Rays*, die das Verfahren des *Raytraced (refractive)* ermöglicht, nicht notwendig. Da das *Scattering* starkes Rauschen im Shader verursacht, müssen die *Single Scatter Subdivs* bis auf einen Wert von 128 erhöht werden, um ein möglichst rauschfreies Rendering zu gewährleisten.

Für die Beleuchtung werden *Front-* und *Back lighting* aktiviert. Es gibt zwar keine Lichtquelle, die das Objekt wirklich rein von hinten anstrahlt, aber die Lichtquellen, die oberhalb des Objektes positioniert sind, durchdringen das Material teilweise auch von der Rückseite aus. Um möglichst realistische Ergebnisse zu erhalten, wird außerdem auch das *Scatter GI* aktiviert.



**Abbildung 4.8:** Das finale Ergebnis der zusammengesetzten Renderings der Kerze mit dem *FastSSS2*-Shader und der Flamme.

### 4.2.3 Der V-Ray Material Shader



**Abbildung 4.9:** Die Anordnung der zusätzlichen Lichtquelle, die bei der Umsetzung mit dem *V-Ray Material* für den Effekt des Durchleuchtens des *Subsurface Scatterings* notwendig ist.

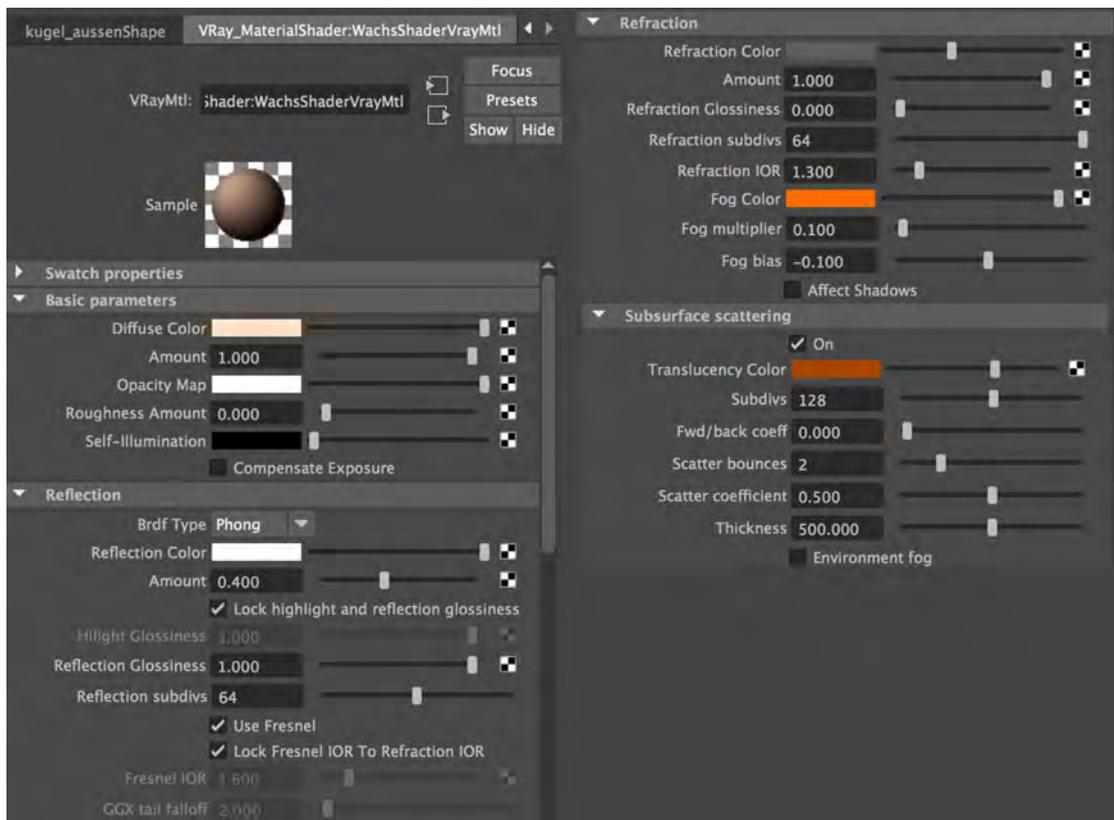


**Abbildung 4.10:** Eine Visualisierung des finalen Wachs-Shaders in der *V-Ray Material*-Variante auf einem *Shader Template*.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

Für die *V-Ray Material*-Variante des Wachs-Shaders wird für die Grundfarbgebung das Attribut der *diffuse Color* auf einen cremefarbenen Farbton gesetzt. Als *Brdf Type* wird *Phong* gewählt und der *Reflection Amount* wird mit 0,4 relativ schwach eingestellt, um der Oberfläche einen weichen, nicht zu stark reflektierenden Eindruck zu geben. Um das Rauschen in den Reflektionen gering zu halten, werden die *Reflection Subdivs* auf 64 erhöht.

Um einen relativen starken *Subsurface Scattering*-Effekt zu erhalten, wird dem Material mit einem *Refraction Amount* von 1 und einer *Refraction Color* in einem hellen Grauton mit den RGB Werten 0,374/0,374/0,374 ein relativ hoher Transluzenter Anteil gegeben. Wie auch in der *FastSSS2*-Variante wird für den *Index of Refraction* ein Wert von 1,47 gewählt, dieser Wert wird, wie bereits beschrieben, für Öle angegeben und da Wachs, wie bereits erwähnt, in seiner Beschaffenheit der von Öl entspricht, kann hier dieser Wert übernommen werden.



**Abbildung 4.11:** Die Attribute des finalen Wachs-Shaders in der *V-Ray Material*-Variante.

## 4.2. Exemplarische Implementierung einer Kerzen-Szene mit Subsurface Scattering

---

Für die *Fog Color* wird ein ocker/orange-farbener Ton gewählt, der dunkler als die diffuse Farbe, aber etwas heller als die gewählte *Translucency Color* ist. Wichtig ist, dass die gewählte *Fog Color* keinen Helligkeitswert von 1 haben darf, da ohne Schwarzanteil das Licht nicht absorbiert wird, was bedeutet, dass kein *Subsurface Scattering*-Effekt berechnet werden kann.

Der *Fog Multiplier* wird auf 0,1 gesetzt und der *Fog bias* auf -0.1. Die beiden Werte regulieren das *Scattering* abhängig von der Dicke des Objektes und sorgen in dieser Szene so dafür, dass der Effekt des Durchleuchtens nicht auf die ganze Kerze übertragen wird, sondern sich auf den oberen Bereich in der Nähe der Lichtquellen beschränkt.

Für die *Translucency Color* wird ebenfalls ein ockerfarbener Ton bestimmt, der leicht dunkler ist als der Farbwert der *Fog Color*, was einen schöneren Verlauf der Farben ermöglicht.



**Abbildung 4.12:** Das finale Ergebnis der zusammengesetzten Renderings der Kerze mit dem *V-Ray Material-Shader* und der Flamme.

#### 4.2.4 Vergleich

Im folgenden Abschnitt sollen die beiden unterschiedlichen Varianten der prototypischen Umsetzung der Kerzen-Szene miteinander verglichen werden und anhand unterschiedlicher subjektiver und objektiver Kriterien analysiert werden.

Nachdem später beide prototypische Umsetzungen dieses Kapitels in den einzelnen Kategorien "Qualität der Ergebnisse", "Renderzeiten" und "Handhabung" bewertet sind, folgt abschließend in 4.4 eine generelle Schlussfolgerung aus den Ergebnissen mit einer Beurteilung der Shader sowie eine Einordnung für welche Szenarien, welcher Shader der geeignetere ist.

##### Qualität der Ergebnisse

Subjektiv betrachtet liefert der *FastSSS2*-Shader gegenüber dem *V-Ray Material* die qualitativ hochwertigeren und besseren Ergebnisse. Allein durch die Tatsache, dass für das *V-Ray Material* eine zusätzliche Lichtquelle hinter die Kerze gesetzt werden muss, um den Effekt des Durchleuchtens durch *Subsurface Scattering* zu erzeugen, was die physikalische Plausibilität in der Umsetzung der Szene gegenüber dem *FastSSS2*-Shader deutlich herabsetzt, scheint hier der *FastSSS2*-Shader die bessere Wahl zu sein.

Auch der visuelle Eindruck der Ergebnisse fällt bei der Realisierung der Kerzenszene mit dem *FastSSS2*-Shader realistischer aus. Das Material wirkt weicher, der Farbverlauf ist schöner und die Übergänge fließender als bei der Umsetzung mit dem *V-Ray Material*. Wenn die Ergebnisse mit realen Fotos von Kerzenwachs verglichen werden, kommt der Prototyp mit dem *FastSSS2*-Shader an den wachstartigen Eindruck einer echten Kerze deutlich näher heran.



**Abbildung 4.13:** Direkter Vergleich der Renderings der Kerze in den beiden Shader-Varianten. Links die *V-Ray Material*-Variante, rechts die *FastSSS2*-Variante.

### Handhabung der Shader

Auf den ersten Blick wirkt der *FastSSS2*-Shader dadurch, dass er ein auf das *Subsurface Scattering* spezialisierter Shader ist, mit seinen vielen Einstellungsmöglichkeiten bezüglich dieses Effektes deutlich komplexer im Gegensatz zu dem kleinen Bereich des *V-Ray Materials*, der sich der Umsetzung dieses Effektes zu widmen scheint. Aufgrund der Tatsache, dass dann doch viele weitere Parameter des *V-Ray Materials* in das Ergebnis des *Subsurface Scatterings* mit einfließen und deren Zusammenspiel, aufgrund kaum existierender Dokumentationen diesbezüglich, zunächst eigenständig erarbeitet werden muss, nimmt die Einarbeitung in das Verständnis der Parameter und ihres Zusammenwirkens hier einige Zeit in Anspruch. Da zum Zeitpunkt der Umsetzung der Kerzen-Szenen beide Shader in Bezug auf die Realisierung von *Subsurface Scattering*-Effekten unbekannt waren, dauerte die Einarbeitung in das Verständnis der Shaderaufbauten und das Anfertigen der beiden Shader in etwa gleich lang.

### Renderzeiten

Die Renderzeiten der beiden finalen Shadervarianten werden für die Gegenüberstellung jeweils an einem *Shader Template* und am finalen Modell mit jeweils unterschiedlichen *Rendersettings* in *High Quality* und *Low Quality*-Einstellungen gemessen, um einen aussagekräftigen Vergleich zu gewährleisten.

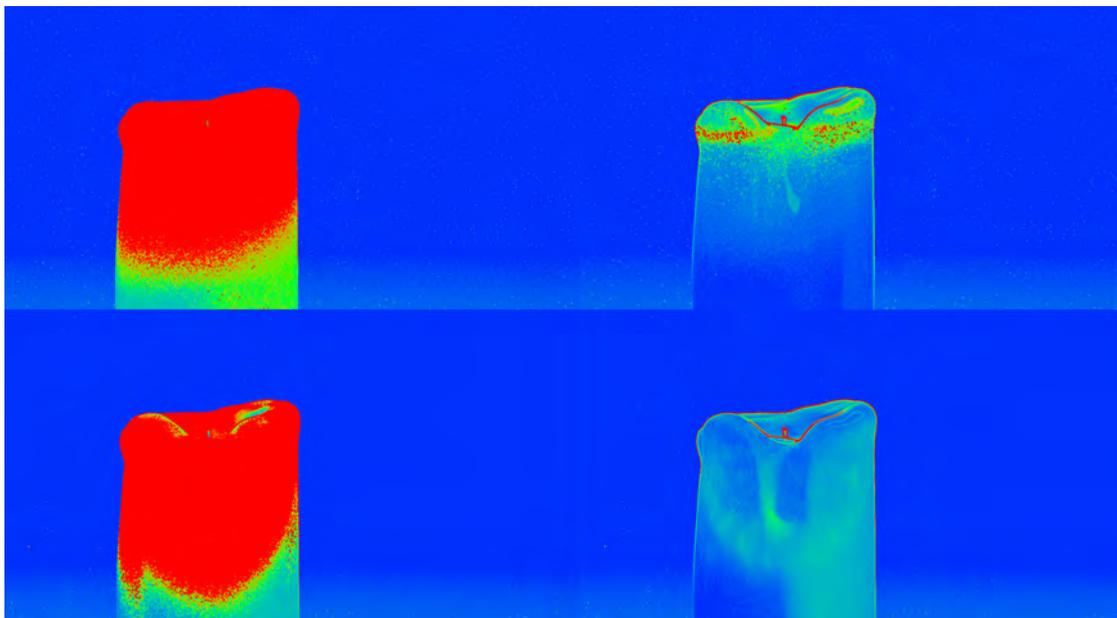
Die Renderings für die Gegenüberstellung der Renderzeiten der Wachs-Shader werden alle auf einem Rechner mit 2,3 GHz Intel Core i7 mit 16GB 1600 MHz DDR3 und einer NVIDIA GeForce GTX 650M 1024MB durchgeführt. Die *Shader Templates* werden mit einer Auflösung von 640 mal 800 Pixeln gerendert. Die finalen Renderings haben in beiden Varianten eine Auflösung von 1280 x 720 Pixeln.

	FastSSS2-Shader	V-Ray Material
Shader-Template Low Quality	0h12m53,6s	0h08m06,5s
Shader-Template High Quality	0h55m09,3s	0h27m45,5s
Final Rendering Low Quality	0h07m52,7s	0h08m40,7s
Final Rendering High Quality	2h48m34,3s	0h41m11,4s

**Tabelle 4.1:** Vergleich der gemessenen Renderzeiten für die Wachs-Shader

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---



**Abbildung 4.14:** Die *Sample Rates* der Kerzenrenderings der verschiedenen Varianten *FastSSS2*-Shader (oben) und *V-RayMtl* (unten), jeweils in hoher und niedriger Render-Qualität. Dabei bedeutet rot niedriges *Sampling* über Grüntöne, die für mittleres *Sampling* stehen, bis hin zu Blautönen, die perfektes *Sampling* visualisieren. Es wurde versucht, für die beiden *High Quality*-Varianten der beiden Shader ähnliche Sampleraten zu erhalten, um eine gute Vergleichbarkeit in den Renderzeiten zu gewährleisten.

Die Renderzeiten zeigen, dass im Vergleich die Kerze mit dem *FastSSS2*-Shader deutlich länger für die Bildsynthese benötigt, als die Kerze, welche mit dem *V-Ray Material* geshadet worden ist. Für alle *Shader Template*-Renderings hat das *FastSSS2*-Material länger gerendert. Lediglich die *Low Quality*-Variante des finalen Renderings bestätigt mit einer Ausnahme die Regel. Hier liegt die Zeit für das *FastSSS2*-Shader-Rendering knapp unterhalb der des *V-Ray Materials*. Werden aber für beide Shader die Einstellungen so optimiert, dass das Ergebnis möglichst rauschfrei ist, so ist das *V-Ray Material* über 2 Stunden schneller als der *FastSSS2*-Shader. Im Punkt Renderzeiten ist also ganz deutlich das *V-Ray Material* der bessere der beiden Shader.

### 4.3 Exemplarische Implementierung einer Marmordrachen-Szene mit Subsurface Scattering



**Abbildung 4.15:** Eine Reihe von realen Aufnahmen von weißem Marmor in unterschiedlichen Szenarien, welche als Referenzen dienen, um ein möglichst fotorealistisches Rendering zu erzeugen. Bildquellen von links nach rechts und von oben nach unten: <http://goo.gl/113ht6> Stand: 27. Januar 2016 , <https://goo.gl/GMtSeU> Stand: 27. Januar 2016, <http://goo.gl/dn5wOV> Stand: 27. Januar 2016, <http://goo.gl/pkeU8X> Stand: 27. Januar 2016, <http://goo.gl/CckWZT> Stand: 27. Januar 2016, <http://goo.gl/z1QIYb> Stand: 27. Januar 2016.

Die zweite prototypische Umsetzung befasst sich mit dem Material Marmor, welches ebenfalls eines der typischen Substanzen mit *Subsurface Scattering* ist, in diesem Fall jedoch meist ohne den Effekt des Durchleuchtens. Das Durchleuchten tritt bei Marmor nur an wirklich dünnen Teilen des Objektes auf. Hauptsächlich liegt der Fokus bei der Umsetzung eines fotorealistischen Marmor-Renderings in der Ausarbeitung der besonderen Weichheit der Oberfläche, wie sie durch das *Scattering* des Lichtes nach dem Eindringen in das Material entsteht. Für die zweite prototypische Implementierung soll eine Drachenfigur aus Marmor auch jeweils mit dem *V-Ray Material* und dem *FastSSS2*-Shader realisiert werden.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

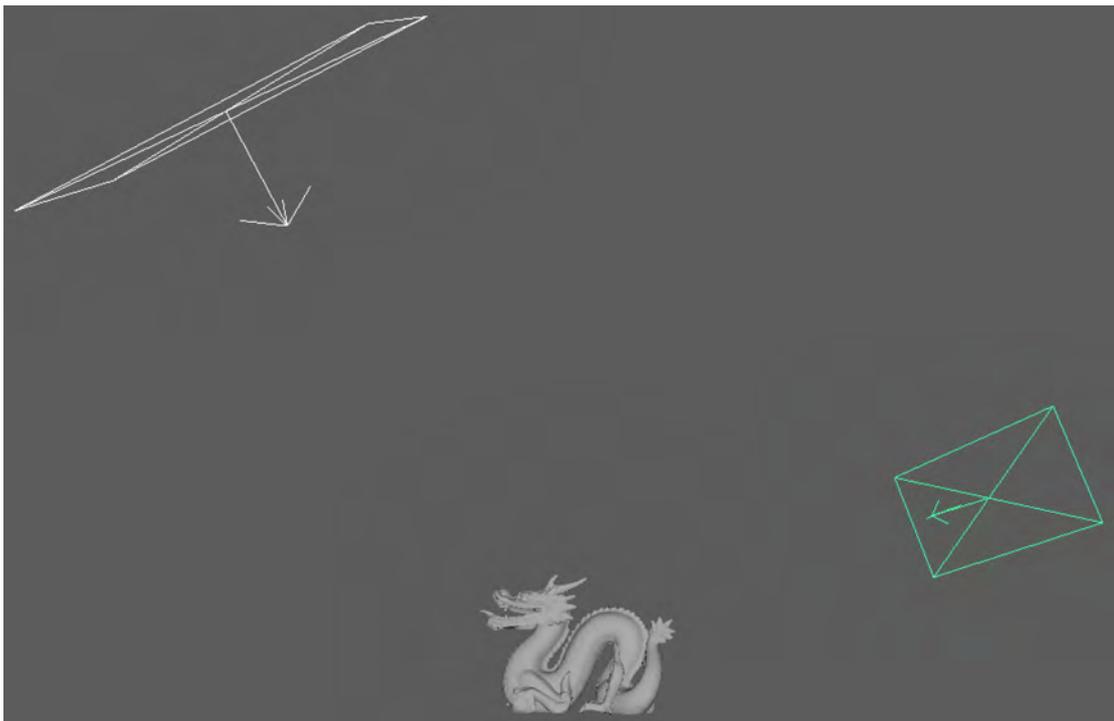
---

##### 4.3.1 Modell und Szenenaufbau

Als Modell wird der Stanford-Drache<sup>4</sup> aus der Reihe der Stanford Referenzmodelle verwendet. Das Modell besteht aus 100000 Polygonen.



**Abbildung 4.16:** Screenshot des *Meshes* des Drachen-Modelles und zwei *Ambient Occlusion*-Renderings zur Kontrolle des Modells.



**Abbildung 4.17:** Das *Light-Setup* der Drachen-Szene.

Wie auch schon zuvor bei der Kerzenszene, soll auch bei der Beispielszene des Marmordrachsens der Fokus des Vergleichs auf den beiden unterschiedlichen Shadervarianten für

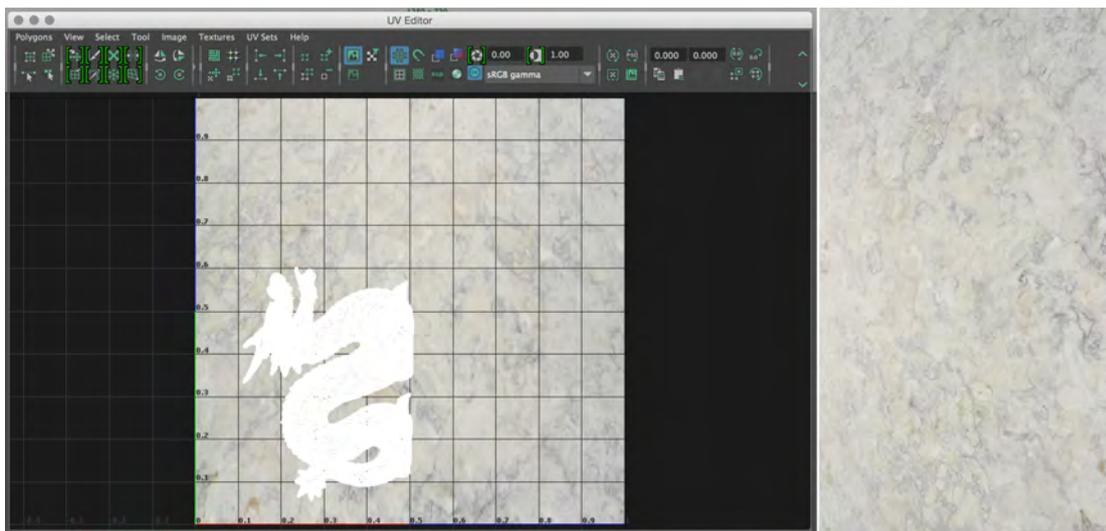
<sup>4</sup><http://graphics.stanford.edu/data/3Dscanrep/>

### 4.3. Exemplarische Implementierung einer Marmordrachen-Szene mit Subsurface Scattering

die Umsetzung des *Subsurface Scatterings* liegen und so wird auch bei dieser Szene ein Studio-Setup verwendet, um eine gute Ausgangssituation für den Vergleich der objektiven Parameter wie Renderzeiten zu schaffen. Als Grundbeleuchtung der Szene wird mit zwei *V-Ray Rectangle Lights* gearbeitet. Einem Hauptlicht von oben rechts mit einer Intensität von 0,025 Watt und einem Fülllicht von vorne links mit einer Intensität von 0,003 Watt. Für die Aufnahmen wird eine physikalische *V-Ray-Kamera* mit einer Blende von 1,2 einem ISO von 100 und einer *Shutterspeed* von 250 angelegt.

#### 4.3.2 Der FastSSS2 Shader

Da das Drachenmodell kein *UV-Map* hat, für den Shader aber mit Texturen gearbeitet werden soll, für deren verzerrungsfreie Darstellung im Rendering ein sauberes *UV-Map* von Nöten ist, wird als erstes ein *UV-Map* erstellt. Da das Modell nur aus einer Perspektive gerendert werden soll, wird an dieser Stelle auf ein aufwendiges *UV-Mapping* verzichtet und lediglich ein *Camera Based Mapping* erstellt (siehe Abbildung 4.18).



**Abbildung 4.18:** Das Kamera basierte *UV-Map* des Drachen-Modells (links) und die originale Marmor-Textur (rechts).

Für das Material Marmor stellt der *FastSSS*-Shader ein vorgefertigtes *Preset* bereit. Dieses wird als Grundlage für den Aufbau des Shaders übernommen und ein Paar kleine Änderungen vorgenommen, um den Shader in seinen Feinheiten an die gewünschten Ergebnisse anzupassen.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---



**Abbildung 4.19:** Das finale Ergebnis des Marmor-Shaders in der *FastSSS2*-Variante auf einem *Shader Template*.

So wird eine Marmor-Textur von Textures.com<sup>5</sup> in den Diffuse-Kanal des Shaders gelegt und der *Diffuse Amount* auf 0,1 gesetzt, damit der Shader einen kleinen diffusen Anteil erhält, welcher mit dem *Subsurface*-Anteil verrechnet wird. Außerdem wird die gleiche Textur auch in die *Sub-surface color* gelegt. Wichtig ist, dass für die Texturen der *linear Workflow* eingehalten wird und dementsprechend eine Gammakorrektur durchgeführt werden muss, da sie sonst zu dunkel berechnet werden.

Die *Specular glossiness* wird ein wenig reduziert auf einen Wert von 0,63 und die *specular subdivs* werden auf einen Wert von 32 erhöht, um das Rauschen in den Reflektionen zu minimieren. Als *Single Scatter*-Verfahren wird *Raytraced Solid* statt *Simple* gewählt, da besonders an den dünnen Stellen des Modells doch stärkeres *Scattering* visualisiert werden soll und *Raytraced solid* hier bessere Ergebnisse erzielt.

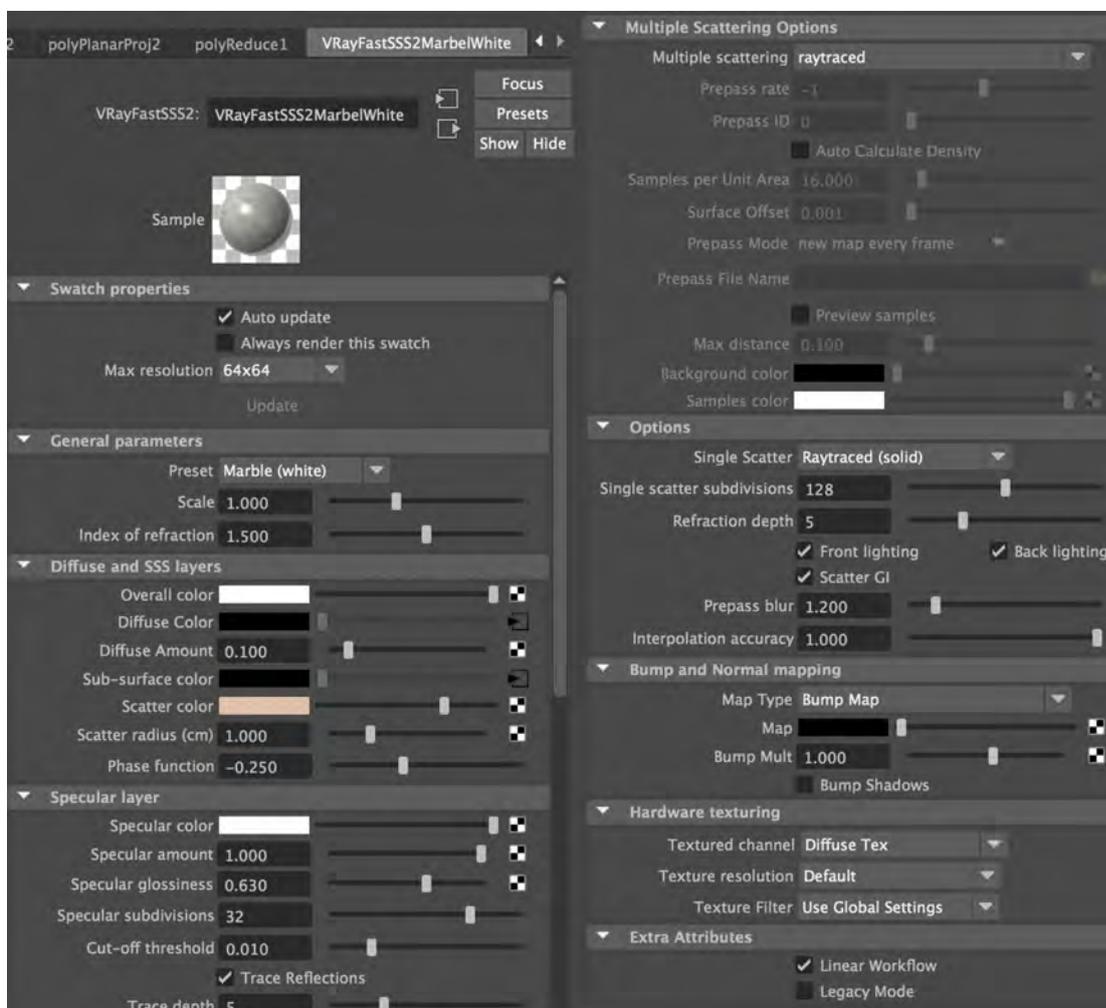
Da besonders an den Stellen, an denen das Material starkes *Scattering* des Lichtes aufweist, Rauschen im Bild entsteht, wird der Wert für die *Single Scatter Subdivisions* auf 128 erhöht, da ab diesem Wert das Bild nahezu rauschfrei ist.

Um zu gewährleisten, dass das *multiple Scattering* für das Licht sowohl berechnet wird, wenn das Objekt von hinten als auch wenn es von vorne angestrahlt wird, werden sowohl

<sup>5</sup><http://www.textures.com/download/marblewhite0021/9637> Stand: 28.Februar 2016

### 4.3. Exemplarische Implementierung einer Marmordrachen-Szene mit Subsurface Scattering

*front* als auch *back lighting* aktiviert. Es wurde zwar keine Lichtquelle direkt hinter das Objekt positioniert, aber das *Rectangle Light*, welches das Objekt von oben rechts beleuchtet, strahlt die Figur eher von seitlich-hinten an. Auch für diese Szene soll die *Global Illumination* für das *Subsurface Scattering* zur Gewährleistung eines möglichst realistischen und physikalisch plausiblen Ergebnisses berechnet und nicht über eine *Map* approximiert werden, daher wird die *scatter GI* aktiviert.



**Abbildung 4.20:** Die gewählten Shader-Attribute des finalen Marmor-Shaders in der *FastSSS2*-Variante.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

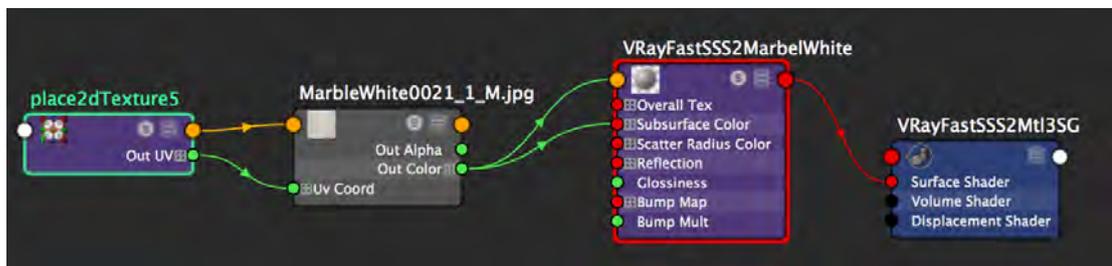


Abbildung 4.21: Das Shader-Netzwerk des Marmor-Shaders in der *FastSSS2*-Variante.



Abbildung 4.22: Das fertige Rendering des Marmor-Drachens in der *FastSSS2*-Shader-Variante.

### 4.3.3 Der V-Ray Material Shader



**Abbildung 4.23:** Das finale Ergebnis des Marmor-Shaders in der *V-Ray Material*-Variante auf einem *Shader Template*.

Um eine möglichst gute Vergleichbarkeit zu gewährleisten, wird für das *V-Ray Material* die gleiche Marmor-Textur verwendet, wie auch für den *FastSSS2*-Shader. In diesem Fall wird sie in den *Diffuse Color*-Kanal und in den Textur-Kanal für die *Translucency Color* gelegt.

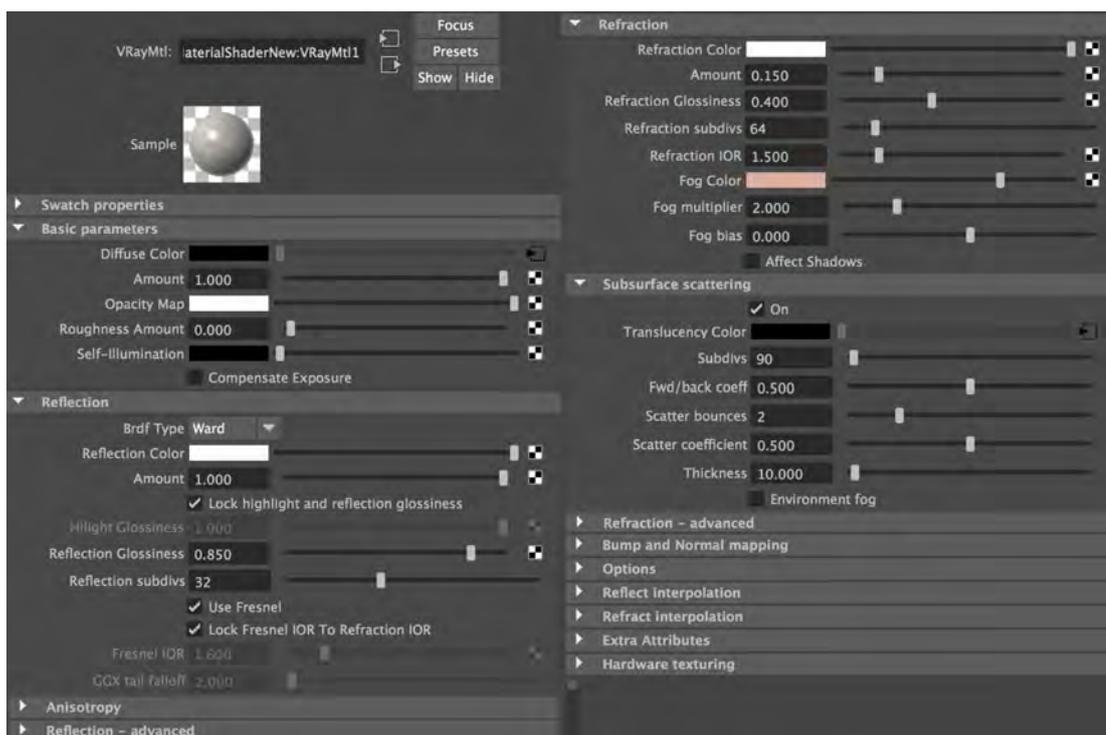
Als *Reflection Type* des Shaders wird *Ward* gewählt, da er weichere Reflektionen erzeugt, als *Phong* oder *Blinn* und somit der besonders weiche Effekt der Marmoroberfläche etwas besser visualisiert werden kann.

Die *Reflection Glossiness* wird auf einen Wert von 0,85 reduziert, was den Reflektionen etwas Unschärfe verleiht. Um dennoch kein Rauschen auf den Reflektionen zu haben, werden die *Reflection Subdivs* auf 32 erhöht.

Damit *Subsurface Scattering* berechnet werden kann, muss das Material einen gewissen Anteil an Transluzenz haben, dafür wird die *Refraction Color* auf weiß gestellt, was für vollständig transluzent steht, der *Amount* jedoch nur auf einen Wert von 0,15, was dafür sorgt, dass der Shader noch einen wesentlichen diffusen Anteil behält, damit die Textur richtig zur Geltung kommt und das Material nicht zu transluzent wirkt.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

Der *Refraction IOR* wird auf 1,5 gestellt, was dem Wert aus dem *Preset* des *FastSSS2*-Shaders für *marble* entspricht und auch in entsprechenden Listen als *Refraction Index* für Marmor nachgeschlagen werden kann. Für die *Fog Color* wird ein beige/rosa Farbton gewählt, was das *scatternde* Licht ein wenig tönt. Wobei bei der Farbauswahl, wie schon zuvor beschrieben, darauf geachtet werden muss, dass der Helligkeitswert der Farbe kleiner als 1 ist. Für den *Fog Multiplier* wird ein Wert von 2 gewählt.



**Abbildung 4.24:** Die gewählten Shader-Attribute des finalen Marmor-Shaders in der *V-Ray Material*-Variante.

Das *Subsurface Scattering* wird natürlich aktiviert und wie schon beschrieben wurde, erhält die *Translucency color* ebenfalls die Textur als Farbwert in den Kanal gelegt. Auch hier muss ebenfalls darauf geachtet werden, dass der *linear Workflow* eingehalten und die Textur gammakorrigiert wird. Da auch im *V-Ray Material* das *Scattering* schnell zu Rauschen im Material führt, muss dieses durch eine Erhöhung der *Subdivs* ausgeglichen werden, in diesem Fall reicht eine Erhöhung auf einen Wert von 90 für ein rauschfreies Bild. Der *Forward backward Coefficient* erhält einen Wert von 0,5, um zu gewährleisten, dass *Scattering* in beide Richtungen berechnet wird, sowohl weiter in die Richtung, in der der Lichtstrahl gestartet ist, als auch in die entgegengesetzte Richtung.

### 4.3. Exemplarische Implementierung einer Marmordrachen-Szene mit Subsurface Scattering

Mit einer Erhöhung der *Scatter Bounces* auf einen Wert von 2 wird der Lichtstrahl unterhalb der Oberfläche des Objektes öfter reflektiert und damit der *Scattering-Effekt* verstärkt. Dadurch erhöht sich auch das Rauschen des Shaders, was durch die bereits beschriebene Erhöhung der *Subdivs* aber mit ausgeglichen wird. Der *Scatter Coefficient* 0,5 beeinflusst, wie die Lichtstrahlen innerhalb des Objektes abgelenkt werden. Eine *Thickness* von 10 sorgt dafür, dass das *Scattering* für das komplette Objekt berechnet wird.



**Abbildung 4.25:** Das finale Ergebnis des Marmor-Shaders in der *V-Ray Material*-Variante.

#### 4.3.4 Vergleich

##### Qualität der Ergebnisse

Bei der Betrachtung der beiden jeweiligen Ergebnisse, ist auch bei dieser exemplarischen Umsetzung der subjektive Eindruck, dass die Qualität der Realisierung mit dem *FastSSS2*-Shader eine deutlich höhere ist. Beim Vergleich der Renderings mit realen Fotos kommt diese Variante näher an den *Look* und das *Feeling* von echtem Marmor heran als die Variante, die mit dem *V-Ray Material* umgesetzt wurde. Insgesamt ist der Eindruck der weichen Oberfläche, die das *Subsurface Scattering* ausmacht, deutlich stärker bei der *FastSSS2*-Variante und auch der filigrane *Look* der Teile bei denen ein leichter Effekt des Durchleuchtens entsteht, wie beispielsweise an den Krallen und den dünnen Stellen des Kopfes, ließ sich hier

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---

deutlich schöner umsetzen. Die Marmor-Textur kommt in der *V-Ray Material*-Variante weniger zur Geltung, obwohl das *Scattering* schon leicht reduziert wurde, damit sie überhaupt zu erkennen ist.



**Abbildung 4.26:** Direkter Vergleich der Marmordrachen-Renderings beider Shader-Varianten. Links in der *V-Ray Material* Variante, rechts in der *FastSSS2*-Shader-Variante.

#### Handhabung

In dieser exemplarischen Umsetzung gibt es in der Handhabung der Shader deutliche Unterschiede. Dadurch, dass es für den *FastSSS2*-Shader ein vorgefertigtes *Preset* für Marmor gibt, mit dem sich schon sehr gute Ergebnisse erzielen lassen, sind nur einige Feinabstimmungen für das Endergebnis notwendig und so ist die Erstellung dieser Shader-Variante deutlich einfacher und schneller, als die des *V-Ray Materials*. In die Einstellungen des *V-Ray Materials* wurde ein erheblicher größer Zeitaufwand investiert, welcher dennoch nicht zu einem gleich guten Ergebnis geführt hat. Das Zusammenspiel der eigentlichen *Subsurface Scattering*-Attribute des Shaders mit der *Refraction* und den *Fog*-Attributen sowie der *Diffuse Color* sind deutlich schwerer zu handhaben und die richtigen Einstellungen zu finden ist deutlich komplexer, als beim *FastSSS2*-Shader. Auffällig ist, dass obwohl die Textur in zwei Kanäle gelegt wurde, sie deutlich weniger zur Geltung kommt, als bei der *FastSSS2*-Variante. Damit sie überhaupt richtig erkennbar wird, musste der *Scattering*-Effekt, wie oben beschrieben schon etwas reduziert werden. Fazit ist also, dass deutlich mehr analytische Schritte und Nachdenken notwendig waren, um halbwegs sinnvolle Einstellungen für das *V-Ray Material* zu erhalten, während die Handhabung des *FastSSS2*-Shaders deutlich intuitiver ist und somit sehr viel schneller zu einem guten Ergebnis geführt hat.

#### Renderzeiten

Die Renderzeiten der beiden finalen Shader-Varianten werden für die Gegenüberstellung jeweils an einem *Shader Template* und am finalen Modell mit jeweils unterschiedlichen Rendersettings in *High Quality*- und *Low Quality*-Einstellungen gemessen, um einen aussagekräftigen Vergleich zu gewährleisten.

### 4.3. Exemplarische Implementierung einer Marmordrachen-Szene mit Subsurface Scattering

Die Renderings für den Vergleich der Renderzeiten der Marmor-Shader werden alle auf einem Rechner mit 2,3 GHz Intel Core i7 mit 16GB 16QO MHz DDR3 und einer NVIDIA GeForce GTM 650M 1024MB durchgeführt. Die *Shader Templates* werden mit einer Auflösung von 640 mal 800 Pixeln gerendert. Die Finalen Renderings haben in beiden Varianten eine Auflösung von 1280 x 720 Pixeln.

Im ersten Durchgang, den *Low Quality*-Varianten, wird alles auf den minimalsten *Sampling*-Einstellungen gerendert. Für den zweiten Durchgang wird das *Sampling* jeweils entsprechend so erhöht, dass ein ausreichendes *Sampling* für ein möglichst rauschfreies Bild erzeugt wird.

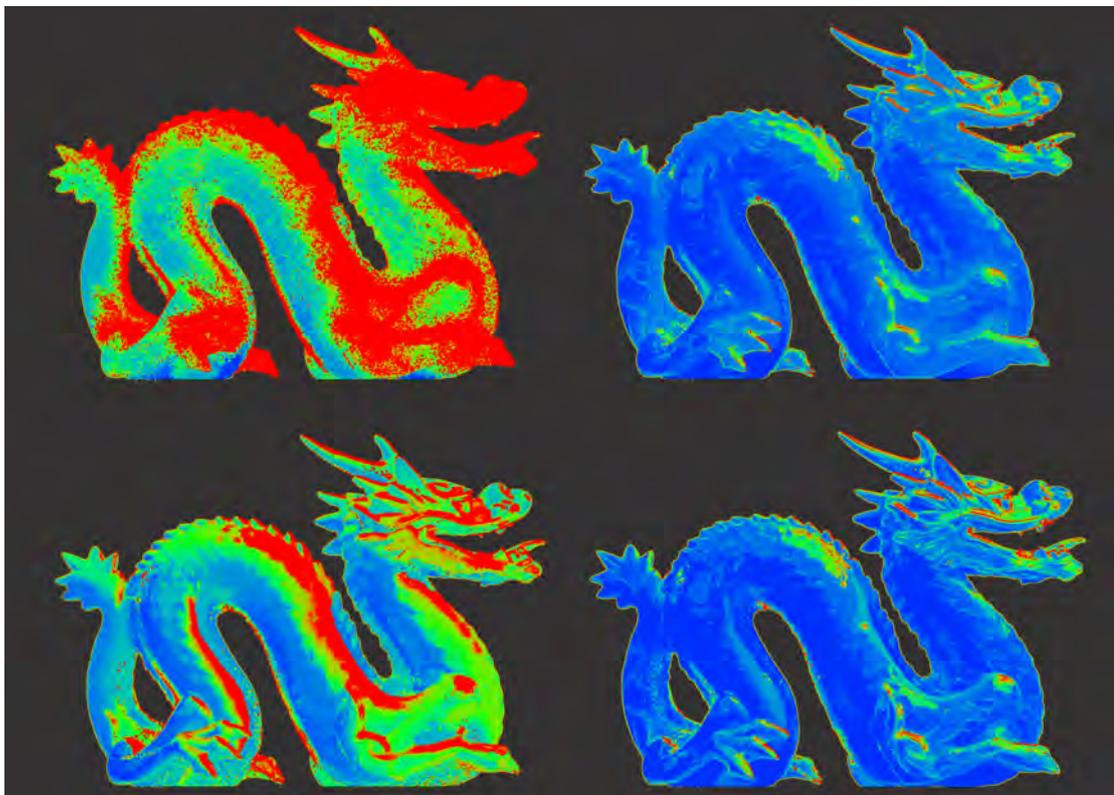
	FastSSS2-Shader	V-Ray Material
Shader-Template Low Quality	0h11m56,7s	0h08m5,3s
Shader-Template High Quality	1h34m36s	0h55m36,3s
Final Rendering Low Quality	0h08m45,8s	0h05m59s
Final Rendering High Quality	1h41m11,2s	0h20m11s

**Tabelle 4.2:** Vergleich der gemessenen Renderzeiten für die Marmor-Shader

Beim Vergleich der Renderzeiten wird schnell deutlich, dass besonders bei Shader-Einstellungen, die eine gute Qualität im Ergebnisbild gewährleisten, der *FastSSS2*-Shader deutlich höhere Renderzeiten benötigt als das *V-Ray Material*. Beim *Shader Template* in *High Quality* Einstellungen benötigt das Rendering mit dem *FastSSS2*-Shader etwa ein Drittel mehr Zeit als das *V-Ray Material*. Beim finalen Marmordrachen-Modell benötigt das Rendering in hoher Qualität mit dem *FastSSS2*-Shader etwa 5 Mal so lange wie mit dem *V-Ray Material*. Dies bestätigt, dass das *V-Ray Material* nicht nur in der ersten prototypischen Umsetzung, bei der auch der zusätzliche Durchleucht-Effekt eine große Rolle gespielt hat, bei den Renderzeiten vorne liegt, sondern auch bei der zweiten Umsetzung, bei der es hauptsächlich um die Realisierung der typischen Weichheit des *Subsurface Scatterings* geht, bessere Ergebnisse erzielt.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---



**Abbildung 4.27:** Gegenüberstellung der Sample Raten jeweils der Low Quality- und High Quality -Renderings der FastSSS2-Variante (oben) und der V-Ray Material-Variante (unten). In den *Sampling*-Raten-Visualisierungen steht rot für *undersampling* und blau für perfektes *sampling*. Es ist deutlich zu erkennen, dass die roten Flächen sich besonders an den Stellen häufen, an denen das *Scattering* stärker ist oder das Licht stark reflektiert wird. Daher werden spezifisch auch diese *Samples* in den Shadern erhöht und nicht das globale *Sampling* des Renderings. Um eine gute Vergleichbarkeit der Renderzeiten zu gewährleisten, wurde versucht, sehr ähnliche *Sample*-Raten für die *High Quality*-Varianten der beiden Renderings zu erhalten.

#### 4.4 Zusammenfassung und Ergebnis

Die Hypothese, dass der auf *Subsurface Scattering*-Umsetzungen spezialisierte *FastSSS2*-Shader, aufgrund seiner individuell auf diese Problematik angepassten Algorithmen, bessere Resultate für Anwendungen mit *Subsurface Scattering*-Effekten gegenüber dem *V-Ray Material* liefert, konnte belegt werden. Lediglich der Aspekt der kürzeren Renderzeiten wurde widerlegt.

Sowohl für die prototypische Umsetzung der Marmordrachen-Szene, bei der es um die Darstellung der typischen Weichheit, die *Subsurface Scattering* der Oberfläche verleiht, ging,

als auch bei der Kerzen-Szene, bei welcher zusätzlich der Effekt des Durchleuchtens hinzukam, waren die visuellen Ergebnisse, die die Umsetzungen mit dem *FastSSS2*-Shader geliefert haben, von deutlich besserer Qualität, als die, die aus den Realisierungen mit dem *V-Ray Material* hervorgegangen sind. Bei der Kerzen-Szene kommt hinzu, dass eine zusätzlich erforderliche Lichtquelle zur Erzeugung des Effektes des Durchleuchtens in der *V-Ray Material*-Variante für eine geringere physikalische Plausibilität der *FastSSS2*-Umsetzung gegenüber spricht.

Im Vergleich der Renderzeiten liegt das *V-Ray Material* mit 0,41 Std. zu 2,48 Std. für die Kerzenszene in *High Quality* und 0,20 Std. zu 1,41 Std. für die Marmordrachen-Szene in *High Quality* deutlich vorne. An dieser Stelle sollte noch einmal betont werden, dass der Name "*FastSSS2*" nicht so zu verstehen ist, dass das Material ein schnelleres *Subsurface Scattering* Rendering gegenüber dem herkömmlichen *V-Ray Material* gewährleistet. Der Name entstammt aus der Historie des Shaders selbst, der zu Beginn, seiner Implementation in den Renderer *V-Ray* den Namen *SSS Shader* hatte, später dann nach einer Überarbeitung, welche eine Beschleunigung der Renderzeiten bewirkte *FastSSS* hieß und nun nach einer weiteren Verbesserung den Namen *FastSSS2* trägt. Dennoch ließe sich vermuten, dass aufgrund der entsprechend implementierten Algorithmen für das Rendering des *Subsurface Scattering*-Effektes mit diesem Shader bessere Renderzeiten erzielbar wären. Dies ist nicht der Fall, was schlicht und einfach daran zu liegen scheint, dass beim *V-Ray Material* der *Subsurface Scattering*-Effekt nicht so stark ausgeprägt ist, wie bei den Resultaten, die mit dem *FastSSS2*-Shader erzielt wurden. Außerdem wird der Effekt beim *V-Ray Material* scheinbar nicht, wie beim *FastSSS2* über die Approximierte *BSSRDF* nach Jensen berechnet, sondern über die *Fog*-Einstellungen simuliert.

Welcher der beiden Shader zu empfehlen ist, ist sehr abhängig von der tatsächlichen Umsetzung, die realisiert werden soll. Steht das *Subsurface Scattering* als solches im Vordergrund, geht es beispielsweise um *Close Up*-Renderings dieses Effektes und wird bei der Realisierung Wert auf die physikalische Plausibilität und Realismus gelegt, so ist der *FastSSS2*-Shader mit Sicherheit die erste Wahl, da er in Bezug auf die Qualität deutlich bessere Ergebnisse liefert. Und auch die bessere Handhabung des Shaders spricht deutlich für die Verwendung des *FastSSS2*-Shaders.

Bei Szenen bei denen das Objekt mit dem *Subsurface Scattering*-Effekt jedoch eher im Hintergrund steht, muss abgewogen werden, ob die doch bemerkenswerte Einsparung der Renderzeit gegebenenfalls die Einbußen in der Qualität rechtfertigen. Besonders wenn es nicht mehr um Stilrenderings, sondern um Animationen geht, bei denen eine große Zahl an Einzelbilder gerendert werden muss, kann es sinnvoll sein, etwas mehr Zeit, die für den Shader-Aufbau beim *V-Ray Material* von Nöten ist, zu investieren, um dafür aber in den Renderzeiten später enorm Zeit einsparen zu können. Auch wenn eine Vielzahl von Objekten mit *Subsurface Scattering*-Effekten in der Szene platziert werden soll, kann ein Abwägen von Qualitätseinbußen gegen schnellere Renderzeiten in einigen Fällen zweckmäßig sein.

#### 4. ANALYSE DES V-RAY MATERIALS UND DES FASTSSS2-SHADERS FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING

---

In Kapitel 7 folgt mit der Durchführung einer Evaluation und dessen statistischer Auswertung der Versuch die subjektiven Parameter objektiv messbar zu machen und damit die Hypothese, dass der *FastSSS2*-Shader gegenüber dem Standard *V-Ray Material* bessere Resultate in Bezug auf die fotorealistische Visualisierung von *Subsurface Scattering*-Effekten liefert, statistisch zu belegen.

Das folgende Kapitel befasst sich mit der Fragestellung, ob der *FastSSS2*-Shader oder das neue *Skin Material* die bessere Wahl bei der Umsetzung von Renderings menschlicher Haut ist.

## Kapitel 5

# Analyse unterschiedlicher Methoden für die Visualisierung von Subsurface Scattering für das Rendering menschlicher Haut

### 5.1 Einleitung

Dieses zweite methodische Kapitel hat die Realisierung zweier Beispielszenarien für die Visualisierung von *Subsurface Scattering* bei menschlicher Haut für die Gegenüberstellung des *FastSSS2*-Shaders und des *Skin Materials* zum Inhalt. Wie schon im vorherigen Kapitel erläutert wurde, werden auch hier alle Umsetzungen mit der *Software Autodesk Maya 2016*<sup>1</sup> und dem *Renderer V-Ray 3.1*<sup>2</sup> der *Chaosgroup* implementiert.

In diesem Kapitel wird die Hypothese aufgestellt, dass für die Umsetzung von Renderings menschlicher Haut das neue *Skin Material*, welches von der *Chaosgroup* mit *V-Ray 3.1* für *Autodesk Maya* zur Verfügung gestellt wurde, durch seine spezielle, an das Rendering von Haut angepasste Aufbereitung, zwar eine Vereinfachung in der Handhabung gegenüber dem *FastSSS2*-Shader bietet, jedoch durch die Vereinfachung und die damit einhergehende Restriktion der Parameter auch schnell an seine Grenzen stößt. Dies ist insbesondere dann der Fall, wenn es um die spezielle Umsetzung der *Subsurface Scattering*-Effekte mit durchleuchtendem Licht geht. Ein Beispiel für derartige Umsetzungen in Bezug auf *Digital Humans* sind von hinten beleuchtete und dadurch rot leuchtende Ohren.

Um diese Hypothese zu belegen sollen das Szenario einer menschlichen Hand und eines menschlichen Gesichts jeweils mit dem *V-Ray FastSSS2*-Shader und dem speziell auf Hautrenderings ausgelegten *Skin Material* umgesetzt und sowohl die Handhabung der Shader, als

---

<sup>1</sup><http://www.autodesk.de/products/maya/overview> Stand: 10. Februar 2016

<sup>2</sup><http://docs.chaosgroup.com/category/vray> Stand: 10. Februar 2016

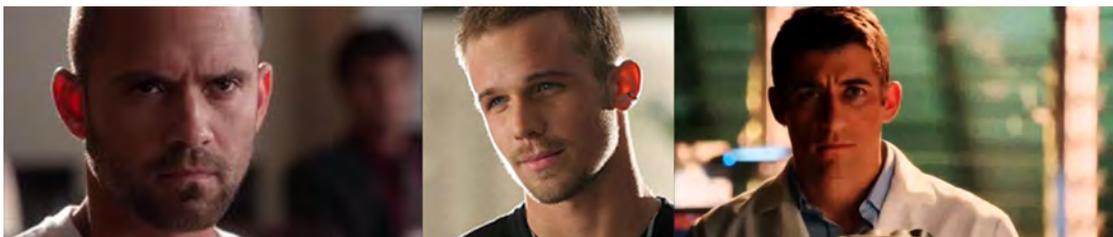
## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

auch die jeweiligen Ergebnisse sowie die Renderzeiten miteinander verglichen werden. Als zusätzlicher Versuch die subjektiven Parameter objektiv zu messen und bewertbar zu machen, folgt im Kapitel 7 die statistische Auswertung einer Evaluation, bezüglich der Frage, ob und wenn ja welcher der beiden Shader mit seinen Resultaten näher an den gewünschten Fotorealismus herankommt.

### 5.2 Implementierung einer Szene mit einem menschlichen Gesicht mit Subsurface Scattering

Das menschliche Gesicht ist und bleibt die Königsklasse des Renderings. Wie schon in Kapitel 2 erläutert wurde, wird der Mensch von Geburt an darauf trainiert selbst die kleinsten Feinheiten in den extrem detailreichen Gesichtern seiner Mitmenschen wahrzunehmen, sodass auch die feinsten Ungereimtheiten in der Umsetzung eines Abbildes sofort als Störung ins Auge fallen [BL03]. *Subsurface Scattering* spielt hier eine außerordentlich wichtige Rolle, da die Haut, die das Gesicht umhüllt transluzent und hochgradig *scatternd* ist. Das Rendering eines menschlichen Gesichts kombiniert nun die Schwierigkeit der Transluzenz mit einem enormen Detailreichtum, was die Komplexität zusätzlich erhöht, weil zum einen die Visualisierung der typischen Weichheit des *Subsurface Scatterings* gefordert ist, aber zum anderen die Vielzahl der Details, wie zum Beispiel die Poren der Haut, dadurch nicht verloren gehen dürfen. Hinzu kommt, dass die menschliche Haut aus unterschiedlichen Schichten besteht, die ein unterschiedliches *Scattering*-Verhalten aufweisen. Außerdem ist die Haut unterschiedlich dick an den unterschiedlichen Gesichtsteilen, was ebenfalls je nach Lichteinfall zu verschiedenartigem *Scattering*-Verhalten führt. So zum Beispiel an den Ohren, wenn es zu den bekannten "glühenden Ohren" kommt. Ein Effekt, wie er von Portrait-Fotografen gefürchtet und unter allen Umständen vermieden, aber in anderen Situationen auch als gewolltes Stil-Mittel bewusst eingesetzt wird, so beispielsweise auffällig häufig in der TV-Serie *CSI Miami* (siehe Abbildung 5.1).



**Abbildung 5.1:** Beispiele für den Einsatz glühender Ohren als gewolltes Stilmittel in den TV-Serien *One tree hill* (<http://goo.gl/ayxpKB> Stand: 10. Februar 2016), *OC California* (<http://goo.gl/NLKjFm> Stand: 10. Februar 2016) und *CSI Miami* (<https://goo.gl/aGhbKF> Stand: 10. Februar 2016).

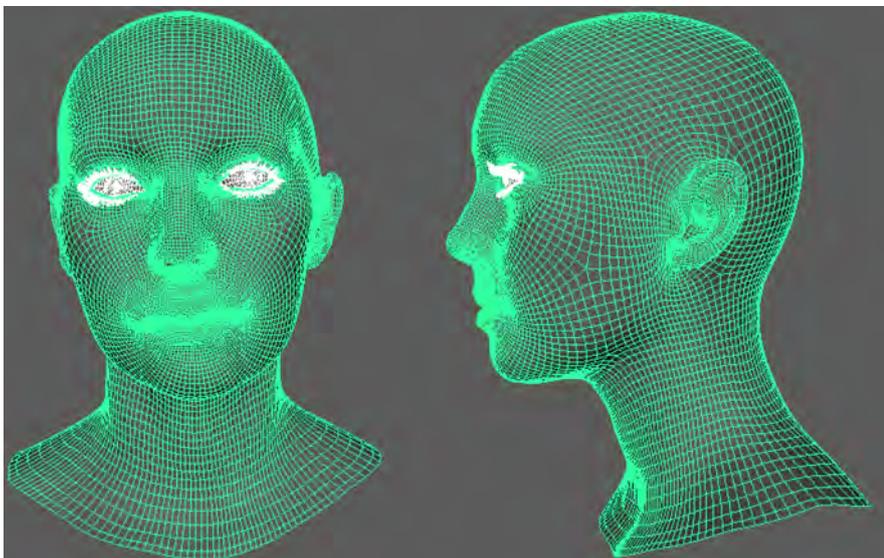
Die in diesem Kapitel umgesetzten prototypischen Implementierungen sollen zum einen den für Renderings menschlicher Haut enorm wichtigen typischen weichen Effekt der Oberfläche über das *Subsurface Scattering* realisieren, aber auch der etwas spezielleren zuvor beschrie-

## 5.2. Implementierung einer Szene mit einem menschlichen Gesicht mit Subsurface Scattering

den Effekt der rot “glühenden Ohren” umsetzen. Auf diese Weise werden die beiden Shader in der vollen Breite des für *Digital Humans* elementaren *Subsurface Scattering*-Umfanges analysiert.

### 5.2.1 Modell und Szenenaufbau

Für die prototypische Umsetzung wird das in Kapitel 2 bereits erwähnte Modell “Digital Emily 2.1” der Schauspielerin Emily O’ Brien verwendet. Welches von der *Digital Human League* über das *Open Source*-Projekt *Wikihuman*<sup>3</sup> zur freien Verfügung gestellt wird. Das Modell besteht aus einem *Mesh* mit 448110 Faces.



**Abbildung 5.2:** Das 3D-Modell des Kopfes der Schauspielerin Emily O’ Brien, welches im Rahmen des *Wikihuman* Projektes der *Digital Human League* an der *University of Southern California* erstellt wurde und das in dieser Arbeit für die prototypische Implementierung einer Szene mit *Subsurface Scattering* bei der Umsetzung eines menschlichen Kopfes verwendet wird.

Zusätzlich zu dem Modell der Emily 2.1 Version sind im Download einige Texturen enthalten. Eine Farbtextur, eine *Scattermap* und eine *Specularmap* bilden dabei die drei Haupttexturen (siehe Abbildung 5.3). Außerdem stehen zwei *Displacement Maps* für die Berechnung der Vertiefungen im Gesicht zur Verfügung. Eine generelle *Displacementmap* für die größeren Erhebungen, über diese werden auch die Kopfbehaarung, Augenbrauen berechnet und eine *Micro-Displacementmap* für die Feinheiten, wie beispielsweise feine Hautporen. Die Augen haben eine innere und eine äußere Farbtextur und jeweils eine entsprechend dazugehörige *Bumpmap*.

<sup>3</sup><http://gl.ict.usc.edu/Research/DigitalEmily2/> Stand: 10. Februar 2016

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---



**Abbildung 5.3:** Die farbgebende Textur, die *Scatter*-Textur und die *Specular*-Textur.



**Abbildung 5.4:** *Displacement*- und *Micro-Displacement*-Texturen

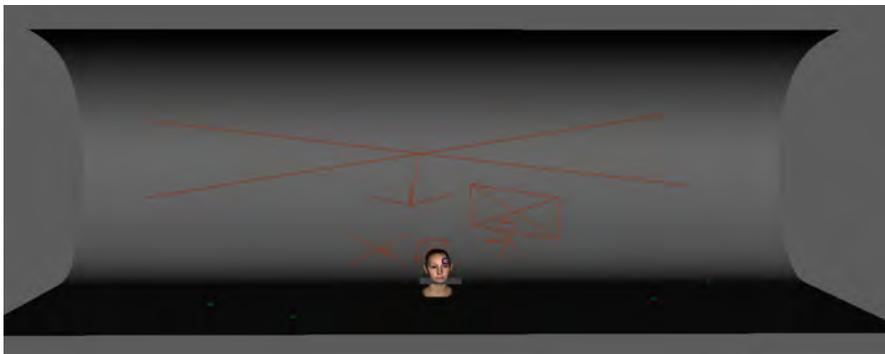


**Abbildung 5.5:** Innere und äußere Augentexturen mit jeweils dazugehörigen *Bump*-Texturen.

Für die Szene wird, um den Fokus ausschließlich auf das Gesichtsrendering legen zu können, ohne viel Aufwand in *Umgebungs-Modeling* oder ähnliches zu investieren, ein Studio-Setup gewählt. Das hat auch den Vorteil, dass die Renderzeiten, die gemessen werden, nicht von zusätzlichem Beiwerk der Szene beeinflusst werden. Für das Setup wird, wie auch zuvor schon bei den anderen prototypischen Implementierungen, eine Hohlkehle in neutralem diffusem grau verwendet, da diese nicht von dem Hauptobjekt ablenkt, nicht unnötig reflektiert und

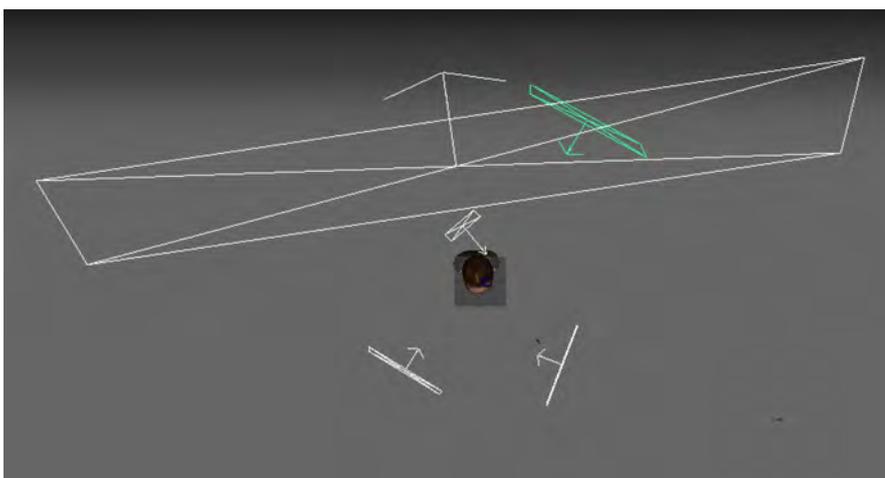
## 5.2. Implementierung einer Szene mit einem menschlichen Gesicht mit Subsurface Scattering

für eine eventuelle *Postproduktion* leicht in *Photoshop* auszutauschen ist. Das Modell des Kopfes wird mittig in diese Hohlkehle platziert. Das *Lightset* für die Ausleuchtung des Kopf-Modelles ist etwas aufwendiger gestaltet als die bisherigen *Lightrigs*. Wieder werden ausschließlich *V-Ray Rectangle Lights* für die Beleuchtung verwendet. Zwei Lichter strahlen dabei das Gesicht jeweils von links vorne mit 0,007 Watt und rechts vorne mit 0,01 Watt an, eine weitere Lichtquelle mit einer Intensität von 0,025 Watt beleuchtet den Kopf von rechts hinten. Ein großes *Rectangle Light* mit 0,03 Watt bestrahlt nur den Hintergrund, um hier unerwünschte Schatten und Farbverläufe zu verhindern. Ein fünftes Licht mit einer Intensität von 0,075 Watt beleuchtet das Ohr von links hinten, damit das Ohr genug Licht erhält, um den Effekt des Durchleuchtens hervorzurufen.



**Abbildung 5.6:** Szenenaufbau mit Hohlkehle, *Lightrig* und Modell.

Die Renderings werden mit einer *V-Ray Physical Camera* mit einer Blende von 2,8, einem ISO von 200 und einer *Shutterspeed* von 250 erzeugt.



**Abbildung 5.7:** Lichtset, welches zur Beleuchtung der Szene verwendet wird.

### 5.2.2 Der FastSS2-Shader

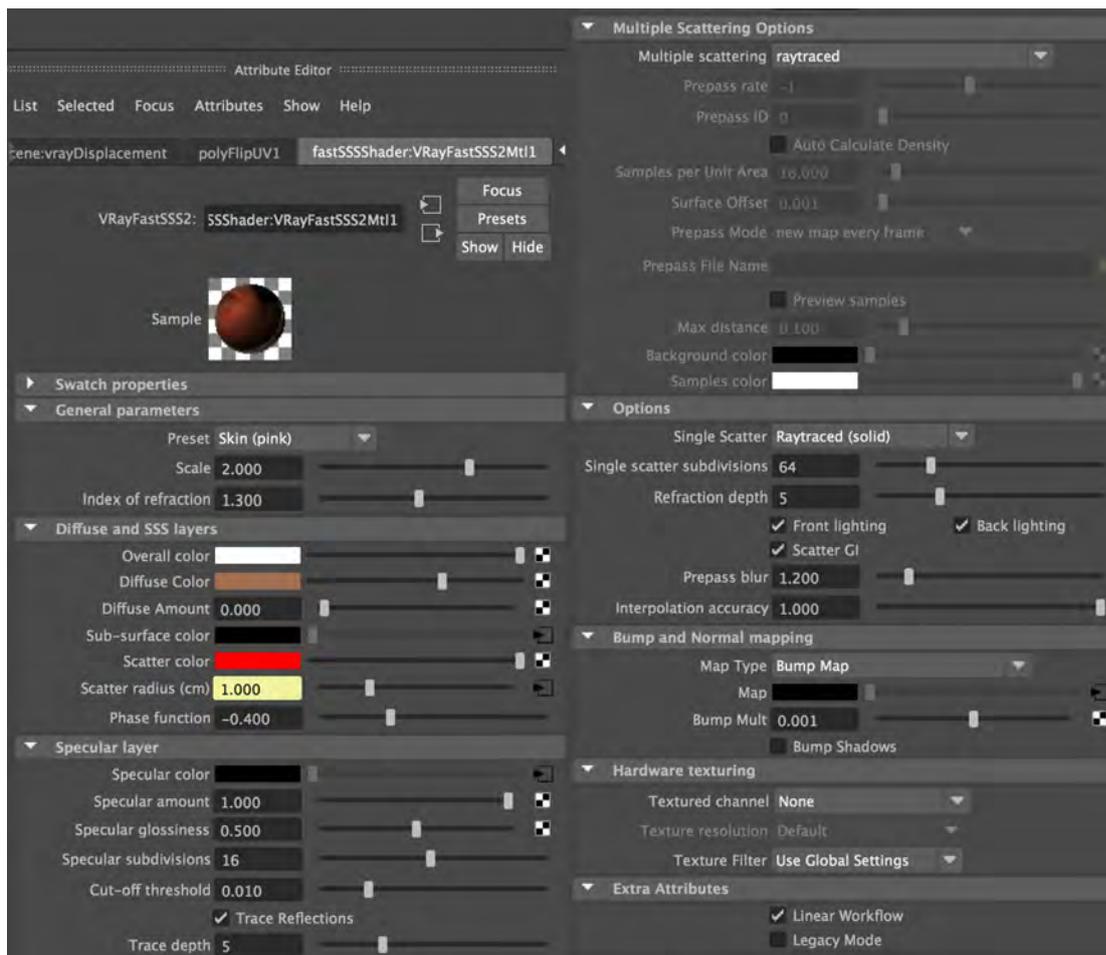


**Abbildung 5.8:** Der finale Kopf-Shader in der *FastSS2*-Variante auf dem *Shader Template*.

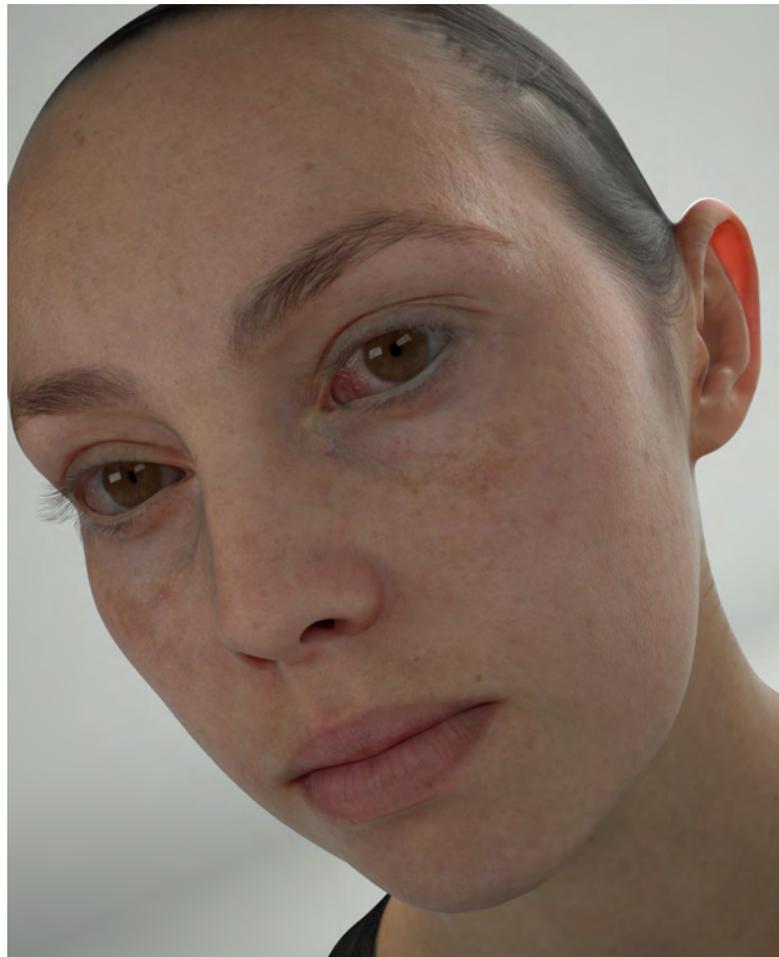
Für die *FastSS2*-Shader-Variante wird das *Preset* "Skin (Pink)" als Grundlage für den Aufbau des Materials verwendet. Um den *Scattering*-Effekt zu erhöhen wird der *Scale*-Parameter auf 2 heraufgesetzt. Die *Overall color* bleibt auf weiß, um die volle Helligkeit des Shaders zu gewährleisten. Der *Diffuse Amount* bleibt auf 0, so wird kein diffuser Anteil beigemischt. In den Kanal der *Sub-surface color* des Shaders wird die Farbtextur gelegt (siehe Abbildung 5.3). Für die Realisierung des Effektes, dass das Ohr rot leuchtet, wenn es von hinten angestrahlt wird, wird die *Scatter color* auf einen roten Farbton gesetzt. Für den *Scatter Radius* wird in dessen Kanal die *Scatter*-Textur eingelesen. Die *Phasefunction* wird auf -0,4 gestellt, einen recht kleinen negativen Wert, der gewährleistet, dass der Effekt des Durchleuchtens am Ohr seitens der Kamera gut sichtbar wird. In die *Specular color* wird die *Specular*-Textur gelegt. Die *Glossiness* wird auf 0,5 reduziert, damit die Glanzpunkte nicht zu scharf gezeichnet werden und damit die Haut etwas matter wirkt und nicht zu stark glänzt. Je nachdem welcher Effekt für die Haut gewünscht ist, könnte man hierüber regeln wie matt oder glänzend die Haut visualisiert wird. Damit an den Stellen, an denen das Licht stark reflektiert wird kein Rauschen im Bild entsteht, werden die *specular Subdivs* auf 16 erhöht. Das *Mutiple Scattering*-Verfahren wird, wie bei allen prototypischen Umsetzungen dieser Arbeit, auf *raytraced* gestellt, da dieses, wie bereits erörtert wurde, das einzige Verfahren ist, das physikalisch plausibel das *Scattering* berechnet, anstatt es, wie die andere Verfahren, anhand einer *Map* zu approximieren. Da Haut ein stark *scatterndes* aber nicht transparentes Organ ist, wird als *Single Scattering*-Methode der *Raytraced (solid)* Modus

## 5.2. Implementierung einer Szene mit einem menschlichen Gesicht mit Subsurface Scattering

gewählt. Um auch hier das Rauschen des Shaders in den Griff zu bekommen, werden die *Single scatter subdivisions* auf 64 erhöht. *Front* und *Back lighting* werden aktiviert, damit alle Lichter des *Lightrigs* in die Berechnung des *Subsurface Scatterings* involviert werden. *Scatter GI* wurde aktiviert, um auch bei dieser prototypischen Umsetzung wieder möglichst physikalisch plausibel zu arbeiten und die *Global Illumination* des *Scatterings* nicht über eine *Map* approximieren zu lassen, auch wenn dies die Renderzeiten etwas erhöht.



**Abbildung 5.9:** Die Attribute des finalen Haut-Shaders für das Rendering des Kopfes der Emily O'Brien in der *FastSS2*-Variante.



**Abbildung 5.10:** Finales Rendering des Emily O'Brien Modells in der *FastSSS2*-Variante.

### 5.2.3 Das Skin Material

Für das von der *Chaosgroup* seit der aktuellen Version *V-Ray 3.1* für *Maya* speziell für das Rendering menschlicher Haut zur Verfügung gestellte *Skin Material* stellt sich folgende Problematik.

Der Shader soll eine intuitive Handhabung und damit gegebenenfalls auch einen erleichterten Einstieg für *3D-Artists* in das Rendering von *Subsurface Scattering* in Bezug auf *Digital Humans* gewährleisten. Dies ist mit der Umsetzung des Shaders auch gelungen. Vor allem bei der Arbeit mit Texturen, was in Bezug auf Hautrenderings in der heutigen Zeit, durch die Verwendung der aktuellen *Scanning*-Technologien zur Erstellung der verschiedenen *Maps*, der häufigste Anwendungsfall ist, lässt sich relativ zügig eruieren in welche der Kanäle welche Textur eingelesen werden muss, um schnell gute Ergebnisse zu erzielen.



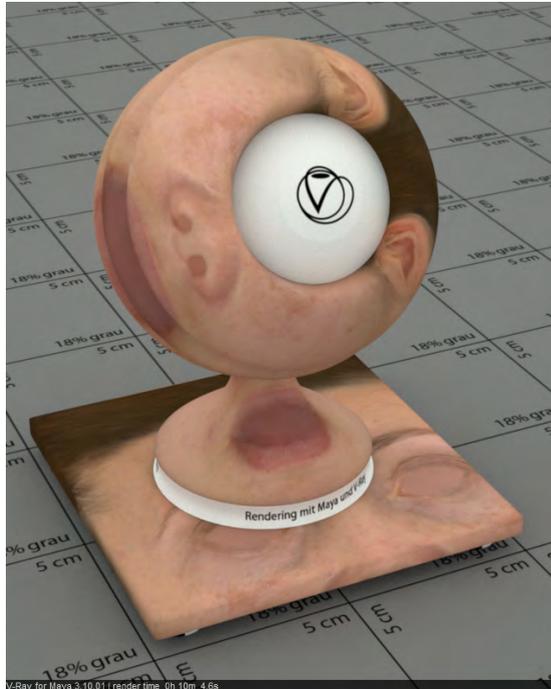
**Abbildung 5.11:** Nachweis für ein gutes Renderergebnis durch relativ schnellen Shader-aufbau mit Hilfe intuitiver Zuweisung der vorhandenen Texturen in die richtigen Kanäle. Allerdings fällt hier schon auf, dass der Effekt des Durchleuchtens am Ohr weißlich erscheint, statt der von Fotos bekannten und daher zu erwartenden orange, roten oder pinken Einfärbung.

Durch diese Vereinfachungen wurden allerdings auch starke Restriktionen in den Implementierungen der Parameter vorgenommen und besonders gegenüber dem bisher für *Subsurface Scattering*-Visualisierungen prädestinierten *FastSSS2*-Shader, haben sich dadurch Einschnitte in der Kontrollierbarkeit der Effekte ergeben. Attribute wie beispielsweise die *Phasefunction*, die *Scatter color* oder das *Front* und *Back lighting* wurden nicht mehr als für den Anwender beeinflussbare Parameter implementiert, vermutlich um die Parameterliste, die durch die drei *Scatter Layer* und die zwei *Reflection Layer* ohnehin schon sehr lang ist, noch übersichtlich zu halten. Besonders für die Umsetzung des Effektes des Durchleuchtens durch *Subsurface Scattering* scheint dies große Auswirkungen zu haben, da das *Skin Material* hier schnell an seine Grenzen zu stoßen scheint. Mit einfachen Bordmitteln des Shaders ist es nicht möglich diesen Effekt intuitiv richtig zu kontrollieren und so in adäquater Zeit ein realistisch wirkendes und zufriedenstellendes Ergebnis zu erzielen. Im Folgenden wird daher ein zur Lösung des beschriebenen Problems entwickelter Ansatz vorgestellt, welcher mit Hilfe einer speziell angefertigten Maske und einer Farbtextur, den Effekt mit dem *Skin Material* umsetzt und die höhere Kontrollierbarkeit für die Realisierung des *Subsurface Scattering*-Effektes des Durchleuchtens zu einem gewissen Maß wieder herstellt, um zufriedenstellende Ergebnisse zu erzielen.

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

Der Shader für die prototypische Implementation der *Skin Material*-Variante wird wie folgt aufgebaut.



**Abbildung 5.12:** Der Kopf-Shader in der *SkinMtl*-Variante auf dem *Shader Template*.

Die Farbtextur wird in den *Color*-Kanal der *Shallow Scattering*-Ebene gelegt. Der *Amount* dieses *Layers* wird auf 1 belassen, um eine gute Darstellung der Details aus der Textur, ohne Abschwächung oder Abdunkelung zu gewährleisten. In den *Radius*-Kanal der *Shallow Scattering*-Ebene wird die *Scattering*-Textur gelegt, so wird über diese *Map* bestimmt, an welchen Stellen das *Scattering* dieser Ebene welchen *Scattering Radius* unter der Oberfläche besitzt. Da die Textur durch das *Scattering* der drei *Layer* dennoch etwas ausblasst und zu viel des Detailreichtums verloren geht, wird die farbgebende Textur in dieser Variante zusätzlich in den *Color* Kanal der *Diffus color* gelegt und ein *Amount* von 0,15 eingestellt, um eine geringe Beimischung des diffusen Anteils sicherzustellen, der aber den *Scattering*-Anteil nicht zu sehr überlagert. Der *Medium Layer* wird für diese Implementation nur in einem kleinen Anteil verwendet, damit er in die Berechnung des Shaders minimal miteinfließt, die Textur aber nicht zu sehr verwaschen und die Haut allgemein nicht zu wachsig wirkt. Der *Amount* wird dafür auf 0,1 gesetzt. Da keine *Deep-Scattering*-Texturen mit Aderverläufen oder Ähnlichem für das Modell vorhanden sind, wird der *Deep Scattering Layer* in diesem Szenario ausschließlich für den Effekt der "glühenden" Ohren verwendet.

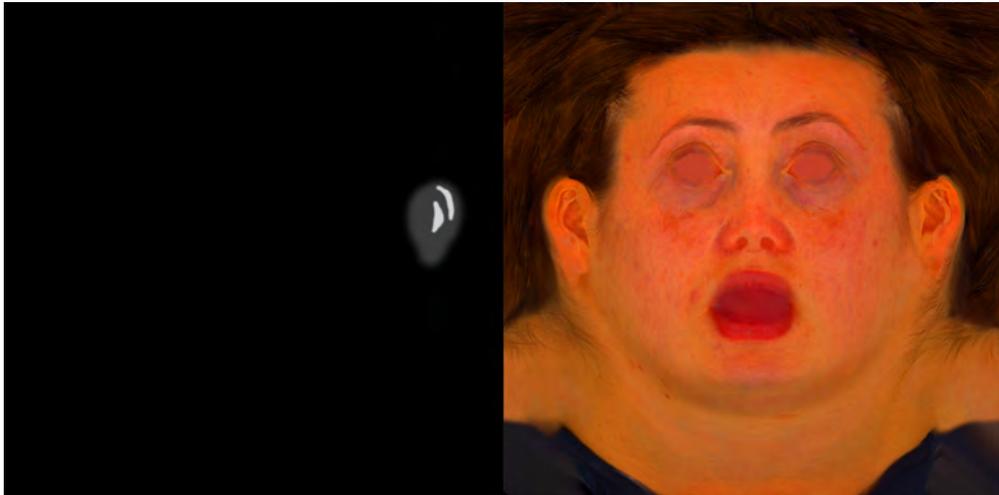


**Abbildung 5.13:** Nachweis für die komplette Rotfärbung des Gesichts, wenn der *Deep Layer* einen roten Farbton und einen erhöhten *Amount* zugewiesen bekommt. Der Effekt des Durchleuchtens selbst hingegen färbt sich nur geringfügig verstärkt rötlich ein.

Da eine einfache Farbzweisung des gewünschten Rottons für den *Color*-Kanal des *Layers* aber den gesamten Kopf rötlich einfärbt und sich dieser Effekt auch nicht über unterschiedliche Einstellungen für den Radius in gewünschter Weise kontrollieren lässt, muss an dieser Stelle über einen Texturansatz die notwendige Kontrolle herbeigeführt werden. Dafür wird in *Photoshop* eine entsprechende Maske angefertigt, welche in den *Amount*-Kanal des *Layers* eingelesen wird (siehe Abbildung 5.14). Die schwarzen Flächen der Maske setzen den *Amount* des *Layers* an diesen Stellen der Textur auf einen Wert von 0 und lassen somit keine Farbinformationen aus dem *Color*-Kanal und keine Berechnung des *Scatterings* dieses *Layers* an dieser Stelle zu. Die weißen Farbflächen entsprechen einem *Amount* von 1 was einer vollen Durchlässigkeit der Farbinformation und des *Scatterings* dieses *Layers* entspricht. Um nun auch für das rote Leuchten den gewünschten Effekt zu erzielen, wird die farbgebende Textur rot eingefärbt und in den *Color* Kanal des *Layers* gelegt. Durch die entsprechende Maske wird die Farbe nur an den erwünschten Stellen am Ohr durchgelassen. Für diesen Prototyp ist vorab bekannt, dass die Kameraperspektive so gewählt wird, dass nur das linke Ohr im Rendering sichtbar sein wird, daher wird die Maske in diesem Fall auch lediglich für das linke Ohr angefertigt.

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---



**Abbildung 5.14:** Speziell erstellte Maske (links) und rot eingefärbte Textur (rechts) für die Erstellung des rot leuchtenden Ohres in der *Skin Material*-Variante. Zu Testzwecken wurde die Maske hier lediglich für das linke Ohr erstellt, da die Kameraperspektive so gewählt wird, dass auch nur dieses im Rendering zu sehen ist.



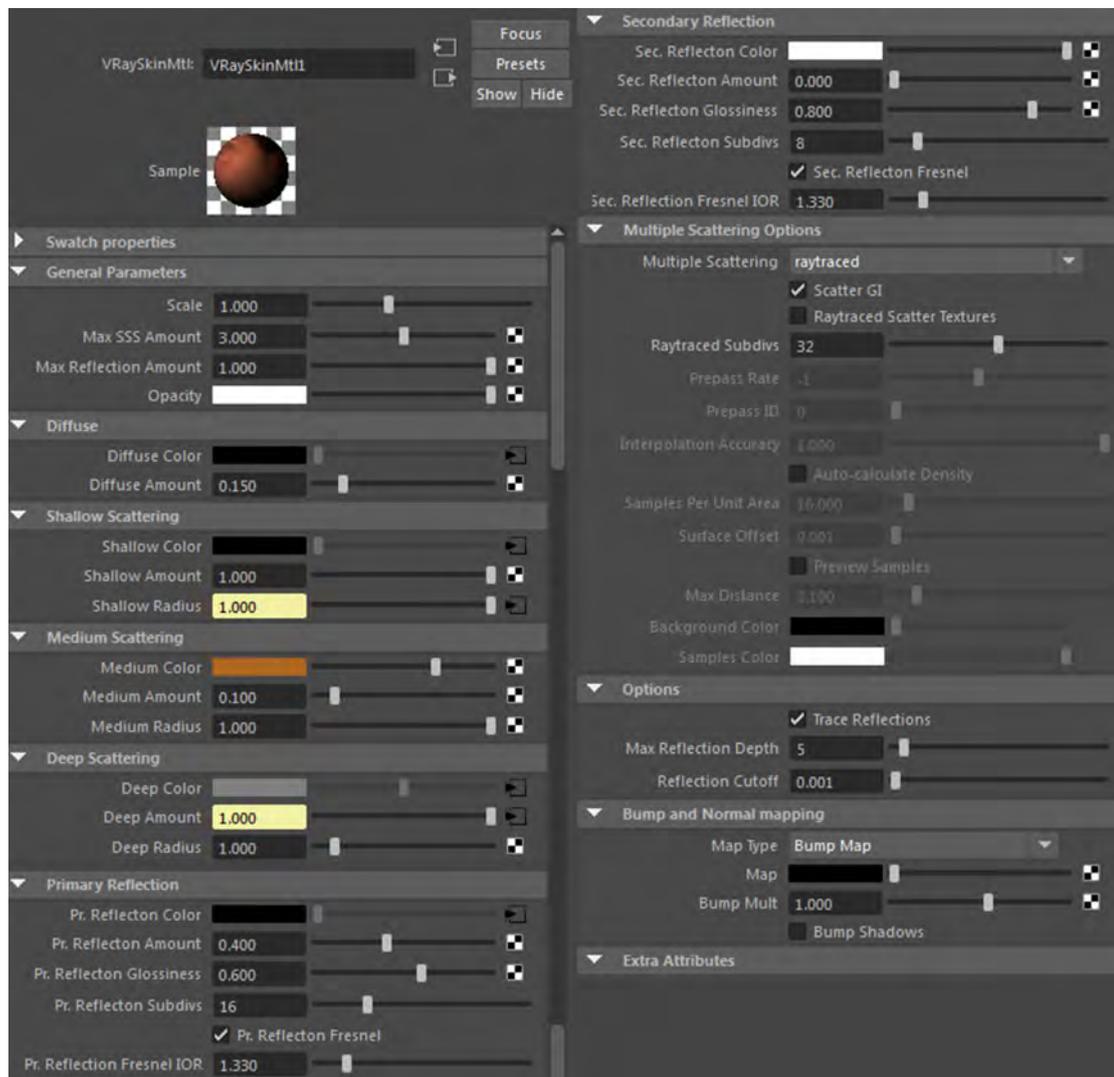
**Abbildung 5.15:** Die linke Hälfte des Gesichtes zeigt, wie sich das *Raytraced Scatter Texture*-Verfahren auf das Rendering-Ergebnis auswirkt. Die Textur wird deutlich weicher, aber auch sehr viele Details gehen verloren.

## 5.2. Implementierung einer Szene mit einem menschlichen Gesicht mit Subsurface Scattering

---

Die Textur mit der *Specularmap* wird in den Farbkanal der *Primary Reflection* gelegt. Die *Glossiness* wird auf 0,6 reduziert, um den Glanz etwas herauszunehmen und die Reflektionen etwas diffuser und weniger scharf zu zeichnen. Um Rauschen in den Reflektionen zu reduzieren werden die *Reflection Subdivs* auf 16 erhöht. Auch im *Vray Skin Material* lässt sich über die Option *Scatter GI* die physikalische Berechnung für die *Global Illumination* aktivieren. Da auch bei dieser Umsetzung ein möglichst physikalisch plausibles Ergebnis erzielt werden soll, wird diese Option verwendet, auch wenn sie etwas längere Renderzeiten zur Folge haben kann. Auf die Option der *Raytraced Scatter Textures*, welche bestimmt, ob die Texturen, welche in die Scattering Layer gelegt werden, in die Berechnung des Scatterings miteinbezogen werden, wird verzichtet. Diese Option kann zu akkurateren Ergebnissen führen, wenn dafür deutlich höhere Renderzeiten in Kauf genommen werden. Verzichtet wird auf die Verwendung dieser Berechnung in diesem Fall jedoch nicht zugunsten der Renderzeiten, sondern weil durch das zusätzliche *Scattering* zu viel an Detailreichtum der Textur verloren geht (siehe Abbildung 5.15). Da durch das *Scattering* auch im *V-Ray Skin Material* Rauschen im Shader entsteht, werden auch hier die *Raytraced Subdivisions* erhöht. Ein Wert von 32 ist in diesem Fall ausreichend, um ein möglichst rauschfreies Ergebnis zu erzeugen.

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT



**Abbildung 5.16:** Die Attribute des finalen Haut-Shaders für das Rendering des Kopfes der Emily O'Brien in der *Skin Material*-Variante.



**Abbildung 5.17:** Finales Rendering des Emily O'Brien Modells in der *Skin Material*-Variante.

#### 5.2.4 Vergleich

Nach der Umsetzung der Emily-Szene in beiden Shader-Varianten folgt nun im folgenden Abschnitt der Vergleich der beiden Shader unter den Aspekten der "Qualität der Ergebnisse", "Renderzeiten" und "Handhabung".

Nachdem beide prototypische Umsetzungen dieses Kapitels in den unterschiedlichen Kategorien differenziert bewertet sind, wird in Abschnitt 5.4 ein generelles Resümee gezogen, bei welchem eine generelle Beurteilung der Shader auf Grundlage der Ergebnisse sowie eine Klassifizierung für welche Szenarien, welcher Shader der geeignere ist, erfolgt.

##### Qualität der Ergebnisse

Beide Shader liefern auf den ersten Blick gute Ergebnisse. Legt man die Renderings für einen direkten Vergleich nebeneinander wird schnell deutlich, dass die Resultate des *FastSSS*-Shaders gegen die des *V-Ray Skin Materials* dennoch eine Klasse besser sind. Die Hauttextur

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

an sich behält einen Hauch mehr an Detailreichtum gegenüber dem *Skin Material*, aber in Bezug auf die Weichheit der Haut gibt es im Generellen wenig Unterschiede und durch die drei *Layer* im *Skin Material* ließen sich auch hier dem jeweiligen Geschmack entsprechend noch eine Reihe von unterschiedlichen Einstellungen finden.

Große Unterschiede hingegen gibt es bei der Umsetzung des Durchleuchtens am Ohr des Emily-Models. Hier kann das *Skin Material* mit dem *FastSSS2*-Shader scheinbar nicht mithalten. Das Leuchten ist in der *FastSSS2*-Variante deutlich stärker ausgeprägt und auch der Farbton in einem viel satteren Rot. Im Vergleich mit einem realen Foto wirkt das leuchtende Ohr mit dem *FastSSS2*-Shader deutlich fotorealistischer. Für das *Skin Material* konnte hier mit dem vorgestellten, zur Lösung des Problems entwickelten, Texturansatz Abhilfe geschaffen werden, sodass auch in Bezug auf den Effekt des Durchleuchtens nur noch geringe Qualitätsunterschiede wahrzunehmen sind. Jedoch kommt es dann bei Verwendung der Texturmethode für das *Skin Material* sehr auf die Qualität der Maske an, die gewährleisten muss, dass der Leuchteffekt an den richtigen Stellen auftritt und keine harten unnatürlichen Kanten, sondern weiche Verläufe entstehen. Da bei der *FastSSS2*-Shader-Variante der Effekt allein durch die Lichtquelle und die Berechnung des Shaders erzeugt wird, sind die Verläufe von sich aus natürlicher und physikalisch plausibler.



**Abbildung 5.18:** Direkter Vergleich der Emily-Renderings in der *FastSSS*-Variante und in der *Skin Material*-Variante sowohl ohne als auch mit der Verwendung der texturbasierten Lösung.

Wichtig zu erwähnen ist an dieser Stelle auch die Tatsache, dass um wirklich fotorealistische Ergebnisse erzielen zu können die Verwendung von Texturen unumgänglich ist. Dies bringt für die an sich physikalisch plausible Grundidee des *Skin Materials* einen großen Nachteil mit sich. Denn die für das *Skin Material* benötigten Texturen lassen sich nicht erzeugen. Lediglich für den obersten der drei *Layer*, den *Shallow Scattering Layer*, können mit Hilfe von Phasenkameras und Oberflächenfotografie Texturen erstellt werden. Für den *Medium*- und den *Deep Layer* ist dies nicht möglich.



**Abbildung 5.19:** Vergleich einiger Handyschnappschüsse und eines Ausschnittes einer Konzertaufnahme (rechts) bei denen der *Subsurface Scattering*-Effekt des Durchleuchtens bei den Ohren sehr ausgeprägt zu sehen ist (oben), mit Renderings (unten). Bei den Renderings handelt es sich um zwei Varianten mit dem *FastSSS2*-Shader (links) sowie eine Variante mit dem *Skin Material* ohne Texturlösung (zweites von rechts) und eine Variante mit *Skin Material*, bei der der entwickelte Texturansatz angewendet wurde (rechts).

### Handhabung der Shader

Über die Handhabung des *FastSSS2*-Shaders wurde schon im vorherigen Kapitel zu den beiden anderen prototypischen Umsetzungen berichtet, daher soll hier nur kurz spezifisch auf die Umsetzung der Emily-Szene eingegangen werden, ausführlicher wird anschließend die Handhabung des *Skin Materials* dargestellt. Die Umsetzung der Emily-Szene mit dem *FastSSS2*-Shader war, wie von der Marmor-Drachen Implementation bereits gewohnt, sehr intuitiv und schnell. Mit Kenntnis der Funktionsweise der umfangreichen Parameter des Shaders, erreicht man gezielt die gewünschten Effekte. Dabei bietet einem der Shader eine hohe Kontrollierbarkeit, um spezifisch und präzise Einfluss nehmen zu können. Die bereitgestellten *Presets* stellen außerdem besonders für den Einstieg eine große Hilfe dar. Nach kurzen Überlegungen welche Textur sinnvoll in welchen Kanal eingebunden werden muss, konnte der Shader logisch aufgebaut werden. Lediglich in Bezug auf die *Scattering*-Textur waren die Überlegungen ein wenig aufwendiger, da diese nicht wie zuerst angenommen in

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

die *Scatter color*, sondern in den *Scatter Radius*-Kanal gelegt werden muss, damit über die *Scatter color* noch eine Farbwahl für den Effekt des Durchleuchtens der Ohren erfolgen kann.

Die Idee hinter dem Aufbau des *V-Ray Skin Materials* die unterschiedlichen Schichten der menschlichen Haut nachzubilden und dafür verschiedene *Layer* bereitzustellen und dies innerhalb eines Shaders und nicht über ein *Blendmaterial* oder ähnliches zu lösen ist innovativ und wirkt zunächst auch intuitiv in der Handhabung, was in besonderem Maße für den Shader auch aufgrund seiner reduzierten Parameterzahl gegenüber dem *FastSS2*-Shader gilt. Allerdings ist aufgrund der schlechten Dokumentation des Shaders und der Attribut-, die Erlangung des Verständnisses wie genau diese drei, beziehungsweise fünf Schichten, die zwei zusätzlichen Reflektionsschichten berücksichtigt, miteinander verrechnet werden und wie sie sich gegenseitig beeinflussen nicht trivial. Durch die starke Restriktion der Attribute verliert der *3D-Artist* leider auch viel Kontrolle über die Resultate der Effekte, besonders wenn es um den Effekt des Durchleuchtens durch *Subsurface Scattering* geht. Der vorgestellte Texturansatz stellt einen wichtigen Anteil dieser verlorengegangenen Kontrollierbarkeit wieder her. Ein großer Nachteil ist jedoch, dass es viel Zeit und Aufwand kostet die entsprechend notwendigen Masken und Farbtexturen nicht nur anzufertigen, sondern sie vor allem in zahlreichen Iterationen zwischen Bildbearbeitungs- und der 3D-Software, bis hin zu einem gewünschten Resultat nachzubessern und entsprechend anzupassen. Da der benötigte Farbwert für den *Deep Layer* von vielen unterschiedlichen Faktoren, wie beispielsweise der Beleuchtung der Szene oder dem generellen Hautton des Shaders abhängt, und weder berechnet werden kann, noch entsprechende Texturen dafür fotografiert werden können, muss hier solange wirklich "ausprobiert" werden, bis der richtige Farbton gefunden ist. Dies ist ohne Texturmethode innerhalb der Software mit dem *Color Picker* natürlich deutlich schneller, als wenn für jede Anpassung zunächst erst wieder in die Bildbearbeitungssoftware zurück-gewechselt werden muss. Und auch die genaue Erstellung der Maske, mit der der Bereich festgelegt wird, an dem die Berechnungen für das *Scattering* des *Deep Layers* zugelassen werden, ist aufwendig. Hier kann als Hilfestellung eine Auswahl im *UV-Snapshot* helfen, um eine Vorauswahl des richtigen Bereichs zu treffen, die in *Photoshop* weiterverarbeitet werden kann. Dennoch muss hier sehr genau gearbeitet werden und gegebenenfalls müssen Feinheiten in mehreren Iterationen nachgearbeitet werden.

### Renderzeiten

Die Renderzeiten der beiden finalen Shadervarianten wurden für die Gegenüberstellung jeweils an einem *Shader Template* und am finalen Modell mit jeweils unterschiedlichen Rendersettings in *High Quality*- und *Low Quality*-Einstellungen gemessen, um einen aussagekräftigen Vergleich zu gewährleisten.

Die Renderings für den Vergleich der Renderzeiten wurden alle auf einem Rechner mit 3,47 GHz (2 Prozessoren) Intel Xenon X5690 mit 96GB Arbeitsspeicher und zwei NVIDIA Tesla C2070 und einer NVIDIA Quadro 6000 Grafikkarten durchgeführt. Die *Shader Templates* wurden mit einer Auflösung von 640 mal 800 Pixeln gerendert. Die finalen Renderings haben in beiden Varianten eine Auflösung von 960 x 1200 Pixeln.

### 5.3. Implementierung einer Szene mit einer menschlichen Hand mit Subsurface Scattering

	FastSSS2-Shader	Skin Material
Shader-Template Low Quality	0h04m14,3s	0h05m5,5s
Shader-Template High Quality	0h48m29,2s	0h10m4,6s
Final Rendering Low Quality	0h57m17,5s	0h21m35,1s
Final Rendering High Quality	2h08m27,6s	0h38m27,0s

**Tabelle 5.1:** Vergleich der gemessenen Renderzeiten für die Kopf-Shader

Die Renderzeiten zeigen, mit einer kleinen Ausnahme bei der Testreihe des *Shader Templates* in der *Low Quality*-Variante, dass das *Skin Material* deutlich kürzere Zeiten verzeichnen kann, als der *FastSSS2*-Shader. Dabei braucht das Rendering in der *FastSSS2*-Variante immer mindestens doppelt so lange, in den *High Quality*-Varianten sogar fast viermal so lange wie das *Skin Material*. Bezüglich der Renderzeiten schneidet das *Skin Material* also deutlich besser ab, als der *FastSSS2*-Shader.



**Abbildung 5.20:** Gegenüberstellung der Sample Raten der Kopf-Renderings des Emily-Models in der *Skin Material*-Variante (die zwei linken Bilder) und in der *FastSSS2*-Shader-Variante (die zwei rechten Bilder), jeweils in der *Low Quality*-Variante (jeweils das linke der beiden Bilder) und *High Quality*-Variante (jeweils das rechte der beiden Bilder). In der *Samplerate*-Visualisierung steht die Farbe Rot für ein zu niedrig gesampeltes Pixel und Blau für ein perfekt gesampeltes Pixel.

### 5.3 Implementierung einer Szene mit einer menschlichen Hand mit Subsurface Scattering

Zur Überprüfung der aus der ersten prototypischen Umsetzung in Bezug auf die Realisierung menschlicher Haut mit den beiden Shadern, dem *FastSSS2*-Shader und dem *Skin Material*, gewonnenen Erkenntnisse, insbesondere in Bezug auf das *Skin Material*, wurde eine

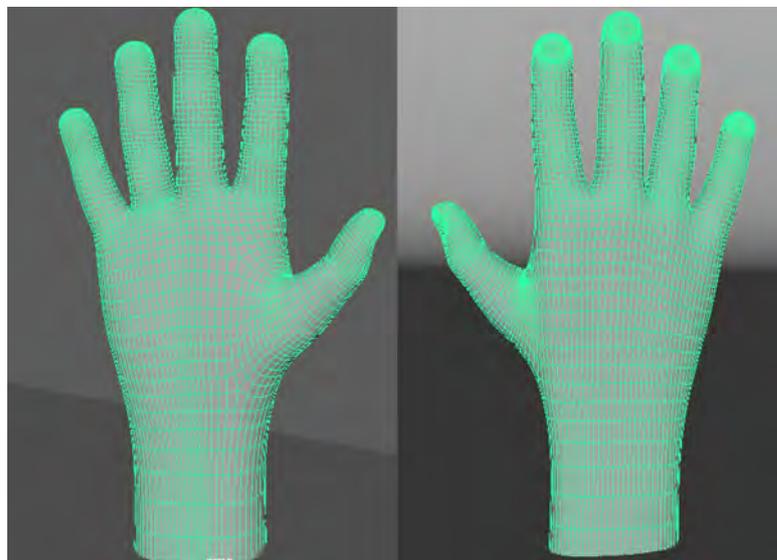
## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

zweite Implementierung mit ebenfalls kombinierten Effekten der generellen Weichheit der Haut mit einem rot leuchtenden Effekt realisiert. Neben den "leuchtenden" Ohren tritt ein typischer Effekt des Durchleuchtens häufig bei Händen auf, wenn entweder ein Finger vor eine Lampe gehalten wird, oft ist dies aktuell bei den modernen Pulsmess-Applikationen in Smartphones zu beobachten, bei denen durch Auflegen des Fingers auf das Kameralicht des Handys der Puls gemessen wird. Oder auch die ganze Hand die angestrahlt wird und in den Fingerzwischenräumen das rote Leuchten aufweist.

### 5.3.1 Modell und Szenenaufbau

Das Modell, das für diese Szene verwendet wird, wird von Dennis Haupt über Sketchfab<sup>4</sup> zur Verfügung gestellt. Dabei handelt es sich eigentlich, um ein geriggtes Doppelhand-Modell für Blender. Für die Zwecke dieser prototypischen Umsetzung wird eine der beiden Hände entfernt, da lediglich eine Hand gerendert werden soll. Auch das *Rig*, das leider in *Autodesk Maya* nicht zu verwenden ist, wird aus der Szene entfernt. Auf das erneute *Riggen* der Hand, wird an dieser Stelle verzichtet da diese Szene lediglich der Überprüfung der Ergebnisse der Emily-Szene dienen soll und der Zeitaufwand daher unverhältnismäßig wäre. An dieser Stelle sei aber erwähnt, dass durch eine realistischere Fingerstellung, die mit Hilfe eines *Rigs* erreicht würde, ein höherer Grad an Realismus erzielt werden könnte. Die Einzelne Hand besteht aus 8288 Faces. Das Modell enthält drei Texturen (siehe Abbildung 5.22), eine farbgebende diffuse Textur, eine *Normal-Map* für das *Displacement* oder *Bump* und eine *Specular-Map* für die Reflektionen.

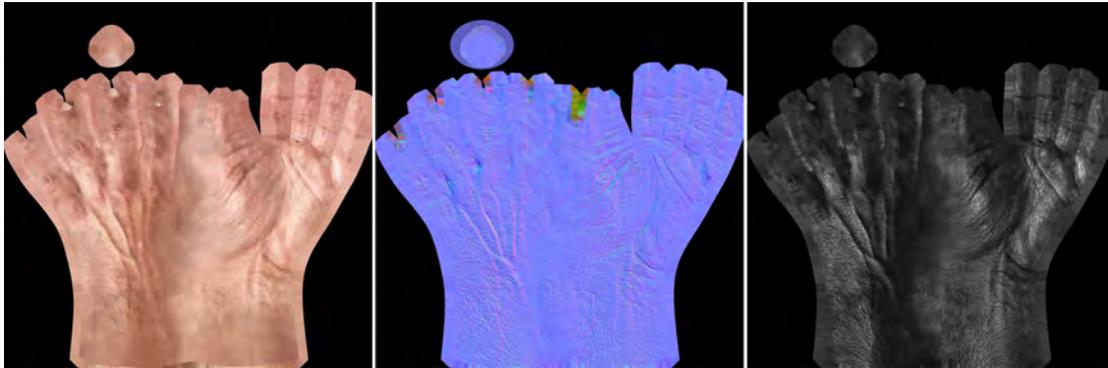


**Abbildung 5.21:** Das Modell der Hand.

---

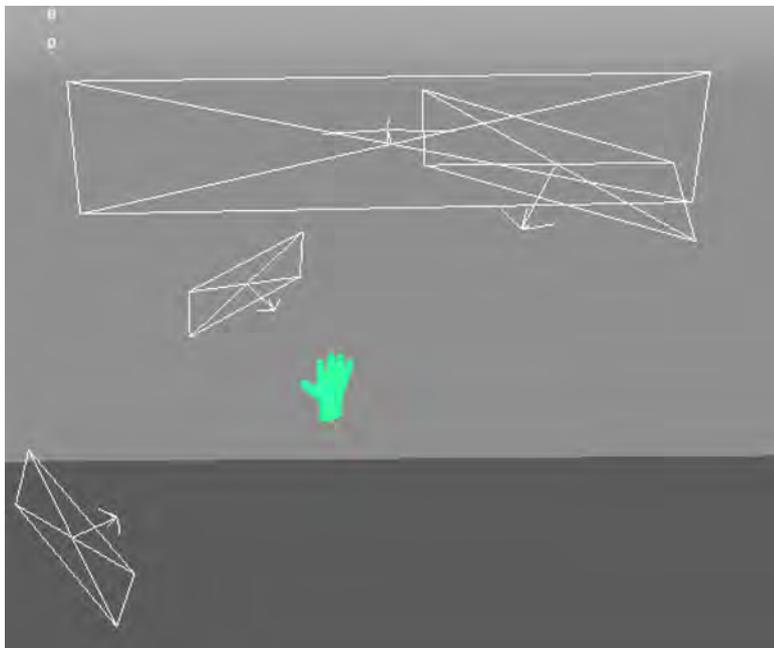
<sup>4</sup><https://sketchfab.com/models/86dff766e51d4f2084a3768be7f08b8c> Stand: 27. Februar 2016

### 5.3. Implementierung einer Szene mit einer menschlichen Hand mit Subsurface Scattering



**Abbildung 5.22:** Die drei Texturen für den Handshader. Von rechts nach links sind die diffuse Farbtextur, die *Normalmap* und die *Specularmap* zu sehen.

Das Modell wird in eine Hohlkehle gestellt, welche mit einem difusen grauen Shader geshadet wird. Die Grundbeleuchtung besteht aus zwei *Rectangle Lights* die das Handmodell von links vorne mit einer Intensität von 0.003 Watt und rechts hinten mit einer Intensität von 0.020 anstrahlen. Außerdem wird eine Lichtquelle in Form eines *Rectangle Lights* hinter der Hand so platziert, dass sie den *Subsurface Scattering*-Effekt des Durchleuchtens erzeugt. Dieses Licht hat eine Intensität von 0.030 Watt. Eine vierte Lichtquelle leuchtet mit ebenfalls 0.030 Watt den Hintergrund aus, um hier ungewollte Schattenbildung zu vermeiden.

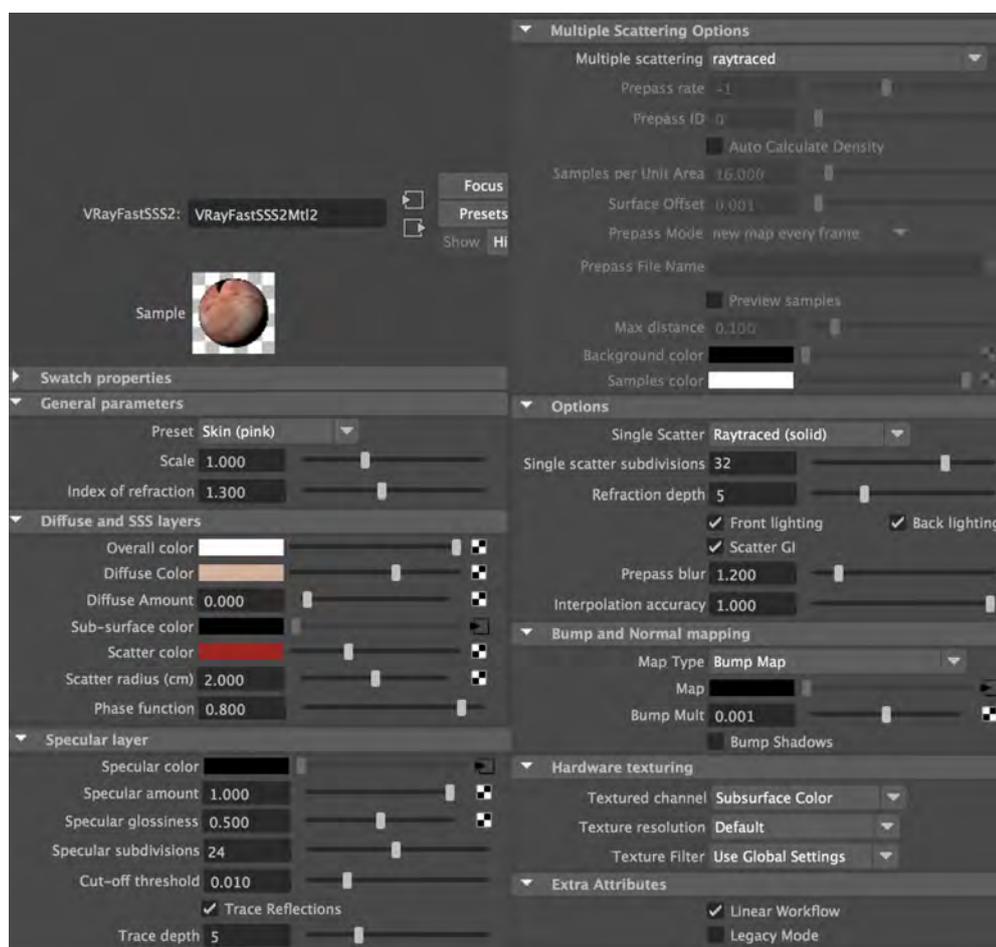


**Abbildung 5.23:** Szenenaufbau und Beleuchtung der Handszene.

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

### 5.3.2 Der FastSSS2-Shader

Für die Umsetzung mit der *FastSSS2*-Shader-Variante wird das *Skin (pink)*-Preset als Grundlage zum Shaderaufbau verwendet. In die *Sub-surface color* wird die farbgebende Textur gelegt. Für die *Scatter color* wird ein roter Farbton gewählt. Dieser soll später dem Effekt des Durchleuchtens seine Farbe geben. Die *Reflection-Map* wird in den Kanal der *Specular color* gelegt und die *Glossiness* auf 0,5 reduziert, um einen etwas matteren und nicht zu stark glänzenden Effekt zu erhalten. Die *Specular subdivisions* werden auf 24 erhöht, um das Rauschen in den Reflektionen möglichst gering zu halten. Für einen rauschfreien Shader besonders an den stark *scatternden* Bereichen des Modells ist eine Erhöhung der *Single scatter subdivisions* auf 32 notwendig. Die *Normal-Map* wird in den *Bump and Normal mapping Map*-Knoten gelegt und ein *Bump mult* von 0,001 eingestellt, um einen subtilen Effekt zu gewährleisten.



**Abbildung 5.24:** Einstellungen der Attribute für den Shader der Hand in der *FastSSS2*-Variante.



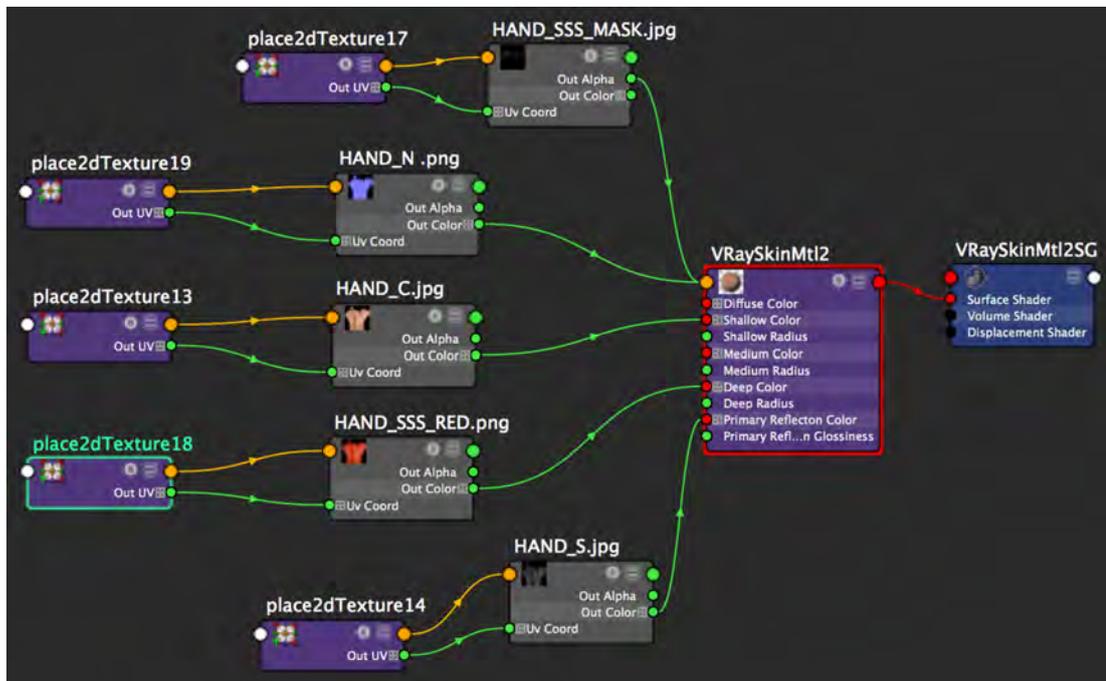
**Abbildung 5.25:** Finales Ergebnis des Handrenderings in der Variante mit dem *FastSSS2*-Shader.

### 5.3.3 Das Skin Material

Für die *Skin Material*-Variante der Umsetzung wird die farbgebende Textur in den *color*-Kanal des *Shallow Scattering Layers* gelegt. Dabei wird der *Amount* mit einem vollen Wert von 1,0, festgelegt, damit auch hier, wie schon bei der Emily-Umsetzung beschrieben, nichts von den Details der Textur verloren geht und sie mit vollem Umfang und nicht abgedunkelt berechnet wird. Der *Medium Layer* soll auch in dieser Anwendung wieder nur zu einem kleinen Teil in die Berechnung mit einfließen und erhält daher einen *Amount* von 0,1. Er behält die Farbe, die er per Default eingestellt hat. Über den *Deep Layer* soll nun der Effekt des Durchleuchtens zwischen den Fingern realisiert werden. Mit dieser zweiten Umsetzung kann die Erkenntnis bestätigt werden, dass durch die reine Farbuweisung der Effekt nicht wie gewünscht umsetzbar ist (siehe Abbildung 5.28). Daher wird der zuvor entwickelte Texturansatz erneut angewendet und gezeigt, dass er auch in dieser Umsetzung funktioniert. Dafür wird, wie im vorherigen Abschnitt dieses Kapitels beschrieben, die farbgebende Textur rot eingefärbt und in den *color*-Kanal des *Deep Layers* gelegt. Um nun die Kontrolle darüber zu erlangen an welchen Stellen des Renderings diese rote Textur ihre Wirkung entfaltet, wird eine Maske erstellt, die nur an den Fingerzwischenräumen den Effekt zulässt. In der Abbildung 5.28, in der die Renderings mit und ohne Maske gegenübergestellt werden, wird deutlich, wie sich der Effekt verstärkt und sich der Farbton ändert, und damit einem

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

realistischeren Ergebnis näher kommt. Die Texturvariante zieht eine geringfügige Steigerung der Renderzeiten mit sich. In diesem Fall hat die Variante mit Maske 31 Sekunden länger gerendert als die Variante ohne Maske.



**Abbildung 5.26:** Das Shadernetzwerk des *Skin Materials* zeigt welche Textur in welchen Knoten des Shaders gelegt wird und wie der Shader aufgebaut ist.



**Abbildung 5.27:** Speziell erstellte Maske und rot eingefärbte Textur zur Erstellung des *Subsurface Scattering*-Effektes des Durchleuchtens mit dem *Skin Material* für das Rendering der menschlichen Hand.



**Abbildung 5.28:** Vergleich der Renderings des *Skin Materials* mit (rechts) und ohne (links) Masken-Variante zur Erzeugung des roten Effektes des Durchleuchtens.

#### 5.3.4 Vergleich

##### Qualität der Ergebnisse

Der Eindruck der ersten prototypischen Umsetzung bezüglich des Renderings menschlicher Haut bestätigt sich auch in der zweiten Szene. Bei der Qualität der Visualisierung des generellen *Subsurface Scattering*-Effektes, der die Weichheit der Haut umsetzt, liefern beide Shader vergleichbare Ergebnisse. Wenn es um die Umsetzung des spezielleren Zusatzeffektes des Durchleuchtens geht, wirkt dieser bei der *FastSSS2*-Variante deutlich stärker ausgeprägt, sowohl in der Leuchtkraft an sich, als auch in der Farbgebung. Um diesem entgegenzuwirken und die Kontrolle darüber zu erhalten an welchen Stellen sich der Effekt und die Farbe ändern, kann für die *Skin Material*-Variante der entwickelte Texturansatz eingesetzt werden. Nach Einsatz der Maske und der speziell eingefärbten Textur für die Realisierung des Effektes des Durchleuchtens sind kaum noch erkennbare Unterschiede in der Qualität der Ergebnisse vorhanden.

##### Handhabung der Shader

Auch in der Handhabung bestätigen sich die bereits gesammelten Eindrücke aus dem ersten prototypischen Haut-Szenario. Zwar geht die Erstellung des Shaders beim zweiten Anlauf, wie zuvor bereits auch beim *FastSSS2* (siehe Kapitel 4) beschrieben, schneller, da die analytischen Versuche welche Einstellungen die richtigen sind, aufgrund der vorangegangenen Erfahrungen, minimiert werden konnten, dennoch bleibt es dabei, dass die Einschränkung in der Kontrollierbarkeit des *Skin Materials* störend ist. Und besonders für derartige Effekte der zusätzliche Parameter der *Scatter color* wünschenswert wäre. Der entwickelte Textur-

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

ansatz überwindet diese Hürde, bedeutet aber zusätzlichen Aufwand, da die Maske und die eingefärbte Textur erstellt und bis zum gewünschten Ergebnis angepasst werden müssen.

### Renderzeiten

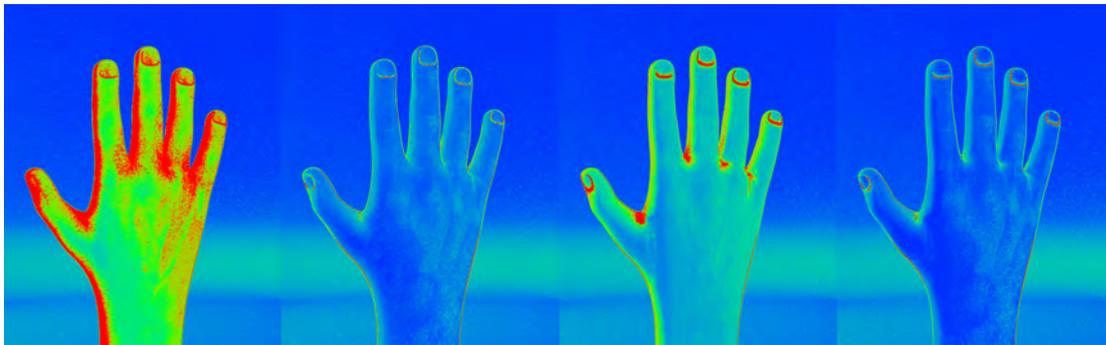
Die Renderzeiten der beiden finalen Shader-Varianten wurden für die Gegenüberstellung jeweils an einem *Shader Template* und am finalen Modell mit jeweils unterschiedlichen Rendersettings in *High Quality*- und *Low Quality*-Einstellungen gemessen, um einen aussagekräftigen Vergleich zu gewährleisten.

Die Renderings für den Vergleich der Renderzeiten wurden alle auf einem Rechner mit 2,3 GHz Intel Core i7 mit 16GB 1600 MHz DDR3 und einer NVIDIA GeForce GTX 650M 1024MB durchgeführt. Die *Shader Templates* wurden mit einer Auflösung von 640 mal 800 Pixeln gerendert. Die Finalen Renderings haben in beiden Varianten eine Auflösung von 960 x 1200 Pixeln.

	FastSSS2-Shader	Skin Material
Shader-Template Low Quality	0h05m31,9s	0h03m33,4s
Shader-Template High Quality	0h12m0,5s	0h07m8,7s
Final Rendering Low Quality	0h04m17,3s	0h02m6,3s
Final Rendering High Quality	0h07m41,7s	0h4m27,1s

**Tabelle 5.2:** Vergleich der gemessenen Renderzeiten für die Hand-Shader

Im Vergleich der Renderzeiten ist zu erkennen, dass das *Skin Material* auch in dieser zweiten Umsetzung deutlich niedrigerer Zeiten verzeichnen kann, als der *FastSSS Shader*. Dabei sind die Zeiten bei allen Durchläufen mindestens ein Drittel höher, bis hin zu einer Verdoppelung bei der *Low Quality*-Variante des finalen Renderings.



**Abbildung 5.29:** Sample Rates für die Renderings der Hand. Von links nach rechts handelt es sich dabei um die *FastSSS2*-Shader-Variante in *Low*- und in *High*-Quality sowie die *Skin Material*-Variante in *Low*- und in *High* Quality.

## 5.4 Zusammenfassung und Ergebnis

Die Hypothese, dass das *Skin Material*, welches von der *Chaosgroup*, seit der Version *V-Ray 3.1*, speziell für das Rendering von Haut-Visualisierungen zur Verfügung gestellt wird, zwar eine vereinfachte Handhabung gegenüber dem *FastSSS2*-Shader bietet, aber aufgrund seiner Parameter-Restriktionen besonders in Bezug auf die Umsetzung des spezielleren Effektes des Durchleuchtens durch *Subsurface Scattering* schnell an seine Grenzen stößt, konnte in zwei prototypischen Implementierungen belegt werden. Durch die bereits gesammelten Vorkenntnisse aus den vorherigen Implementierungen mit dem *FastSSS2*-Shader konnte diese Variante problemlos umgesetzt werden und liefert gute Ergebnisse. Das neue *Skin Material* ist aufgrund seiner schlichter gehaltenen Funktionen und den drei *Layern*, die den drei Hautschichten nachempfunden sind, intuitiv zu bedienen, die schlechte Dokumentation der Funktionsweise ist hingegen ein echtes Hindernis bei der Einarbeitung. Auch sind die Grenzen der Kontrollierbarkeit des Shaders schnell erreicht, besonders wenn es um den *Subsurface Scattering*-Effekt des Durchleuchtens geht. Dieser ließ sich in gewünschter Qualität mit den Bordmitteln des Shaders nicht umsetzen. Dieses Problem konnte gelöst werden, indem ein texturbasierter Ansatz entwickelt wurde, mit dessen Hilfe ein Teil dieser Kontrolle wiederhergestellt wird. Dafür wird eine Maske gezeichnet, welche bestimmt an welchen Stellen die Farbinformationen und der *Scattering*-Effekt des *Deep Layers*, beziehungsweise des jeweiligen *Layers* für den die Maske verwendet wird, zugelassen wird und an welchen Stellen diese blockiert wird. Diese Maske wird dann in den *Amount* des entsprechenden *Layers* gelegt. Mit einer entsprechend eingefärbten Textur im *color* Kanal, kann so der gewünschte Effekt umgesetzt werden. In einer zweiten Umsetzung konnten sowohl die Problematik als auch die entwickelte texturbasierte Lösung nochmals verifiziert werden. In Bezug auf die Handhabung des Shaders gibt es dennoch Abstriche dem *FastSSS2*-Shader gegenüber, da das Erstellen und Anpassen der Textur in der Bildbearbeitungs-Software bis zum gewünschten Ergebnis zusätzlichen Aufwand und damit mehr Zeit erfordert.

## 5. ANALYSE UNTERSCHIEDLICHER METHODEN FÜR DIE VISUALISIERUNG VON SUBSURFACE SCATTERING FÜR DAS RENDERING MENSCHLICHER HAUT

---

Die Entscheidung welcher der beiden Shader für die Umsetzung einer *Subsurface Scattering*-Visualisierung eingesetzt werden sollte, ist sehr spezifisch von der Art des Szenarios abhängig. Der *FastSSS2*-Shader ist sowohl in seiner Handhabung dem *Skin Material* gegenüber überlegen, als auch die Ergebnisse sind von herausragender Qualität, besonders was die physikalische Plausibilität angeht. Die Handhabung des *Skin Materials* ist besonders dann deutlich aufwendiger, wenn es um Anwendungen geht, bei denen der Effekt des Durchleuchtens durch *Subsurface Scattering* umgesetzt werden soll, da hier der Shader mit seinen Bordmitteln schnell an seine Grenzen stößt und der entwickelte texturbasierte Ansatz einigen Aufwand bedeutet und der Shader durch dessen Verwendung an physikalischer Plausibilität deutlich einbüßt. Bei den Renderzeiten, ist das *Skin Material* allerdings dem *FastSSS2*-Shader deutlich überlegen.

Die Verwendung des *FastSSS2*-Shaders ist immer dann empfehlenswert, wenn es um *Close-Ups* oder Detailaufnahmen geht und der *Subsurface Scattering*-Effekt im Fokus der Umsetzung liegt. Auch wenn besonderer Wert auf physikalischer Plausibilität, besonders in Bezug auf den Effekt des Durchleuchtens, liegt, ist die Verwendung des *FastSSS2*-Shaders zu bevorzugen. Bei allen Anwendungen, bei denen gegebenenfalls Abstriche der Qualität zugunsten der Renderzeiten gemacht werden können, ist das *Skin Material* die richtige Wahl. Besonders für Animationen, bei denen viele Frames gerendert werden müssen, kann diese Entscheidung Sinn machen, vor allem dann wenn das Objekt mit dem *Subsurface Scattering*-Effekt eher im Hintergrund der Szene liegt.

Das *Skin Material* hat auch dann Vorteile, wenn es um die Überlagerung mehrere Scattering-Ebenen mit gegebenenfalls unterschiedlichen Texturen geht, die aufgrund der Layer-Struktur des Shaders dann nicht mehr über ein Blend-Material gelöst werden müssen. Auch die beiden Reflection-Layer können für die ein oder andere Umsetzung von Vorteil sein. Beispielsweise könnte für die Reflektionen von Schweiß eine andere Reflektionmap mit Glanzpunkten an unterschiedlichen Stellen und einer anderen Reflektionsstärke eingestellt werden, als für die generell reflektierende Fettschicht der Haut.

Ein weiteres Anwendungsbeispiel für das die Verwendung des *Skin Materials* prädestiniert sein könnte, sind Renderings menschlicher Haut bei denen es nicht um Fotorealismus geht, sondern die einen *Cartoon*-Charakter aufweisen. Wenn für den *Medium*- und den *Deep Layer* keine fotorealistischen Texturen benötigt werden, die wie bereits beschrieben wurde, nicht erzeugt werden können, sondern stattdessen für einen *Cartoon Skin Shader* tatsächlich die Farbwerte für den Aufbau des Shaders genutzt werden, könnten für eine derartige Anwendung mit dem *Skin Material* sehr gute Ergebnisse erzielt werden.

Die statistische Auswertung der Evaluation bezüglich der Frage, ob einer der beiden Shader nachweisbar fotorealistischere Ergebnisse liefert, folgt in Kapitel 7.

## Kapitel 6

# Konzept und Entwicklung eines Plugins für Hautrendering

### 6.1 Einleitung

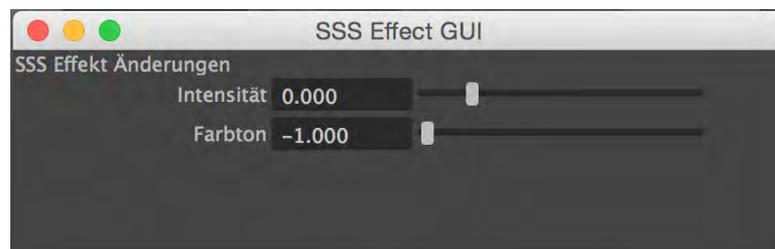
Die Ergebnisse des vorherigen Kapitels haben gezeigt, dass sich das neue *Skin Material* durch seine an den biologischen Aufbau der menschlichen Haut angelehnte Struktur mit den drei *Layern* für die drei entsprechenden Hautschichten sowie durch seine vereinfachte Implementation, die mit deutlich weniger, für den Benutzer regelbaren, Parametern gegenüber dem *FastSSS2*-Shader umgesetzt wurde, zwar intuitiv bedienen lässt, aber vor allem in Bezug auf den speziellen Effekt des Durchleuchtens des *Subsurface Scatterings* hier dem Benutzer schnell Grenzen aufgezeigt werden. Aufgrund der Restriktion der Parameter lassen sich diese Art von Effekte in ihren Renderergebnissen mit diesem Shader nicht mehr hinreichend kontrollieren, so dass die gewünschten Resultate nicht erzielt werden können. Nachdem im Kapitel 5 bereits ein Lösungsansatz entwickelt und vorgestellt wurde, der mithilfe einer speziell angefertigten Maske, sowie einer speziell eingefärbten Textur am Beispiel des Emily-Modells einen Teil der Kontrolle über den Effekt wiederhergestellt hat, soll nun in diesem Kapitel noch einen Schritt weiter gegangen werden. Im folgenden Kapitel wird ein Konzept für ein *Plugin* entwickelt und umgesetzt, welches es ermöglicht, den für fotorealistische Renderings so elementaren Durchleucht-Effekt der Haut mit einem Interface zu steuern.

### 6.2 Konzept und Idee

Um in Bezug auf das Rendering menschlicher Haut Fotorealismus erreichen zu können, ist es unumgänglich, beim Aufbau der jeweiligen Shader mit Texturen zu Arbeiten, um die feinen Strukturen der Haut abbilden zu können. In Bezug auf das neue *Skin Material* wurde im vorherigen Kapitel die Problematik, dass dieser Shader in Bezug auf den *Subsurface Scattering*-Effekt des Durchleuchtens schnell an seine Grenzen stößt, ebenfalls mit einem texturbasierten Ansatz gelöst. Der entsprechend notwendige Farbton für den *Deep Layer* kann weder als Textur fotografiert noch berechnet werden und aufgrund der Tatsache, dass er von vielen unterschiedlichen Faktoren, wie beispielsweise dem Lichteinfall in der Szene abhängt, kann

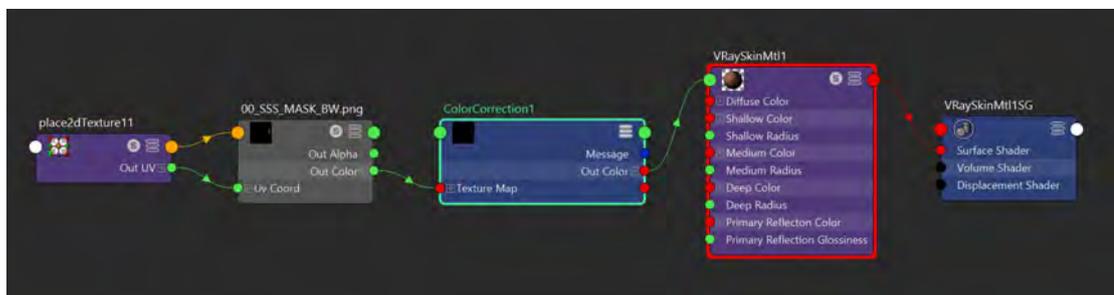
## 6. KONZEPT UND ENTWICKLUNG EINES PLUGINS FÜR HAUTRENDERING

zunächst nur eine grobe Abschätzung für den richtigen Farbton gemacht werden. Oft ist sehr aufwendig zu einem zufriedenstellenden Ergebnis dieser filigranen Effekte zu gelangen, da sich nur in minimalen Anpassungen der Textur in der Bildbearbeitungs-Software dem gewünschten Ziel angenähert werden kann. So ist es in diesem Fall ein ständiger Wechsel zwischen der Bildbearbeitungs-Software und dem *Skin Shader* in der 3D-Software *Maya* unumgänglich. Diese Vorgänge können schnell extrem zeitintensiv werden. Aus diesem Grund soll ein *Interface* entwickelt werden, welches für den *Skin Shader* den Effekt des Durchleuchtens wieder kontrollierbar macht, ohne ständig die Software wechseln zu müssen, um die entsprechenden Texturen anzupassen.



**Abbildung 6.1:** Interface zur Anpassung des *Subsurface Scattering*-Effektes mit zwei Slidern.

Es muss zwar noch im ersten Schritt einmal die entsprechende Maske und die entsprechend eingefärbte Textur erstellt werden, jegliche weitere Anpassung kann dann aber innerhalb der Software *Maya* erfolgen. Dies birgt auch einen weiteren großen Vorteil, nämlich, dass sich die Änderungen, die an der Textur vorgenommen werden, direkt im *Realtime*-Renderer anzeigen lassen. Es geht also darum, die nachträglichen Anpassungen der Textur nicht in *Photoshop* durchführen zu müssen, sondern sie direkt in *Maya* vorzunehmen. Hierzu existiert ein undokumentierter Knoten, welcher in der *V-Ray*-Bibliothek versteckt ist, der "ColorCorrection"-Knoten. Dieser ermöglicht Änderungen einer Textur in Bezug auf typische Attribute wie Helligkeit, Sättigung, Kontrast, Farbwert im HSV-Farbraum und weitere Parameter. Die eigentliche Anpassung kann also durch die Änderung der Attributwerte dieses *ColorCorrection*-Knotens bewirkt werden, der im Netzwerk des Shaders hinter die entsprechende Textur geschaltet wird.



**Abbildung 6.2:** Shadernetzwerk zur Einbindung des *ColorCorrection*-Knoten

Wenn dieser nicht dokumentierte Knoten gefunden und korrekt angelegt ist, ist er auch ohne Skript bedienbar. Das hierzu entwickelte Skript soll nun:

1. Die Arbeitsabläufe beschleunigen
2. Das Zusammenspiel verschiedener Parameter für den *Subsurface Scattering*-Effekt kombinieren

Um die gewünschten Effekte ohne ein Skript zu erzielen, ist schon ein großer Vorteil der direkten Bedienung durch die Verwendung des *ColorCorrection*-Knotens in *Maya* gegeben. Dennoch sind zum einen eine Vielzahl kleiner redundanter Einzelschritte notwendig, um zu einem entsprechenden Ergebnis zu gelangen. Die hohe Anzahl von Fenstern, die benötigt werden, bringt Schwierigkeiten mit sich, besonders bei der Bedienung mit nur einem Monitor und es sind permanente Wechsel von Knoten im Attribut Editor notwendig. Ziel ist es, all diese Gegebenheiten mit dem Skript deutlich zu vereinfachen. Zum Anderen soll es dem *3D-Artist* durch das Skript ermöglicht werden, effektorientiert arbeiten zu können und sich nicht laufend mit Fragen der Bildbearbeitung konfrontiert zu sehen. Daher sollen die wichtigsten Attribute des Knotens herausgearbeitet und so kombiniert werden, dass sie die bestmöglichen Resultate für den Durchleucht-Effekt erzielen. Das *Interface* soll so lediglich auf zwei Regler reduziert werden, die die wichtigsten Eigenschaften des Effektes des Durchleuchtens bei menschlicher Haut repräsentieren, die Intensität des Effektes und den Farbton. Die Einzelattribute wie Helligkeit, Kontrast und Sättigung sollen für den Anwender automatisch korrekt miteinander verrechnet werden.

### 6.3 Maya Embedded Language-Skripte

Bei *Maya Embedded Language*, kurz *MEL*, handelt es sich um eine Skript-Sprache mit der Abläufe in *Maya* automatisiert werden können [Gou03]. *MEL* erlaubt, alle Befehle und Interaktionen des Anwenders in *Maya* auch als programmierbaren Befehl zu verwenden. Eine besondere Vereinfachung bietet *MEL* dadurch, dass jede im *Viewport* ausgeführte Aktion im *Maya* Skript-Editor als *MEL*-Befehl protokolliert wird. So lässt sich ein unbekannter zu programmierender Befehl schnell und einfach durch dessen normale Ausführung und Aufsuchen im Skript-Editor in Erfahrung bringen.

## 6. KONZEPT UND ENTWICKLUNG EINES PLUGINS FÜR HAUTRENDERING

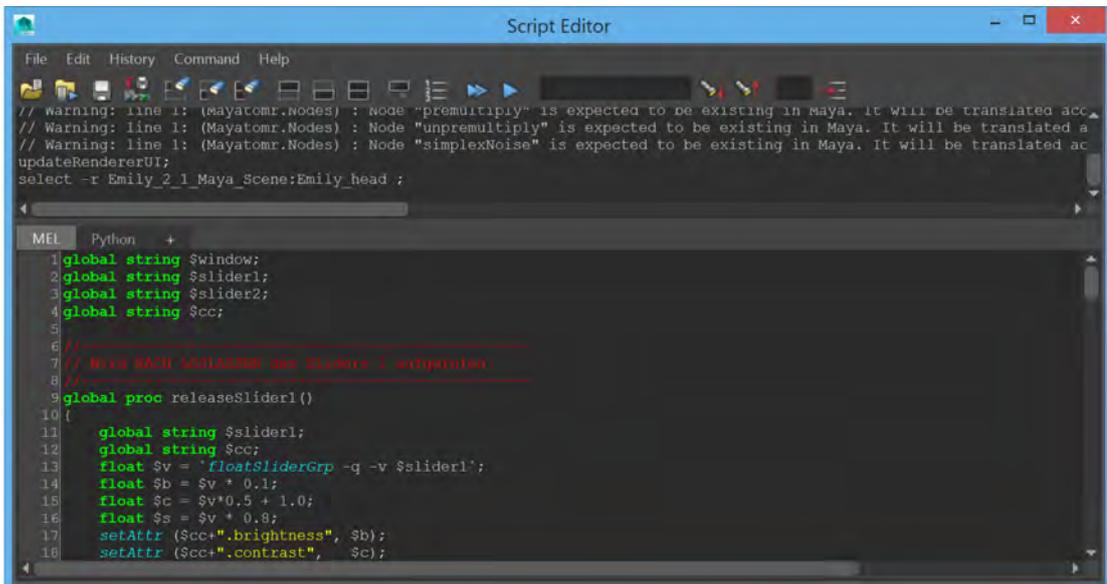


Abbildung 6.3: Skript-Editor, welcher in der Software Autodesk Maya bereitgestellt wird.

Es wird keine eigene Entwicklungsumgebung für die Programmierung mit *MEL* benötigt. Jeder einfache Text-Editor ist ausreichend. *Maya* stellt aber auch einen komfortablen Editor zur Verfügung, den Skript-Editor. Dieser beherrscht *MEL* und *Python*. Programmiert man in einem anderen Texteditor wie *Wordpad* oder ähnlichem, wird die Textdatei anschließend in den *Maya*-Skript-Editor geladen und von dort aus gestartet. Dieser dient gleichzeitig auch als Ausgabe-Konsole auf der zum Beispiel mit dem *Print*-Befehl Informationen auf dem Bildschirm ausgegeben werden können.

### 6.4 Installation

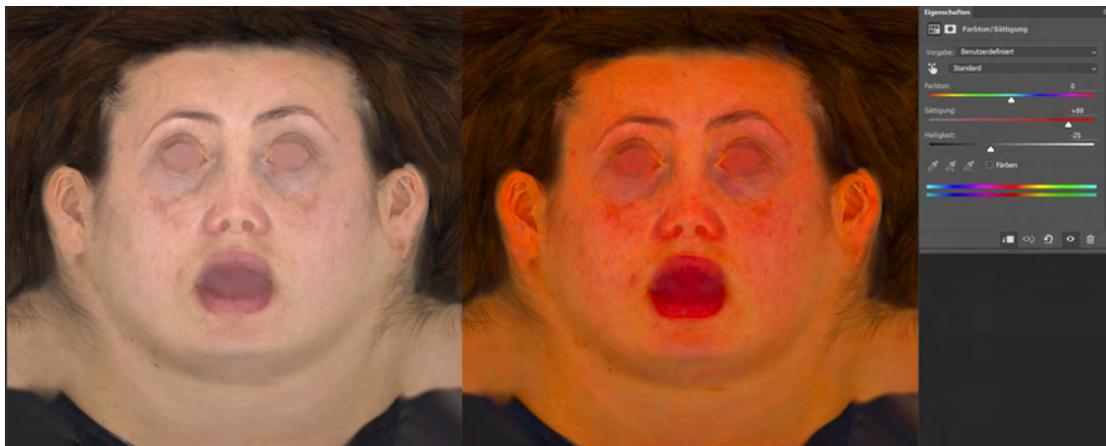
Die Installation ist denkbar einfach. Das Skript wird in den *Maya*-Skript-Editor geladen, der Text vollständig markiert und in das bevorzugte *Shelf* gezogen. *Maya* erzeugt dann dort automatisch eine Schaltfläche, deren Betätigung die Ausführung des Skriptes bewirkt.



Abbildung 6.4: Shelf-Button der für die Ausführung des Skriptes angelegt wird.

## 6.5 Vorbereitung einer Szene

Die Idee ist es, den speziellen Effekt des Durchleuchtens des *Subsurface Scatterings*, den das *Skin Material* nicht in gewünschter Form umsetzen kann, über eine Textur zu realisieren, welche in den *Deep Layer*-Kanal eingelesen wird. Wie bereits in Kapitel 5 beschrieben, wird in *Photoshop* dafür die Hauttextur rot eingefärbt. Der perfekte Farbton kann aber nicht im Vorfeld ermittelt werden, da er abhängig von einer Vielzahl anderer Faktoren wie beispielsweise der Beleuchtungssituation in der Szene, den Render Settings oder der Originaltextur ist. Daher erfolgt die Annäherung an einen gewünschten Farbton durch eine Abschätzung. Die Einfärbung erfolgt typischerweise durch Anwendung von Korrekturfiltern für Farbton und Sättigung.



**Abbildung 6.5:** Einfärben der Textur in einer Bildbearbeitungs-Software wie beispielsweise Adobe Photoshop.

Außerdem wird eine Graustufen-Maske angefertigt, welche die Intensität des Effektes lokal begrenzt. In diesem Fall wird der Effekt auf das rechte Ohr begrenzt. Je heller der Maskenwert gewählt wird, desto stärker ist der Effekt im Rendering später ausgeprägt.

## 6. KONZEPT UND ENTWICKLUNG EINES PLUGINS FÜR HAUTRENDERING

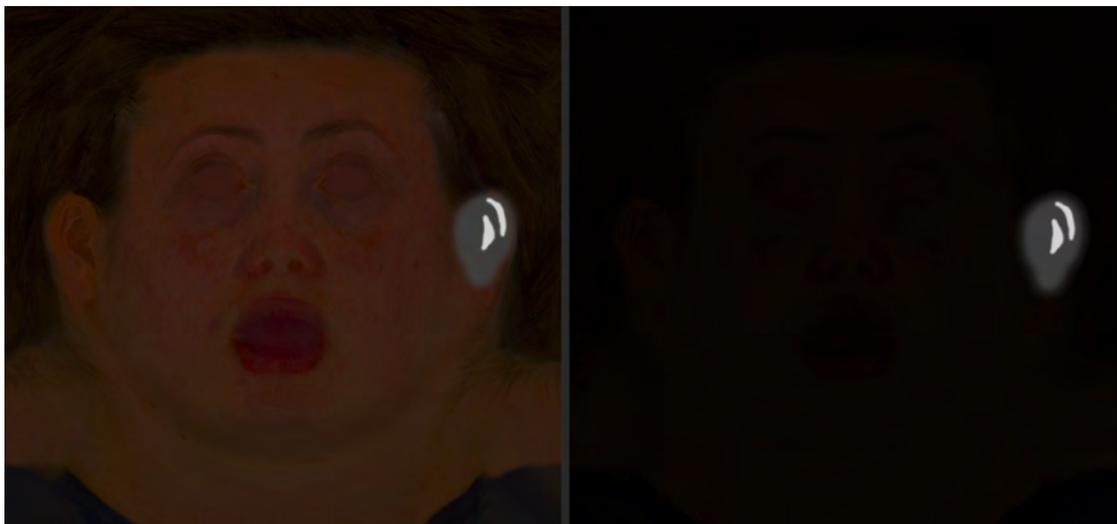


Abbildung 6.6: Erstellen der Maske

Im *Skin Material* wird, wie bereits im vorherigen Kapitel beschrieben, nun die rot eingefärbte Textur für das Attribut *color* im *Deep Layer* verwendet und die Maske für den *Amount*-Parameter des selben *Layers* eingesetzt.

### 6.6 Implementierung

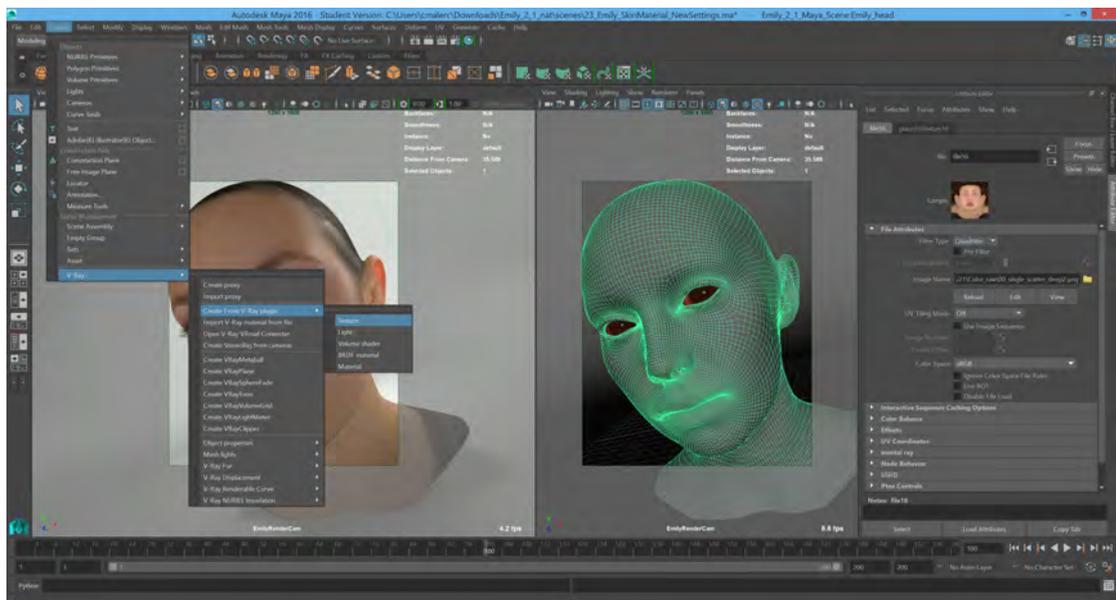


Abbildung 6.7: Anlegen des *ColorCorrection*-Knotens



## 6.7 Aufbau des Skriptes

Das Skript besteht aus drei Teilen:

- einer Initialisierungsmethode `start()`
- einer Methode zur Erzeugung des Interface mit den beiden *Slidern*
- einer Gruppe aus *Callback*-Funktionen, die bei Interaktionen die Knoten-Attribute ändern

Die Start-Funktion übernimmt die Aufgabe aller sicherheitsrelevanten Überprüfungen und ermittelt den für das ausgewählte Objekt angelegten *ColorCorrection*-Knoten. Als erstes wird sichergestellt, dass genau eine Geometrie selektiert ist.

Listing 6.1: Funktion zum Starten des SSS-Interfaces

```
global proc start() {
    global string $cc; // Variable für den ColorCorrection-Knoten
    string $objects[] = 'ls -sl';

    int $count = size($objects);
    if ($count <= 0) {
        deleteUI SSS_GUI;
        error("Bitte_ein_Objekt_auswählen!");
    }
    if ($count > 1) {
        deleteUI SSS_GUI;
        error("Bitte_nur_ein_Objekt_auswählen!");
    }
    for ( $one_obj in $objects ) {
        if ( 'nodeType $one_obj' != "transform" ) {
            deleteUI SSS_GUI;
            error("Bitte_nur_Geometrien_anwählen!");
        }
    }
}
```

Danach wird das zur Geometrie gehörende Material gesucht und geprüft, dass hier bereits ein *ColorCorrection*-Knoten angelegt wurde.

Listing 6.2: Suche nach ColorCorrection-Knoten

```
// Suche nach ColorCorrection-Knoten in der Selektion
hyperShade -smn;
string $getMaterial[] = 'ls -sl';
string $mat = $getMaterial[0];
print($mat + "\n");
string $list[] = 'listConnections -s true -d false $mat';

for( $node in $list ) {
    if( 'nodeType $node' == "VRayPluginNodeTex" ) {
        print("VRayPluginNodeTex_node_found:_" + $node + "\n");
        $cc = $node;
        break;
    }
}
```

Der ermittelte Knoten wird als nächstes als Namens-Variable der Methode übergeben, welche das *GUI* erzeugt und den Verbindungsaufbau zu den einzelnen Attributen des *ColorCorrection*-Knotens herstellt.

Listing 6.3: Aufruf der CreateGui-Methode mit Übergabe des CC-Variablen-Namens

```

if($cc == "") {
    deleteUI SSS_GUI;
    error("Noch kein ColorCorrection-Node angelegt");
} else { // Wenn cc-Knoten gefunden, GUI erzeugen
    // Dafür übergeben wir den Knoten, damit die Werte ausgelesen
    // und für die Slider gesetzt werden können.
    createGUI($cc);
}
}

```

Die Methode `createGUI` erzeugt das *Interface*, in dem die beiden Schieberegler für die Intensität und den Farbton des Leucht-Effekts dargestellt werden sollen. Eventuell noch geöffnete Skript-Fenster werden geschlossen, um zu verhindern, dass mit der Zeit eine unübersichtliche große Zahl an *Interfaces* geöffnet ist.

Listing 6.4: Erzeugung des Interface

```

global proc createGUI(string $cc_node)
{
    global string $window;
    global string $slider1;
    global string $slider2;

    if ( 'window -exists SSS_GUI' ) { // Wenn noch ein Fenster offen ist...
        deleteUI SSS_GUI;           // ...schließen und löschen
                                   // bevor wir ein neues aufmachen
    }
    string $window = 'window -title "SSS-Effect-GUI" SSS_GUI';
    print($window + "_created.\n");
    columnLayout; // Für horizontale Slider
    text -label "SSS-Effekt-Änderungen";
}

```

Vor der Erzeugung der *Slider* liest die Methode die aktuellen Werte aus dem *ColorCorrection*-Knoten aus, um die richtigen Werte nach der Erzeugung der Schieberegler setzen zu können. Für die Berechnung der Effekt-Intensität werden die Attribute *Brightness* (Helligkeit), *Contrast* (Kontrast, Brillanz) und *Saturation* (Sättigung) verwendet. Für den Farbton des Effekts dient das Attribut *Hue Shift*, das den reinen Farbgrad einer Farbe im HSV-Farbraum beschreibt.

Listing 6.5: Auslesen der aktuellen Werte für die Slider-Attribute

```

// Auslesen der aktuellen Knoten-Werte
float $brightness_value = 'getAttr($cc_node + ".brightness")';
float $contrast_value   = 'getAttr($cc_node + ".contrast")';
float $saturation_value  = 'getAttr($cc_node + ".saturation")';
float $hue_shift_value   = 'getAttr($cc_node + ".hue_shift")';

float $intensity_value = 0.0;
float $color_value     = $hue_shift_value * 0.1;

$slider1 = 'floatSliderGrp -label "Intensität" -field true
-minValue -0.2 -maxValue 1.0 // initiales Min/Max des Sliders
-fieldMinValue -1.0 -fieldMaxValue 1.0 // höchstes Min/Max
-value $intensity_value // Startwert aus dem CC-Knoten ausgelesen
-step 0.001 // Schrittweite hier mit drei Nachkommastellen
-dragCommand "dragSlider1()" // bei gedrückter Maustaste
-changeCommand "releaseSlider1()"'; // Mausloslassen-Callback

$slider2 = 'floatSliderGrp -label "Farbton" -field true
-minValue -1.0 -maxValue 1.0 // initiales Min/Max des Sliders
-fieldMinValue -1.0 -fieldMaxValue 1.0 // höchstes Min/Max
-value $color_val // Startwert aus dem CC-Knoten ausgelesen
-step 0.001 // Schrittweite hier mit drei Nachkommastellen
-dragCommand "dragSlider2()" // bei gedrückter Maustaste
-changeCommand "releaseSlider2()"'; // Mausloslassen-Callback

showWindow $window; // Zeigt nun das Fenster auch an...
}

```

In der *createGUI*-Methode wurden beim Erzeugen der Schieberegler die *Callback*-Funktionen deklariert, die bei Bedienung der *Slider* aufgerufen werden sollen. Jeder der beiden *Slider* hat zwei *Callbacks*: Eine Funktion, die bei Wertänderung während des Schiebevorgangs (mit noch gedrückter Maustaste aufgerufen wird) und eine, die beim Loslassen des *Sliders* aufgerufen wird. Beide Methoden berechnen aus der aktuellen *Slider*-Einstellung die entsprechenden Werte für den *ColorCorrection*-Knoten und setzen diese dort. Da bildverarbeitende Methoden insbesondere bei großen Texturen lange Rechenzeiten verursachen können, ist eine Trennung von *drag*- und *release*-Operationen sinnvoll (wenn auch zur Zeit noch nicht implementiert). Der *Callback* für die Intensität setzt aus den drei genannten Attributen einen kombinierten Wert zusammen, der dem Knoten übergeben wird. Dabei beträgt die Sättigung 80% des *Slider*-Wertes, die Sättigung nimmt 50% des *Slider*-Wertes und verschiebt den Wert um 1.0 und die Helligkeit wird mit 10% mit einer leichten Anhebung angepasst.

Listing 6.6: Callback für Intensitäts-Slider

```

// Callback für Intensitäts-Slider
global proc releaseSlider1()
{
    global string $slider1;
    global string $cc;
    float $v = 'floatSliderGrp -q -v $slider1';
    float $b = $v * 0.1;
    float $c = $v * 0.5 + 1.0;
    float $s = $v * 0.8;
    setAttr ($cc + ".brightness", $b);
    setAttr ($cc + ".contrast", $c);
    setAttr ($cc + ".saturation", $s);
}

```

Die Methode für die Anpassung des Farbtons macht sich zunutze, dass der *ColorCorrection*-Knoten neben dem RGB-Farbraum auch im HSV-Farbraum arbeiten kann, und hier einer der drei Kanäle für die Grundfarbe (ungeachtet der Sättigung und der Helligkeit) verändert

werden kann. Alle Farbwerte sind laut Definition auf einem Farbkreis angeordnet. Dementsprechend können Grundfarben durch eine Gradangabe zwischen  $0^\circ$  und  $360^\circ$  bestimmt werden. Laut Definition des Farbraumes ist die Farbe Rot bei  $0^\circ$  gesetzt. Der *Slider* ermöglicht nun Werte von  $\pm 40^\circ$ , wobei in negative Richtung dem Rot ein höhere Blauanteil und in positiver Richtung ein höhere Gelbanteil beigemischt wird.

Listing 6.7: Callback für Farbton-Slider

```
// Callback für Farbton-Slider
global proc releaseSlider2()
{
    global string $slider2;
    global string $cc;
    float $v = 'floatSliderGrp -q -v $slider2';
    float $v_calc = $v * 40.0;
    setAttr ($cc+".hue_shift", $v_calc);
}
```



**Abbildung 6.9:** Ergebnis des *Sliders* in einer kombinierten Anwendung mit dem *Realtime-Renderer*. Die obere Reihe zeigt die Veränderung des Effektes bei Betätigung des *Sliders* für unterschiedliche Intensitäten des Leuchtens und die untere Reihe unterschiedliche Farbtöne. Hier wurden zur besseren Veranschaulichung relativ extreme Farbvarianten gewählt. In der Realität würde mit subtileren Abstufungen gearbeitet werden.

### 6.8 Zusammenfassung

Um den im vorherigen Kapitel 5 entwickelten texturbasierten Lösungsansatz für die verbesserte Umsetzung des Effektes des Durchleuchtens des *Subsurface Scatterings* mit dem neuen *V-Ray Skin Material* zu vereinfachen und die Arbeitsvorgänge zu beschleunigen, wurde ein *Plugin* konzeptioniert und prototypisch implementiert, welches mithilfe des versteckten *V-Ray ColorCorrection*-Knotens die direkte Bildbearbeitung in *Maya* ermöglicht. Dadurch kann dem *3D-Artist* bei der Anwendung des texturbasierten Ansatzes viel Zeit erspart werden, da er nicht für jede Anpassung die Software wechseln und zurück in die Bildbearbeitungssoftware gehen muss, um die Textur, welche für die Erzeugung des Durchleucht-Effektes eingesetzt wird, entsprechend seiner Vorstellung in vielen Iterationen bis zum gewünschten Resultat anzupassen. Um die Abläufe zu beschleunigen, der Problematik mit zu vielen Fenstern auf einem Bildschirm entgegen zu wirken und dem *3D-Artist* eine effektorientiertere Arbeit zu ermöglichen, wurde ein *MEL-Skript* geschrieben, welches ein *Interface* mit lediglich zwei *Slidern* erzeugt, die in einer bereits aufeinander abgestimmten Kombination von Einzeleffekten der Attribute Kontrast, Helligkeit und Sättigung, sowie des HSV-Farbraum-*Shifts* die Regelung der Intensität und des Farbtons des Leuchteffektes mithilfe von zwei *Slidern* zur Verfügung stellen. So kann mit wenigen Klicks das *Interface* geöffnet und direkt am *Realtime*-Renderer die Feinabstimmung der Textur vorgenommen werden. Es handelt sich bei der vorliegenden Umsetzung um einen Prototypen, der zeigen soll, dass eine Vereinfachung der Arbeitsabläufe in der Form funktionieren. Sinnvolle, noch nicht umgesetzte Erweiterungen wären in jedem Fall, dass der *ColorCorrection*-Knoten, der bisher noch vorab manuell angelegt werden muss, automatisch über das Skript erstellt wird. Des Weiteren wäre es wünschenswert, nicht nur die Farbtextur, sondern auch die Graustufenmaske nachträglich direkt in *Maya* anpassen zu können. Das nächste Kapitel beinhaltet mit der Evaluation die statistische Auswertung der Fragestellung, welche der jeweiligen gegenübergestellten Shaderpaare: *FasstSSS2* und *V-Ray Mtl.* sowie *FastSSS2* und *Skin Mtl.* die Ergebnisse mit dem höheren Fotorealismus liefern.

# Kapitel 7

## Evaluation

### 7.1 Einleitung

Nachdem in den drei methodischen Kapiteln dieser Arbeit zunächst das *V-Ray Material* und der *FastSSS2*-Shader an den Materialien Wachs und Marmor und darauffolgend der *FastSSS2*-Shader und das *Skin Material*, für die *Subsurface Scattering*-Visualisierung beim Rendering menschlicher Haut in jeweils zwei prototypischen Umsetzungen realisiert wurden und bezüglich ihrer Qualität, Handhabung und Renderzeiten analysiert worden sind und anschließend für das *Skin Material* ein *Plugin* für die Vereinfachung des nötigen texturbasierten Ansatzes für die Realisierung des Effektes des Durchleuchtens entwickelt wurde, folgt nun die Evaluation der Ergebnisse.

Das hier vorliegende Kapitel beinhaltet den Versuch, die subjektiven Parameter greifbar und durch eine statistische Auswertungen objektiv bewertbar zu machen. Dabei soll der Nachweis erbracht werden, dass mit Hilfe des Renderers *V-Ray* der *Chaosgroup* und der in dieser Arbeit analysierten Methoden zur Umsetzung von *Subsurface Scattering*-Visualisierungen fotorealistische Bilder erzeugt werden können. Des Weiteren wird nach einer Antwort auf die Frage gesucht, ob jeweils einer der beiden für die Analysen miteinander verglichenen Shader auch statistisch nachweisbar bessere Ergebnisse liefert.

### 7.2 Konzept

Die Evaluation besteht aus drei Abschnitten:

1. Demografische Erhebung (zur Einordnung der Tests)
2. Fake or Foto-Test (zum Nachweis des Fotorealismus der Methoden und zum Vergleich der Shader)
3. Vergleichstest (zur direkten Gegenüberstellung der Shader)

## 7. EVALUATION

---

Um möglichst viele Probanden für die Durchführung der Tests gewinnen zu können, wurde eine Online Plattform<sup>1</sup> genutzt. So konnten in relativ kurzer Zeitspanne viele Personen erreicht werden. Die Teilnehmer konnten die Evaluation in Ruhe zu Hause oder unterwegs auf dem Smartphone durchführen und sie mit ihren Freunden und Bekannten direkt auf sozialen Netzwerken wie *Facebook*<sup>2</sup>, *Twitter*<sup>3</sup>, *GooglePlus*<sup>4</sup> oder per E-Mail teilen und so weiterverbreiten. Die Evaluation ist anonym. Die demografische Erhebung wurde auf das nötigste beschränkt. Abgefragt wurden das Alter und das Geschlecht der Probanden sowie, ob Vorkenntnisse im Bereich der Computergrafik bestehen. Diese Daten wurden erhoben, da getestet werden soll, ob das Alter, das Geschlecht, vor allem aber die Vorkenntnisse in der Computergrafik gegebenenfalls Auswirkungen auf die Beantwortung der Testfragen haben.

**Abbildung 7.1:** Oberfläche des Webportals für die demografische Erhebung.

Nach der kurzen Befragung zur Erhebung der genannten Daten folgt der erste Test. Dieser ist inspiriert von "Fake or Foto" von *Autodesk*<sup>5</sup>. Dem Probanden wird dabei ein Bild gezeigt und dieser soll dann entscheiden, ob es sich bei dem Bild um ein reales Foto oder ein am Computer erstelltes Rendering handelt. Für die Befragung wurden die für diese Arbeit in den prototypischen Umsetzungen erstellten Renderings unter reale Fotos und weitere am Computer erstellte Grafiken gemischt.

---

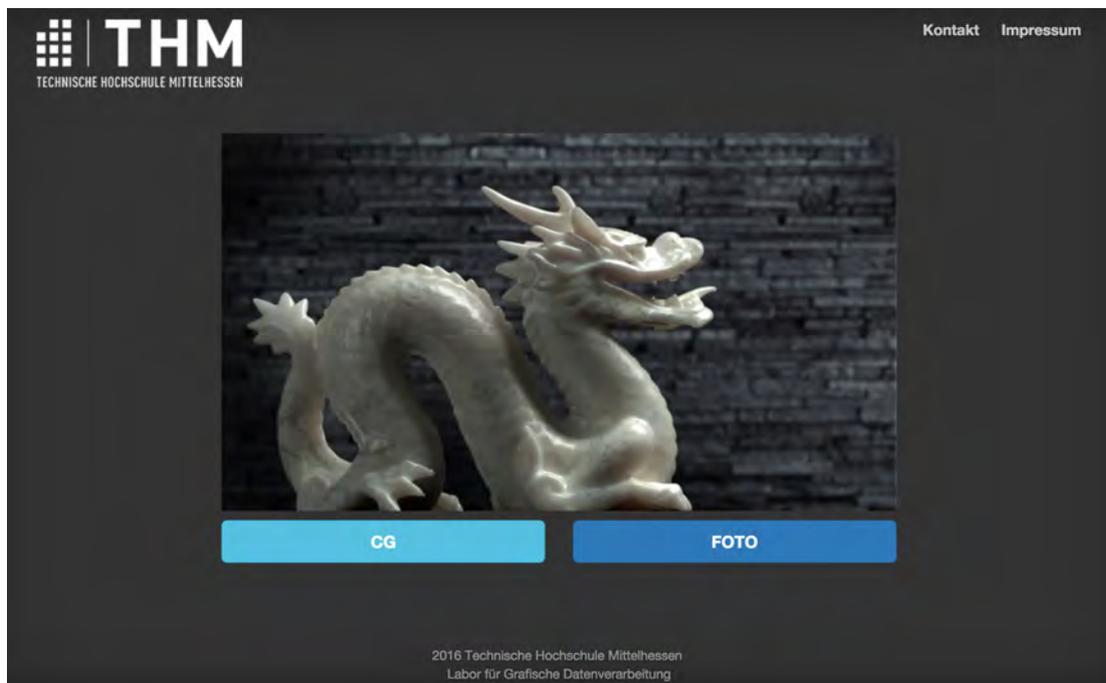
<sup>1</sup><http://evaluation.ngoepfert.visioulounge.eu/> Stand: 22. März 2016

<sup>2</sup><https://www.facebook.com/> Stand:25:03.2016

<sup>3</sup><https://twitter.com>

<sup>4</sup><https://plus.google.com>

<sup>5</sup><http://area.autodesk.com/fakeorfoto> Stand: 22. März 2016



**Abbildung 7.2:** Oberfläche des Webportals für den “Fake oder Foto”-Test.

Bei der Auswahl der Bilder wurde darauf geachtet, dem Probanden eine repräsentative Auswahl an unterschiedlichen Bildern zu zeigen, damit sich möglichst keine Muster abzeichnen, die die für diese Arbeit produzierten Renderings in irgendeiner Weise herausstellen. Unter den Bildern sind sowohl Fotos als auch Renderings mit und ohne *Subsurface Scattering*. Des Weiteren wurde darauf geachtet, dass typische Objektfotos aber auch Renderings mit aufwendigeren Hintergründen dabei sind. Damit sich die für diese Arbeit produzierten Renderings durch die sehr ähnlichen Hintergründe nicht zu stark von den anderen Bildern abheben und direkt auffallen, wurden die Kerze und der Marmordrachen vor einen anderen Hintergrund gesetzt (siehe Abbildung 7.3). Für die beiden Haut-Renderings wurde der graue “Studio-Hintergrund” gelassen. Unter die Bildersammlung wurden außerdem zwei “Kontrollbilder” gemischt (siehe Abbildung 7.4). Dabei handelt es sich um ein Bild (Rendering der Kamera), das sehr eindeutig als computergeneriertes Bild zu erkennen ist, um zu überprüfen, ob die Probanden in der Lage sind, ein nicht fotorealisiertes Rendering auch als solches zu klassifizieren. Mit der mit deutlichem Abstand höchsten Erkennungsrate aller im Test verwendeten Renderings von 84% konnte dies bestätigt werden. Beim zweiten Kontrollbild handelt es sich um ein reales Foto, welches aber mit großer Wahrscheinlichkeit von den Testpersonen für ein am Computer erzeugtes Bild gehalten wird. Dabei geht es um das Foto der Schneeflocke. Ohne die Kenntnis darüber, wie das Foto entstanden ist, wird es für unmöglich gehalten, eine solche Aufnahme zu produzieren. Mit einer Erkennungsrate von 24,5%, die der mit Abstand niedrigsten Erkennungsrate aller im Test verwendeten Fotos entspricht, konnte auch diese Annahme bestätigt werden.



**Abbildung 7.3:** Die Renderings des Marmordrachens und der Kerze jeweils in den beiden Shader-Varianten mit den für die Evaluation neu bearbeiteten Hintergründe.



**Abbildung 7.4:** Die zur Kontrolle in die Evaluation eingefügten Bilder. Links die Schneeflocke, die für ein Rendering gehalten werden soll, obwohl es sich um ein Foto handelt. Ohne die spezielle Technik zu kennen, lässt es sich nur schwer vorstellen, dass so eine Aufnahme real fotografiert werden kann. Rechts die Kamera, die ganz eindeutig am Computer generiert wurde.

Da mit dieser Evaluation der Frage auf den Grund gegangen werden soll, ob einer der beiden Shader fotorealistischere Ergebnisse liefert als der andere, wurden zwei Bild-Sets erstellt. Dabei enthält Set 1 alle Renderings, die mit dem *FastSSS2*-Shader umgesetzt wurden und Set 2 die Renderings, die mit dem *V-Ray Material* oder dem *Skin Material* umgesetzt wurden. Abgesehen von den für diese Arbeit produzierten Renderings enthalten beide Sets die

gleichen Bilder, lediglich diese Renderings sind je nach Set unterschiedlich. Jeder Proband, der an der Evaluation teilnimmt, bekommt für den "Fake or Foto"-Test nur eines dieser Sets per Zufallsalgorithmus zugewiesen.



**Abbildung 7.5:** Gegenüberstellung der beiden Bildsets. In Bild-Set 1 (links) werden die FastSSS2-Renderings unter die anderen Bilder gemischt, in Bild-Set 2 (rechts) die VRay Material-Renderings beziehungsweise die Skin Material-Renderings.

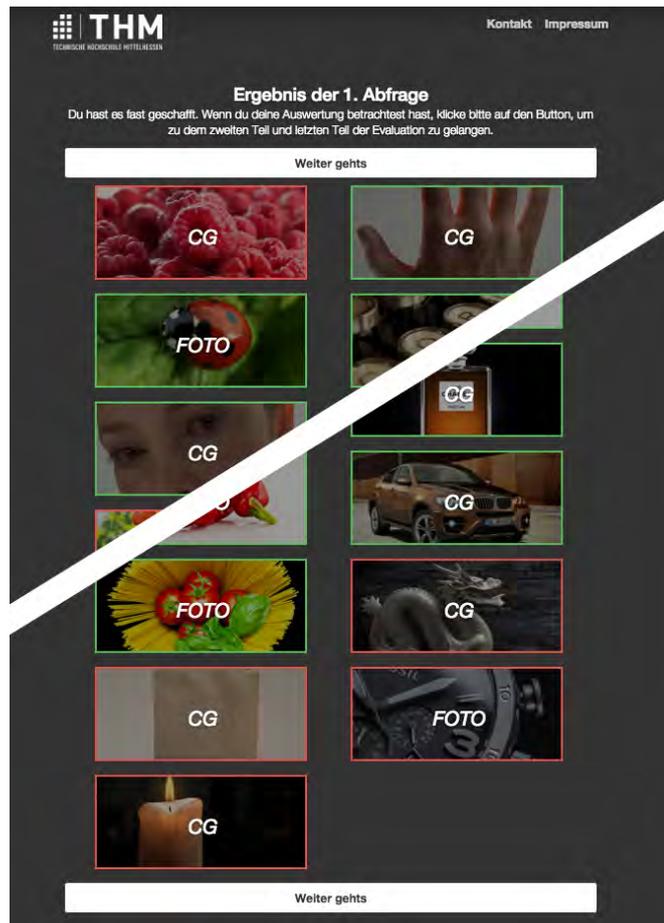
Um möglichst auszuschließen, dass aufgrund der Reihenfolge der Bilderanordnung Muster bei der Entscheidung entstehen, wurde auch die Anzeigereihenfolge für jeden Probanden erneut mit Hilfe eines Zufallsalgorithmus bestimmt.

Bevor es zum zweiten Test geht, bekommt der Proband die Auflösung, welche seiner Entscheidungen richtig waren und welche nicht und damit auch, bei welchen Bildern es sich um reale Fotos handelt und welche mit Hilfe von moderner Computergrafik erstellt wurden.

## 7. EVALUATION

---

Die Auflösung hat keine Bewandnis für die Evaluation an sich, aber wer den Test gemacht hat, der möchte vermutlich auch gerne die Auflösung wissen. Außerdem regt die Auflösung und gegebenenfalls die Feststellung, in wie vielen Fällen sich hinter den vermeintlichen Fotos dann doch Renderings verborgen haben, die Motivation der Probanden an, die Evaluation weiter zu verbreiten.



**Abbildung 7.6:** Anzeige der Auflösung des “Fake oder Foto”-Tests für den Probanden. Der rote oder grüne Rahmen um das Bild, gibt an, ob seine Wahl richtig oder falsch war. Der Text im Bild verrät, ob es sich um ein reales Foto oder eine am Computer erzeugte Grafik handelt.

Beim zweiten Test weiß der Proband dann vorab schon, dass es sich um am Computer generierte Renderings handelt. Im direkten Vergleich werden die prototypischen Umsetzungen jeweils in beiden Shader-Varianten angezeigt und der Tester soll entscheiden, welches der beiden Bilder realistischer ist. Beim “Fake oder Foto”-Test können noch eine ganze Reihe anderer Faktoren Einfluss auf die Entscheidung des Probanden nehmen, die mit den Shadern an sich nichts zu tun haben, wie beispielsweise Reihenfolgen der angezeigten Bilder (“wenn gerade zwei Fotos gezeigt wurden, muss ja auch mal ein Rendering folgen”), Hintergründe,

das *Uncanny Valley* bei den Haut-Renderings. Im direkten Vergleich geht es nun darum, nochmal Aufschluss darüber zu erlangen, welcher der beiden Shader-Varianten jeweils als die realistischere angesehen wird, um darüber für den 3D-Artisten eine Entscheidung treffen zu können, welcher Shader für die Produktion fotorealistischer Renderings zu bevorzugen ist.

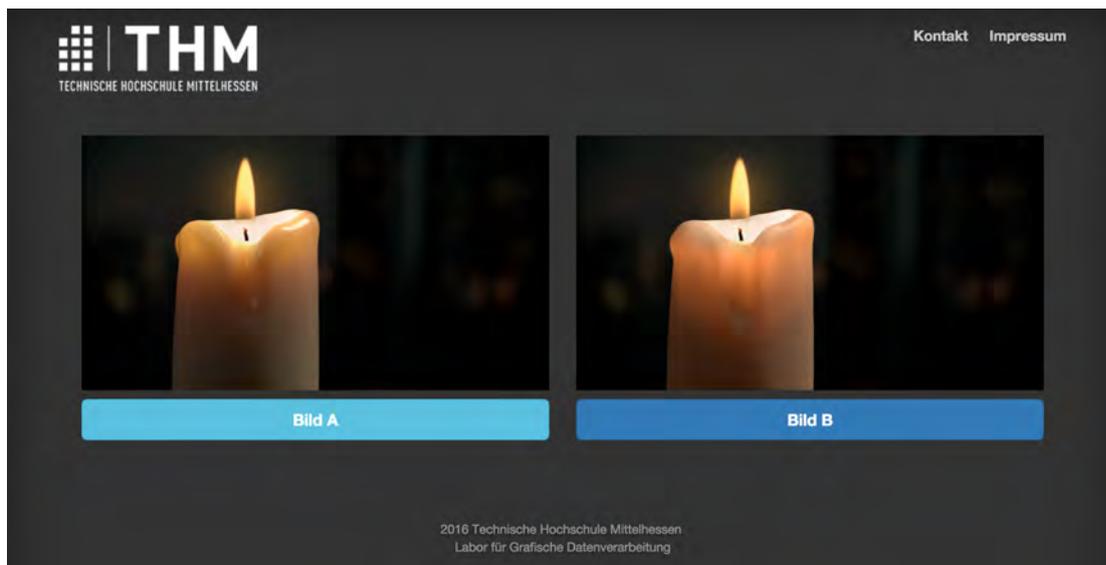


Abbildung 7.7: Oberfläche des Webportals für den "Compare"-Test.

## 7.3 Statistische Auswertung

### 7.3.1 Verwendete Software zur statistischen Analyse

Nach dem Export der in der Datenbank gesammelten Daten wurde das Datenhandling und die deskriptive Statistik in Excel durchgeführt. Die schließende Statistik (Teststatistik) zum Beantworten der aufgestellten Hypothesen wurde mit der Statistiksoftware Systat<sup>6</sup> durchgeführt [WBG96] [Sac13].

### 7.3.2 Erkennungsrate

Die Ergebnisse für die Evaluation sind keine messbaren Parameter, sondern subjektive Einschätzungen der Probanden. Menschliche Probanden verhalten sich oft aber nicht mathematisch logisch. So würde man in der Wahrscheinlichkeitsrechnung davon ausgehen, dass bei einem guten Rendering, das nicht mehr von einer Fotografie unterscheidbar ist, eine 50:50-Chance existiert, für welche der beiden *Buttons* "Fake oder Foto" sich ein Proband entscheidet. Es ist aber zu erwarten, dass sich die Testpersonen anders verhalten.

<sup>6</sup> <http://www.systat.de/>

## 7. EVALUATION

---

Es wird für diese Evaluation von der Annahme ausgegangen, dass die Testperson, wenn sie zu einer Entscheidung gezwungen wird (auf einen dritten *Button* für den Fall, dass sich ein Proband nicht entscheiden kann, wurde daher bewusst verzichtet), versuchen wird, einen Hinweis darauf zu finden, der verrät, dass es sich nicht um ein Foto handelt, sondern um ein am Computer erzeugtes Bild. Das bedeutet im Umkehrschluss, wenn der Proband diesen "Fehler" im Bild nicht findet, und damit das Rendering fotorealistisch genug ist, bei seiner Wahl zum Foto tendiert. Bei Richtigkeit dieser Annahme bedeutet das, dass die Erkennungsrate für die Fotos insgesamt besser ausfällt, als die für gute realistische 3D-Renderings, da hier, wenn kein "Fehler" entdeckt wird, eine falsche Entscheidung getroffen wird. Wichtig ist an dieser Stelle noch zu erwähnen, dass die beiden Kontrollbilder, die Kamera und die Schneeflocke (siehe Abbildung 7.4) für die statistischen Tests bezüglich der Erkennungsrate nicht berücksichtigt werden dürfen, da diese das Ergebnis verfälschen würden. Aufgrund der Tatsache, dass das Bild der Kamera beabsichtigt als sehr offensichtlich am Computer erzeugtes Bild und nicht als fotorealistisches Rendering der Testreihe hinzugefügt wurde, genauso wie die Schneeflocke als Foto, welches aber nicht eindeutig als Foto zu erkennen ist, gehören sie als Kontrollbilder nicht zur Fragestellung. Des Weiteren wurden die für diese Arbeit in den methodischen Kapiteln 4 und 5 erzeugten Bilder (Drache, Kerze, Kopf und Hand) für diesen Test ebenfalls nicht berücksichtigt, weil die Einordnung dieser Renderings bezüglich des Fotorealismus eben erst später im Verlauf dieses Kapitels überprüft werden soll. Alle anderen Renderings (abgesehen von dem Kontrollrendering) sind aufgrund ihrer Einstufung als fotorealistisch in die Liste der repräsentativen Auswahl aufgenommen worden.

### Hypothese

Die Erkennungsrate für die realen Fotos ist statistisch nachweisbar höher als die Erkennungsrate für am Computer erzeugte 3D-Renderings.

	ERKENNUNGSRATE
N of Cases	15
Arithmetic Mean	48,156
Standard Deviation	24,837

Abbildung 7.8: Systat-Datentabelle für die Gesamterkennungsrate.

## Results for STATUS\$ = CG

	ERKENNUNGSRATE
N of Cases	8
Arithmetic Mean	35,705
Standard Deviation	24,923

Abbildung 7.9: Systat-Tabelle für die Erkennungsrate der am Computer erzeugten Bilder.

## Results for STATUS\$ = IM

	ERKENNUNGSRATE
N of Cases	7
Arithmetic Mean	62,386
Standard Deviation	16,484

Abbildung 7.10: Systat-Tabelle für die Erkennungsrate der realen Fotos.

Den Tabellen ist mit 35,7% im arithmetischen Mittel für die Erkennungsrate bei mit Hilfe von Computergrafik erzeugten Bildern und 62,4% im arithmetischen Mittel für die Erkennungsrate bei realen Fotos eine deutliche Tendenz für eine höhere Erkennungsrate bei realen Fotos zu entnehmen.

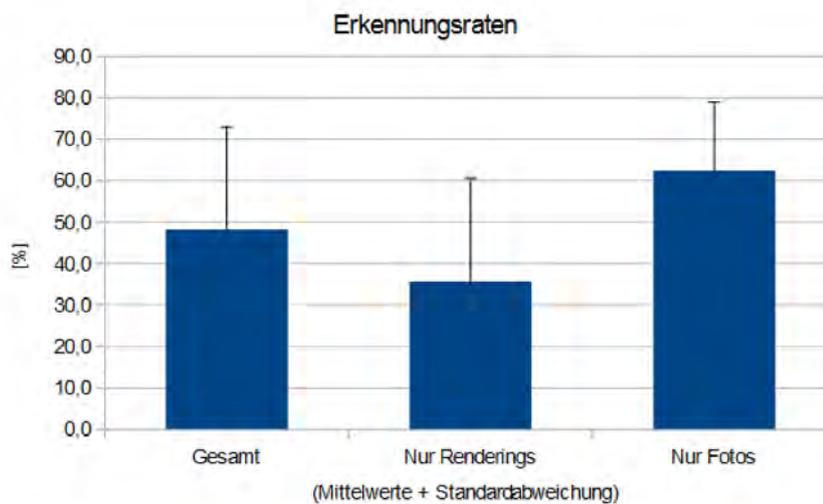


Abbildung 7.11: Grafische Visualisierung für die Ergebnisse der Erkennungsraten.

Da es sich um unverbundene Stichproben handelt und zuvor nicht auf Normalverteilung getestet wurde, wird der nicht-parametrische Kruskal-Wallis Test durchgeführt, um diese Tendenz statistisch zu überprüfen. Als Signifikanzniveau wird eine Irrtumswahrscheinlichkeit von 5% vorausgesetzt.

#### Kruskal-Wallis One-way Analysis of Variance for 49 Cases

Data for the following results were selected according to  
SELECT (SET = 3)

The categorical values encountered during processing are

Variables	Levels
STATUS\$ (2 levels)	CG IM

Dependent variable	ERKENNUNGSRATE
Grouping variable	STATUS\$

Group	Count	Rank Sum
CG	8	47,000
IM	7	73,000

Mann-Whitney U Test Statistic : 11,000  
p-value : 0,049  
Chi-square Approximation : 3,871  
df : 1

**Abbildung 7.12:** Systat-Auswertung des Kruskal-Wallis-Test auf signifikanten Unterschied zwischen den beiden Erkennungsraten der Computergrafik-Bilder und der realen Fotos.

#### Ergebnis

Mit einer Irrtumswahrscheinlichkeit von  $P=0,049$  oder 4,9% kann ein Unterschied in den Erkennungsraten zwischen den realen Fotos und den am Computer generierten Bildern statistisch nachgewiesen werden.

### 7.3.3 Gegenüberstellung V-RayMtl und FastSS2

Aus den Ergebnissen der prototypischen Umsetzungen aus Kapitel 4 leitet sich die Hypothese ab, dass der *FastSS2*-Shader für *Subsurface Scattering*-Visualisierungen qualitativ hochwertigere Ergebnisse in Bezug auf den Fotorealismus liefert als das *V-Ray Material*. Dies gilt es nun statistisch nachzuweisen. Getestet werden die beiden Bildpaare des Marmordrachen und der Kerze.

**Hypothese**

Der *FastSS2*-Shader liefert für das Marmordrachen-Rendering fotorealistischere Ergebnisse als das *V-Ray Material*.

Es wird getestet, ob das Merkmal "Shader" einen Einfluss auf die Entscheidung der Probanden bezüglich der Wahl "Foto oder CG" hat. Als statistischer Test dafür dient der  $\chi^2$ -Vierfeldertest. Da dieser Test verteilungsfrei ist, müssen die Daten zuvor nicht auf Normalverteilung getestet werden.

**Counts****CG\_DRACHE\$(rows) by SET(columns)**

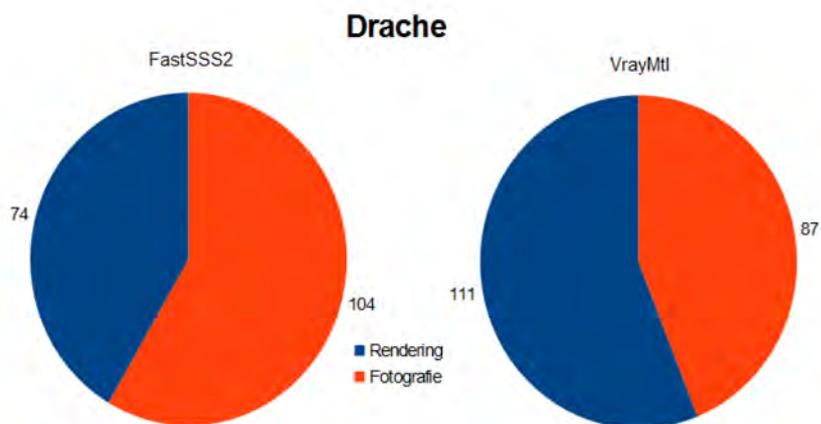
	1	2	Total
FAKE	74	111	185
FOTO	104	87	191
Total	178	198	376

**Percents of Total Count****CG\_DRACHE\$(rows) by SET(columns)**

	1	2	Total	N
FAKE	19,681	29,521	49,202	185,000
FOTO	27,660	23,138	50,798	191,000
Total	47,340	52,660	100,000	
N	178,000	198,000		376,000

**Abbildung 7.13:** Systat Tabellen für den  $\chi^2$ -Vierfelder-Test mit den Merkmalen Marmordrachen-Rendering *FastSS2*-Shader/Marmordrachen-Rendering *V-Ray Material* und *Fake/Foto*.

Wie an den Tabellen abzulesen ist, haben sich mit 27,7% für Foto gegen 19,7% für *Fake* bei dem *FastSS2*-Shader in *Set 1* und 23,1% für Foto gegen 29,5% für *Fake* beim *V-Ray Material* in *Set 2* deutlich mehr Probanden für Foto bei bei dem Marmordrachen-Rendering mit dem *FastSS2*-Shader entschieden.



**Abbildung 7.14:** Visualisierung des Ergebnisses des “Fake oder Foto”-Tests für die Marmordrachen-Renderings.

**Chi-Square Tests of Association for CG\_DRACHE\$ and SET**

Test Statistic	Value	df	p-value
<b>Pearson Chi-square</b>	7,872	1,000	0,005

Number of Valid Cases: 376

**Abbildung 7.15:** Systat-Ergebnis für die Teststatistik dem  $\chi^2$ -Vierfeldertest für den Marmordrachen.

**Ergebnis**

Der  $\chi^2$ -Vierfeldertest unterstützt die Hypothese und zeigt einen signifikanten Einfluss des Shaders auf die Entscheidung “Fake oder Foto” für das Marmordrachen-Rendering nicht am Signifikanzniveau von 5% sondern sogar bei einer Irrtumswahrscheinlichkeit von  $P=0,005$  oder 0,5%.

**Hypothese**

Der *FastSS2*-Shader liefert fotorealistischere Ergebnisse für das Rendering der Kerze als das *V-Ray Material*.

Getestet wird auch für das Rendering der Kerze, ob das Merkmal “Shader” einen Einfluss auf Entscheidung “Foto oder CG” hat.

CG\_KERZE\$(rows) by SET(columns)

	1	2	Total
FAKE	79	83	162
FOTO	99	115	214
Total	178	198	376

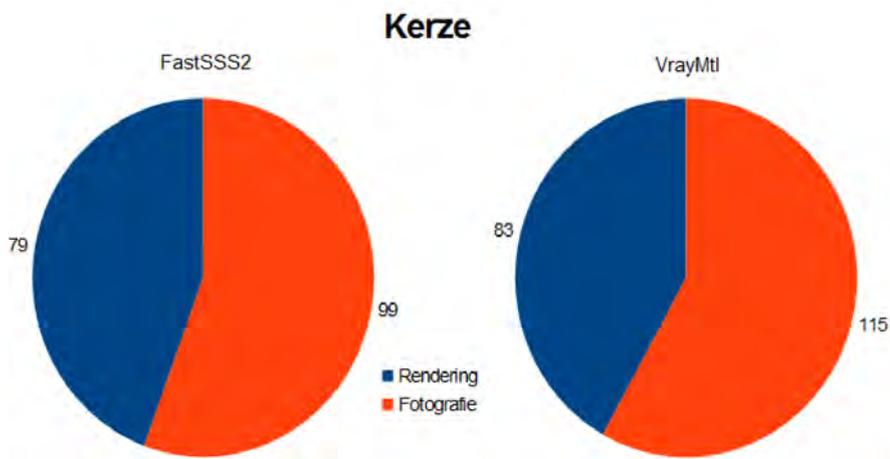
Percents of Total Count

CG\_KERZE\$(rows) by SET(columns)

	1	2	Total	N
FAKE	21,011	22,074	43,085	162,000
FOTO	26,330	30,585	56,915	214,000
Total	47,340	52,660	100,000	
N	178,000	198,000		376,000

**Abbildung 7.16:** Systat Tabellen für den  $\chi^2$ -Vierfeldertest mit den Merkmalen Kerzen-Rendering FastSS2-Shader/Kerzen-Rendering V-Ray Material und Fake/Foto.

An den Tabellen ist abzulesen, dass der Anteil der Probanden, die sich für Foto entschieden haben, für das Kerzen Rendering in beiden Shader-Varianten größer ist als der Anteil derer, die sich für *Fake* entschieden haben. Für den *FastSS2*-Shader liegt der Anteil bei 26,3% für Foto gegen 21,0% für *Fake*. Für das *V-Ray Material* liegt der Anteil bei 30,6% für Foto und 22,1% für *Fake*.



**Abbildung 7.17:** Visualisierung des Ergebnisses des “Fake oder Foto”-Tests für die Kerzen-Renderings.

**Chi-Square Tests of Association for CG\_KERZE\$ and SET**

Test Statistic	Value	df	p-value
Pearson Chi-square	0,232	1,000	0,630

Number of Valid Cases: 376

**Abbildung 7.18:** Systat-Ergebnis für die Teststatistik dem  $\chi^2$ -Vierfeldertest für die Kerze.

**Ergebnis**

Der  $\chi^2$ -Vierfeldertest unterstützt die Hypothese nicht. Ein signifikanter Einfluss des Shaders auf die Entscheidung "Fake oder Foto" kann bei einer Irrtumswahrscheinlichkeit von  $P=0,63$  oder 63% nicht nachgewiesen werden.

### 7.3.4 Einfluss der CG-Vorkenntnisse

Getestet wurde auch, ob die Computergrafik-Vorkenntnisse der Probanden einen Einfluss auf die Entscheidung "Fake oder Foto" haben.

**Hypothese**

Vorhandene Vorkenntnisse in der Computergrafik haben einen Einfluss auf die Entscheidung "Fake oder Foto" für das Marmordrachen-Rendering in der *FastSSS2*-Shader-Variante.

Es wird getestet, ob das Merkmal Computergrafik-Vorkenntnisse einen Einfluss auf die Entscheidung "Fake oder Foto" der Probanden hat. Dafür wird der verteilungsfreie  $\chi^2$ -Vierfeldertest durchgeführt.

## Counts

## CG\_DRACHE\$(rows) by CG(columns)

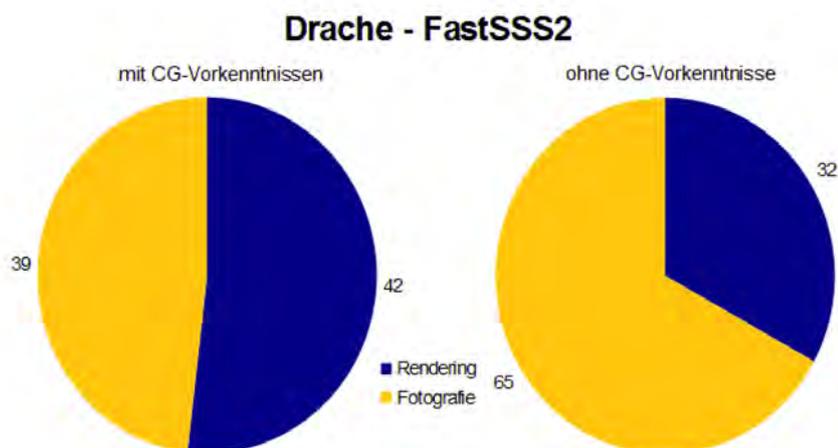
	0	1	Total
FAKE	32	42	74
FOTO	65	39	104
Total	97	81	178

## Percents of Total Count

## CG\_DRACHE\$(rows) by CG(columns)

	0	1	Total	N
FAKE	17,978	23,596	41,573	74,000
FOTO	36,517	21,910	58,427	104,000
Total	54,494	45,506	100,000	
N	97,000	81,000		178,000

**Abbildung 7.19:** Sysstat-Tabellen für den  $\chi^2$ -Vierfeldertest mit den Merkmalen mit/ohne "Computergrafik-Vorkenntnissen" und *Fake/Foto* für die *FastSS2*-Shader-Variante des Marmordrachsens.



**Abbildung 7.20:** Visualisierung der Ergebnisse des Tests bezüglich des Einflusses der Computergrafik-Vorkenntnisse.

## Chi-Square Tests of Association for CG\_DRACHE\$ and CG

Test Statistic	Value	df	p-value
Pearson Chi-square	6,465	1,000	0,011

Number of Valid Cases: 178

**Abbildung 7.21:** Systat Ergebnis für die Teststatistik dem  $\chi^2$ -Vierfeldertest für die Computergrafik-Vorkenntnisse beim Marmordrachen-Rendering in der *FastSS2*-Shader-Variante.

**Ergebnis**

Durch das Ergebnis des  $\chi^2$ -Vierfeldertests wird die Hypothese unterstützt, dass die Computergrafik-Vorkenntnisse einen signifikanten Einfluss auf die Entscheidung "Fake oder Foto" für das Marmordrachen-Rendering in der *FastSS2*- Shader-Variante hat. Dies kann bei einer Irrtumswahrscheinlichkeit von  $P=0,011$  oder 1,1% nachgewiesen werden.

**7.3.5 Gegenüberstellung FastSS2 und Skin Material**

Aus den Ergebnissen den prototypischen Umsetzungen aus Kapitel 5 leitet sich die Hypothese ab, dass der *FastSS2*-Shader für *Subsurface Scattering*-Visualisierungen menschlicher Haut qualitativ hochwertigere Ergebnisse liefert als das *Skin Material*. Dies gilt es nun statistisch nachzuweisen. Getestet werden die beiden Bildpaare des Emily-Renderings und der Hand. Für das *Skin Material* wurden für die Evaluation die Bilder verwendet, bei denen die entwickelte Texturmethode angewendet wurde.

**Hypothese**

Der *FastSS2*-Shader liefert für das Emily-Rendering fotorealistischere Ergebnisse als das *Skin Material*.

Es wird getestet, ob das Merkmal "Shader" einen Einfluss auf die Entscheidung der Probanden bezüglich der Wahl "Foto oder CG" hat. Dafür wird ein  $\chi^2$ -Vierfeldertest durchgeführt. Auch hier gilt wieder, dass da dieser Test verteilungsfrei ist, die Daten zuvor nicht auf Normalverteilung getestet werden müssen.

CG\_KOPF\$(rows) by SET(columns)

	1	2	Total
FAKE	97	135	232
FOTO	81	63	144
Total	178	198	376

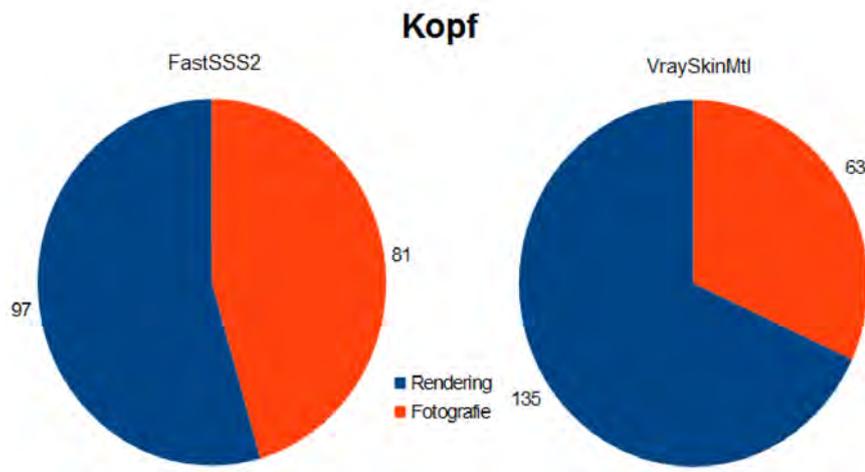
Percents of Total Count

CG\_KOPF\$(rows) by SET(columns)

	1	2	Total	N
FAKE	25,798	35,904	61,702	232,000
FOTO	21,543	16,755	38,298	144,000
Total	47,340	52,660	100,000	
N	178,000	198,000		376,000

**Abbildung 7.22:** Systat-Datentabelle für den  $\chi^2$ -Viefeldertest für die Merkmale Shader-Variante und “Fake oder Foto” für das Emily-Rendering.

Aus den Tabellen lässt sich ablesen, dass sich für beide Varianten des Shaders eine Mehrheit für *Fake* entschieden hat. Der Anteil für den *FastSS2* Shader liegt aber mit 97 zu 81 Probanden deutlich näher an einem 50:50-Ergebnis und bei dem *Skin Material* mit 135 zu 63 bei einer deutlich stärkeren Tendenz zu *Fake* als zu *Foto*. In Prozenten bedeutet das 25,8% für *Fake* für den *FastSS2*-Shader und 35,9 % für *Fake* für das *Skin Material*.



**Abbildung 7.23:** Visualisierung des Ergebnisses des “Fake oder Foto”-Tests für die Kopf-Renderings.

**Chi-Square Tests of Association for CG\_KOPF\$ and SET**

Test Statistic	Value	df	p-value
Pearson Chi-square	7,431	1,000	0,006

Number of Valid Cases: 376

**Abbildung 7.24:** Systat-Ergebnistabelle für den  $\chi^2$ -Vierfeldertest für die Merkmale Shader-Variante und *Fake oder Foto* für das Emily-Rendering.

**Ergebnis**

Der  $\chi^2$ -Vierfeldertest unterstützt die Hypothese und zeigt einen signifikanten Einfluss des Shaders auf die Entscheidung "Fake oder Foto" für das Emily-Rendering bei einer Irrtumswahrscheinlichkeit von  $P=0,006$  oder 0,6%.

**Hypothese**

Der *FastSSS2*-Shader liefert für das Hand-Rendering fotorealistischere Ergebnisse als das *Skin Material*.

Es wird getestet, ob das Merkmal "Shader" einen Einfluss auf die Entscheidung der Probanden bezüglich der Wahl "Foto oder CG" hat. Dafür wird ein  $\chi^2$ -Vierfeldertest durchgeführt. Da dieser Test verteilungsfrei ist, müssen die Daten zuvor nicht auf Normalverteilung getestet werden.

## Counts

## CG\_HAND\$(rows) by SET(columns)

	1	2	Total
FAKE	146	173	319
FOTO	32	25	57
Total	178	198	376

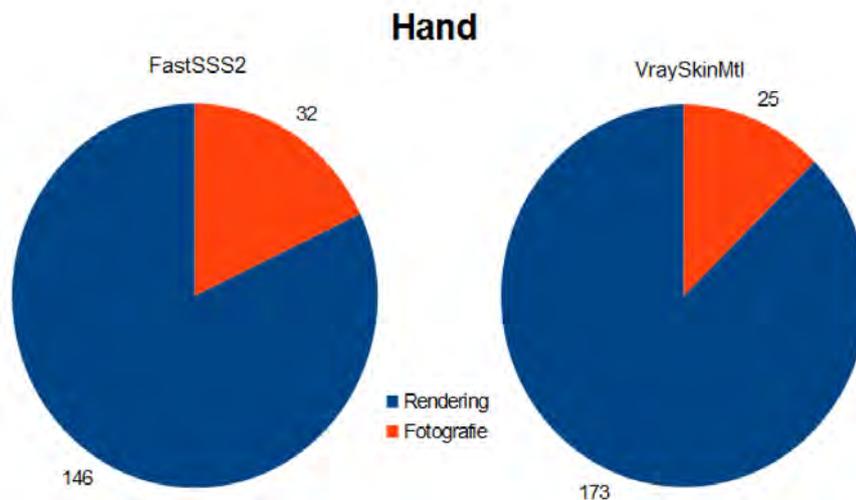
## Percents of Total Count

## CG\_HAND\$(rows) by SET(columns)

	1	2	Total	N
FAKE	38,830	46,011	84,840	319,000
FOTO	8,511	6,649	15,160	57,000
Total	47,340	52,660	100,000	
N	178,000	198,000		376,000

**Abbildung 7.25:** Systat-Datentabelle für den  $\chi^2$ -Viefeldertest für die Merkmale Shader-Variante und *Fake oder Foto* für das Hand-Rendering.

Die Tabelle zeigt für beide Shader eine deutliche Mehrheit für *Fake* bei beiden Shadern. Diese liegt für den *FastSSSS2*-Shader bei 38,8 % und für das *Skin Material* bei 46,0%.



**Abbildung 7.26:** Visualisierung des Ergebnisses des "Fake oder Foto"-Tests für die Hand-Renderings.

## Chi-Square Tests of Association for CG\_HAND\$ and SET

Test Statistic	Value	df	p-value
Pearson Chi-square	2,087	1,000	0,149

Number of Valid Cases: 376

**Abbildung 7.27:** Systat Ergebnistabelle für den Vierfelder- $\chi^2$ -Test für die Merkmale Shader-Variante und *Fake oder Foto* für das Hand-Rendering.

**Ergebnis**

Der  $\chi^2$ -Vierfeldertest unterstützt die Hypothese nicht. Ein signifikanter Einfluss des Shaders auf die Entscheidung "Fake oder Foto" für das Hand-Rendering am Signifikanzniveau von 5% kann nicht nachgewiesen werden. Die Irrtumswahrscheinlichkeit liegt bei  $P=0,149$  oder 14,9%.

**7.3.6 Test auf Fotorealismus**

Für die Prototypen, bei denen sich eine Mehrheit der Probanden für Foto entschieden hat, soll nun getestet werden, ob sie auch im Vergleich zu einem realen Foto bestehen können.

**Hypothese**

Für die Entscheidung der Probanden bezüglich der Wahl "Fake oder Foto" hat die Tatsache, ob es sich bei dem Bild um das Marmordrachen-Rendering oder das reale Pasta-Foto handelt, keinen Einfluss.

Getestet wird hier zunächst punktuell das im vorherigen Test besser bewertete Marmordrachen-Rendering in der *FastSSS2*-Shader-Variante gegen das Pasta-Foto. Das Pasta-Foto wurde ausgewählt, da es eine typische *Food*-Fotografie ist, die durch ihre sehr leuchtenden grellen Farben etwas nachbearbeitet wirkt, aber doch schnell als Foto zu identifizieren ist. Es soll durch den Test der Frage auf den Grund gegangen werden, ob es für die Entscheidung *Fake* oder *Foto* einen Unterschied macht, ob es sich bei dem getesteten Bild, um ein reales Foto oder ein fotorealistisches Rendering handelt.

## Counts

## WAHLS(rows) by BILD\$(columns)

	DRACHE	PASTA	Total
FAKE	74	70	144
FOTO	104	108	212
Total	178	178	356

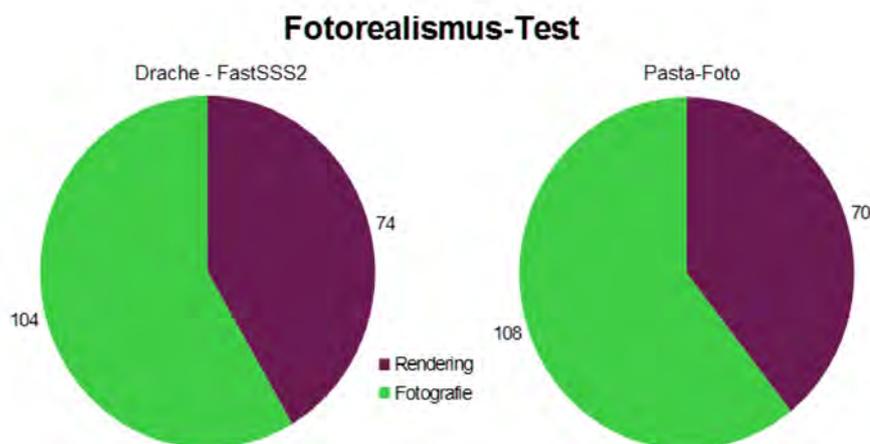
## Percents of Total Count

## WAHLS(rows) by BILD\$(columns)

	DRACHE	PASTA	Total	N
FAKE	20,787	19,663	40,449	144,000
FOTO	29,213	30,337	59,551	212,000
Total	50,000	50,000	100,000	
N	178,000	178,000		356,000

**Abbildung 7.28:** Systat-Datentabelle für den  $\chi^2$ -Vierfeldertest bezüglich der Merkmale Marmordrachen-Rendering und Pasta-Foto und die Entscheidung für "Fake oder Foto."

In der Tabelle ist zu erkennen, dass sich die Werte mit 74 Entscheidungen für *Fake* für das Rendering und 70 für das Foto sowie 104 Entscheidungen für Foto für das Rendering und 108 für das reale Foto sehr nahe kommen. Das spiegeln natürlich auch die Prozentanteile mit 20,8 % für *Fake* und 29,2% für Foto für das Marmordrachen-Rendering und 19,7% für *Fake* sowie 30,3% für Foto für das Pasta-Foto wider.



**Abbildung 7.29:** Visualisierung des Ergebnisses des Tests auf Fotorealismus bei Vergleich des Marmordrachen-Renderings in der *FastSS2*-Variante mit einem Foto.

**Chi-Square Tests of Association for WAHL\$ and BILD\$**

Test Statistic	Value	df	p-value
Pearson Chi-square	0,187	1,000	0,666

Number of Valid Cases: 356

**Abbildung 7.30:** Sytat  $\chi^2$ -Vierfeldertest-Ergebnis bezüglich der Merkmale Marmordrachen-Rendering und Pasta-Foto und die Entscheidung für *Fake* oder Foto.

**Ergebnis**

Der  $\chi^2$ -Vierfeldertest, der den Einfluss des Merkmals Marmordrachen-Rendering oder Pasta-Foto auf die Entscheidung *Fake* oder Foto getestet hat, zeigt, dass bei einer Irrtumswahrscheinlichkeit von  $P=0,666$  oder 66,6% kein signifikanter Einfluss dieser Merkmale auf Entscheidung besteht und unterstützt damit die aufgestellte Hypothese.

Um dieses Ergebnis noch weiter zu verifizieren, wurde der gleiche Test noch einmal mit drei Fotos gleichzeitig gemacht, um zu überprüfen, ob das Rendering auch gegen mehrere Bilder bestehen kann.

**Hypothese**

Für die Entscheidung der Probanden bezüglich der Wahl "Fake oder Foto" hat die Tatsache, ob es sich bei dem Bild um das Marmordrachen-Rendering oder eines von drei realen Fotos handelt, keinen Einfluss.

## Counts

## WAHL\$(rows) by BILD\$(columns)

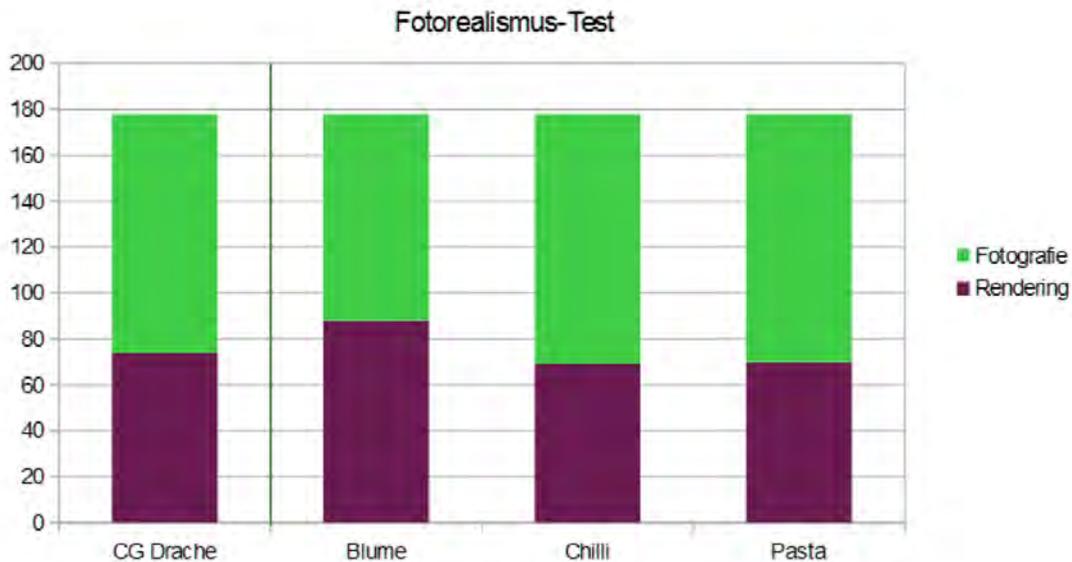
	CG_Drache	IM_Blume	IM_Chilli	IM_Pasta	Total
FAKE	74	88	69	70	301
FOTO	104	90	109	108	411
Total	178	178	178	178	712

## Percents of Total Count

## WAHL\$(rows) by BILD\$(columns)

	CG_Drache	IM_Blume	IM_Chilli	IM_Pasta	Total	N
FAKE	10,393	12,360	9,691	9,831	42,275	301,000
FOTO	14,607	12,640	15,309	15,169	57,725	411,000
Total	25,000	25,000	25,000	25,000	100,000	712,000
N	178,000	178,000	178,000	178,000		712,000

**Abbildung 7.31:** Systat-Datentabelle für den  $\chi^2$ -Vierfeldertest-Ergebnis bezüglich der Merkmale Marmordrachen-Rendering und Pasta-Foto und die Entscheidung für *Fake* oder *Foto*.



**Abbildung 7.32:** Visualisierung des Ergebnisses des Tests auf Fotorealismus bei Vergleich des Marmordrachen-Renderings in der *FastSSS2*-Variante mit drei Fotos.

## Chi-Square Tests of Association for WAHL\$ and BILD\$

Test Statistic	Value	df	p-value
Pearson Chi-square	5,312	3,000	0,150

Number of Valid Cases: 712

**Abbildung 7.33:** Systat-Ergebnis für den  $\chi^2$ -Mehrfeldertest bezüglich der Merkmale Marmordrachen-Rendering und drei reale Fotos und die Entscheidung für *Fake* oder Foto.

**Ergebnis**

Der  $\chi^2$ -Mehrfeldertest, der den Einfluss des Merkmals Marmordrachen-Rendering oder eines der drei realen Fotos auf die Entscheidung *Fake* oder Foto getestet hat, zeigt, dass bei einer Irrtumswahrscheinlichkeit von  $P=0,150$  oder 15% kein signifikanter Einfluss dieser Merkmale auf Entscheidung besteht und unterstützt damit die aufgestellte Hypothese.

Getestet wurden außerdem: Das Rendering der Kerze in der *FastSSS2*-Shader-Variante im Mehrfeldertest auf Fotorealismus gegen drei Fotos mit dem Ergebnis, dass bei einer Irrtumswahrscheinlichkeit von  $P=0,143$  oder 14,3% kein signifikanter Unterschied zwischen dem Rendering und den Fotos besteht und das Rendering damit nachweisbar fotorealistisch ist.

Das Rendering der Kerze in der *V-Ray Material*-Variante im Mehrfeldertest auf Fotorealismus gegen drei Fotos mit dem Ergebnis, dass bei einer Irrtumswahrscheinlichkeit von  $P=0,141$  oder 14,1% kein signifikanter Unterschied zwischen dem Rendering und den Fotos besteht und das Rendering damit nachweisbar fotorealistisch ist.

Das Rendering des Kopfes im Mehrfeldertest auf Fotorealismus gegen drei Fotos mit dem Ergebnis, dass bei einer Irrtumswahrscheinlichkeit von  $P=0,005$  oder 0,5% ein signifikanter Unterschied zwischen dem Rendering und den Fotos besteht und das Rendering damit nachweisbar nicht fotorealistisch ist.

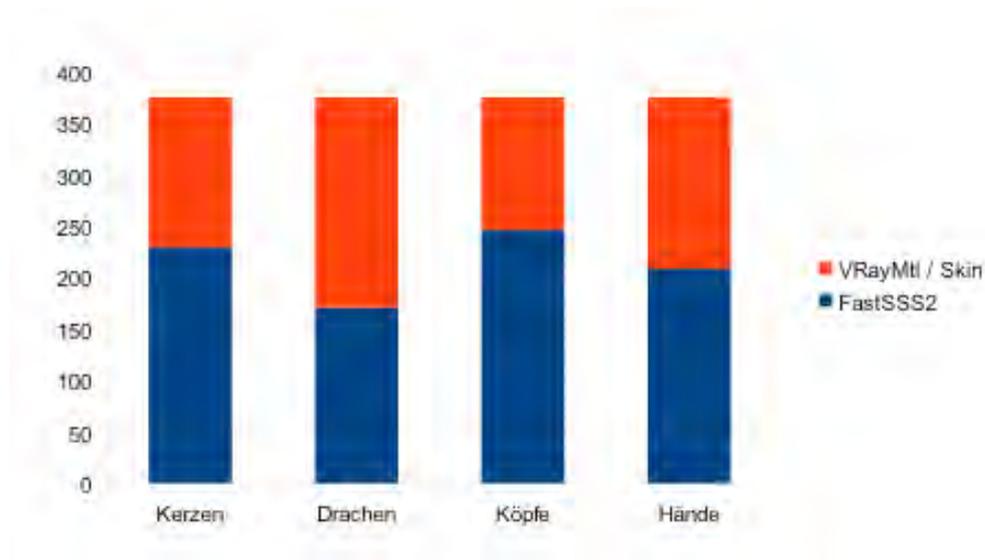
Der Einfluss des Geschlechts auf die Entscheidung *Fake* oder Foto für alle Bilder in der *FastSSS2*-Variante mit dem Ergebnis, dass bei allen Bildern mehr Frauen zu Foto tendieren als Männer.

Der Einfluss der Computergrafik-Vorkenntnisse wurde außer für das Drachen-Rendering auch noch für die anderen Renderings in der *FastSSS2*-Variante getestet, mit dem Ergebnis, dass sich immer mehr Probanden mit, als ohne CG-Vorkenntnisse für Foto entscheiden.

Alle Systat-Tabellen zu diesen Tests finden sich im Anhang.

### 7.3.7 Direkter Vergleich

Im zweiten Test der Evaluation war den Probanden nun bekannt, dass es sich bei den Bildern, die sie gezeigt bekommen, ausschließlich um Renderings handelt. Ihnen wurden jeweils zum direkten Vergleich die Renderings in den unterschiedlichen Shader-Varianten *FastSSS2* und *V-Ray Material* beziehungsweise *FastSSS2* und *Skin Material* gezeigt und sie sollten eine Entscheidung treffen, welches der beiden Bilder sie für realistischer halten.



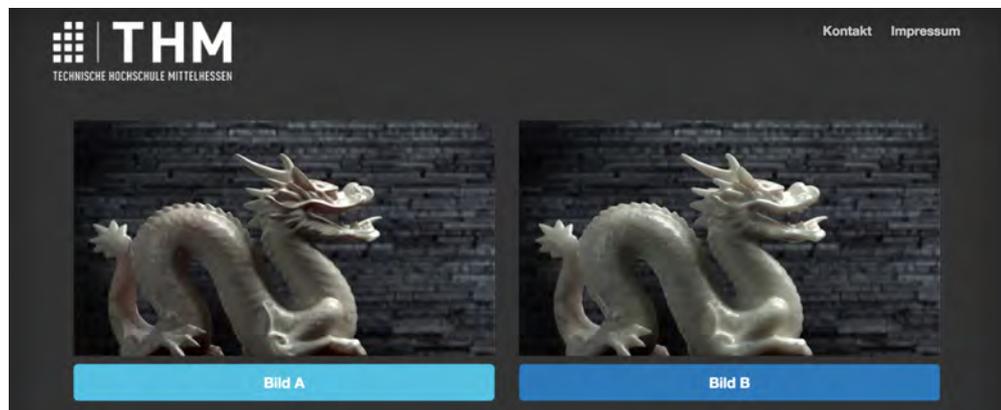
**Abbildung 7.34:** Grafische Darstellung der vier direkten Vergleichstest für die Renderings der Kerzen, der Drachen, der Köpfe und der Hände. Dabei repräsentieren die blauen Teil der Balken jeweils den Anteil der Probanden, die sich für die *FastSSS2*-Shader-Variante als das realistischere Bild entschieden haben und der rote Teil des Balkens den Anteil der Probanden, die sich für die *V-Ray Material*- bzw für die *SkinMtl*-Variante als das realistische Bild entschieden haben.

Die folgenden Tests wurden der Einfachheit halber manuell in Excel berechnet. Da die Anzahl der Testdurchführungen deutlich über 40 lag wurde für die Berechnung folgende Formel verwendet:

$$\hat{z} = \frac{x_1 - x_2}{\sqrt{x_1 + x_2}}$$

Dabei ist  $x_1$  die größere und  $x_2$  die kleinere der beiden Häufigkeiten,  $\hat{z}$  ist der Testwert, welcher mit der kritischen Z-Schranke verglichen wird. Der Wert für die kritische Z-Schranke liegt für Normalverteilung bei 5% Signifikanz-Niveau bei  $Z_{\text{krit}}=1,96$ . [Sac70]

### 7.3.8 Drachen



**Abbildung 7.35:** Direkter Vergleichstest der Marmordrachen-Renderings in beiden Shader-Varianten.

Die Testpersonen haben das Marmordrachen-Rendering in direkter Gegenüberstellung in der *FastSSS2* und in der *V-Ray Material*-Variante gezeigt bekommen und sollten sich für das realistischere der beiden Bilder entscheiden.

#### Hypothese

Im direkten Vergleich liefert der *FastSSS2*-Shader für das Marmordrachen-Rendering fotorealistischere Ergebnisse als das *V-Ray Material*.

Bei 376 Testläufen ergaben sich dabei 171 Wertungen für die *FastSSS2*-Variante und 205 Wertungen für die *V-Ray Material*-Variante als besseres oder fotorealistischeres Ergebnis.

Aus diesen Daten lässt sich die Vermutung ableiten, dass die *V-Ray Material* fotorealistischere Ergebnisse erzielt.

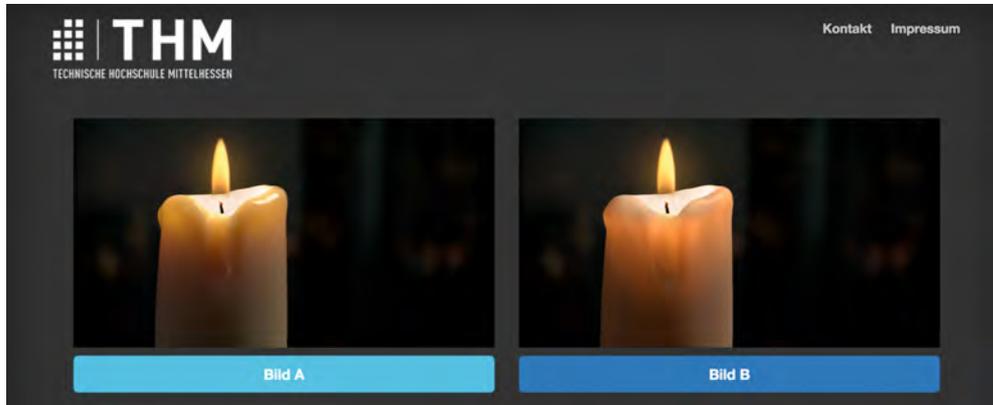
Getestet wird mit einem "Vergleich zweier Häufigkeiten" [Sac70] am 5%-Signifikanzniveau.

Der Test berechnet den Testwert  $\hat{z} = 1,75$ . Da  $\hat{z} = 1,75 < 1,96 = z_{\text{krit}}(1)$ , kann die Nullhypothese  $H_0$  (Gleichheit der Parameter, die für das Ergebnis verantwortlich sind) nicht verworfen werden.

#### Ergebnis

Ein signifikanter Unterschied zwischen den beiden Shadern kann also nicht sicher nachgewiesen werden. Die Hypothese gilt damit als nicht statistisch belegt.

## 7.3.9 Kerzen



**Abbildung 7.36:** Direkter Vergleichstest der Kerzen-Renderings in beiden Shader-Varianten.

Die Testpersonen haben das Kerzen-Rendering in direkter Gegenüberstellung in der *FastSS2* und in der *V-Ray Material*-Variante gezeigt bekommen und sollten sich für das realistischere der beiden Bilder entscheiden.

#### Hypothese

Im direkten Vergleich liefert der *FastSS2*-Shader für das Kerzen-Rendering fotorealistischere Ergebnisse als das *V-Ray Material*.

Bei 376 Testläufen ergaben sich dabei 230 Wertungen für die *FastSS2*-Variante und 146 Wertungen für die *V-Ray Material*-Variante als besseres oder fotorealistischeres Ergebnis.

Aus diesen Daten lässt sich die Vermutung ableiten, dass der *FastSS2*-Shader fotorealistischere Ergebnisse erzielt.

Getestet wird mit einem "Vergleich zweier Häufigkeiten" [Sac70] am 5%-Signifikanzniveau.

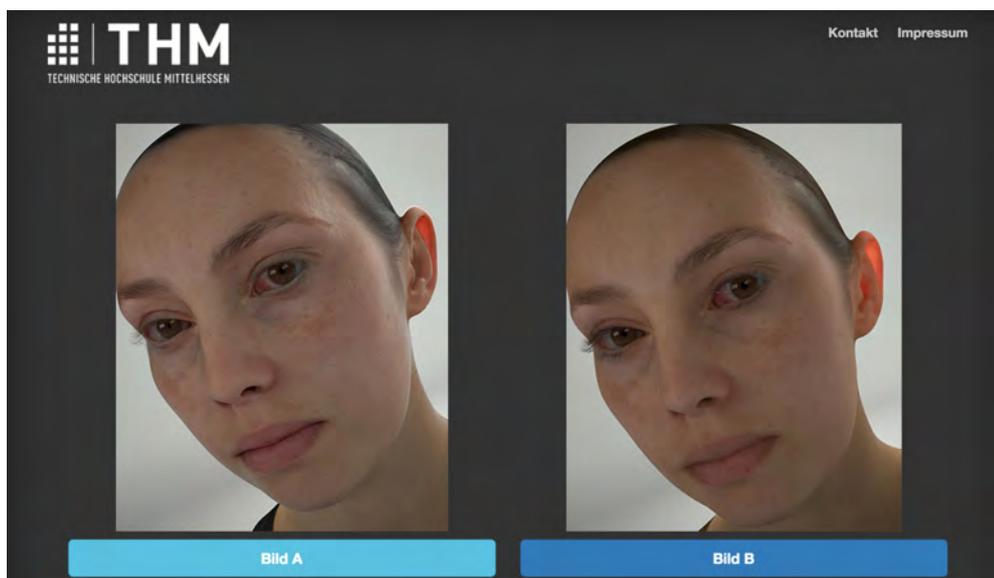
Der Test berechnet den Testwert  $\hat{z} = 4,33$ . Da  $\hat{z} = 4,33 > 1,96 = z_{\text{krit}}(1)$ , muss die Nullhypothese  $H_0$  (Gleichheit der Parameter, die für das Ergebnis verantwortlich sind) verworfen werden.

Das gleiche Ergebnis ergibt sich sogar für eine Signifikanzschranke von 0,1%. Hier ist der entsprechende kritische Z-Wert 3,291.

### Ergebnis

Es besteht bei einer Irrtumswahrscheinlichkeit von unter 0,1% ein signifikanter Unterschied zwischen den beiden Shadern. Damit wird die aufgestellte Hypothese unterstützt und das Rendering der Kerze mit dem *FastSS2*-Shader nachweisbar besser als das mit *V-Ray Material*.

### 7.3.10 Emily-Kopf



**Abbildung 7.37:** Direkter Vergleichstest der Kopf-Renderings in beiden Shader-Varianten.

Die Testpersonen haben das Kopf-Rendering in direkter Gegenüberstellung in der *FastSS2* und in der *SkinMtl*-Variante gezeigt bekommen und sollten sich für das realistischere der beiden Bilder entscheiden.

### Hypothese

Im direkten Vergleich liefert der *FastSS2*-Shader für das Kopf-Rendering der Emily fotorealistischere Ergebnisse als das *VRay Skin*-Material.

Bei 376 Testläufen ergaben sich dabei 247 Wertungen für die *FastSS2*-Variante und 129 Wertungen für die *V-Ray Skin Material*-Variante als besseres oder fotorealistischeres Ergebnis.

Aus diesen Daten lässt sich die Vermutung ableiten, dass der *FastSS2-Shader* fotorealistischere Ergebnisse erzielt.

Getestet wird mit einem "Vergleich zweier Häufigkeiten" [Sac70] am 5%-Signifikanzniveau.

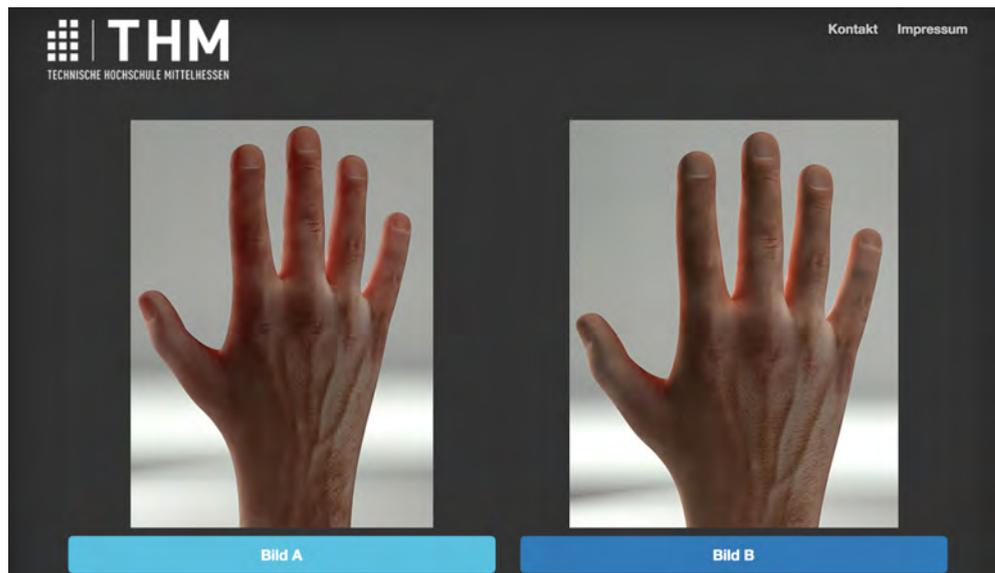
Der Test berechnet den Testwert  $\hat{z} = 6,09$ . Da  $\hat{z} = 6,09 > 1,96 = z_{\text{krit}}(1)$ , muss die Nullhypothese  $H_0$  (Gleichheit der Parameter, die für das Ergebnis verantwortlich sind) verworfen werden.

Das gleiche Ergebnis ergibt sich sogar für eine Signifikanzschranke von 0,1%. Hier ist der entsprechende kritische Z-Wert 3,291.

### Ergebnis

Es besteht also bei einer Irrtumswahrscheinlichkeit von 0,1% ein signifikanter Unterschied zwischen den beiden Shadern. Das Rendering der Emily mit dem *FastSSS2*-Shader ist nachweisbar besser als das mit *V-Ray Skin Material*. Damit wird die zuvor aufgestellte Hypothese unterstützt.

### 7.3.11 Hand



**Abbildung 7.38:** Direkter Vergleichstest der Hand-Renderings in beiden Shader-Varianten.

Die Testpersonen haben das Hand-Rendering in direkter Gegenüberstellung in der *FastSSS2*- und in der *Skin Material*-Variante gezeigt bekommen und sollten sich für das realistischere der beiden Bilder entscheiden.

### Hypothese

Im direkten Vergleich liefert der *FastSS2*-Shader für das Hand-Rendering fotorealistischere Ergebnisse als das *V-Ray Skin Material*.

Bei 376 Testläufen ergaben sich dabei 209 Wertungen für die *FastSS2*-Variante und 167 Wertungen für die *V-Ray Skin Material*-Variante als besseres oder fotorealistischeres Ergebnis.

Aus diesen Daten lässt sich die Vermutung ableiten, dass der *FastSS2*-Shader fotorealistischere Ergebnisse erzielt.

Getestet wird mit einem "Vergleich zweier Häufigkeiten" [Sac70] am 5%-Signifikanzniveau.

Der Test berechnet die Testwert  $\hat{z} = 2,17$ . Da  $\hat{z} = 2,17 > 1,96 = z_{\text{krit}}(1)$ , muss die Nullhypothese  $H_0$  (Gleichheit der Parameter, die für das Ergebnis verantwortlich sind) verworfen werden.

### Ergebnis

Es besteht bei einer Irrtumswahrscheinlichkeit von unter 5% ein signifikanter Unterschied zwischen den beiden Shadern. Das Rendering der Hand mit dem *FastSS2*-Shader ist nachweisbar besser als das mit *V-Ray Skin Material*.

## 7.4 Zusammenfassung und Interpretation der Ergebnisse

Mit Hilfe einer Online-Evaluation wurden in 376 Testdurchläufen in zwei unterschiedlichen Testreihen, einem "Fake oder Foto"- Test und einem Vergleichstest, Daten gesammelt, die es ermöglicht haben, aus den subjektiven Parametern möglichst objektiv bewertbare Teststatistiken zu generieren. Zunächst wurde eine für die Evaluation grundlegende Annahme bestätigt, die besagt, dass ein Proband, der zu einer Entscheidung gezwungen wird, ob es sich bei einem Bild um ein reales Foto oder ein am Computer generiertes Bild handelt, sich immer dann für Foto entscheidet, wenn er in einem realistischen Bild keinen "Fehler" entdeckt, der ihm einen Hinweis darauf gibt, dass es sich nicht um ein reales Foto handelt. Mithilfe der Erkennungsraten, die statistisch nachweisbar besser bei den realen Fotos sind als bei den am Computer generierten Bildern, konnte diese Annahme belegt werden.

Für das Marmordrachen-Rendering konnte durch den  $\chi^2$ -Vierfeldertest statistisch nachgewiesen werden, dass sich die Probanden bei der *FastSS2*-Shader-Variante häufiger für Foto entscheiden als bei der *V-Ray Material*-Variante und damit, dass in Bezug auf diesen Prototypen, der *FastSS2*-Shader die besseren Ergebnisse liefert. Beim Vergleichstest konnte dies im direkten Vergleich der beiden Bilder in den unterschiedlichen Varianten nicht bestätigt werden hier gab es für diesen Prototyp in den beiden Varianten keinen signifikanten Unterschied am 5%-Niveau.

Für das Kerzen-Rendering konnte durch den  $\chi^2$ -Vierfeldertest kein signifikanter Unterschied der beiden Shader-Varianten in Bezug auf die Entscheidung der Probanden auf die Wahl, ob es sich um ein Foto oder ein am Computer generiertes Rendering handelt, nachgewiesen werden. Für beide Varianten besteht eine deutliche Tendenz zu der Entscheidung Foto. Im direkten Vergleich bei beiden Varianten konnte dann jedoch signifikant nachgewiesen werden, dass der *FastSSS2*-Shader besser abschneidet als das *V-Ray Material*.

Aus diesen Resultate lässt sich der Schluss ziehen, dass sich auch aus den statistischen Tests die Empfehlung ableiten lässt, den *FastSSS2*-Shader immer dann für die Produktion von *Subsurface Scattering*-Visualisierungen zu verwendet, wenn eine hohe Qualität dieser Effekte gefordert und physikalische Plausibilität und Fotorealismus für die Umsetzung von Bedeutung sind.

Für das Kopf-Rendering der Emily konnte mit Hilfe des  $\chi^2$ -Vierfeldertests signifikant nachgewiesen werden, dass der *FastSSS2*-Shader bessere Ergebnisse bezüglich des Fotorealismus liefert als das *Skin Material*. Dies konnte auch im direkten Vergleichstest bestätigt werden.

Das gleiche gilt für das Rendering der Hand. Auch hier schneidet in beiden Testreihen jeweils der *FastSSS2*-Shader signifikant besser ab als das *Skin Material*.

newline Auch im direkten Vergleich zu Fotos konnte nachgewiesen werden, dass zwischen der Kerze in beiden Varianten, sowie dem Marmordrachen in der *FastSSS2*-Variante und jeweils drei realen Fotos kein signifikanter Unterschied besteht.

Die Renderings des Marmordrachen und der Kerze haben in beiden Varianten eine deutliche Tendenz für eine Entscheidung zum Foto. Anders sieht es da bei den Renderings in der Kategorie der *Digital Humans* aus. Hier haben alle vier Renderings eine Tendenz zu einer Entscheidung zum Rendering.

Die Renderings der Hand werden dabei etwas außer Konkurrenz betrachtet, da sie angefertigt wurden, um die Problematiken bezüglich des *Skin Materials* noch einmal zu überprüfen und den entwickelten texturbasierten Ansatz in seiner Funktion zu belegen. Die Renderings sind bezüglich des Fotorealismus nicht auf einem Stand, wie es die anderen Prototypen sind, da hier schon allein für die Erzeugung einer wirklich natürlich wirkende und dadurch glaubhaften Pose Zeit in das *Rigging* des Models hätte investiert werden müssen, um ein Beispiel zu nennen. So war ein deutlich schlechteres Ergebnis in der Evaluation für diese Renderings zu erwarten. Dennoch wurde sich bewusst dazu entschieden, das Rendering mit in die Testreihe aufzunehmen, um auch darüber Aufschluss über das Verhalten der Probanden zu erhalten.

Wird nun das Kopf Rendering betrachtet, so ist auffällig, dass zwar auch beide eine Tendenz der Entscheidung zu "Fake" zu verzeichnen haben, aber dennoch die *FastSSS2*-Shader-Variante deutlich besser abschneidet als das *Skin Material* und nahezu ein 50:50-Ergebnis erzielt hat. Die Ergebnisse zeigen, dass das *Uncanny Valley*, wie mehrfach in dieser Arbeit beschrieben, für die Beurteilung des Fotorealismus eine große Rolle spielt. Dabei haben

## 7. EVALUATION

---

noch viele weitere Faktoren Einfluss auf die Entscheidung als das *Subsurface Scattering*, Beispielsweise zu starr wirkende Augen sind ein oft genannter Faktor, der den unheimlichen Effekt ausmachen kann. Das ein qualitativ besser umgesetzter und physikalisch plausibler *Subsurface Scattering*-Effekt, erzeugt durch einen vermeintlich besseren Shader, aber die Entscheidung der Probanden das Bild für ein reales Foto zu halten positiv beeinflussen kann, zeigt, dass *Subsurface Scattering* in Bezug auf das Rendering von *Digital Humans* einen wichtigen Teil zur Überwindung des *Uncanny Valley* beitragen kann. Wenngleich auch gezeigt wurde, dass noch mehr dazu gehört, um das *Uncanny Valley* tatsächlich überwinden zu können und eine Mehrheit der Entscheidungen für Foto zu erhalten.

## Kapitel 8

# Zusammenfassung und Ausblick

### 8.1 Zusammenfassung

Eine Vielzahl transluzenter also lichtdurchlässiger Materialien umgibt uns täglich in unserer Umwelt. Dies führt dazu, dass eine akkurate Nachbildung der Streuung des Lichts in transluzenten Materialien elementar für die Produktion fotorealistischer Renderings ist. Dabei gibt es starke Variationen betreffend des Grades der Lichtdurchlässigkeit, jedoch verbindet diese Materialien alle die Gemeinsamkeit, dass Licht nicht nur an ihrer Oberfläche reflektiert wird, sondern es auch unter die Oberfläche eindringt, wo es gestreut wird, um gegebenenfalls auch an einer anderen Stelle des Objektes wieder auszutreten. Dieser Vorgang wird als *Subsurface Scattering* bezeichnet. Durch dieses besondere Streuverhalten der Lichtstrahlen erhalten alle transluzenten Materialien ein sehr weiches Erscheinungsbild. Ein weiterer Effekt, der durch *Subsurface Scattering* auftreten kann ist das Durchleuchten von Licht durch das Objekt hindurch, was vor allem dann auftritt, wenn die Beleuchtung von hinten auf das Objekt gerichtet ist.

Im Rahmen dieser Arbeit wurde die Problemstellung behandelt, dass der für die Realisierung fotorealistischer und physikalisch plausibler Renderings so bedeutungsvolle Effekt des *Subsurface Scatterings* in seiner Umsetzung nicht trivial ist. Die Schwierigkeiten, die die Erzeugung dieses Effektes in der 3D Computergrafik bereitet, liegen zum einen in der Tatsache, dass er dadurch entsteht, dass Licht unter die Oberfläche eines Materials eindringt und dort diffus gestreut wird, bevor es wieder austritt. Die Streuung findet also innerhalb des Volumens statt. In der 3D-Software, wie beispielsweise *Autodesk Maya*, welche in dieser Arbeit verwendet wurde, werden die Materialeigenschaften aber lediglich über einen Shader der Oberfläche zugewiesen und somit hat das Material an sich kein Volumen in dem die Streuung des Lichts berechnet werden könnte. So muss dieser Effekt möglichst physikalisch plausibel simuliert werden. Ein weiteres Problem ist, dass die *Bidirectional Reflectance Distribution Function* den meisten Modellen zur Simulation dieses Effektes zu Grunde gelegen hat. Die Verwendung dieses Modells ist jedoch problematisch, da die Funktion davon ausgeht, dass das Licht das Material an genau derselben Stelle wieder verlässt, an der es auftritt. Aus diesem Grund kann diese Funktion nur für die Berechnung der Streuung an

einem einzigen Punkt, nicht aber für die Simulation des Transportes von Licht unter die Materialoberfläche verwendet werden. Die dafür geeignete *Bidirectional Surface Scattering Distribution Function* ist in ihrer Berechnung zu komplex und zu ressourcenintensiv, als dass sie in der 3D-Computergrafik eingesetzt werden könnte. Erst mit einer Approximation an diese Funktion gelang im Jahre 2001 Jensen et al. mit ihrer Arbeit "A practical model for subsurface light transport" der Durchbruch für einen Algorithmus des *Subsurface Scattering* sowohl in guter Qualität als auch bei für die Produktion akzeptablen Renderzeiten.

Auch die Software *V-Ray* hat diese Algorithmen in ihrem *FastSSS2*-Shader implementiert. *V-Ray* der Firma *Chaosgroup* ist Vorreiter auf dem Markt, wenn es um physikalisch plausibles Rendering geht und auch bezüglich der Realisierung des *Subsurface Scattering*-Effektes hat dieser Renderer einiges zu bieten. So hat zum einen der Standard-Shader, das *V-Ray Material*, Funktionen zum Rendern dieses Effektes implementiert aber auch zwei spezielle Shader in Form des bereits erwähnten *FastSSS2-Shaders* sowie eines neuen *Skin Materials* werden für das Erzeugen dieser Effekte zur Verfügung gestellt.

Leider ist eine Dokumentation darüber, wann welcher dieser Shader für die Produktion die richtige Wahl ist oder inwiefern sie sich in ihrer Handhabung, ihrer Qualität und ihren Renderzeiten voneinander unterscheiden, nicht vorhanden. Aus diesem Grund wurden in dieser Arbeit Prototypen erstellt, anhand dessen diese Shader analysiert und jeweils der *FastSSS2*-Shader und das *V-Ray Material* sowie der *FastSSS2*-Shader und das *Skin Material* miteinander verglichen - mit dem Ergebnis, dass bezüglich der Qualität der Ergebnisse und der Handhabung in beiden Vergleichen der *FastSSS2*-Shader deutlich besser abschneidet. Bezüglich der Renderzeiten hingegen liegt der *FastSSS2*-Shader wiederum jeweils bei beiden Vergleichen deutlich zurück. Damit lässt sich die Empfehlung aussprechen, dass wenn immer es um die besondere Qualität der *Subsurface Scattering*-Effekte geht und diese im Vordergrund der Szene sowie auch der Umsetzung stehen, und wenn physikalische Plausibilität und Fotorealismus im Fokus stehen, der *FastSSS2*-Shader für die Produktion zu bevorzugen ist. Sollte der *Subsurface Scattering*-Effekt eher Beiwerk der Szene sein oder kaum sichtbar im Hintergrund liegen oder ist die Produktion sehr Renderzeiten-kritisch, beispielsweise weil es um eine Animation mit sehr vielen zu rendernden Einzelframes geht, dann ist es ratsam, das *V-Ray Material* oder für das Rendering von Haut das *Skin Material* für die Umsetzung zu verwenden.

Beim *Skin Material* handelt es sich um einen neuen Shader der speziell für das Rendering menschlicher Haut entwickelt wurde und dafür in seinem Aufbau an die biologische Struktur der Hautschichten des Menschen angepasst worden ist. Das Konzept des Shaders ist innovativ und zunächst ist seine Nutzung auch sehr intuitiv. Leider stößt der Shader aufgrund starker Restriktionen bezüglich der vom Benutzer kontrollierbaren Parameter aber schnell an seine Grenzen, besonders in Bezug auf den *Subsurface Scattering*-Effekt des Durchleuchtens. Hierfür wurde ein texturbasierter Lösungsansatz entwickelt, der mit Hilfe einer eingefärbten Textur sowie einer speziell angefertigten Maske die nötige Kontrolle für das gewünschte Ergebnis des Effektes wiederherstellt. Der texturbasierte Lösungsansatz ist aufgrund der Tatsache, dass die Textur in einer Bilbearbeitungs-Software eingefärbt und

dort auch die Maske erstellt werden muss mit deutlichem Mehraufwand verbunden. Hinzu kommt noch der ständige Wechsel zwischen 3D-Software und der Bildbearbeitungs-Software für die vielen Iterationen, die meist notwendig sind, um die Texturen in ihren Feinheiten bis zu einem gewünschten Ergebnis anzupassen. Um dem entgegenzuwirken und die Arbeit für den *3D-Artist* effektiver zu gestalten wurde im nächsten Schritt ein *Plugin* konzipiert und prototypisch umgesetzt, welches es ermöglicht, die Anpassungen der eingefärbten Textur direkt in der 3D-Software über ein Interface mit zwei *Slidern* für eine Effektverstärkung sowie eine Farbtonveränderung vorzunehmen.

Mit einer abschließenden Online-Evaluation, welche aus einem *Fake oder Foto-Test* und einem direkten Vergleichstest bestand, konnte mit einer Reihe von Teststatistiken nachgewiesen werden, dass der *FastSSS2-Shader* die besseren Ergebnisse bezüglich des Fotorealismus liefert. Außerdem konnte gezeigt werden, dass eine gute Umsetzung von *Subsurface Scattering* für das Rendering menschlicher Haut die Glaubwürdigkeit eines Renderings unterstützt. Jedoch gibt es bezüglich des Renderings von *Digital Humans* noch eine Reihe weiterer Faktoren, die zur Überwindung des *Uncanny Valleys* von Nöten sind. Ein Beispiel sind das oft genannte Starren der Augen, wodurch Renderings von Gesichtern unheimlich auf den Betrachter wirken..

## 8.2 Ausblick

In dieser Arbeit wurde gezeigt, dass bezüglich des neuen *Skin Materials* Probleme auftreten, wenn es um die Erzeugung des *Subsurface Scattering*-Effektes des Durchleuchtens geht, da durch die starke Restriktion der für den Benutzer regulierbaren Parameter, gegenüber dem *FastSSS2-Shader*, die Kontrollierbarkeit dieser Effekte stark eingeschränkt ist. Anhand von zwei prototypischen Implementierungen konnte aufgezeigt werden, dass ein texturbasierter Lösungsansatz, der für diese Problematik entwickelt wurde, funktioniert. Auch die Funktionalität eines dafür konzipierten *Plugins*, welches die Arbeitsschritte deutlich beschleunigen und eine effektivere Handhabung für den *3D-Artist* gewährleisten soll, in dem mit Hilfe eines *Interfaces* direkt in der *3D-Software* die Anpassung der Textur gesteuert werden kann, konnte dargelegt werden. An dieser Stelle gibt es noch einige Möglichkeiten der Erweiterung. Zum einen muss der *ColorCorrection*-Knoten für das *Plugin* derzeit noch manuell angelegt werden. Eine Automatisierung dessen wäre wünschenswert. Außerdem ist derzeit nur die nachträgliche Anpassung der eingefärbten Textur möglich. Vorstellbar wäre auch, dass die Maske ebenfalls nachträglich direkt in der *3D-Software* angepasst werden kann. In der aktuellen Umsetzung muss die erste Einfärbung der Textur noch in einer Bildbearbeitungs-Software vorgenommen werden und nur die danach folgenden Anpassungsschritte können dann in der *3D-Software* vorgenommen werden. Gegebenenfalls könnte ein *Workflow* entwickelt werden, bei dem auch dieser erste Schritt schon direkt in der *3D-Software* stattfinden kann. So wäre gegebenenfalls ein Interface ähnlich dem *Skin Weights-Tools* aus *Autodesk Maya*, welches für das *Rigging* von Charakteren eingesetzt wird, vorstellbar, um zu bestimmen, an welcher Stelle der Einfluss des *Subsurface Scattering*-Effektes für das Durchleuchten in welcher Stärke greifen soll.

In Bezug auf das *Skin Material* wurde im Verlauf der Arbeit mehrfach die Problematik angesprochen, dass für fotorealistisches Rendering Texturen nicht nur für den *Shallow Layer*, sondern auch für den *Medium Layer* und den *Deep Layer* benötigt würden, diese aber nicht hergestellt werden können, weil mit herkömmlichen Kameras nicht unter die Hautoberfläche fotografiert werden kann. Ein Aspekt, mit dem sich in zukünftigen Forschungsarbeiten auseinandergesetzt werden könnten, wäre also die Fragestellung, wie solche Texturen möglichst physikalisch plausibel erzeugt oder aber simuliert werden könnten.

In weiterführenden Arbeiten könnten weitere Teststatistiken durchgeführt werden. Besonders bezüglich des *Uncanny Valley* könnten noch eine Reihe spannender Hypothesen untersucht werden, beispielsweise, indem *Closeups* gegen komplette Aufnahmen getestet werden oder auch Bilder mit und ohne *Subsurface Scattering*-Effekte direkt miteinander verglichen werden. Gegebenenfalls könnte auch getestet werden, ob für den Betrachter plausible Hintergründe großen Einfluss auf die Entscheidung nehmen. Damit ist beispielsweise gemeint, dass die Lichtquelle, welche den Effekt des Durchleuchtens beim Ohr erzeugt, für den Betrachter sichtbar gemacht wird.

Ein wichtiges Thema für eine weiterführende Untersuchung ist auch, wie sich die hier analysierten Shader beim Rendern von bewegten Bildern in Form von Animationen verhalten.

# Anhang A

Für alle Systat-Tabellen in diesem Anhang gilt bei den Tests auf Unterscheidungen bezüglich der Vorkenntnisse in der Computergrafik steht "0" für keine Vorkenntnisse in der Computergrafik und "1" für vorhandene Vorkenntnisse in der Computergrafik. Entsprechend steht bei den Systat-Tabellen bezüglich einer Unterscheidung des Geschlechtes wegen "0" für weiblich und "1" für männlich.

## Counts

### WAHL\$(rows) by BILD\$(columns)

	CG_Kerze	IM_Blume	IM_Chilli	IM_Pasta	Total
FAKE	79	88	69	70	306
FOTO	99	90	109	108	406
<b>Total</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>178</b>	<b>712</b>

## Percents of Total Count

### WAHL\$(rows) by BILD\$(columns)

	CG_Kerze	IM_Blume	IM_Chilli	IM_Pasta	Total	N
FAKE	11,096	12,360	9,691	9,831	42,978	306,000
FOTO	13,904	12,640	15,309	15,169	57,022	406,000
<b>Total</b>	<b>25,000</b>	<b>25,000</b>	<b>25,000</b>	<b>25,000</b>	<b>100,000</b>	
<b>N</b>	<b>178,000</b>	<b>178,000</b>	<b>178,000</b>	<b>178,000</b>		<b>712,000</b>

## Chi-Square Tests of Association for WAHL\$ and BILD\$

Test Statistic	Value	df	p-value
Pearson Chi-square	5,433	3,000	0,143

Number of Valid Cases: 712

A.

---

### Counts

#### WAHL\$(rows) by BILD\$(columns)

	CG_Kerze	IM_Blume	IM_Chilli	IM_Pasta	Total
FAKE	83	99	78	81	341
FOTO	115	99	120	117	451
Total	198	198	198	198	792

### Percents of Total Count

#### WAHL\$(rows) by BILD\$(columns)

	CG_Kerze	IM_Blume	IM_Chilli	IM_Pasta	Total	N
FAKE	10,480	12,500	9,848	10,227	43,056	341,000
FOTO	14,520	12,500	15,152	14,773	56,944	451,000
Total	25,000	25,000	25,000	25,000	100,000	
N	198,000	198,000	198,000	198,000		792,000

### Chi-Square Tests of Association for WAHL\$ and BILD\$

Test Statistic	Value	df	p-value
Pearson Chi-square	5,454	3,000	0,141

Number of Valid Cases: 792

---

Counts

WAHL\$(rows) by BILD\$(columns)

	CG_Kopf	IM_Blume	IM_Chilli	IM_Pasta	Total
FAKE	97	88	69	70	324
FOTO	81	90	109	108	388
Total	178	178	178	178	712

Percents of Total Count

WAHL\$(rows) by BILD\$(columns)

	CG_Kopf	IM_Blume	IM_Chilli	IM_Pasta	Total	N
FAKE	13,624	12,360	9,691	9,831	45,506	324,000
FOTO	11,376	12,640	15,309	15,169	54,494	388,000
Total	25,000	25,000	25,000	25,000	100,000	
N	178,000	178,000	178,000	178,000		712,000

Chi-Square Tests of Association for WAHL\$ and BILD\$

Test Statistic	Value	df	p-value
Pearson Chi-square	12,913	3,000	0,005

Number of Valid Cases: 712

A.

---

### Counts

#### CG\_DRACHE\$(rows) by SEX(columns)

	0	1	Total
FAKE	27	47	74
FOTO	58	46	104
Total	85	93	178

### Percents of Total Count

#### CG\_DRACHE\$(rows) by SEX(columns)

	0	1	Total	N
FAKE	15,169	26,404	41,573	74,000
FOTO	32,584	25,843	58,427	104,000
Total	47,753	52,247	100,000	
N	85,000	93,000		178,000

### Chi-Square Tests of Association for CG\_DRACHE\$ and SEX

Test Statistic	Value	df	p-value
Pearson Chi-square	6,443	1,000	0,011

Number of Valid Cases: 178

---

### Counts

#### CG\_KERZE\$(rows) by CG(columns)

	0	1	Total
FAKE	38	41	79
FOTO	59	40	99
Total	97	81	178

### Percents of Total Count

#### CG\_KERZE\$(rows) by CG(columns)

	0	1	Total	N
FAKE	21,348	23,034	44,382	79,000
FOTO	33,146	22,472	55,618	99,000
Total	54,494	45,506	100,000	
N	97,000	81,000		178,000

### Chi-Square Tests of Association for CG\_KERZE\$ and CG

Test Statistic	Value	df	p-value
Pearson Chi-square	2,341	1,000	0,126

Number of Valid Cases: 178

A.

---

### Counts

CG\_KERZE\$(rows) by SEX(columns)

	0	1	Total
FAKE	24	55	79
FOTO	61	38	99
Total	85	93	178

### Percents of Total Count

CG\_KERZE\$(rows) by SEX(columns)

	0	1	Total	N
FAKE	13,483	30,899	44,382	79,000
FOTO	34,270	21,348	55,618	99,000
Total	47,753	52,247	100,000	
N	85,000	93,000		178,000

### Chi-Square Tests of Association for CG\_KERZE\$ and SEX

Test Statistic	Value	df	p-value
Pearson Chi-square	17,183	1,000	0,000

Number of Valid Cases: 178

---

## Counts

### CG\_KOPF\$(rows) by CG(columns)

	0	1	Total
FAKE	46	51	97
FOTO	51	30	81
Total	97	81	178

## Percents of Total Count

### CG\_KOPF\$(rows) by CG(columns)

	0	1	Total	N
FAKE	25,843	28,652	54,494	97,000
FOTO	28,652	16,854	45,506	81,000
Total	54,494	45,506	100,000	
N	97,000	81,000		178,000

## Chi-Square Tests of Association for CG\_KOPF\$ and CG

Test Statistic	Value	df	p-value
Pearson Chi-square	4,299	1,000	0,038

Number of Valid Cases: 178

A.

---

### Counts

#### CG\_KOPF\$(rows) by SEX(columns)

	0	1	Total
FAKE	34	63	97
FOTO	51	30	81
Total	85	93	178

### Percents of Total Count

#### CG\_KOPF\$(rows) by SEX(columns)

	0	1	Total	N
FAKE	19,101	35,393	54,494	97,000
FOTO	28,652	16,854	45,506	81,000
Total	47,753	52,247	100,000	
N	85,000	93,000		178,000

### Chi-Square Tests of Association for CG\_KOPF\$ and SEX

Test Statistic	Value	df	p-value
Pearson Chi-square	13,783	1,000	0,000

Number of Valid Cases: 178

---

### Counts

CG\_HANDS\$(rows) by CG(columns)

	0	1	Total
FAKE	73	73	146
FOTO	24	8	32
Total	97	81	178

### Percents of Total Count

CG\_HANDS\$(rows) by CG(columns)

	0	1	Total	N
FAKE	41,011	41,011	82,022	146,000
FOTO	13,483	4,494	17,978	32,000
Total	54,494	45,506	100,000	
N	97,000	81,000		178,000

### Chi-Square Tests of Association for CG\_HANDS\$ and CG

Test Statistic	Value	df	p-value
Pearson Chi-square	6,615	1,000	0,010

Number of Valid Cases: 178

A.

---

### Counts

#### CG\_HAND\$(rows) by SEX(columns)

	0	1	Total
FAKE	66	80	146
FOTO	19	13	32
Total	85	93	178

#### Percents of Total Count

#### CG\_HAND\$(rows) by SEX(columns)

	0	1	Total	N
FAKE	37,079	44,944	82,022	146,000
FOTO	10,674	7,303	17,978	32,000
Total	47,753	52,247	100,000	
N	85,000	93,000		178,000

### Chi-Square Tests of Association for CG\_HAND\$ and SEX

Test Statistic	Value	df	p-value
Pearson Chi-square	2,112	1,000	0,146

Number of Valid Cases: 178

# Literaturverzeichnis

- [BGBC05] BRENTON, Harry ; GILLIES, Marco ; BALLIN, Daniel ; CHATTING, David: D.: The uncanny valley: does it exist. In: *In: 19th British HCI Group Annual Conference: workshop on human-animated character interaction*, 2005
- [BKIH09] BARTNECK, Christoph ; KANDA, T. ; ISHIGURO, H. ; HAGITA, N.: My Robotic Doppelgänger - A Critical Look at the Uncanny Valley Theory. In: *18th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN2009*. Toyama : IEEE, 2009, S. 269–276
- [BL03] BORSHUKOV, George ; LEWIS, J. P.: Realistic Human Face Rendering for “The Matrix Reloaded”. In: *ACM SIGGRAPH 2003 Sketches & Applications*. New York, NY, USA : ACM, 2003 (SIGGRAPH '03), 1–1
- [Dan03] DANFERBER: The man who mistook his girlfriend for a robot. (2003). <http://www.popsci.com/scitech/article/2003-08/man-who-mistook-his-girlfriend-robot>
- [DEJ<sup>+</sup>06] DORSEY, Julie ; EDELMAN, Alan ; JENSEN, Henrik W. ; LEGAKIS, Justin ; PEDERSEN, Hans K.: Modeling and rendering of weathered stone. In: *ACM SIGGRAPH 2006 Courses* ACM, 2006, S. 4
- [DGFK02] DISALVO, Carl F. ; GEMPERLE, Francine ; FORLIZZI, Jodi ; KIESLER, Sara: All Robots Are Not Created Equal: The Design and Perception of Humanoid Robot Heads. In: *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. New York, NY, USA : ACM, 2002 (DIS '02). – ISBN 1–58113–515–7, 321–326
- [DJ05] DONNER, Craig ; JENSEN, Henrik W.: Light diffusion in multi-layered translucent materials. In: *ACM Transactions on Graphics (TOG)* 24 (2005), Nr. 3, S. 1032–1039
- [DWd<sup>+</sup>08] DONNER, Craig ; WEYRICH, Tim ; D'EON, Eugene ; RAMAMOORTHI, Ravi ; RUSINKIEWICZ, Szymon ; LAYERED, A: Heterogeneous Reflectance Model for Acquiring and Rendering Human Skin. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2008)* 27 (2008), Nr. 5, S. 140
- [fre]

- [Geb03] GEBHARDT, Nikolaus: Einige brdf modelle. In: <http://www.irrlicht3d.org/papers/BrdfModelle.pdf> (2003)
- [Gou03] GOULD, D.A.D.: *Complete Maya Programming: An Extensive Guide to MEL and the C++ API*. Morgan Kaufmann Publishers, 2003 (Complete Maya Programming). <https://books.google.de/books?id=z1jjFzxsc8IC>. – ISBN 9781558608351
- [GRK07] GERGLE, Darren ; ROSE, Carolyn P. ; KRAUT, Robert E.: Modeling the Impact of Shared Visual Information on Collaborative Reference. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2007 (CHI '07). – ISBN 978-1-59593-593-9, 1543-1552
- [Has06] HASLAM, Nick: Dehumanization: An integrative review. In: *Personality and social psychology review* 10 (2006), Nr. 3, S. 252-264
- [HBF11] HUNTER, Fil ; BIVER, Steven ; FUQUA, Paul: *Light Science and Magic: An Introduction to Photographic Lighting*. 4th. Focal Press, 2011. – ISBN 0240812255, 9780240812250
- [HBV03] HAO, Xuejun ; BABY, Thomas ; VARSHNEY, Amitabh: Interactive subsurface scattering for translucent meshes. In: *Proceedings of the 2003 symposium on Interactive 3D graphics* ACM, 2003, S. 75-82
- [HFM<sup>+</sup>10] HAŠAN, Miloš ; FUCHS, Martin ; MATUSIK, Wojciech ; PFISTER, Hanspeter ; RUSINKIEWICZ, Szymon: Physical reproduction of materials with specified subsurface scattering. In: *ACM Transactions on Graphics (TOG)* Bd. 29 ACM, 2010, S. 61
- [HK93] HANRAHAN, Pat ; KRUEGER, Wolfgang: Reflection from layered surfaces due to subsurface scattering. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* ACM, 1993, S. 165-174
- [HM10] HO, Chin-Chang ; MACDORMAN, Karl F.: Revisiting the uncanny valley theory: Developing and validating an alternative to the Godspeed indices. In: *Computers in Human Behavior* 26 (2010), Nr. 6, S. 1508-1518
- [JB02] JENSEN, Henrik W. ; BUHLER, Juan: A rapid hierarchical rendering technique for translucent materials. In: *ACM Transactions on Graphics (TOG)* 21 (2002), Nr. 3, S. 576-581
- [Jen97] JENTSCH, Ernst: On the psychology of the uncanny (1906) 1. In: *Angelaki: Journal of the Theoretical Humanities* 2 (1997), Nr. 1, S. 7-16
- [JLD99] JENSEN, Henrik W. ; LEGAKIS, Justin ; DORSEY, Julie: Rendering of wet materials. In: *Rendering Techniques 99*. Springer, 1999, S. 273-281

- [JMLH01] JENSEN, Henrik W. ; MARSCHNER, Stephen R. ; LEVOY, Marc ; HANRAHAN, Pat: A practical model for subsurface light transport. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques ACM*, 2001, S. 511–518
- [Kan09] KANG, Minsoo: The ambivalent power of the robot. In: *Antennae, The Journal of Nature in Visual Culture* 1 (2009), Nr. 9, S. 47–58
- [LGB<sup>+</sup>03] LENSCH, Hendrik ; GOESELE, Michael ; BEKAERT, Philippe ; KAUTZ, Jan ; MAGNOR, Marcus A. ; LANG, Jochen ; SEIDEL, Hans-Peter: Interactive Rendering of Translucent Objects. In: *Computer Graphics Forum* Bd. 22 Wiley Online Library, 2003, S. 195–205
- [Mac05] MACDORMAN, Karl F.: Androids as an experimental apparatus: Why is there an uncanny valley and can we exploit it. In: *CogSci-2005 workshop: toward social mechanisms of android science*, 2005, S. 106–118
- [Mac06] MACDORMAN, Karl F.: Subjective ratings of robot video clips for human likeness, familiarity, and eeriness: An exploration of the uncanny valley. In: *ICCS/CogSci-2006 long symposium: Toward social mechanisms of android science*, 2006, S. 26–29
- [Mis09a] MISSELHORN, Catrin: Empathy with inanimate objects and the uncanny valley. In: *Minds and Machines* 19 (2009), Nr. 3, S. 345–359
- [Mis09b] MISSELHORN, Catrin: Empathy with inanimate objects and the uncanny valley. In: *Minds and Machines* 19 (2009), Nr. 3, S. 345–359
- [MKB<sup>+</sup>05] MERTENS, Tom ; KAUTZ, Jan ; BEKAERT, Philippe ; VAN REETH, Frank ; SEIDEL, Hans-Peter: Efficient rendering of local subsurface scattering. In: *Computer Graphics Forum* Bd. 24 Wiley Online Library, 2005, S. 41–49
- [MMK12] MORI, Masahiro ; MACDORMAN, Karl F. ; KAGEKI, Norri: The Uncanny Valley [From the Field]. In: *IEEE Robot. Automat. Mag.* 19 (2012), Nr. 2, 98–100. <http://dx.doi.org/10.1109/MRA.2012.2192811>. – DOI 10.1109/MRA.2012.2192811
- [NB03] NOWAK, Kristine L. ; BIOCCA, Frank: The Effect of the Agency and Anthropomorphism of Users' Sense of Telepresence, Copresence, and Social Presence in Virtual Environments. In: *Presence: Teleoper. Virtual Environ.* 12 (2003), Oktober, Nr. 5, 481–494. <http://dx.doi.org/10.1162/105474603322761289>. – DOI 10.1162/105474603322761289. – ISSN 1054–7460
- [NRH<sup>+</sup>77] NICODEMUS, FE ; RICHMOND, JC ; HSIA, JJ ; GINSBERG, IW ; LIMPERIS, T: Geometric considerations and nomenclature for reflectance, volume 161 of Monograph. In: *National Bureau of Standards (US)* (1977)

- [PH00] PHARR, Matt ; HANRAHAN, Pat: Monte Carlo evaluation of non-linear scattering equations for subsurface reflection. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* ACM Press/Addison-Wesley Publishing Co., 2000, S. 75–84
- [PK06] POWERS, Aaron ; KIESLER, Sara: The Advisor Robot: Tracing People's Mental Model from a Robot's Physical Attributes. In: *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*. New York, NY, USA : ACM, 2006 (HRI '06). – ISBN 1–59593–294–1, 218–225
- [PR12] PÉREZ ROIG, FRANCISCO: *Photorealistic physically based render engines: a comparative study*, Diss., 2012
- [Sac70] SACHS, L.: *Statistische Methoden: Ein Soforthelfer Fur Praktiker in Naturwissenschaft, Medizin Technik, Wirtschaft, Psychologie Und Soziologie*. Springer, 1970 <https://books.google.com.pe/books?id=jNemKQEACAAJ>. – ISBN 9783540050032
- [Sac13] SACHS, L.: *Angewandte Statistik: Anwendung statistischer Methoden*. Springer Berlin Heidelberg, 2013. – ISBN 9783662057483
- [SBS03] SLATER, Mel ; BROGNI, Andrea ; STEED, Anthony: Physiological Responses to Breaks in Presence: A Pilot Study. In: *Proceedings of Presence 2003: The 6th Annual International Workshop on Presence*, 2003
- [SG09] STECKENFINGER, Shawn A. ; GHAZANFAR, Asif A.: Monkey visual behavior falls into the uncanny valley. In: *Proceedings of the National Academy of Sciences* 106 (2009), Nr. 43, S. 18362–18366
- [Sla02] SLATER, Mel: Presence and the Sixth Sense. In: *Presence: Teleoper. Virtual Environ.* 11 (2002), August, Nr. 4, 435–439. <http://dx.doi.org/10.1162/105474602760204327>. – DOI 10.1162/105474602760204327. – ISSN 1054–7460
- [SN07] SEYAMA, Jun'ichiro ; NAGAYAMA, Ruth S.: The Uncanny Valley: Effect of Realism on the Impression of Artificial Human Faces. In: *Presence: Teleoper. Virtual Environ.* 16 (2007), August, Nr. 4, 337–351. <http://dx.doi.org/10.1162/pres.16.4.337>. – DOI 10.1162/pres.16.4.337. – ISSN 1054–7460
- [SS00] SLATER, Mel ; STEED, Anthony: A Virtual Presence Counter. In: *Presence* 9 (2000), Nr. 5, 413–434. <http://dblp.uni-trier.de/db/journals/presence/presence9.html#SlaterS00>
- [Sta95] STAM, Jos: Multiple scattering as a diffusion process. In: *Rendering Techniques* 95. Springer, 1995, S. 41–50
- [Sta01] STAM, Jos: *An illumination model for a skin layer bounded by rough surfaces*. Springer, 2001

- [Tin14] TINWELL, Angela: Applying psychological plausibility to the Uncanny Valley phenomenon. In: *Oxford handbook of virtuality*, Oxford University Press, New York (2014), S. 173–186
- [Tin15] TINWELL, Angela: Is the Uncanny Valley a Universal or Individual Response? (2015)
- [TS14] TINWELL, Angela ; SLOAN, Robin J.: Childrens perception of uncanny human-like virtual characters. In: *Computers in Human Behavior* 36 (2014), 286 - 296. <http://dx.doi.org/http://dx.doi.org/10.1016/j.chb.2014.03.073>. – DOI <http://dx.doi.org/10.1016/j.chb.2014.03.073>
- [TSP<sup>+</sup>06] TRAFTON, J. G. ; SCHULTZ, Alan C. ; PERZNOWSKI, Dennis ; BUGAJSKA, Magdalena D. ; ADAMS, William ; CASSIMATIS, Nicholas L. ; BROCK, Derek P.: Children and Robots Learning to Play Hide and Seek. In: *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*. New York, NY, USA : ACM, 2006 (HRI '06). – ISBN 1–59593–294–1, 242–249
- [VSS05] VINAYAGAMOORTHY, Vinoba ; STEED, Anthony ; SLATER, Mel: Building characters: Lessons drawn from virtual environments. In: *Proceedings of Toward Social Mechanisms of Android Science: A CogSci 2005 Workshop*, 2005, S. 119–126
- [Wat93] WATT, Alan: *3D Computer Graphics*. 2nd. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1993. – ISBN 0201631865
- [WBG96] WILKINSON, Leland ; BLANK, Grant ; GRUBER, Christian: *Desktop Data Analysis SYSTAT*. 1st. Upper Saddle River, NJ, USA : Prentice Hall PTR, 1996. – ISBN 0135693101
- [Wes06] WESCHLER, Lawrence: Why Is This Man Smiling? Digital animators are closing in on the complex system that makes a face come alive. (2006). [https://archive.wired.com/wired/archive/10.06/face\\_pr.html](https://archive.wired.com/wired/archive/10.06/face_pr.html)
- [Win06] WINDSOR, Brian: Capturing Puppets: Using the age old art of puppetry combined with motion capture. (2006), S. 49–55
- [ZPB13] ZLOTOWSKI, Jakub ; PROUDFOOT, Diane ; BARTNECK, C.: More Human Than Human: Does The Uncanny Curve Really Matter? In: *HRI2013 Workshop on Design of Humanlikeness in HRI from uncanny valley to minimal design*, 2013, S. 7–13

