Physically Based Rendering

PBR vs. NON PBR in Blender 2.79b und Cylces

Diplomarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades
Dipl.-Ing. für technisch-wissenschaftliche Berufe

am Masterstudiengang Digitale Medientechnologien an der
Fachhochschule St. Pölten, **Masterklasse Postproduktion**

von:
Lukas Rapf, BEng
1510262525

Betreuer/in und Erstbegutachter/in: Mag. Franz Schubert
Zweitbegutachter/in: FH-Prof. Dipl.-Ing. (FH) Mario Zeller

[Ort, TT.MM.JJJJ]

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

....

Ort, Datum Unterschrift

Kurzfassung

In dieser Arbeit soll der Fragestellungen der Effizienz des PBR (Physically Based Rendering) Workflows gegenüber des/eines NON PBR Workflows innerhalb der 3D Software Blender 2.79b und der Render Engine Cycles nachgegangen werden.

Beginnend mit der grundsätzlichen Auseinandersetzung mit der Funktion von Render Engines und in weiterer Folge speziell der Render Engine Cycles werden die jeweiligen Workflows definiert und hergeleitet.

Anschließend werden die Forschungsfragen definiert.

Die auszuarbeitenden Fragestellungen beziehen sich auf die Renderzeiten sowie die benötigten Hardware Ressourcen der verschiedenen Methoden. Des Weiteren sollen auch Aussagen über etwaige Unterschiede, vor allem auch bei den Renderzeiten und den benötigten Hardware Ressourcen wie VRAM und RAM, beim Rendering über die GPU (Graphics Processing Unit oder Grafikkarte) bzw. die CPU (Central Processing Unit) gemacht werden.

Um eben jene Fragestellungen beantworten zu können, wurden im praktischen Teil dieser Arbeit zwei 3D Szenen erstellt, welche mit den verschiedenen Workflows, Lichtsituationen und über die GPU bzw. die CPU gerendert wurden.

Dabei stellte sich vor allem heraus, dass der NPR Workflow langsamer beim Set Up allerdings in der Regel schneller beim Rendering als der PBR Workflow ist. Alternativ wurden die 3D Szenen jeweils noch mit PBR Texturen aus einer externen PBR Texture Painting Software (Allegorithmic Substance Painter) geshadet. Der PBR Workflow zeichnete sich neben der schnelleren Set Up Zeit vor allem durch die, visuell eindeutig erkennbarer, physikalisch korrekteren Darstellung der Reflektionen und Specular Highlights aus.

Abstract

The main purpose of this diploma thesis is to analyze the PBR workflow in comparison to the NON PBR workflow unsing the 3D software package blender 2.79b and the render engine cycles and to analyze the difference of the two workflows and working out where their strengths and weaknesses are.

Starting with the overall definition of render engines and especially the render engine cycles the different workflows are getting illustrated and explained.

Followed by the core questions of the thesis wich include the measurement and comparison of the render times, the hardware ressources used and the difference in rendering either with the GPU or the CPU.

Therefore two 3D scenes with different 3D models where built, shaded, lit and rendered using the different workflows.

Alternatively, the 3D scenes were shaded with PBR textures exported from a different software called Allogerithmic Substance Painter.

According to the collected data from this examination the NON PBR workflow can be considered slower in terms of set up time, but faster in rendering. The PBR workflow on the other hand has its benefits in faster and easier set up times as well as in the visual outcome as light is behaving physically more accurate on the 3D models surface.

Inhaltsverzeichnis

1 Einleitung

Physically Based Rendering oder kurz PBR bekam vor allem mit dem immer mehr steigenden Einfluss von Real Time Render Engines, wie in den Game Engines Unity und Unreal, Aufmerksamkeit. Doch schon Walt Disney Pictures hatte mit dem Film *Wreck- It Ralph* und dem darauf folgenden von Walt Disney Animation Studios veröffentlichten Paper, auf welches im späteren Verlauf dieser Arbeit noch im Detail eingegangen wird, über ihren für diesen Film entwickelten Principled Shader die Weichen für eine Veränderung im Shading auch abseits von Game/Real Time Engines gestellt.

3D Texture Painting Software wie *Allegorithmic Substance Painter* oder Quixels *DDO* unterstützen auch zwei in der Industrie als Standard angesehene Softwarelösungen diesen Workflow.

Mit dem 2.79b Release hat auch das 3D Open Source Software Package *Blender* eine eigene Adaptierung dieses Disney Principled Shaders implementiert und so auch für die Render Engine *Cycles* den PBR Workflow, wie er von den Disney Animation Studios beschrieben wird, ermöglicht.

Um ähnliche Ergebnisse zu erziehlen war es im Vorfeld notwendig mehrere Shader Nodes zu kombinieren und so war es dem NON PBR Workflow geschuldet, schon bei relativ simplen Materialien mit komplexen Node Trees konfrontiert worden zu sein.

Intension und Ziel dieser Arbeit soll es sein, die Unterschiede dieser beiden Workflows zu untersuchen und deren Vor- und Nachteile in der Produktion auszuarbeiten.

Ausgehend von der grundlegenden Definition von 3D Render Engines und deren Funktion über die verschiedenen klassischen Shading Methoden bis zu dem von

den Disney Animations Studios publizierten Paper über deren Principled Shader und Definition von PBR. Gefolgt von den Möglichkeiten des Shadings in Blender 2.79b und der, in dieser Version neuen, Principled Shader Node und deren Funktionen. Anschließend werden Referenzfotos von einer Petroleum Laterne und einem Stillleben erstellt und diese in Blender 2.79b nachgebaut. Mittels der 3D Render Engine Cycles werden die 3D Szenen gerendert um so Aussagen bezüglich der Unterschiede in den Set Up Zeiten, den Renderzeiten und den benötigten Hardware Ressourcen zu treffen. Auch etwaige Unterschiede beim Rendering über die GPU bzw. die CPU im Zusammenspiel der beiden verschiedenen Workflows sollen in dieser Arbeit Beachtung finden. Des Weiteren werden diese 3D Szenen auch in unterschiedlichen Lichtsituationen gerendert um auch über diesen Prozess Aussagen treffen zu können.

2 Grundlagen: Licht und Rendering

Um den Vorgang des physikalisch korrekten Shadings in 3D Render Engines erläutern zu können, ist es im Vorfeld notwendig, Licht im Allgemeinen in seinem Verhalten, zu beschreiben und zu verstehen.

2.1 Licht allgemein

2.1.1 Definition von Licht

„Licht ist eine elektromagnetische Strahlung, angesiedelt in einem schmalen Wellenlänge-Spektrum, das wir mit den Augen wahrnehmen können."

(Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.21)

„Bei diesem Spektrum handelt es sich um den Bereich von ca. 380 nm (nm= Nanometer = 1 Milliardstel Meter) bis ca. 720 nm. Dieser Bereich des sichtbaren Lichts wird begrenzt vom unsichtbaren Infrarotbereich (>720 nm) und dem ebenfalls nicht sichtbaren, ultravioletten Bereich (<380 nm). Das dazwischenliegende, sichtbare Spektrum repräsentiert das „weiße Licht", das uns in Abhängigkeit der Wellenlänge, dann in unterschiedlichen Farben den so genannten Spektralfarben, erscheint."

(Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.21f)

Licht kann durch natürliche als auch durch künstliche Prozesse entstehen. Voraussetzung für eine Lichtquelle ist, dass sie durch die Zuführung von Energie angeregt wird (Licht auszusenden). Unabhängig von ihrer Art haben alle Lichtquellen gemeinsam, dass sich das von ihnen abgegebene Licht mit derselben Geschwindigkeit ausbreitet: der Lichtgeschwindigkeit.

(vgl. Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.22)

2.1.2 Spektralfarben

Das oben genannte „weiße Licht“ setzt sich aus unterschiedlichen Spektralfarben zusammen. Alle zusammen ergeben Weiß. Durch das Hinzufügen (additive Farbmischung) oder das Wegnehmen (subtraktive Farbmischung) kann jede beliebige Farbe generiert werden.

(Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.27)

2.1.3 Reflektion, Streuung und Brechung

Die Reflexion ist eine der wichtigsten Gesetzmäßigkeiten welcher das Licht unterliegt. Gegenstände können vom menschlichen Auge wahrgenommen werden, weil das Licht (ausgehend von einer Lichtquelle) an diesen reflektiert und so auf das Auge treffen. Wie viel und mit welcher Intensität das Licht reflektiert beziehungsweise absorbiert wird hängt sowohl von der Oberflächenbeschaffenheit als auch von der Färbung des Objektes ab. So absorbieren dunkle Oberflächen mehr Licht, dadurch wird weniger Licht in das Auge reflektiert und wird daher als dunkel wahrgenommen. Bei der Oberflächenbeschaffenheit gilt: Je glatter eine Oberfläche desto gleichmäßiger ist gestaltet sich die Reflexion. Je rauer die Oberfläche, desto diffuser ist das Licht und die Oberfläche wird als matter wahrgenommen.

(vgl. Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.27)

Für glatte Oberflächen gilt außerdem, dass das Licht mit dem gleichen Winkel in dem es auftrifft auch reflektiert wird. Streng genommen gilt dieses Gesetz auch für raue Oberflächen, allerdings nur für den jeweiligen Bereich in dem das Licht auf der Oberfläche auftritt.

(vgl. Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.27)

Diese Theorie liegt auch der Microfacet Theorie zugrunde, auf welche unter dem Punkt 4.2 Das Microfacet Modell und BRDF eingegangen wird.

„Eine diffuse Reflexion bezeichnet man als Streuung.“

(Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.27)

„Die Brechung ist eine weitere Gesetzmäßigkeit der Lichtablenkung. An der Berührungs- bzw. Grenzfläche zweier Medien mit unterschiedlichen Ausbreitungskonstanten wie etwas Luft und Wasser reflektiert ein auftreffender Lichtstrahl zum Teil, strahlt aber auch „gebrochen“ in das neue Medium hinein. Dabei verändert die Grenzfläche den Winkel, so dass der Lichtstrahl seinen Weg „geknickt“ in eine neue Richtung fortsetzt. Ist die Ausbreitungskonstante des zweiten Mediums größer als die des ersten (Übergang vom optisch dünneren zum optisch dichteren Medium), wird das Licht zum Lot hin gebrochen. Eine geringere Ausbreitungskonstante des zweiten Mediums hingegen bricht das Licht vom Lot weg.“

(Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.28)

Die Stärke der Brechung bei einer bestimmten Oberfläche wird von dem Brechungsindex IOR (Index Of Refraction) definiert. Jede Oberfläche eines bestimmten Materials hat einen eigenen definierten IOR.

Die allgemeine Formel lautet: $n = \frac{c}{v}$, wobei ‘c‘ für die Lichtgeschwindigkeit im Vakuum steht und ‘v‘ für die Phasengeschwindigkeit.

(vgl. https://en.wikipedia.org/wiki/Refractive_index)

2.1.4 Transmission (und Absorption)

Bestimmte Materialien, wie zum Beispiel Glas, lässt Licht hindurchpassieren, allerdings wird dabei auch ein bestimmter Teil absorbiert und/oder reflektiert. Ausschlaggebend dafür sind Oberflächenbeschaffenheit und Farbe der Oberfläche auf welcher das Licht auftrifft.

(vgl. Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München, S.29f)

2.2 3D Render Engine allgemein

3D Render Engines sind Softwarelösungen bzw. innerhalb einer 3D Software implementierte Algorithmen, welche die in 3D Software erstellten Szenen oder Objekte in ein zweidimensionales Bild für (in der Regel) den Bildschirm berechnen

und darstellen. Diese Algorithmen projizieren also die 3D Koordinaten des 3D Objektes in Display Koordinaten abhängig von der Position der Kamera, der Lichtquellen, des verwendeten Farbmodelles/ -raumes, Anti-Aliasing und weiteren für die jeweilige 3D Render Engine im Code festgelegten Parameter und Eigenschaften.
(vgl. http://gamma.cs.unc.edu/courses/graphics-s09/LECTURES/3DModels_SurveyPaper.pdf)

Um 3D Szenen in Display Koordinaten umzuwandeln gibt es verschiedene Arten von 3D Render Engines.

2.2.1 Rasterization:

Bei diesem Verfahren werden im Vorfeld Vektoren definiert welche später als 2D Abbildung gerendert werden. Danach werden diese mit dem Algorithmus unter Berücksichtigung der Lichtquellen usw. gerendert. Diese Art von Engines findet heutzutage vor allem in der Computer Spiele Industrie Anwendung da sie besonders schnelle Renderzeiten garantieren. Aufgrund der vorhergehenden Eliminierung der im Endbild nicht sichtbaren Objekte kommt es allerdings zu detailärmeren Endergebnissen als bei anderen 3D Render Engines. Des Weiteren basiert die Darstellung der 3D Objekte in der 3D Software selber (im sogenannten Viewport) auf der Rasterization Methode.

2.2.2 Ray Casting:

Bei dieser Methode, wird pro Pixel des Endbildes ein Light-Ray von der Kamera in die 3D Szene projiziert. Bei Objekten welche sich zum Beispiel hintereinander, ausgehend von der Kameraperspektive, befinden, wird jeweils der sich zur Kamera am nächsten befindende Punkt gerendert. Das macht diese Render Methode zu einer sehr ressourcen- und rechenintensiven Methodik des 3D Renderings. Diese Art des 3D Renderings definiert die Farbdarstellung der gerenderten Objekte rein durch die auf dem Objekt angewendeten Texturen. Daher werden bei dieser Methode keine detailreichen Endbilder mit zum Beispiel Spiegelungen, Reflektionen oder Schatten erzeugt.

(vgl. https://inst.eecs.berkeley.edu/~cs283/fa10/lectures/283-lecture2.pdf)

2.2.3 Ray Tracing:

Die Ray Casting Methode wurde mit der von Tunrer Whitted im Jahr 1980 entwickelten Ray Tracing Methode revolutioniert. Wie beim Ray Casting wird

pro Pixel ein Light Ray projiziert allerdings werden zusätzlich noch die Light Ray Bounces berechnet welche beim Auftreffen des Light Rays auf einer Oberfläche zu den jeweiligen Lichtquellen projiziert werden. Dank dieser revolutionären Weiterentwicklung ist es bei der Ray Tracing Methode möglich Spiegelungen, Reflektionen oder Schatten zu berechnen und darzustellen.

(vgl. https://inst.eecs.berkeley.edu/~cs283/fa10/lectures/283-lecture2.pdf)

2.2.4 Path Tracing:

Das Path Tracing kann als weitere Weiterentwicklung des Raytracing gesehen werden. Im Unterschied zu Ray Tracing, wird der von der Kamera projizierte Light Ray nicht direkt an die Lichtquellen weiter geleitet sondern kann mehrmals (je nach Einstellungsmöglichkeit) von den sich in der 3D Szene befindenden Objekten reflektiert, gebrochen oder absorbiert werden.

In sogenannten 3D Ray Tracing Engines, wie zum Beispiel Mental Ray, Vray, Octane oder Cycles, wird ein dem menschlichen visuellen Wahrnehmen ähnliches Verfahren angewandt. Dieses nennt sich Path Tracing. Der größte Unterschied ist allerdings die Richtung aus der das Licht kommt, da in der Realität unendlich viele Lichtwellen von der Lichtquelle (zum Beispiel der Sonne) ausgehen, werden in der 3D Ray Tracing Engine die Wellen/Strahlen von der 3D Kamera zur Lichtquelle berechnet. Dadurch wird der Rechenaufwand für die Engine maßgeblich reduziert. Weitere Einstellungsmöglichkeiten zur Reduktion der Rechen- und Renderzeiten, sind die Zahl der Bounces. Bounces definieren wie oft ein Lichtstrahl von Objekten reflektiert wird. Für die, der realen Welt am nächsten kommende Einstellung wird in der Regel der Ausdruck „Full Global Illumination“ verwendet.

(vgl. https://www.cs.utah.edu/~shirley/books/fcg2/rt.pdf)

Da sich im empirischen Teil dieser Arbeit der Path Tracing Engine Cycles von Blender bedient wird, ist es sinnvoll die Einstellungsmöglichkeiten zu definieren und erklären, da diese erheblichen Einfluss auf das finale Rendering haben.

2.3 Cycles (Blender 2.79)

Im Zuge der Arbeit wird die Cycles Render Engine aus der Blender Version 2.79 verwendet.

Blender ist eine Open Source 3D Software welche unter der GNU General Public License veröffentlicht wird. Das 3D Package besitzt neben Modellierungs- Simulations- und Animations- Möglichkeiten auch verschiedene Render Engines. Neben der Blender Internal Engine, der Blender Game Engine gibt es in Blender 2.79 auch standardmäßig die Render Engine Cycles. Für die Auswertungen in dieser Arbeit wird die Path Tracing Render Engine Cycles verwendet.

Die Cycles Render Engine ist eine „Unidirectional path tracing with multiple importance sampling“(https://www.cycles-renderer.org/about/) Render Engine.

Beim Rendervorgang wird sowohl Multithreading beim CPU Rendering als auch (Multi-) GPU Rendering unterstützt. In den generellen Einstellungen für das Rendering kann man zwischen Path Tracing und Branched Path Tracing wählen.

(vgl. https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render)

2.3.1 Path Tracing in Cycles

Die Render Engine Cycles ist ein wie im Punkt 2.2.4 definierter Path Tracer. Jeder Light Hit strahlt in eine bestimmte Richtung und wird von einer bestimmten Lichtquelle beleuchtet. Diese Variante hat den Vorteil dass die einzelnen Samples schneller berechnet werden, allerdings sind in der Regel höher Sampleraten notwendig um den Rauschanteil als auch das Anti-Aliasing im finalen Rendering zu minimieren.

(vgl. https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render)

2.3.2 Branched Path Tracing in Cycles

Der Branched Path Tracing Integrator ist dem Path Tracing sehr ähnlich, mit dem Unterschied, dass Cycles die Einstellungsmöglichkeit besitzt die Samples pro Lightray eigens zu definieren.

Daraus resultiert eine langsamere Berechnungszeit pro Sample im Vergleich zu der herkömmlichen Path Tracing Methode in Cycles, der Rauschanteil ist

aber deutlich reduziert, besonders in Szenen, in denen vor allem Direct Lighting bzw. One Bounce Lighting Anwendung finden.

(vgl. https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render)

Die genauen Einstellungen sind folgendermaßen definiert:

AA Render Samples

Definiert die Anzahl der Samples für jeden Pixel im finalen Rendering. Je höher die Anzahl der Samples ist, desto qualitativ höher ist das Anti-Aliasing (Kantenglättung bei der Quantisierung/Abtastung).

AA Preview Samples

Definiert die Anzahl der Samples im Preview Modus, dies dient zur Steigerung der Performance.

Diffuse Samples

Definiert die Anzahl der Diffuse Bounce Samples für jeden AA Sample.

Glossy Samples

Definiert die Anzahl der Glossy (Reflexion) Bounce Samples für jeden AA Sample.

Transmission Samples

Definiert die Anzahl der Transmission Bounce Samples für jeden AA Sample.

AO Samples

Definiert die Anzahl der Ambient Occlusion (physikalisch nicht korrekte Darstellung von Schattierungen am Objekt) Samples für jeden AA Sample.

Mesh Light Samples

Definiert die Anzahl der Mesh Light Samples für jeden AA Sample.

Subsurface Samples

Definiert die Anzahl der Subsurface Scattering (Untergrundstreuung, das Licht kann teilweise durch das Objekt hindurchdringen und wird darin gebrochen) Samples für jeden AA Sample.

Volume Samples

Definiert die Anzahl der Volume Scattering (Licht wird in einem in seiner Dichte vordefinierten Volumen gebrochen) Samples für jeden AA Sample.

Sample All Direct Lights

Bei dieser Einstellungsmöglichkeit wird jede Lichtquelle in der Szene für die Direct Bounces gesampelt. Im Umkehrfall wird für die Direct Bounces eine Lichtquelle per Zufall ausgewählt.

Sample All Indirect Lights

Bei dieser Einstellungsmöglichkeit wird jede Lichtquelle in der Szene für die Indirect Bounces gesampelt. Im Umkehrfall wird für die Indirect Bounces eine Lichtquelle per Zufall ausgewählt.

(vgl.https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render)

2.3.3 Ray Types

2.3.3.1 Definition

Die verschiedenen Light Rays kann man allgemein in vier verschiedene Kategorien unterscheiden:

Camera: Der Light Ray kommt direkt von der Kamera

Reflection: Der Light Ray wird von einer reflektierenden Oberfläche generiert

Transmission: Der Light Ray wird generiert beim Durchqueren einer durchlässigen Oberfläche (Transmissive Surface)

Shadow: Der Light Ray definiert (transparente) Schatten

Des Weiteren können Reflection und Transmission Rays folgende Eigenschaften aufweisen:

Diffuse: Der Light Ray wird durch eine diffuse Reflektion oder Transmission (Translucency) generiert

Glossy: Der Light Ray wird durch eine Glossy Specular Reflection (glänzende spiegelnde Reflektion) oder Transmission generiert

Singular: Der Light Ray wird durch eine „perfekte scharfe Reflektion“ generiert

(vgl.
https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render
/light_paths.html)

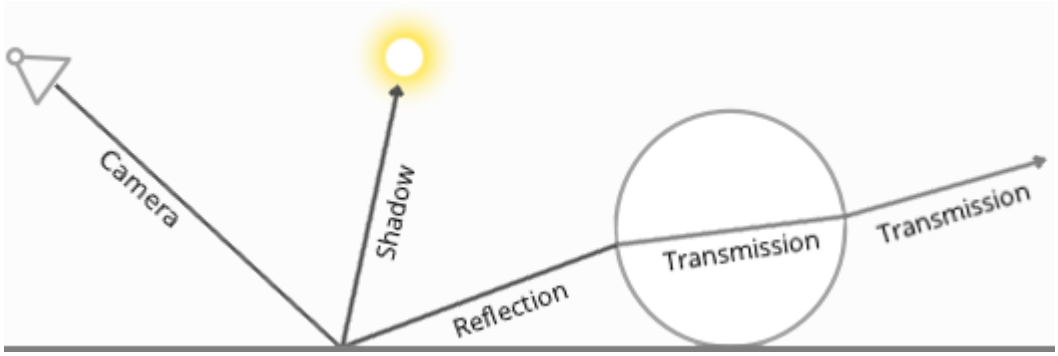

Abbildung 1. Ray Types (http://builder.openhmd.net/blender-hmd-viewport-temp/render/cycles/settings/light_paths.html)

2.3.4 Light Paths

Unter dem Überbegriff Light Paths werden die Einstellungsmöglichkeiten der Bounces, der Transparency Bounces und der Reflective und Refractive Caustics behandelt.

2.3.4.1 Bounces

Mittels der Einstellungsmöglichkeiten der Bounces wird zuerst die Anzahl an Bounces definiert. Dafür werden primär die minimalen und maximalen Bounces anhand von zwei Einstellungsmöglichkeiten ausgewählt. Sind die maximalen Bounces etwa auf 0 so handelt es sich rein nur um Direct Lighting. Des Weiteren finden sich auch Einstellungsmöglichkeiten um die Bounces für Diffuse, Glossy, Transmission und Volume extra zu definieren.

2.3.4.2 Transparency

Unter den Einstellungen für die Transparency Bounces können jeweils, wie bei den Bounces, Minimum und Maximum Werte angegeben werden, als auch ein eigener Wert für die Transparency Shadows. Dies findet bei Direct Light Sampling für die Schattenberechnung zwischen zwei Oberflächen mit Transparenz Anwendung.

2.3.4.3 Caustics

„Katakaustiken sind leicht zu beobachten und einem breiteren Publikum dadurch bekannt, daß sie bei schräg von der Sonne beleuchteten und hinreichend gefüllten und geeigneten Kaffeetassen auftreten (sog. Kaffeetassenkaustik). Natürlich funktioniert das auch mit Tee, vorausgesetzt, man tut etwas Milch hinein."

(https://users.ph.tum.de/cucke/ftp/lectures/jena96.pdf S1.)

Kaustik ist ein Abbildungsfehler, welcher bei optischen Instrumenten wie zum Beispiel bei Linsen auftreten kann. Die Häufigste Form ist die sogenannte sphärische Aberration. (vgl. Ausgewählte Kapitel der Technik, Viktor Ritter von Niesiolowski-Gawin, Verlag von L.W. Seidel & Sohn, 1908, S.385)

„Sind die Linsen nicht dünn im Vergleich zu ihren Krümmungshalbmessern und die Einfalls- und Brechungswinkel der Strahlen nicht klein, dann bleiben homozentrische Strahlenbüschel nach der Brechung nicht mehr homozentrisch. Es haben vielmehr, wie Konstruktion und Rechnung lehren, Randstrahlen eine kürzere Vereinigungsweite als Zentralstrahlen. Die Gesamtheit der konischen Lichtbüschel hat eine einhüllende Fläche (Brennfläche oder Kaustik), deren engste Stelle (Zerstreuungskreis) durch ihren Schnitt mit den verlängerten Randstrahlen bestimmt ist. Die Differenz der Schnittweiten von Rand- und Zentralstrahlen A''A' heißt Längenabweichung. Da die Erscheinung von der Kugelgestalt der Linsenfläche herführt, bezeichnet man sie als sphärische Aberration."

(Ausgewählte Kapitel der Technik, Viktor Ritter von Niesiolowski-Gawin, Verlag von L.W. Seidel & Sohn, 1908, S.385f)

In der Render Engine Cycles befinden sich zwei Checkboxen welche als Einstellungsmöglichkeit bzw. An- und Abwahlmöglichkeit für Refractive und Reflective Caustics dienen.

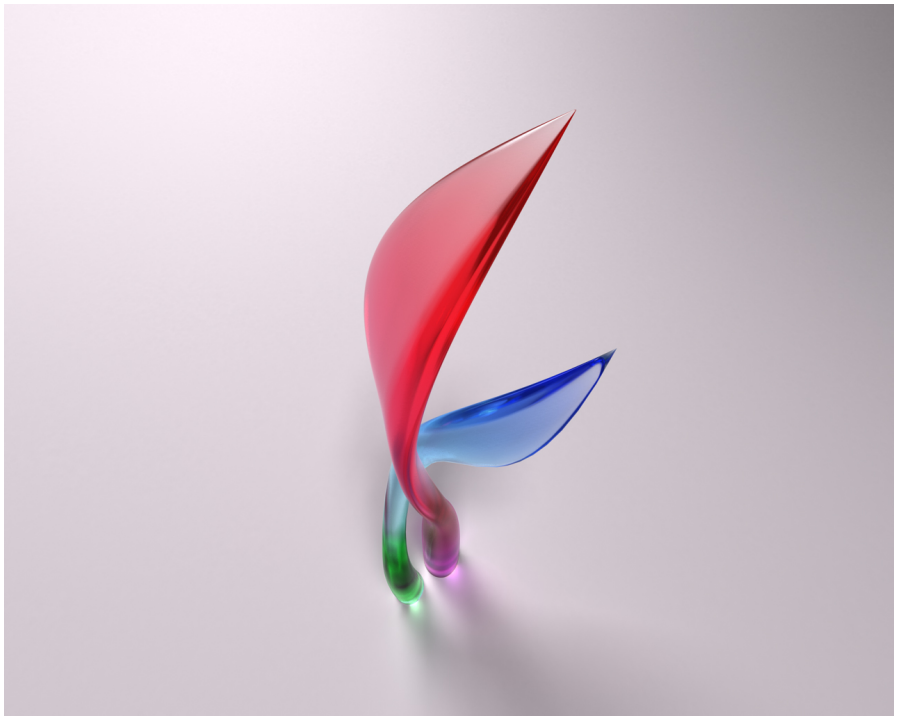

Abbildung 2. Gerenderte Caustics (https://de.wikipedia.org/wiki/Datei:Seduce_Me_by_DonBertone.jpg)

2.3.4.4 *Filter Glossy*

Die Einstellungsmöglichkeit Filter Glossy ist eine Funktion welche Glossy („glänzende“) Reflektionen weichzeichnet.

Manche Light Paths haben eine sehr geringe Wahrscheinlichkeit von der Render Engine gesehen zu werden, geben aber gleichzeitig extrem viel Licht pro Pixel ab. Daraus entstehen einige Pixel die extrem hell sind, diese Pixel nennt man Fire Flies. Diese sind besonders auf Glossy Materials als Specular Highlights sichtbar.

Da es mit Path Tracing Einiges schwer ist Specular Highlights zu finden, vergrößert man diese über das Erhöhen der Roughness (Rauheit, Oberflächenbeschaffenheit). Hierzu dient die Filter Glossy Einstellung.

(vgl. https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render/integrator.html#light-paths)

Bei Einzelbildern und mit freiem Auge sind diese weichgezeichneten Oberflächen schwer zu erkennen (siehe Abbildung), allerdings kann vor allem bei gerenderten Animationen von einer deutliche reduzierten Fire Fly Dichte ausgegangen werden.

Abbildung 3. Vergleich von Spheres mit Glossy BSDF und verschiedenen Roughness Werten mit dem Filter Glossy Wert 0.0 und 10.0 (vom Autor mit Blender 2.79 erstellt und mit Cycles gerendert)

3 Klassische Beleuchtungsmodelle

3.1 Allgemein

Um die Unterschiede beziehungsweise die Vorteile des PBR Shadings definieren zu können, ist es notwendig, die bisher wichtigsten und vorherrschenden Beleuchtungsmodelle zu benennen und zu erklären. Diese werden in den nachstehenden Punkten definiert.

Prinzipiell unterscheidet man in Diffuse Reflection Shader und Specular Reflection Shader.

3.2 Diffuse Reflection Shading Modelle

3.2.1 Lambert

Das Lambert Shading Modell zeichnet sich vor allem dadurch aus, dass es eine ideal matte beziehungsweise diffuse Oberfläche ohne „Specular Highlights“ darstellt.

Dem Lambert Shading Modell liegt das Lambertsche Gesetz (auch Lambertsches Kosinusgesetz genannt) von Johann Heinrich Lambert aus dem Jahre 1760 zugrunde. Dieses Gesetz beschreibt die Abnahme der Strahlungsstärke bei flacher werdendem Abstrahlwinkel. Unterliegt eine Oberfläche zur Gänze diesem Gesetz spricht man von einem Lambert-Strahler.

In der Natur kommt es allerdings nie dazu, dass das Gesetz auf eine Oberfläche zutrifft, vor allem weil die Strahldichte jeder Oberfläche Richtungsabhängig ist und somit dem Gesetzt widerspricht. Allerdings gibt es Materialien die dem Lambert Strahler sehr nahe kommen, wie zum Beispiel mattes Papier.

In 3D Render Engines allerdings dient genau dieses Modell zur Darstellung matter und diffuser Oberflächen. Es handelt sich dabei auch um eine relativ schnelle und ressourcenschonende Art des Shadings.

(vgl. http://iopscience.iop.org/article/10.1088/0026-1394/47/3/021/meta)

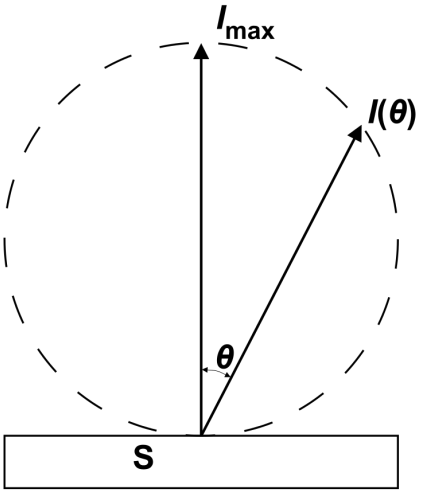

Abbildung 4. Lambertsches Gesetz, Winkelabhängigkeit der Strahlstärke (Winkel zur Oberflächennormale; I: Strahlstärke; S: Quelle oder Reflexionsfläche) (https://commons.wikimedia.org/wiki/File:LambertCosineLaw.svg)

3.2.2 Oren-Nayar

„Oren-Nayar materials are another generalisation of the basic diffuse material type,…

The appearance is quite similar to diffuse materials, but with less darkening at grazing angles; this makes it suitable for modelling very rough or porous surfaces such as clay or the moon's surface.“

(http://www.indigorenderer.com/documentation/manual/indigo-scenes/materials/material-types/oren-nayar)

Das Oren-Nayar Modell wird auch als die Veralgemeinerung des Lambertschen Gesetzes bezeichnet. Im Gegensatz zu dem Lambertschen Modell allerdings kann mittels des Oren-Nayar Modelles auch Mehrfachreflexion dargestellt werden. Dadurch wirkt die Oberfläche heller und pysikalisch akkurater.

Entwickelt wurde dieses Modell, im Jahr 1993, von Michael Oren und Shree K. Nayar.

(vgl. http://www.indigorenderer.com/documentation/manual/indigo-scenes/materials/material-types/oren-nayar)

Abbildung 5. Vergleich des Lambertschen Modelles mit dem Oren Nayar Modell (https://www.semanticscholar.org/paper/Generalization-of-the-Lambertian-model-and-for-Oren-Nayar/45d4f4f956f5a618a233836edfd1686ed4ee68da/figure/24)

3.2.3 Lambert und Oren-Nayar in Blender 2.79

Das Lambert Shading Modell in Blender, ist vor allem bei der zweiten standardmäßig in Blender 2.79 integrierten Render Engine der Blender Internal Render Engine, als am häufigsten genutzter Diffuse Shading Algorithmus im Einsatz. Hierbei hat man nur die Möglichkeit, abgesehen von der Farbwahl, mittels eines Sliders die Intensity einzustellen, damit definiert man wie viel des verfügbaren Lichtes reflektiert wird. Standardmäßig ist der Wert mit 0,8 vordefiniert.

(vgl. https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/diffuse_shaders.html)

Eine zusätzliche Einstellungsmöglichkeit bietet sich bei dem Oren-Nayar Modell. Mittels des Roughness Wertes kann die Menge der diffusen Streuung manipuliert werden.

(vgl.
https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/diffuse_shaders.html)

In der Render Engine Cycles (Version 2.79) ist der Standard Diffuse Shader ein Mix aus den beiden oben erklärten Shader Modellen. Bei einem Roughness Wert von 0,0 benutzt der Shader oder auch die Diffuse BSDF (Bidirectional Scattering Distribution Function) Node das Lambert Modell, bei anderen Werten wird das Oren-Nayar Modell verwendet.

(vgl.
https://docs.blender.org/manual/en/dev/render/cycles/nodes/types/shaders/diffuse.html)

3.3 Specular Reflection Shading Algorithmen

3.3.1 Phong

Das 1975 entwickelte Phong Beleuchtungsmodell, benannt nach seinem Entwickler Bùi Tường Phong, ist eine ressourcenschonende Methode glänzende/reflektierende Oberflächen darzustellen. Hauptsächlich werden Phong Shader für Materialien wie Plastik und ähnliches verwendet. Das Modell ist nicht physikalisch akkurat, das Glanzlicht (Specular Highlight) wird durch $\cos n(0)$ definiert. Der Wert n bestimmt die Rauheit (Roughness) der Oberfläche.

(vgl. http://www.cs.northwestern.edu/~ago820/cs395/Papers/Phong_1975.pdf)

3.3.2 Blinn-Phong

Das Blinn Beleuchtungsmodell, auch Blinn-Phong Modell genannt, ist ein weiteres Modell zur Darstellung reflektierender/glänzender Oberflächen. Als Grundlage dient das Phong Modell, allerdings werden durch die sogenannten Halfway-Vektoren die Berechnungen schneller durchgeführt wobei gleichzeitig kaum Unterschiede im Endergebnis (zum Phong Modell) erkennbar sind. Der Halfway-Vektor H wird statt dem Reflektionsvektor verwendet, die Formel zur Berechnung des Halfway-Vektors lautet $H = \frac{V+L}{\|V+L\|}$ wobei V der normierte Vektor vom Punkt zum Betrachter ist und L der normierte Vektor vom Punkt zu der (Punkt-) Lichtquelle ist.

Entwickelt wurde das Blinn-Phong Modell 1977 von James F. Blinn.

(vgl. https://de.wikipedia.org/wiki/Blinn-Beleuchtungsmodell)

3.3.3 Cook-Torrance

Das Cook-Torrance Modell oder auch das Torrance-Sparrow Modell ist ein Modell zur physikalisch korrekten Darstellung der Specular Highlights auf Oberflächen bei der Darstellung bzw. dem Rendering von 3D Modellen.

Die Oberfläche wird dabei in kleine sogenannte Facetten unterteilt, dies ermöglicht die Berechnung der Rauheit, des Brechungsverhältnisses und der Abschattung.

Zur Anwendung kommen die Beckmannsche Wahrscheinlichkeitsdichtefunktion auch Microfacettenverteilung genannt, mittels welcher die Roughness berechnet wird: $D = \frac{e^{-(\frac{\tan ß}{m})^2}}{4m^2 cos^4 ß}$.

Wobei ß den Winkel zwischen der Oberflächennormalen $\vec{N}$ und der Winkelhalbierenden zwischen der Lichteinfallsrichtung $\vec{L}$ und der Beobachtungsrichtung $\vec{V}$ beschreibt. Die Standardabweichung der Steigung der einzelnen Facetten der Oberfläche wird mit dem Wert m beschrieben.

Für das Brechungsverhältnis wird die Fresnel Gleichung, welche sich aus den maxwellschen Gleichungen herleitet, angewandt. Daraus ergibt sich folgende Gleichung zur Berechnung von Abschattung, Brechungsverhältniss und Rauheit $\rho = \frac{F_\lambda DG}{\pi(\vec{N}+\vec{V})(\vec{N}+\vec{L})}$ wobei F_λ als sogenannter Fresnelterm die Reflektion des Lichtes von den Facetten beschreibt und G der Abschwächungsfaktor ist.

(vgl. http://inst.eecs.berkeley.edu/~cs283/sp13/lectures/cookpaper.pdf)

3.3.4 Phong in Blender 2.79

Die Option beim Diffuse Shading das Phong Modell auszuwählen, gibt es bei der Version 2.79 von Blender nur bei der Internen Render Engine. Laut dem Blender 2.79 Manual eignet sich das Phong Modell besonders für die Darstellung von Atmosphären von Planeten (bei niedrigen Werten bei der Einstellungsmöglichkeit der Hardness). Der Hardness Slider bestimmt die Größe des Specular Highlights.

(vgl.
https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/specular_shaders.html)

3.3.5 Blinn in Blender 2.79

Im Gegensatz zum Phong Modell bietet das Blinn-Phong Modell in Blender 2.79 mit dem IOR (Index Of Refraction) eine zusätzliche Einstellungsmöglichkeit und somit etwas mehr Kontrolle über das Endergebnis. Das Blinn Modell wird oft in Kombination mit dem Oren-Nayar Modell verwendet.

(vgl.
https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/specular_shaders.html)

3.3.6 Cook-Torrance in Blender 2.79

Das Cook-Torrance Modell ist in Blender 2.79 in der Internen Render Engine als standardmäßiges Modell zur Darstellung der Specular Highlight voreingestellt.
Der Hardness Slider bestimmt die Größe des Specular Highlights.

(vgl.
https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/specular_shaders.html)

4 PBR nach Disney

In dem am 31. August 2012 veröffentlichten Paper von den Walt Disney Animation Studious wird die die Grundüberlegung beschrieben welche zur Entwicklung ihres PBR Shaders, dem Disney “principled” BRDF führte.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.1)

4.1 Erste Überlegungen

Basierend auf dem Erfolg des Physically Based Hair Shading des Disney Filmes Tangled (im deutschsprachigen Raum unter „Rapunzel neu verföhnt“ erschienen), machten sich die Entwickler bei Disney Animation Studios Gedanken darüber das Physically Based Shading auch auf andere Materialien und Modellen anzuwenden. Das Ziel sollte dabei sein, die Reaktion der Beleuchtung über das gesamte 3D Modell oder Szenerie, mit seinen verschiedenen Materialien möglichst einheitlich zu gestalten. Gleichzeitig sollte der durch den vereinfachten Workflow auch die Produktivität der Artists gesteigert werden.

Mittels eines eigens entwickelten BRDF-Viewers dem *BRDF-Explorer* wurde auch ein Tool entwickelt um verschiedene BRDF Modelle zu vergleichen.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.1)

4.2 Das Microfacet Modell und BRDF

Die sogenannte Bidirectional Refraction Distribution Function kurz BRDF unter Berücksichtigung des Microfacet Modell geht davon aus, dass sich auf der reflektierenden Oberfläche von 3D Objekten sogenannte Microsurface Normals befinden, welche die Reflektion beeinflussen. Bei einem gegebenen Lichtvektor l und dem Viewvector (Blickwinkel Vektor) v liegt die Microsurface Normal genau dazwischen und ist mit $h = \frac{l+v}{|l+v|}$ definiert.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.1f)
Als generelle Formel für isotrope Materialien im Microfacet Model gilt:

$$f(l,v) = \text{diffuse} + \left(\frac{D(0_h)F(0_d)G(0_l,0_v)}{4\cos 0_l \cos 0_v}\right)$$

„Diffuse“ steht für eine unbekannte/unbestimmte Funktion, in der Praxis wird meist die Lambert Diffuse Funktion angenommen und mit einem konstanten Wert angegeben. Die Reflektion (Specularity) D, definiert durch die Microfacet Distribution Function, beschreibt die Spiegelung und deren Spitzen. F ist der Fresnel Reflection Coefficient und G steht für den „geometric attenuation or shadowing factor“ welcher mit „Abschattungsfaktor“ übersetzt werden kann. Viele Modelle welche nicht spezifisch das Microfacet Modell verwenden können dennoch als solches interpretiert werden. Voraussetzung dafür ist eine Distribution Function (Verteilungsfunktion), ein Fresnel Faktor und ein weiterer Faktor welcher als Abschattungsfaktor beschrieben/verwendet werden kann. Die Ableitung der Microfacet Funktion ist das ausschlaggebendste Alleinstellungsmerkmal des Microfacet Modells. Es entspricht dem Faktor $\frac{1}{4\cos 0_l \cos 0_v}$. Für andere Modelle kann ein Abschattungsfaktor angenommen werden indem man diese mit $4\cos 0_l \cos 0_v$ multipliziert nachdem man D und F gekürzt hat.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.2.)

4.3 Die MERL BRDF Database Dateien

4.3.1 MERL 100

Die MERL (Mitsubishi Electric Research Laboratories) BRDF Database beinhaltet Reflexionsfunktionen für 100 verschiedene Materialien. Jede Reflexionsfunktion wurde als eine gemessene Bidirektionale Reflexionsverteilungsfunktion (BRDF) gespeichert. Diese Daten sind unter https://www.merl.com/brdf/ für wissenschaftliche Untersuchungen freigegeben.

In dieser Datenbank sind Materialien wie Stein, Plastik, Gummi, Farbe, Holz und andere synthetische Materialien enthalten. Bei der Beurteilung neuer BRDF Modelle werden diese Daten oft herangezogen. Jedes der BRDF in den MEL 100 ist als Würfel gesampelt und hat die gleichen Abmessungen.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.2f)

Zur Untersuchung und Überprüfung der Informationen aus den MEL 100, wurde mit dem BRDF Explorer ein eigenes Tool entwickelt. Dieses Tool ist als Open Source Software unter github.com/wdas/brdf verfügbar. Der BRDF Explorer besitzt folgende Eigenschaften:

- Die Eigenschaft mehrere BRDF auf einmal zu laden (in GLSL)
- Die Eigenschaft bereits gemessene BRDF zu laden
- Die Eigenschaft mehrere Datendiagramme zu haben (zum Beispiel die 3D-Halbkugelansicht)
- Berechnung des Albedo (der reinen Farbinformation der Oberfläche des Materials)
- Image Slice View mit Exposure Control (Belichtungskontrolle)
- Belichtete Objektansicht mit importance-sampled IBL (Image Based Lighting) Beleuchtung
- UI (User Interface) Steuerelemente

Der BRDF Explorer half nicht nur bei dem Untersuchen und Vergleichen bestehender und neu entwickelter Modelle, er wurde auch als interaktiver BRDF Editor von den Artists bei Disney verwendet.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.3f)

4.3.2 Observations

Mittels der Observations (Beobachtungen) wurden die materialspezifischen Eigenschaften von Diffuse, Specular D, Specular F und Specular G mit dem BDRF-Explorer beobachtet und untersucht.

4.3.2.1 Diffuse Observation

Die sogenannte Diffuse Reflectance beschreibt das Licht, dass von der Oberfläche gebrochen und teilweise absorbiert wird. Aus der teilweisen

Absorption ergibt sich, dass das reflektierte Licht die Farbe der Oberfläche hat. Aus dieser Beobachtung lässt sich ableiten, dass jedes nicht-metallische Material welches diese Eigenschaften aufweist, als diffuses Material beschrieben werden kann.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.5)

4.3.2.2 Specular D Observation

Die Spiegelungen, die vor allem bei metallischen Materialien wie Stahl oder Chrom auftreten sind mit den herkömmlichen Modellen wie Phong, Blinn usw. nicht korrekt darstellbar. Dies betrifft vor allem die Schweife (Ausbreitung des Scheins) der Spiegelungen.
Als Lösung wurde die GGX Verteilung gefunden welche in ihren Eigenschaften den Referenzen am nächsten kam.
(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.7)

„GGX is a microfacet distribution. It has a sharper peak and a larger tail than Beckmann. GGX is suitable for modeling light reflection from surfaces more realistically“
(https://arnold-rendering.com/2016/02/06/ggx-microfacet-distribution/)

4.3.2.3 Specular F Observation

Der Fresnel Reflection Factor definiert die Veränderungen der Specular Reflections wenn sich Lichtquelle und Viewvectors voneinander entfernen. Des Weiteren beschreibt er, dass alle glatten Oberflächen bis zu 100% spiegelnd werden, sobald das einfallende Licht diese streift. Ähnliches gilt für raue Oberflächen, allerdings mit dem Unterschied, dass diese niemals eine 100% spiegelnden Oberfläche werden, aber ebenfalls um ein Vielfaches mehr Licht reflektieren. .

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.8f)

4.3.2.4 Specular G Observation

Die Specular G Observation oder auch der Albedo Observation beschäftigt sich mit der Abschattung. Wichtig dabei ist eine möglichst genaue Schätzung der D und F Faktoren (aus der Specular D Observation und der Specular F

Observation) als auch eine Unterteilung beziehungsweise Isolation von Specular und Diffuse.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.9f)

4.4 Disneys „principled“ BRDF

Anschließend wird der Ensteheungsprozess rund um die Parameter des Shaders und die Definition des Disney Principled BRDF Shader Node erläutert.

4.4.1 Prinzipientreu

Disney entschied sich, anstatt eines 100% physikalisch korrekten Shaders einen „prinzipientreuen“ Shader zu entwickeln. Somit haben die Artists letztendlich die Kontrolle über das Endprodukt.

Disney definierte fünf Prinzipen nach denen der „principled“ Shader ausgerichtet werden soll. Diese lauten:

1. Intuitive anstatt physikalisch korrekter Parameter sollen im Shader (vor allem dem UI) verwendet werden
2. Die Zahl der Parameter sollte so klein wie möglich gehalten werden
3. Jeder Parameter sollte die Einstellungsmöglichkeit von 0 bis 1 haben
4. An Punkt 3 anknüpfend soll es bei der Einstellung bei den Parametern jedoch trotzdem erlaubt sein, dass diese über die vordefinierte Einstellungsmöglichkeit von Null bis Eins hinausgehen kann, wenn es Sinn ergibt.
5. Alle Parameter und Kombinationen sollen so plausibel wie möglich sein
6. Aufbauend auf den fünf Prinzipien wurden 11 Parameter definiert, wobei einer die Farbeinstellung ist und die restlichen 10 skalierbare Parameter von 0 bis 1 (und teilweise darüber hinaus).

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.12)

4.4.2 Parameter

Folgende Parameter wurden für den Disney „principled“ Shader definiert:

- baseColor: definiert die Farbe der Oberfläche, kann auch mit Texturen versehen werden
- subsurface: steuert die Subsurface Approximation
- metallic: ist die Einstellungsmöglichkeit für die Auswahl zwischen zwei verschiedenen Modellen (der Wert 0 steht für dielektrische Materialien und 1 für metallische Materialien). Metallische Materialien haben keine Diffuse Komponente und haben dafür eine getönte/gefärbte Spiegelung vergleichbar mit dem Parameter der baseColor.
  Auch bei diesem Input bietet es sich an, diesen mit Texturen zu versehen, um eine korrekte Darstellung zu garantieren, etwa bei komplexen Materialien die sowohl metallische als auch dielektrische Komponenten haben wie zum Beispiel rostiges oder bemaltes Metall oder ähnliches. Als dielektrische Materialien werden in der Regel alle nicht metallischen Materialien definiert.
- specular: definiert wie sehr das Material spiegelt. Diese Einstellungsmöglichkeit wird anstelle eines konkreten IOR Wertes verwendet.
- specularTint: dient zum Einstellen der Farbe der Spiegelung abhängig von der Farbe der Oberfläche.
- roughness: definiert die Oberflächenrauheit, hat Einfluss auf Diffuse und Specular des Materials.
- anisotropic: kontrolliert das Seitenverhältnis des Specular Highlights und somit die Grad der Anisotropie. Die Reflektionen werden also in eine bestimmte Richtung verzerrt.
- sheen: ist eine spezielle Einstellungsmöglichkeit die ursprünglich für Stoff und ähnliche Materialien entwickelt wurde. Die Reflektionen werden bei bestimmten Glanzwinkeln verändert. Bei höheren Metallic-Werten verringert sich die Ausprägung dieses Effektes.
- sheenTint: bestimmt wie sehr sich der Sheeneffekt farblich an der Grundfarbe (baseColor) orientiert.
- clearcoat: eine Zweite, für spezielle Verwendungszwecke, „Schicht“ von Reflektionen. Diese Einstellung wird zum Beispiel bei der Darstellung von, besonders glänzendem, Autolack verwendet.
- clearcoatGloss: definiert wie spiegelnd der Clearcoat ist. (0 ist ganz diffus und 1 ist 100% spiegelnd)

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.12f)

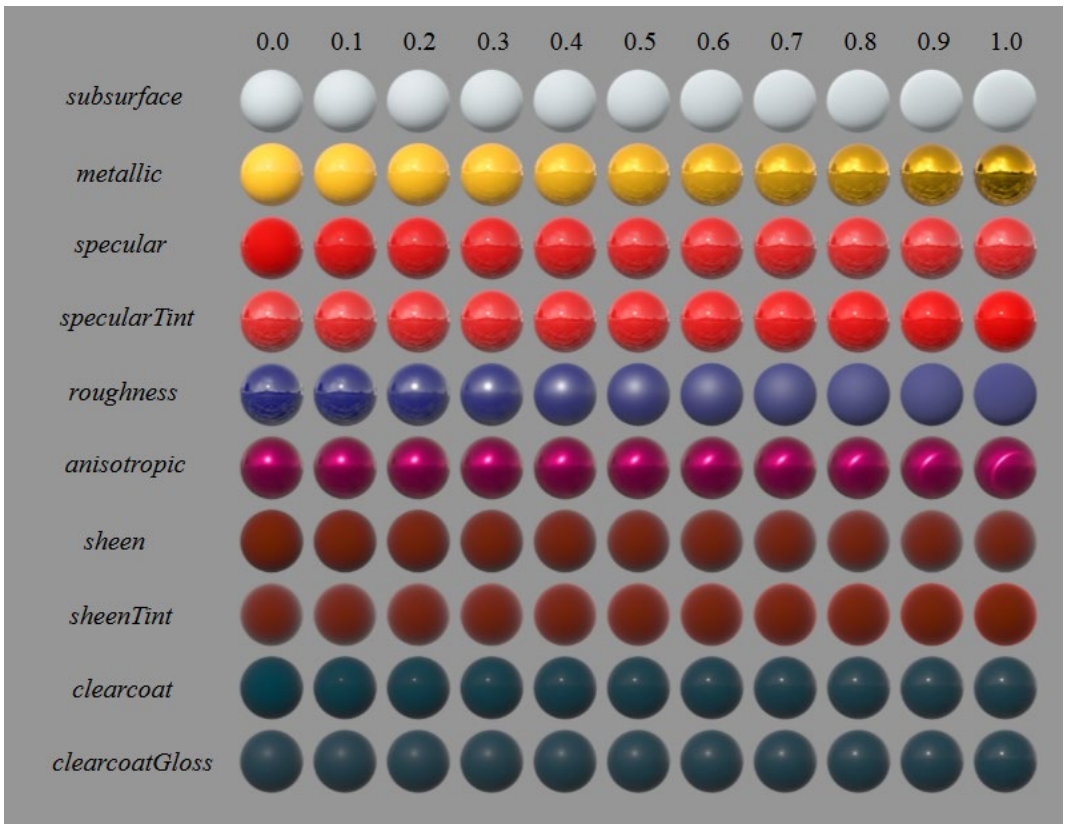

Abbildung 6. Beispiele der verschiedenen Einstellungsmöglichkeiten der Parameter beim „principled“ BRDF Shader von Disney (https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.13)

4.4.3 Erster Einsatz in einer Produktion

Bei dem Film „Wreck-It Ralph“, (im deutschsprachigen Raum unter „Ralph reichts“ erschienen) aus dem Jahr 2012 von Walt Disney Pictures, kam der Shader oder die „Principled Layers“ das erste Mal zum Einsatz. Mit Ausnahme des Hair Shaders (für Haare, Augenbrauen etc.) wurde der neue

Shader für jedes Material in der Produktion benutzt (mit Ausnahme von bestimmten Special Effects).

In weiterer Folge wurden auch für das Belichten beziehungsweise das Lichtsetzten eigene Area-Lights und IBLs (Image Based Lighting) gesampelt. Das war notwendig da zwar der Shader, und mit ihm also auch das 3D Objekt an sich, physikalisch korrekt auf Licht reagiert, die bis dato üblichen Point Lights aber keine physikalisch korrekten Ergebnisse liefern können. Als positiver Nebeneffekt sei noch erwähnt, dass sich das Belichten mit IBLs und Area-Lights als um einiges einfacher für die Light Artists gestaltete, da sie einfacher zu kontrollieren sind, was die Produktion wiederum beschleunigte.

Nach der Produktion von „Wrek-It Ralph“ wurde der Shader für die weiteren Produktionen ohne nennenswerte Änderungen beziehungsweise Modifikationen verwendet.

(vgl. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf S.17)

5 Untersuchungen in dieser Arbeit

5.1 Ausgangssituation

In der Gaming-Industrie hat sich PBR schon länger als Standard durchgesetzt und seit dem Release des Papers des Principled Shaders 2012 und seiner Adaptionen für diverse andere Render Engines auch in Studios etabliert welche nicht auf Gaming spezialisiert sind.

(vgl. https://theblog.adobe.com/gaming-technology-changing-design-visualize-products/)
Ziel dieser Arbeit ist nun, die Unterschiede in den PBR und Non-PBR Workflows mit der Render Engine Cycles aufzuzeigen und die unter dem Punkt 5.3 ausformulierten Forschungsfragen zu beantworten.

5.2 PBR und Non-PBR Workflows

5.2.1 Allgemeiner Non-PBR Workflow

Der allgemeine Non-PBR Workflow in der Render Engine Cycles beinhaltet die verschiedenen Non-PBR Shader wie Diffuse, Glossy, Mix Shader und weitere. Mittels der Fresnel Node kann man PBR ähnliche Effekte erzielen wenn man eben diese Node als Mix Faktor zwischen einem Diffuse und einem Glossy Shader verwendet.

5.2.2 Allgemeiner PBR Workflow

5.2.2.1 Allgemein

Aufbauend auf die Microfacet-Theorie beschreibt der PBR Workflow in die Darstellung von Oberflächen und des damit reagierenden Lichtes mittels sogenannter Maps (Texturen). Somit ist der Grad der Detailierung der Oberfläche, z.B.: der Roughness, abhängig von deren Texel-Space (Texture Pixel). Daraus ergibt sich, je höher die Auflösung der Maps, desto höher ist die Auflösung der Details auf den Oberflächen.

Im Prinzip kann jeder Input des Principled Shaders mit einer eigenen Map versehen werden.

Sogenannte Texture Painting Software, wie zum Beispiel Allegorithmics Substance Painter (https://www.allegorithmic.com/products/substance-painter), versorgen die Render Engine mit genau eben diesen Maps, welche dann mit den dafür vorhergesehenen Inputs der jeweiligen Shader der Render Engine verbunden werden.

Allerdings können für die diversen Inputs auch sogenannte Procedural Textures verwendet werden. Diese bietet den Vorteil, dass das 3D Modell keine UV Maps benötigt sondern sich anderer Texture Mapping Workflows bedient wird.

Generell unterscheidet man allerdings zwischen den zwei Workflows Roughness/Metallic und Specular/Gloss, auf welche in den nächsten beiden Punkten detaillierter Eingegangen wird.

5.2.2.2 Roughness/Metallic und Specular/Gloss

Abhängig von der Software gibt es verschiedene Workflows welche verschiedene Maps als Input benötigen. Beim Roughness/Metallic Workflow wird mittels der Metallic-Map die Menge und Art an Reflektion bestimmt (Metallic/Dielectric) und mittels der Roughness-Map die Streuung/Oberflächenbeschaffenheit definiert.

Der zweite Workflow ist Specular/Gloss welche Maps für diese Inputs benötigt. Man kann die zwei Workflows in der Regel nicht kombinieren.

Allerdings kann man sehr wohl die jeweiligen Maps konvertieren um sie für den jeweilig anderen Workflow einsetzbar zu machen. So kann man zum Beispiel die Gloss-Map invertieren um eine Roughness-Map zu erhalten. (vgl. https://marmoset.co/posts/pbr-texture-conversion/)

Die Art des Workflows wird aber in erster Linie von der benutzen Software vorgegeben. Im Fall der Render Engine Cycles ist es der Roughness/Metallic Workflow.
So muss man zum Beispiel in der Texture Painting Software Substance Painter vor dem Export genau definieren welche Maps und welche Parameter man für die verwendete Render Engine verwendet bzw. welcher der beiden Workflows verwendet wird. Die meisten gängigen Render Engines haben mittlerweile eine Adaption des Principled Shaders von Disney eingeführt.

5.3 Definition der Forschungsfragen

Wie müssen Shader-Kombinationen bei Nicht-PBR Workflows aufgebaut sein um ein vergleichbares Endprodukt wie bei PBR Workflows zu erhalten?

Wie sind die zeitlichen Unterschiede beim Set Up eines Nicht-PBR Workflows und eines PBR Workflows und wie wirken sich diese bei Veränderung des Set- Ups aus am Beispiel der Anpassung an veränderte Lichtsituationen?

Wie groß sind die Unterschiede in den Renderzeiten?

Welche Ressourcen werden beim Rendern benötigt und wie unterscheiden sich diese bei den verschiedenen Workflows?

Wie unterscheiden sich diese Werte bei Rendervorgang über die CPU von den Werten über die GPUs?

6 Praktischer Teil

6.1 PBR Workflow in Cycles

Seit der Blender Version 2.79a beinhaltet die Software mit dem Principled BSDF einen eigenen PBR Shader. Die Principled BSDF Node basiert auf den Grundlagen und Erkenntnissen des von Disney entwickelten Principled Shaders. Dieser Shader ist somit leichter mit anderer Software und Render Engines kompatibel als andere in Blender Cycles integrierte Nodes bzw. Shader. Auch der Workflow mit sogenannten Texture Painting Software, wie etwa Substance Painter von Allegorithmic oder 3DO von Quixel, können direkt mit den dafür vorgesehenen Inputs verbunden werden.

(vgl. https://docs.blender.org/manual/en/latest/render/cycles/nodes/types/shaders/principled.html)

Ähnlich dem Vorbild von Disney sind auch die Inputs des Principled BSDF definiert. Allerdings versteht sich der Principled BSDF auch als sogenannter „Uber Shader“, der mehrere Materialien und deren spezifische Eigenschaften abdecken kann und hat daher mehr Inputs als der originale Principled Shader von Disney. Die Principled BSDF der Blender Version 2.79a hat demnach folgende Inputs:

- Base Color: Definiert die Farbe der Oberfläche (Diffuse oder Metall), kann auch durch eine Textur/Map definiert werden. Der gelbe Input bedeutet, dass drei Werte übertragen werden, bei denen es sich um die RGB Kanäle handelt.
- Subsurface: Dieser Input definiert den Anteil an Diffuse bzw. Subsurface Scattering. Genauer wird dabei der Input als Multiplikator des Subsurface Radius verwendet. Der graue Input bedeutet, dass ein einzelner Wert übertragen wird. Kann auch durch eine Textur/Map definiert werden.
- Subsurface Radius: Die Scattering Distance der einzelnen RGB Kanäle, wird durch den Subsurface Input multipliziert. Der Blaue Input bedeutet, dass drei Werte übertragen werden. Im Gegensatz zu dem gelben Input, handelt es sich aber hierbei nicht um die RGB Kanäle sondern um einen Vektor.

- Subsurface Color: Die Grundfarbe des Subsurface Scattering. Kann auch durch eine Textur/Map definiert werden (gelber Input).
- Metallic: Bestimmt den Anteil von dielektrischem Material (Diffuse und Specular mir möglicher Transparenz) und Metall. In der Regel gibt es keine Abstufungen zwischen diesen beiden Materialien, daher sollte immer entweder der Wert 0 oder 1 gewählt werden. Kann auch durch eine Textur/Map definiert werden (grauer Input).
- Specular: Betrag der dielektrischen Reflexion. Gibt das Reflexionsvermögen entlang der Normalen an. Diese beträgt in den meisten Fällen 0-8%. Da in der Natur aber auch Materialien existieren welche über diese 8% Schwelle kommen, ist beim Input ein Wert über 1 möglich (grauer Input).
- Specular Tint: Färbung der Specular Reflection mittels der Base Color. Da dielektrische Materialien in der Realität keine gefärbten Reflexionen besitzen, ist dieser Input technisch gesehen nicht physikalisch korrekt (grauer Input).
- Roughness: Gibt die Rauheit der Mikrofacettenoberfläche für diffuse und spiegelnde Reflexion an. Im Vergleich ist der Wert des Inputs die Quadratwurzel des Wertes, der in älteren Blender Versionen verwendeten, Glossy BSDF Node (grauer Input).
- Anisotropic: Input für den Betrag der Anisotropie für die Spiegelreflexion (grauer Input).
- Anisotropic Rotation: Betrag der Rotation der Anisotropie (grauer Input).
- Sheen: Einstellungsmöglichkeit für weiche Reflexionen in der Nähe von Ecken und Kanten um Materialien wie Stoff zu simulieren (grauer Input).
- Sheen Tint: Mix aus der Base Color und Weiß für die Reflexion (grauer Input).
- Clearcoat: Extra Layer über den vorhergegangenen Layern für mehr Spiegelung. Besonders für Materialien wie Autolack oder ähnliches (grauer Input).
- Clearcoat Roughness: Rauheit der Clearcoat Specular (grauer Input).
- IOR: Index Of Refraction/Brechungsindex für den Transmission Input (grauer Input).
- Transmission: Einstellungsmöglichkeit von voll deckend/undurchsichtig zu transparent/durchsichtig. Die physikalische Korrektheit wird durch den IOR bestimmt (grauer Input).
- Transmission Roughness: Rauheit für den Transmission Input, verwendet die GGX Distribution (grauer Input).

- Normal: Input für die Normal Maps bzw. Height/Bump Maps der Base Layer. Kann auch durch eine Textur/Map definiert werden, diese müssen allerdings vorher durch eine Normal Node oder Bump Node als eben solche definiert werden (blauer Input).
- Clearcoat Normal: Input für die Normals des Clearcoat Layers. Kann auch durch eine Textur/Map definiert werden, diese müssen allerdings vorher durch eine Normal Node oder Bump Node als eben solche definiert werden (blauer Input).
- Tangent: Extra Einstellungsmöglichkeit für den anisotropen Layer (blauer Input).

(vgl. https://docs.blender.org/manual/en/latest/render/cycles/nodes/types/shaders/principled.html)

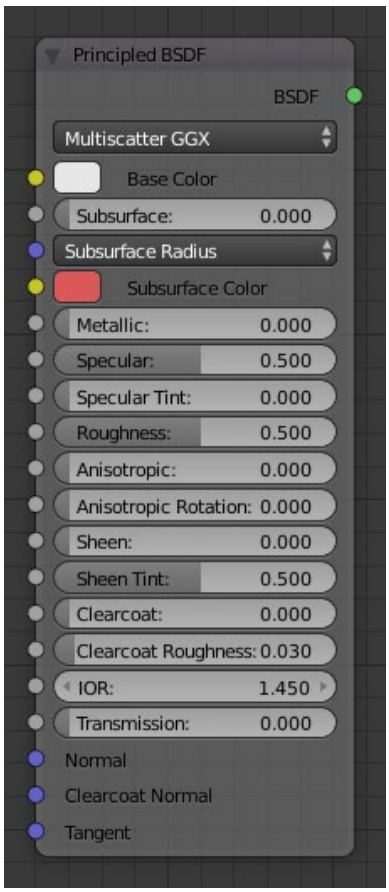

Abbildung 7. Principled BSDF Node mit den Inputs (vom Autor mit Blender 2.79 erstellt)

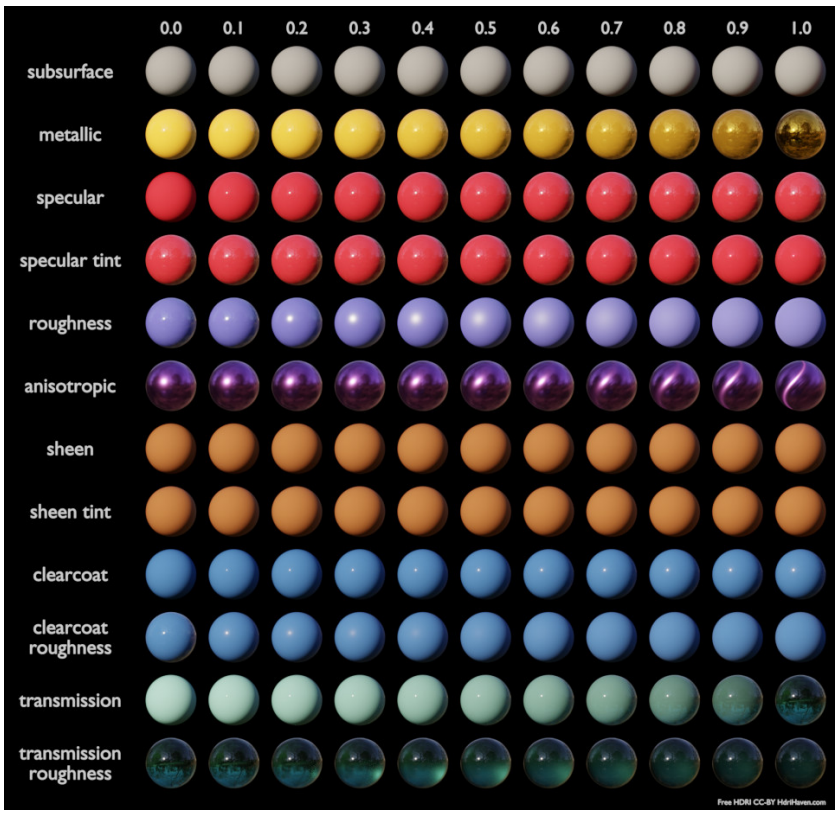

Abbildung 8. Auswirkungen bei Veränderten Werten in den Inputs des Principled BSDF (https://docs.blender.org/manual/en/latest/render/cycles/nodes/types/shaders/principled.html)

Weitere Eigenschaften der Principled BSDF Node:

Distribution: Sowohl die GGX Distribution als auch die Multiple-Scattering GGX Distribution können verwendet werden. Der Unterschied hierbei besteht darin, dass GGX die schnellere Distribution ist, physikalisch aber als weniger akkurat beschrieben wird. Allerdings erlaubt nur die GGX Distribution die Verwendung der Transmission Roughness. Die Multiple-Scattering Distribution, welche auch die standardmäßige Distribution ist, berechnet mehrere Bounces zwischen den Microfacets.

Subsurface Methode: Auch beim Rendering von Materialien mit Subsurface Komponenten gibt es zwei verschiedene Methoden, die Christensen-Burley Methode und die Random Wolk Methode. Die Christensen-Burley Methode ist eine Annäherung an die Streuung in und durch Volumen während die

Random Walk Methode physikalisch akkurater ist da sie keine Annäherung sondern die tatsächliche Streuung innerhalb des Volumens berechnet. Die Random Walk Methode ist daher um einiges ressourcen- und rechenintensiver als die Christensen-Burley Methode.

(vgl.
http://graphics.pixar.com/library/ApproxBSSRDF/approxbssrdfslides.pdf)

Allerdings können bei nicht geschlossenen Meshes, bei sich überlappenden Faces/Flächen oder bei Löchern in einem Mesh Fehler auftreten.

(vgl.
https://docs.blender.org/manual/en/latest/render/cycles/nodes/types/shaders/principled.html)

6.2 Versuchsaufbau

Um die Fragestellung untersuchen zu können, werden folgende Versuche mit der Software Blender 2.79 und der Render Engine Cycles vorgenommen. Es werden zwei 3D Modelle beziehungsweise Szenen erstellt, eine mit sogenanntem Hardsurface-Modeling und eine mit sogenanntem Organischen-Modeling. Die jeweiligen Szenen werden dann jeweils einmal mittels des PBR Shaders (Principled BSDF) und einmal mit der herkömmlichen NON PBR Methode geshadet und mit Texturen versehen. Anschließend werden beide mit der gleichen Lichtsituation in der Render Engine Cycles gerendert. Anschließend findet die Auswertung statt.

In den weiteren Punkten wird genau auf den Aufbau die Lichtsituation und auf die Kriterien für die Auswertung eigegangen.

6.2.1 3D Modelle und deren materialspezifische Unterschiede

In der 3D Modellierung unterscheidet man grundlegend unter zwei verschiedenen Arten, welche auch materialspezifische Unterschiede vorweisen.

6.2.1.1 Hardsurface

Als erste der zwei verschiedenen Arten, sei das sogenannte Hardsurface-Modeling genannt.

Als sogenannte Hardsurface Modelle, werden vor allem 3D Modelle bezeichnet welche statische Materialeigenschaften besitzen, beziehungsweise in der realen Welt von Hand gefertigt wären. So fallen unter diese Kategorie vor allem 3D Modelle wie: Fahrzeuge, Gebäude, Werkzeuge, Möbel, nahezu alle Assets welche mit dem zukunftsorientierten Genre des Science Fiction befassen, etc.

Bei der Modellierung selbst, meist kommt das sogenannte Box- Modeling zum Einsatz. Bei dieser Methode des Modellierens, ist die Ausgangsgeometrie meist ein Würfel oder Quader aus dessen Flächen, in der Software Blender auch Faces genannt, neue Geometrie extrudiert wird und so nach und nach komplexere Strukturen geschaffen werden können. In der 3D Software Blender 2.79 stehen dafür auch extra Tools wie der Boolean Modifier zur Verfügung. Mit diesem kann man komplexe Modelle mittels verschiedener Operationen welche auf dem booleschen Prinzip beruhen, aus zwei separaten Modellen erstellen. Bei dem von George Boole entwickelten Verfahren, kann mittels der boole´schen Variabel ein Wert nur zwei Zustände annehmen.

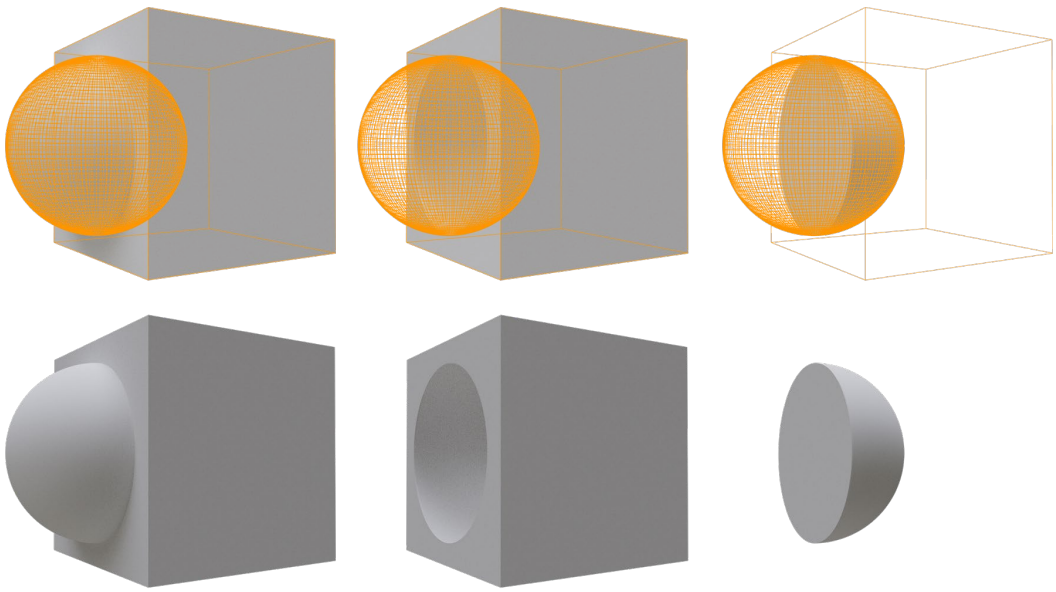

Abbildung 9. Würfel mit Boolean Modifier mit den drei Hauptoperationen Union (beide Meshes werden zu einem Mesh), Difference (aus dem Mesh mit dem Modifier wird das sich überschneidende Volumen des anderen Meshes

ausgeschnitten), Intersect (das Volumen des Meshes mit dem Modifier wird durch das Volumen des sich überschneidenden Meshes definiert) v.l.n.r. (vom Autor mit Blender 2.79 erstellt)

Mit dem Aufkommen von immer besser werdenden Retopology-Algorithmen gewinnen auch diverse Digital-Sculpting Methoden immer mehr an Relevanz im Bereich des Hardsurface-Modeling.

Im Rahmen dieser Arbeit wurde mit der 3D Software Blender 2.79, anhand eines Referenzfotos, eine Gaslaterne modelliert welche einmal mit dem NON PBR und einmal mit dem PBR Workflow geshadet wird. Des Weiteren wird die PBR Texturing Software Substance Painter von Allogerithmic verwendet.

6.2.1.2 Organische 3D Modelle

Die zweite der grundlegenden Arten bei der Konzipierung und Erstellung von 3D Modellen is das sogenannte organische Modeling.

Als organische Modelle werden in der Regel all jene 3D Objekte zusammengefasst welche in ihrer Struktur und Beschaffenheit organische Eigenschaften besitzen. Dazu gehören: Character, Pflanzen, abstrakte sich verformende Objekte, etc. Auch werden manchmal Objekte als „organisch“ bezeichnet, welche eigentlich in die Kategorie des Hardsurface fallen würden, weil sie (oft stilisiert) animierbar/animiert sind und daher aus ästhetischen Gründen eher Eigenschaften organischer Strukturen aufweisen als Eigenschaften von Hardsurface Modellen. Eine weitere typische Materialeigenschaft organischer Modelle ist das Subsurface Scattering welches nahezu ausschließlich bei organischen 3D Modellen auftritt.

Ähnlich dem Hardsurface-Modeling kommen auch bei dieser Art der 3D Objekt Erstellung die Methoden des Box-Modeling zum Einsatz. Jedoch ist vor allem beim Modeling von organischen Objekten der Digital-Sculpting Workflow sehr verbreitet. Beim Digital Sculpting gibt es vor allem zwei Methoden welche auch manchmal kombiniert in unterschiedlichen Stadien des Modellierungsprozesses vorkommen können. Im Sculpting Mode der Software Blender 2.79 heißen diese Sculpting und Dynamic Topology.

Beim regulären Sculpting Modus werden die in dem 3D Mesh vorhandenen Vertecies (Vertex = Punkt in einem Mesh, drei Prunkte bilden ein Trigon, vier Punkte ein Face) mittels der Sculpting Tools, auch Brushes genannt, verschoben. Falls zu wenig Geometrie für die gewünschte Bearbeitung vorhanden ist, kann ein

sogenannter Multiresolution Modifier verwendet werden welcher auf eine höhere Dichte an Vertecies interpoliert. So kann mit jedem Level an Detail dass man hinzufügt ein weiterer Level an Dichte des Meshes geschaffen werden. Dieser Workflow ist weitgehend non-destructive, das heißt, es wird dem Grund-Mesh keine zusätzliche Geometrie hinzugefügt (ausgenommen der Multires. Modifier wird auf das Mesh angewandt).

Die zweite Methode des Sculptings nennt sich Dynamic Topology (in Blender 2.79). Im Unterschied zu dem herkömmlichen Sculpting wird mittels der Brushes tatsächliche neue Geometrie auf dem Mesh erzeugt, außerdem werden etwaige auf dem Ausgangs-Mesh angewandte Modifier ignoriert. Mittels der sogenannten Retopology wird mit dem gesculpteten Mesh mit schlechter Topology ein neues Mesh mit guter (rein aus Quads oder Tris) Topology erstellt. Damit fallen weitere Schritte wie UV Unwrapping, Texture Painting, Rigging und Animation leichter oder werden erst dadurch ermöglicht.

Im Rahmen dieser Arbeit wurde mit der 3D Software Blender 2.79, anhand eines Referenzfotos, eine Obstschale modelliert welche einmal mit dem NON PBR und einmal mit dem PBR Workflow geshadet wird. Des Weiteren wird die PBR Texturing Software Substance Painter von Allogerithmic verwendet.

6.2.2 Belichtung und Lichtsituation

Da die 3D Modelle anhand eines Referenzfotos modelliert und geshadet werden, wird auch die Lichtsituation möglichst an die des Referenzfotos angepasst. Hierfür wird eine 360 Grad Aufnahme des Raumes gemacht in welchem die Referenzfotos aufgenommen werden. Die 3D Szenen werden dann mittels IBL (Image Based Lighting) belichtet.

In der 3D Software Blender 2.79 wird hierfür in dem Node Editor unter dem Reiter World Tab eine neue Environment Node erstellt welche mit dem Input der Background Node verbunden wird. Weitere Steuerungsmöglichkeiten über Rotation, Skalierung und Position hat man über die Texture Coordinate und die Mapping Nodes.

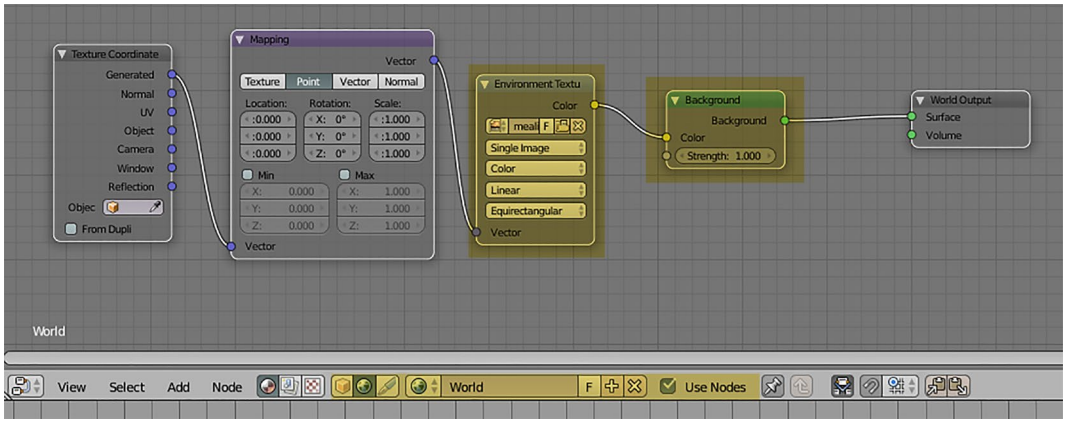

Abbildung 10. Node basiertes IBL Set-Up in der 3D Software Blender 2.79 (vom Autor mit Blender 2.79 erstellt)

Des Weiteren werden zwei Area-Lights welche als Key- und Fill-Light eingesetzt werden in der 3D Szene verwendet.

Außerdem werden die jeweiligen 3D Szenen noch in einer veränderten Lichtsituation gerendert, um die Unterschiede der beiden Shading-Techniken in sich verändernden Lichtsituationen ausarbeiten zu können und vor allem eine Aussage, über die etwaige Notwendigkeit der Veränderung der Shadingtrees bei veränderter Lichtsituation, treffen zu können.

6.2.3 Aufbau

Folgende Unterpunkte behandeln sowohl den Aufbau welcher für die Erstellung der Referenzfotos benötigt wird, als auch das genaue Set-Up welches innerhalb der Software Blender 2.79 verwendet wird und wie die 3D Szene gestaltet und aufgebaut ist. Die 3D Szenen werden der Umgebung und vor allem der Lichtsituation bzw. der Belichtung der Referenzfotos bestmöglichst angeglichen.

6.2.3.1 Referenzfotos

Das Set-Up für die Referenzfotos wird, sowohl für die Objekte welche unter die Kategorie Hardsurface 3D Modell fallen als auch für jene welche der Kategorie organische 3D Modelle zuzurechnen sind.

Innerhalb einer weißen Foto Box, werden die jeweiligen Objekte platziert und zusätzlich zu dem natürlichen Raumlicht wird sowohl mit einem Key- als auch mit einem Fill-Light die Szenerie beleuchtet. Die zwei zusätzlichen Lichter helfen die natürlichen Strukturen der Referenzobjekte visuell besser abzubilden.

Der Aufbau für die Aufnahme der Referenzfotos kann mittels dieser skizierten Darstellung beschrieben werden.

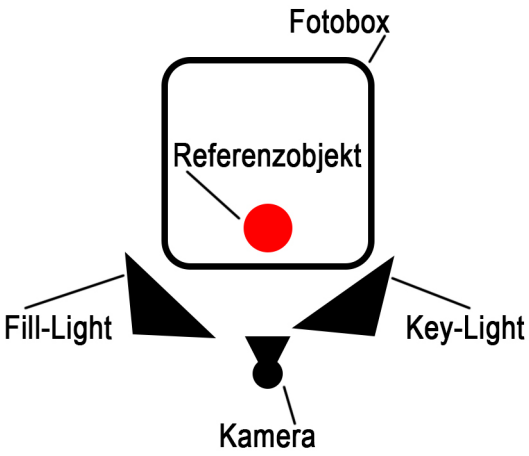

Abbildung 11. Grafische Darstellung des Aufbaus der Aufnahmesituation für die Referenzfotos (vom Autor erstellt)

Für die Aufnahme wird eine Panasonic Lumix GH5 auf einem Stativ mit einem Panasonic Lumix G Vario 14 - 140 mm 3.5 - 5.6 ASPH OIS Objektiv verwendet. Hierbei gilt es zu beachten, dass die Panasonic Lumix GH5 keine Vollformatkamera sondern eine sogenannte Micro Four Thirds Kamera ist und einen CMOS Chip mit den Abmessung 17,3 mm x 13 mm hat, daraus ergibt sich ein Cropfaktor von 2.

Mittels Adobe Lightroom CC wurden die Fotos einer generellen Farbkorrektur unterzogen, die sich vor allem auf Anpassung des Weißabgleiches sowie auf rudimentäre Änderungen in der Belichtung beliefen.

Abbildung 12. Referenzfoto der Lampe welche das Hardsurface Modell darstellt (vom Autor erstellt)

Abbildung 13. Referenzfoto des Obsttellers, welcher die Basis für die organischen Modelle ist (vom Autor erstellt)

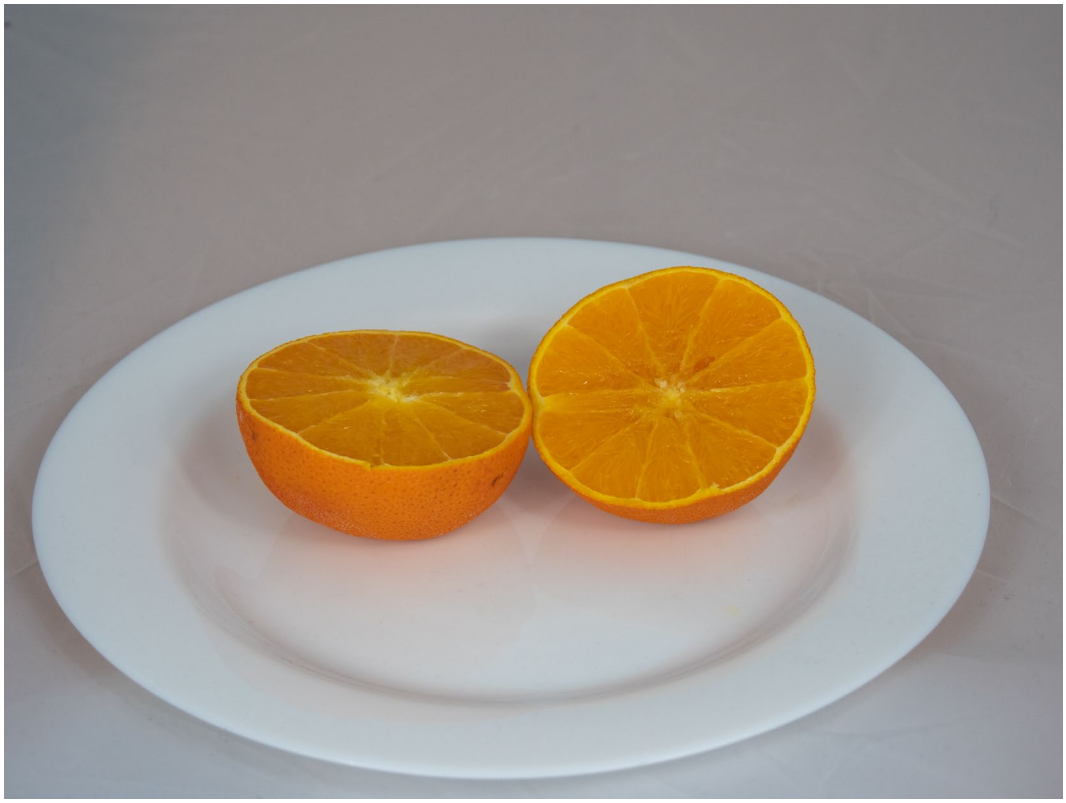

Abbildung 14. Referenzfoto als Ergänzung der organischen Modelle, bei welchem eine halbe Orange dargestellt wird (vom Autor erstellt)

6.2.3.2 3D Szenen

Für die 3D Szenen wird die Kamera bestmöglich an die Spezifikationen der Panasonic Lumix GH5 angepasst. In der 3D Software Blender 2.79 bestehen diese Einstellungsmöglichkeiten aus: Focal Length (in Millimeter oder Field of View) und Sensor Größe.

Bei Bedarf wird das Addon BLAM für Blender 2.79 eingesetzt. BLAM ist ein Tool um die 3D Kamera anhand eines Referenzfotos auszurichten und zu kalibrieren. (Link zum Addon https://stuffmatic.com/blam-blender-camera-calibration-toolkit/).

Die 3D Szene wird an die Skalierungen der Referenzfotos angepasst.

Die bei den Aufnahmen der Referenzfotos zusätzlich aufgenommene 360 Grad Aufnahme des Raumes wird in der 3D Software Blender 2.79 als IBL importiert. Zusätzlich werden zwei Area-Lights als Key- und Fill-Light wie bei den Referenzfotos erstellt.

Die jeweiligen 3D Objekte werden platziert und anschließend geshadet bzw. mit Texturen versehen.

Anschließend wird jede Szene einzeln über zwei NVIDIA GTX 1080 Ti Grafikkarten und anschließend über die CPU mittels der Render Engine Cycles gerendert, wobei zu erwähnen ist, dass die Karten von verschiedenen Herstellern und zwar namentlich Zotac und MSI sind. Weitere relevante Spezifikationen der Workstation sind:

- CPU: AMD Threadripper, 16 Cores
- RAM: 64GB
- Festplatte: 1TB

6.2.4 Erstellung der einzelnen 3D Objekte

Dieser Punkt behandelt die Erstellung der verschiedenen 3D Modelle. Mittels der Screenshots aus dem 3D Programm Blender 2.79 kann man vor allem Rückschlüsse auf die Topologie der einzelnen Modelle schließen. Des Weiteren sind die Seams in Rot dargestellt. Die Seams geben an, an welchen Stellen das 3D Modell für das UV Unwrapping geteilt wird.

6.2.4.1 Hardsurface 3D Modelle

Als Hardsurface 3D Modell wurde eine Petroleum Laterne modelliert. Der Grund für die Wahl dieses Objektes war, dass die Laterne sowohl metallische, dielektrische als auch, mit dem Glaszylinder, transparente Materialeigenschaften aufweist und somit optimal für eine derartige Materialstudie geeignet ist. Für die Modellierung wurde hauptsächlich Box- Modelling zusammen mit einem Subdivision Surface Modifier verwendet. Für bestimmte Teile des Modells wurden auch sogenannte Curves verwendet um etwa den Bügel der Laterne zu modellieren.

Abbildung 15. 3D Modell Petroleum Laterne mit Glaszylinder (vom Autor mit Blender 2.79 erstellt)

Das 3D Modell wurde mittels UV Unwrapping für etwaiges Texturieren mittels der UV Koordinaten vorbereitet, wobei für den Texturierungsprozess auch andere Methoden angewendet werden können.

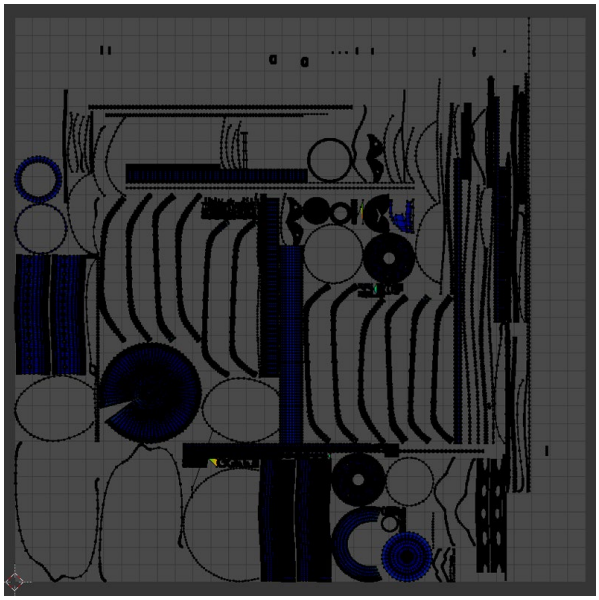

Abbildung 16 Darstellung des UV Unwrappings der einzelnen Teile der Petroleum Laterne ohne Glaszylinder (vom Autor mit Blender 2.79 erstellt)

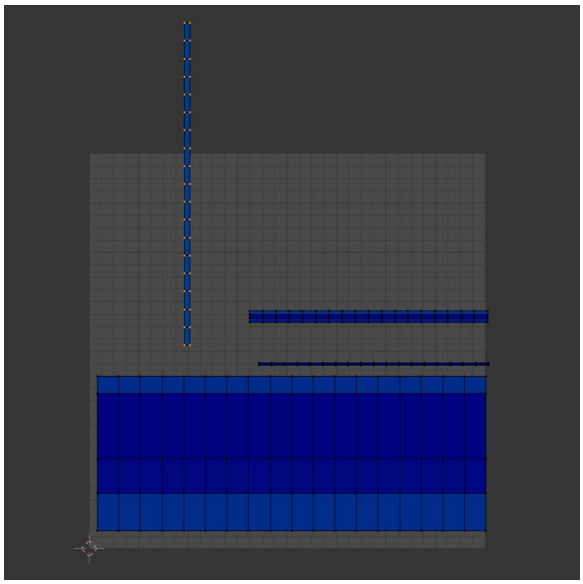

Abbildung 17 Darstellung des UV Unwrappings des Glaszylinders der Petroleum Lampe (vom Autor mit Blender 2.79 erstellt)

Die Einfärbung der UV Meshes sind ein Indikator dafür wie sehr die Textur auf dem eigentlichen 3D Objekt verzerrt wird. Blau bedeutet es gibt keinerlei Verzerrung, welche sich bei Farbveränderung ins rötliche Farbspektrum allerdings immer weiter steigert.

Weiters ist auch der von den UVs eingenommene Platz von Bedeutung, da er den sogenannten Texel-Space definiert. In 3D Texture Painting Softwaren wie Allegorithmic Substance Painter definiert der Texel-Space wieviel Pixel des, in seiner Größe vordefinierten, 2D Bildes auf das 3D Modell angewendet werden.

6.2.4.2 Organische 3D Modelle

Das Stillleben besteht aus einer Birne, einem Apfel, einer Banane, einer Orange, einer halben Orange sowie aus einem Teller auf welchem die Objekte platziert sind. Es wurden sowohl die klassischen Modellierungstechniken wie Box-Modelling aber auch Sculpting verwendet um die Oberfläche als weniger einheitlich und homogene Fläche darzustellen.

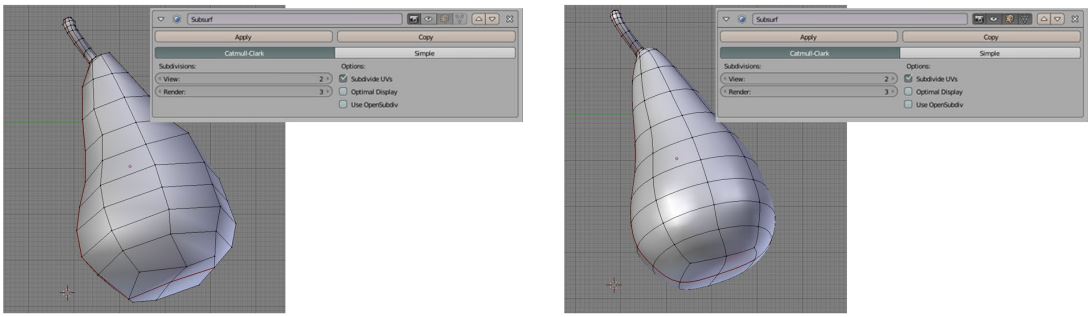

Abbildung 18. 3D Modell Birne (Low- und High- Poly 3D Modell) (vom Autor mit Blender 2.79 erstellt)

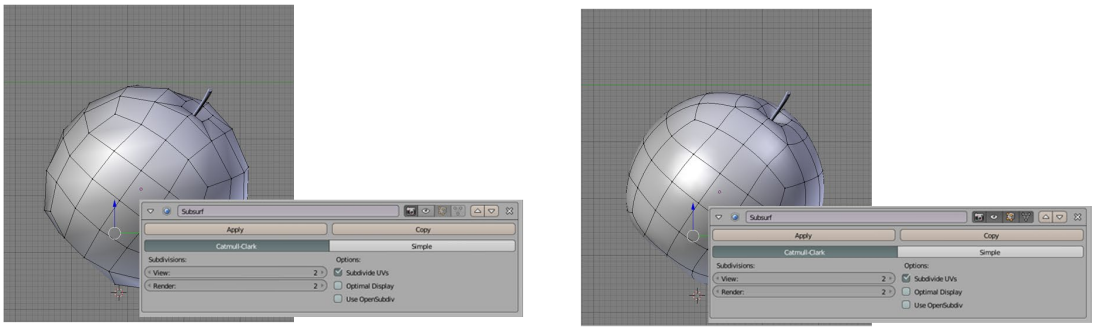

Abbildung 19. 3D Modell Apfel (Low- und High- Poly 3D Modell) (vom Autor mit Blender 2.79 erstellt)

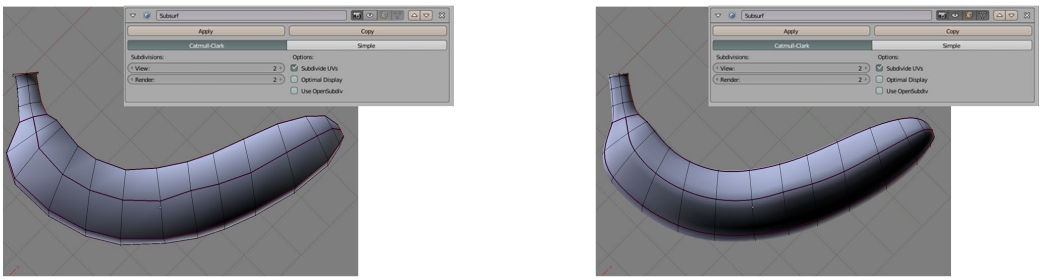

Abbildung 20. 3D Modell Banane (Low- und High- Poly 3D Modell) (vom Autor mit Blender 2.79 erstellt)

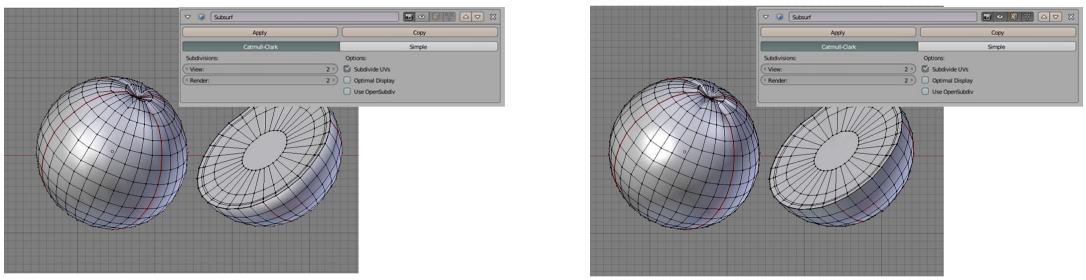

Abbildung 21. 3D Modell Orange und halbe Orange (Low- und High- Poly 3D Modell) (vom Autor mit Blender 2.79 erstellt)

Es wurde bei der Erstellung der 3D Modelle vor allem das sogenannte Box-Modelling sowie einfaches Sculpting verwendet um auf ein langwieriges und für die Zielsetzung dieser Arbeit irrelevantes Retopology zu verzichten.

Jedes 3D Modell wurde mittels UV Unwrapping für etwaiges Texturieren mittels der UV Koordinaten vorbereitet, wobei für den Texturierungsprozess auch andere Methoden angewendet werden können.

Jedes Objekt besitzt eine eigene UV- Map, sodass der Texel- Space bei jedem Objekt individuell transformierbar und einstellbar ist.

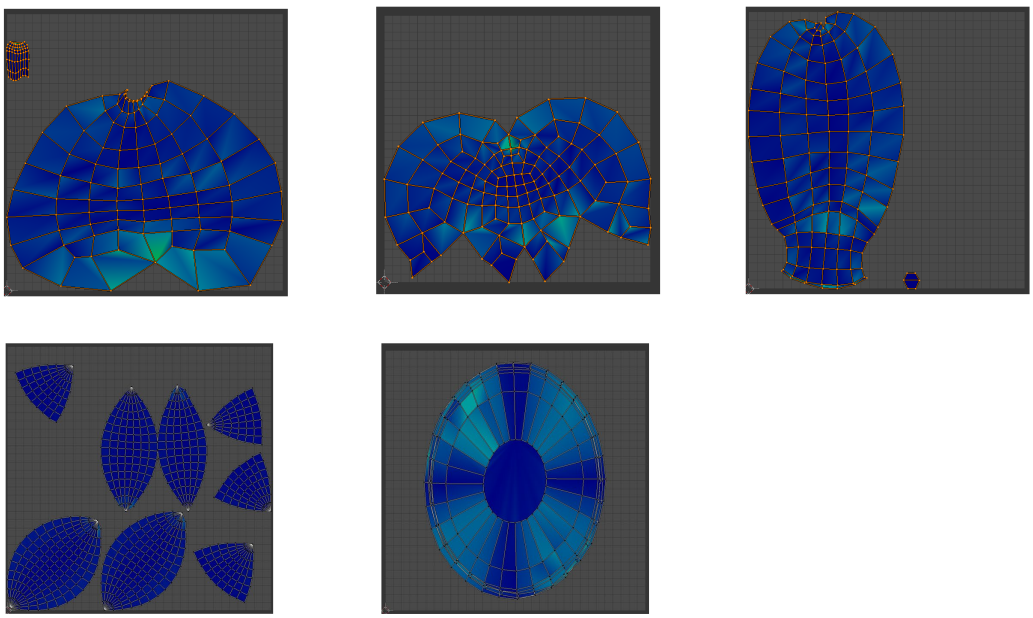

Abbildung 22. Darstellung des UV Unwrappings der einzelnen 3D Modelle (Birne, Apfel, Banane, Orange und Schnittfläche der halben Orange v.l.n.r.). (vom Autor mit Blender 2.79 erstellt)

6.2.5 Kriterien für die Auswertung

Die Kriterien für die Auswertung sind folgend definiert.

- Zeit für das Set-Up des Shaders
- Ressourcenbedarf während des Rendervorgangs
- Renderzeit gesamt für die Szene
- Zeit für Anpassungen bei geänderter Lichtsituation
- Bewertung ob NON PBR und PBR Workflows visuell vergleichbar sind

Anhand dieser Kriterien werden die Ergebnisse bewertet und die Forschungsfragen beantwortet.

6.3 Shading der 3D Objekte

6.3.1 NON PBR in Blender

Prinzipiell wird ein dielektrisches aber auch metallisches Material in der NON PBR Methode wie folgt aufgebaut:

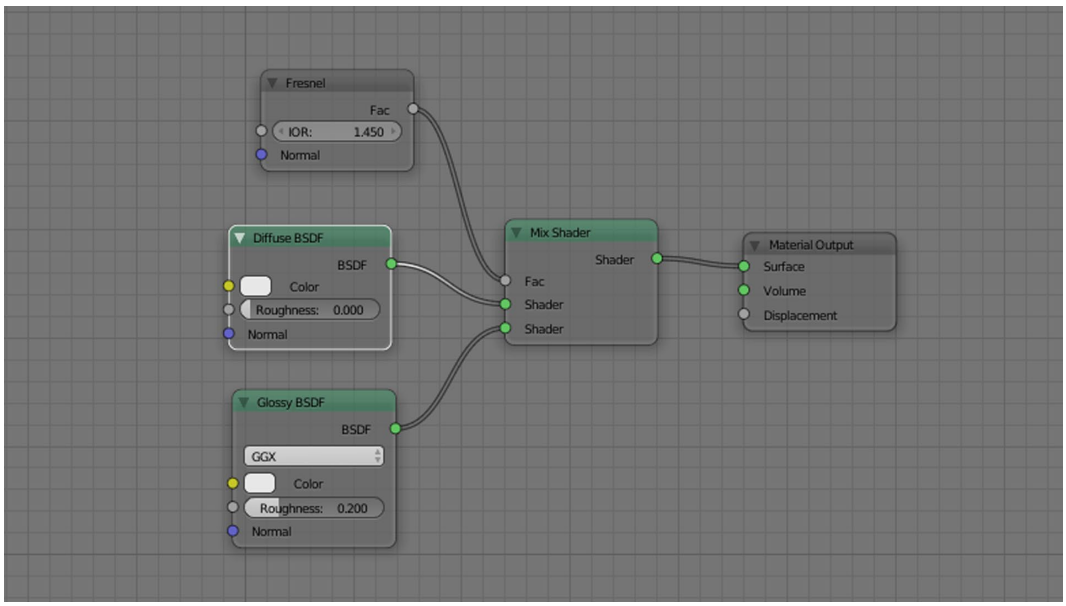

Abbildung 23. Basis eines Materials mit der NON PBR Methode in Blender 2.79 (vom Autor mit Blender 2.79 erstellt)

Ausgehend an der Material Output Node wird ein Mix Shader an den Output angefügt. Der Faktor des Mix Shaders, welcher definiert in welchem Verhältnis die Shader gemixt werden, wird von einer Fresnel Node und deren IOR Wert definiert.

Die beiden weiteren Inputs der Mix Shader Node werden mit jeweils einer Diffuse BSDF Node und einer Glossy BSDF Node versehen.

Der Hauptunterschied bei der Erstellung dielektrischer und metallischer Materialien wird über den Color Input der Glossy BSDF Shader Node definiert. Metallische Materialien haben in der Regel gefärbte Reflektionen während dielektrische Materialien diese Eigenschaft nicht teilen.

Der Roughness Wert muss über beide Nodes (Diffuse und Glossy) definiert werden. Ebenso ist es für Normal und/oder Bump Maps notwendig mit allen drei Inputs der Diffuse, der Glossy und der Fresnel Node versehen zu werden.

6.3.2 PBR in Blender

Der Workflow des PBR Shadings in Blender 2.79 mittels der Principled BSDF Shader Node wurde in Punkt 6.1 dieser Arbeit erläutert.

Der Basis Aufbau der Principled BSDF Shader Node im Node Editor in der 3D Software Blender 2.79 wird in der folgenden Abbildung veranschaulicht.

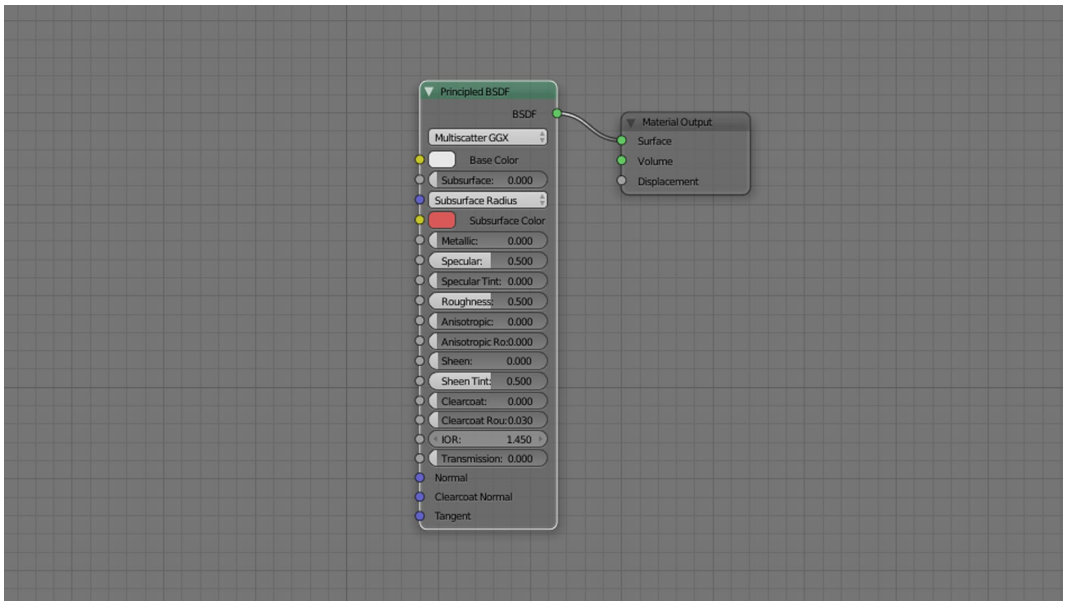

Abbildung 24. Basis eines Materials mit der PBR Methode mittels der Principled BSDF Shader Node in Blender 2.79 (vom Autor mit Blender 2.79 erstellt)

6.3.3 Shading der Laterne

Das 3D Modell der Laterne setzt sich in erster Linie aus zwei Hauptmaterialien zusammen. Ersteres besteht es aus der metallischen Oberfläche des Gehäuses der Lampe, welche allerdings mit dem Rost und den allgemeinen Gebrauchsspuren auch dielektrische Materialeigenschaften aufweist. Das zweite Material, für welches auch ein eigener Shader angelegt wird, definiert den Glaszylinder. Dieser Shader wird vorrangig aus einem klassischen „Glasshader“

(in dem jeweiligen Workflow) bestehen welchem die Gebrauchsspuren und Ähnliches hinzugefügt werden.

Anschließend werden die jeweiligen Shader Trees der NON PBR-, der PBR- und der PBR mittels Substance Painter Methode beschrieben und erklärt.

6.3.3.1 Shading Laterne NON PBR

Um fotorealistische Ergebnisse, welche anhand der Referenzaufnahmen angefertigt gerendert und verglichen werden, ohne einen PBR Shader Node erzielen, müssen mehrere Shader Nodes mittels etwa einem Mix Shader Node kombiniert werden um die physikalisch akkurate Darstellung garantieren zu können (siehe Punkt 6.3.1). Des Weiteren wird der Shadertree bei dem Glaszylinder Material noch um ein Glass Material mittels einer Glass BSDF Shader Node ergänzt.

NON PBR Glas Material:

Ausgehend von dem NON PBR Basis Shader (siehe Abbildung 23) wird dieser mit einem Glas BSDF Shader mittels einer Mix Shader Node gemixt. Der Faktor des Mix Shaders wird mittels einer prozedural generierten Noise Textur determiniert. Hierbei wird mittels Schwarz und Weiß Werten definiert auf welchen Teil der Oberfläche des Objektes der Mix Shader wie angewendet wird. Schwarz steht für den Wert 0 und Weiß für den Wert 1 und der Faktor Input wird über einen Float Wert festgelegt. (siehe Abbildung 25)

Daraus ergibt sich, dass je eher ein Wert gegen 0 geht („schwärzer ist“) desto mehr wird die Glas BSDF Node angewandt, je eher ein Wert gegen 1 geht desto mehr wird die Diffuse BSDF Node angewandt. Die Werte werden anschließend mittels einer Color Ramp (Vergleichbar mit den Levels bei Adobe Photoshop) noch in Kontrast und Verlauf verändert um den gewünschten Look zu erzielen.

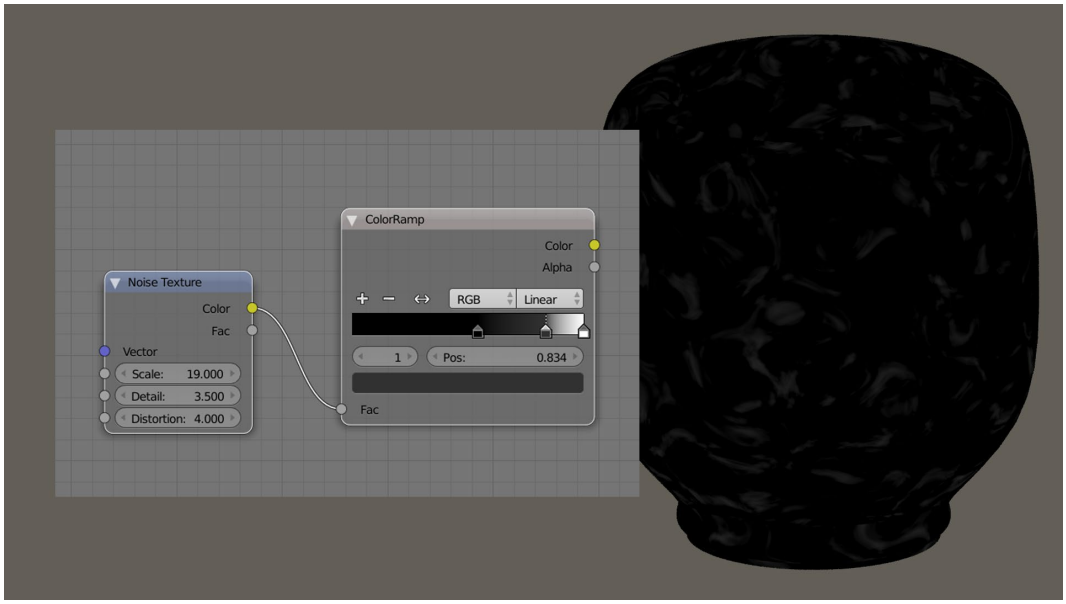

Abbildung 25. Noise Textur für den Faktor Input der ersten Mix Shader Node (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Der zweite Layer an Staub an den Rändern, Wölbungen, Kanten und Enden des Zylinders wurde mittels der Geometrie Node und deren Pointiness Output und einer weiteren prozeduralen Noise Textur generiert. Mittels einer Math Node im Modus Overlay (ähnlich den Layer Blending Modes bei Adobe Photoshop) wurden diese beiden Texturen miteinander zu einer Textur fusioniert. (siehe Abbildung 26)

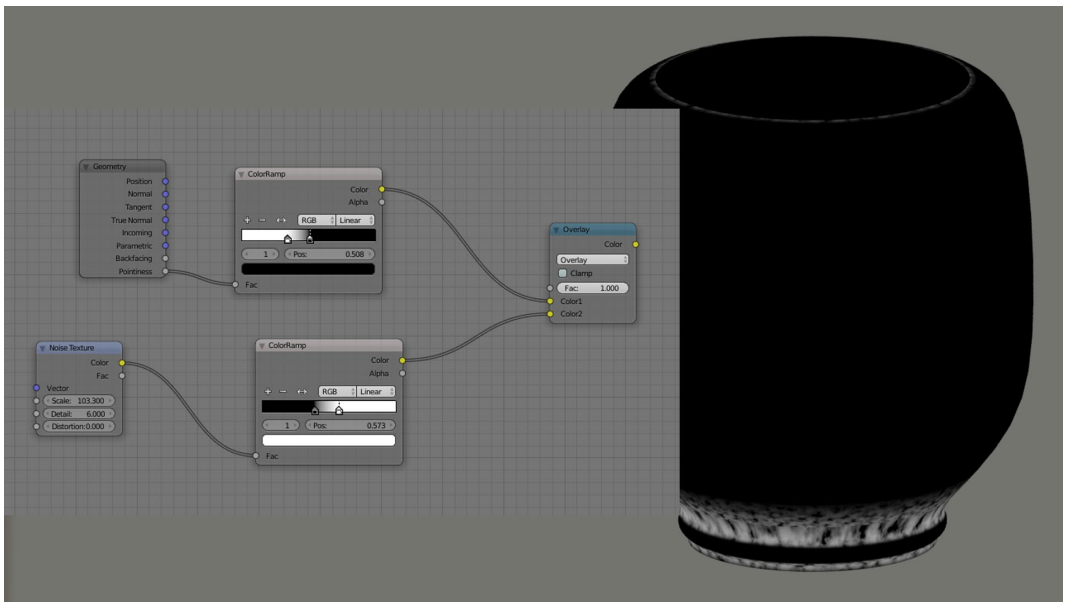

Abbildung 26. Noise Textur mit Geometrie Node und Math Node für den Faktor Input der zweiten Mix Shader Node (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

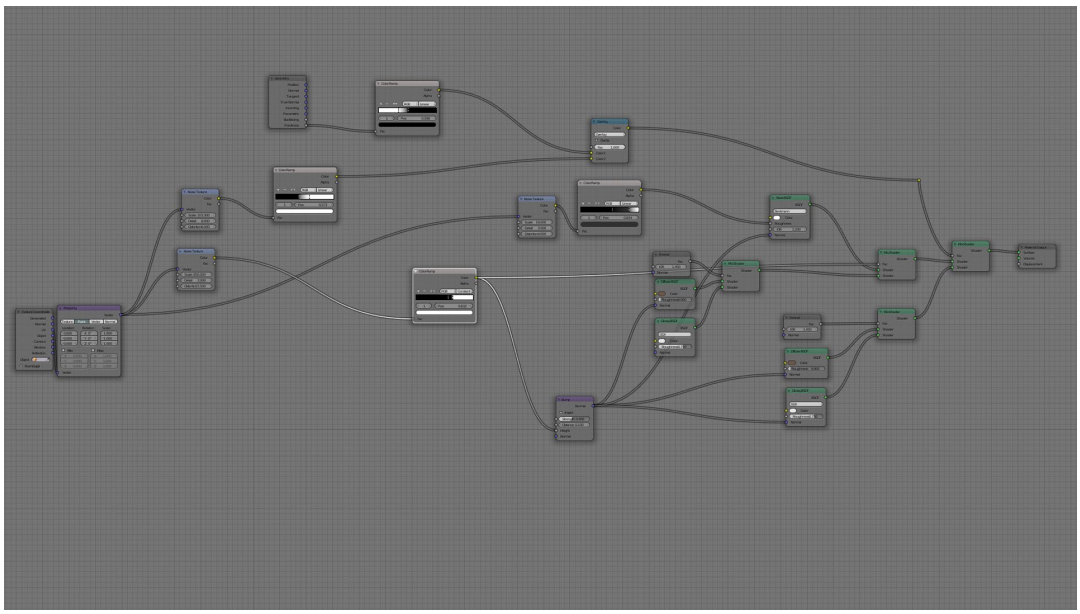

Abbildung 27. Shader Tree des NON PBR Glas Materials (gesamt) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

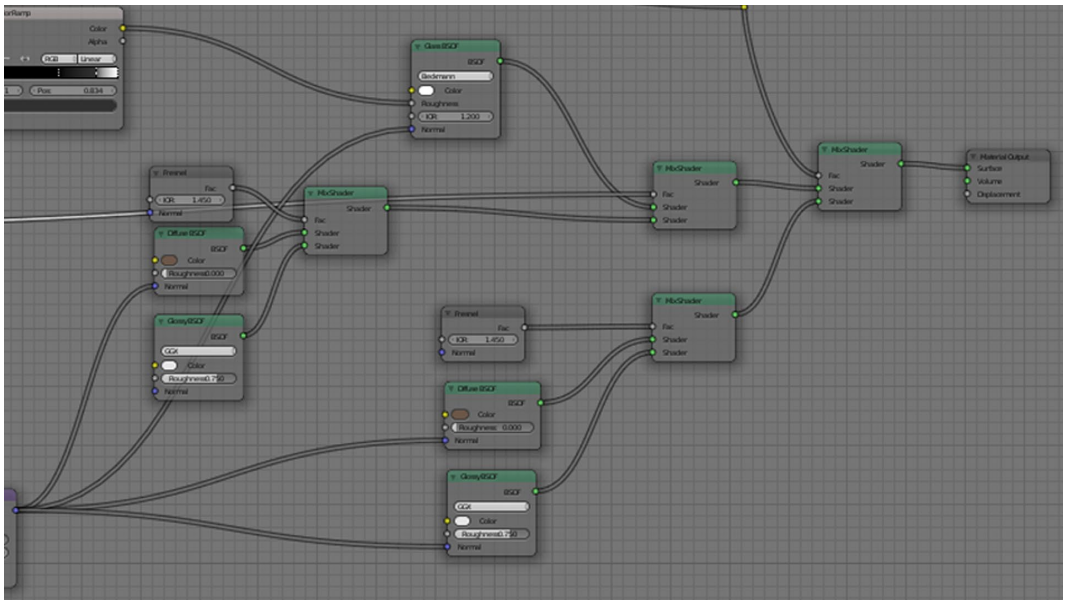

Abbildung 28. Shader Tree des NON PBR Glas Materials (Close Up Basis Shader) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Abbildung 29. Low Resolution Rendering des Glaszylinders mit dem NON PBR Glas Shader (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

NON PBR Gehäuse Laterne Material:

Auch beim Laternengehäuse wird von dem Basis Shader aus Abbildung 23 ausgegangen. Anschließend werden zwei Shader Trees erstellt, eine für den dunklen Lack der Lampe und einen für den Rost und diese dann mit einander mittels Mix Shader Node zum finalen Shader verbunden.

Ersterer besitzt eine prozedurale Noise Textur welche die Roughness Werte der Diffuse und der Glossy Node definiert sowie einer zweiten Noise Textur welche durch den Height Input der Bump Node geschleift und mit den jeweiligen Normal Inputs der Diffuse, Glossy und Fresnel Node verbunden werden. Die Farbe wird durch den Color Input der Diffuse Node mittels eines RGB Wertes angegeben.

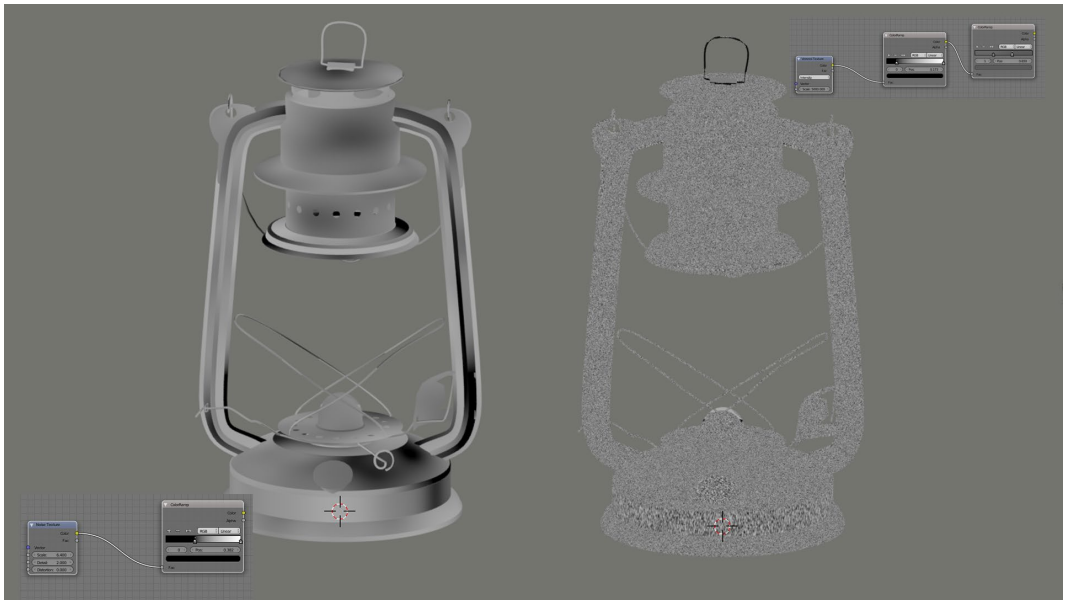

Abbildung 30. Noise Texturen mit Color Ramps als Input für die Roughness Werte (links) und die Bump Node (rechts) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Der zweite Shader Tree welcher den Rost darstellt besitzt ebenfalls eine eigne Noise Textur welche die Roughness und auch den Bump Wert definiert. Zusätzlich wird der Color Input der Diffuse Node ebenfalls durch eine Farbverlauf, mittels Color Ramp Node, und durch den im Faktor Input angeschlossenen eigenen Noise Textur gespeist.

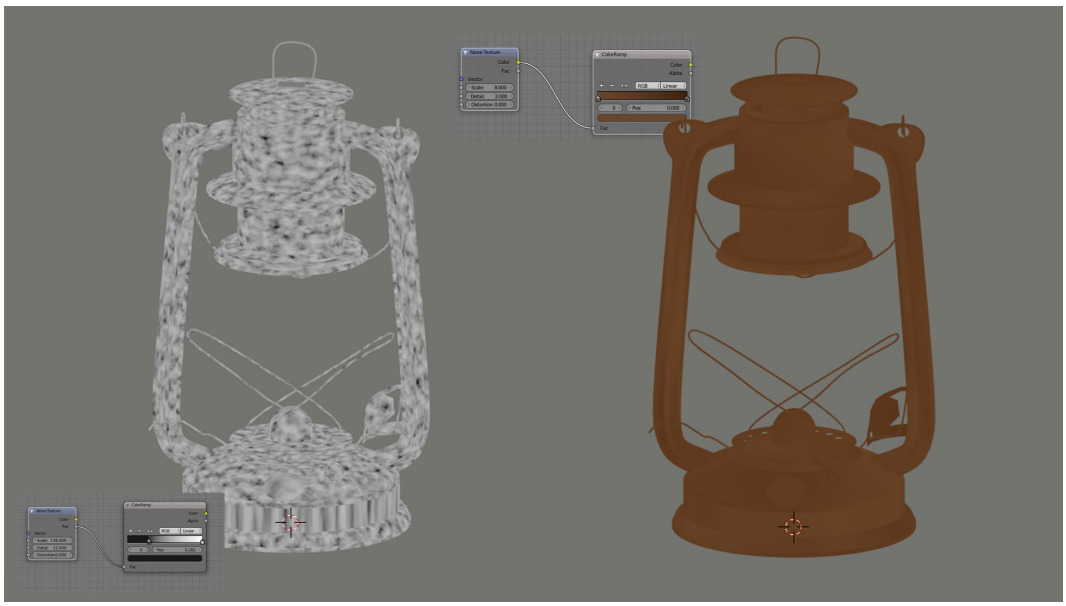

Abbildung 31. Noise Texturen mit Color Ramps als Input für die Roughness und Bump Werte (links) und den Color Input (rechts) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

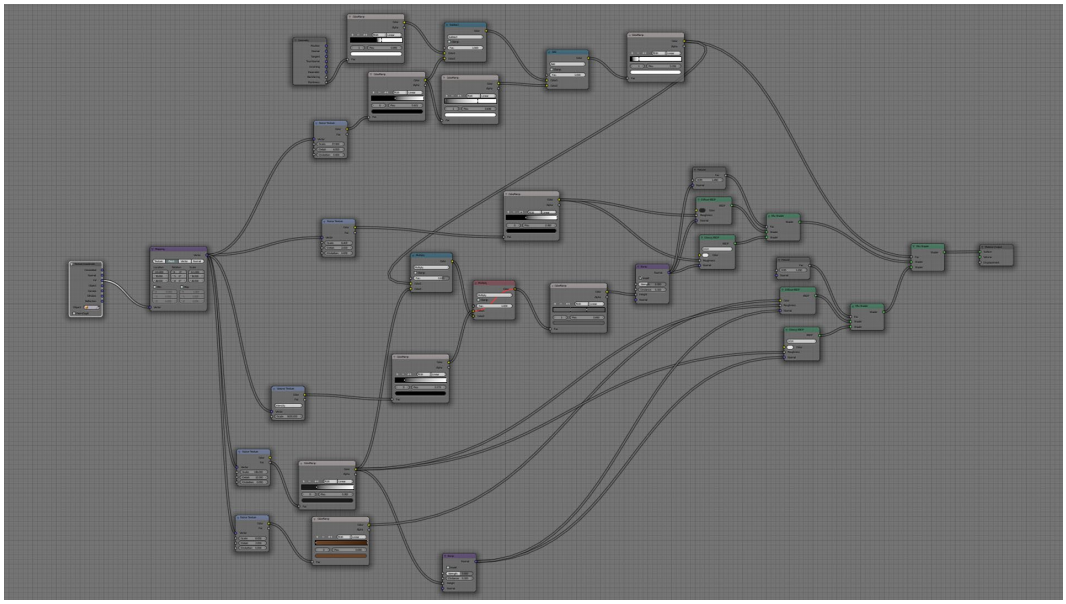

Abbildung 32. Shader Tree des NON PBR Lampen Materials (gesamt) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

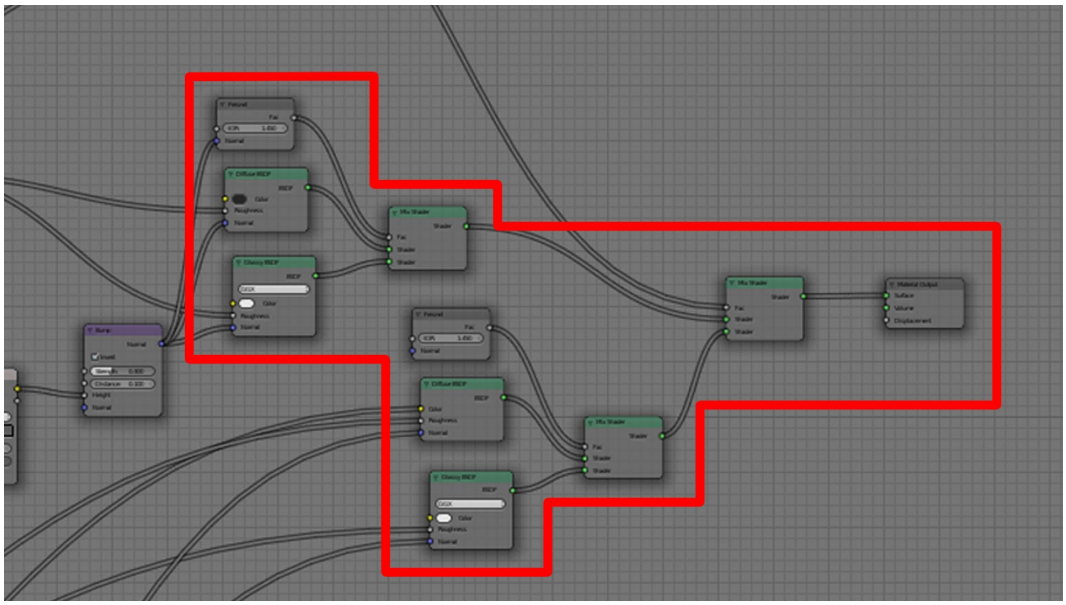

Abbildung 33. Shader Tree des NON PBR Laternen Materials (Close Up Basis Shader) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Abbildung 34. Low Resolution Rendering der zwei Materialien auf das gesamte 3D Modell angewendet (links und rechts) und des finalen Mixes (Mitte) des Laternen Gehäuses mit dem NON PBR Shader (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

6.3.3.2 Shading Laterne PBR

Wie beim NON PBR Shading werden auch beim PBR Shading des 3D Modelles der Laterne zwei Shader angelegt. Ein Shader wird für das Glasmaterial und ein Shader wird für das Material des Laternengehäuses erstellt.

PBR Glas Material:

Die einzige Shadernode ist bei diesem PBR Glas Material zu Anwendung kommt, ist die Principled Shader Node. Der finale Shader wird durch die mit den jeweiligen Inputs verbundenen Maps realisiert.

Die verschiedenen Layer des Materials, zwei Mal Dirt Layer einmal reiner Glas Shader, werden somit nur durch eine Node definiert.

Es werden die gleichen prozeduralen Noise Texturen wie aus dem NON PBR Shader verwendet (siehe Abbildung 25 und Abbildung 26). Allerdings werden diese mit einer Math Node mittels Addition kombiniert und steuern so zum einen die Farbe und zum anderen die Height Map und in deren weitere Folge die Bump Map.

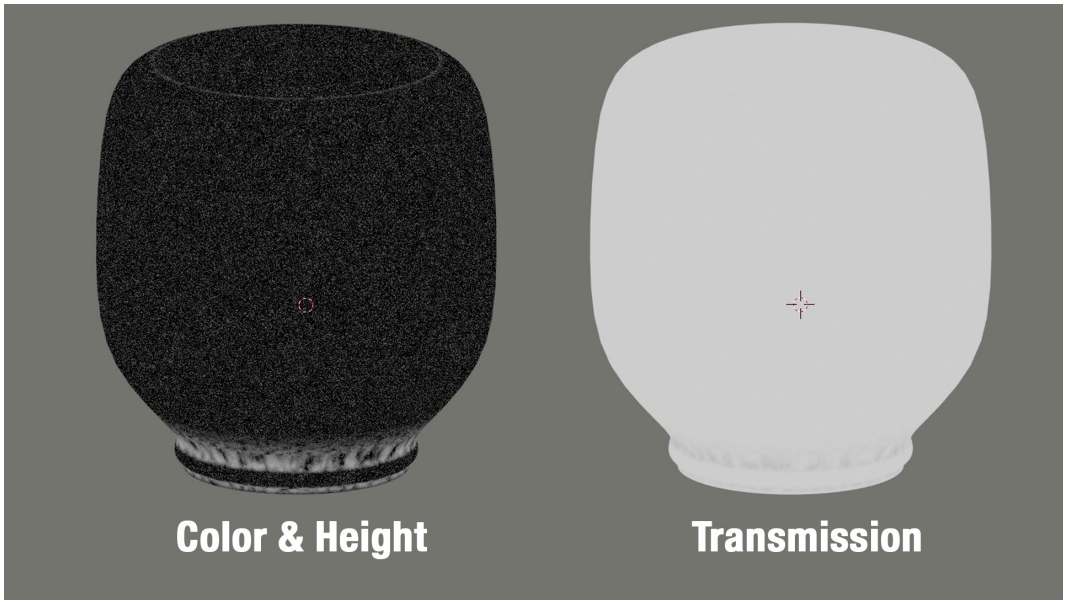

Abbildung 35. Die mittels Math Node und Invert Node kombinierten und veränderten Texturen welche die Inputs der Principled Shader Node steuern (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Außerdem definiert diese aus den verschiedenen Noise Texturen verbundene Textur, welche Bereiche vom Principled Shader über den Transmission Input als Glas Shader gerendert werden. Hierzu wird die Ausgangs Textur mittels Invert Node invertiert.

Der Roughness Input wird über eine separate Noise Texture und Color Ramp gespeist.

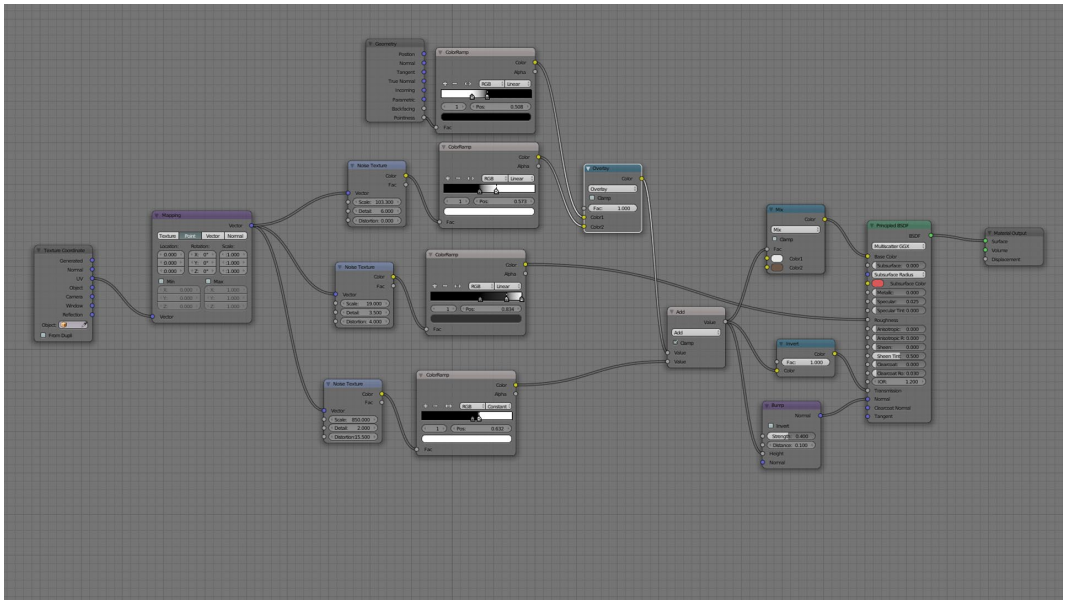

Abbildung 36. Shader Tree des PBR Glas Materials (gesamt) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

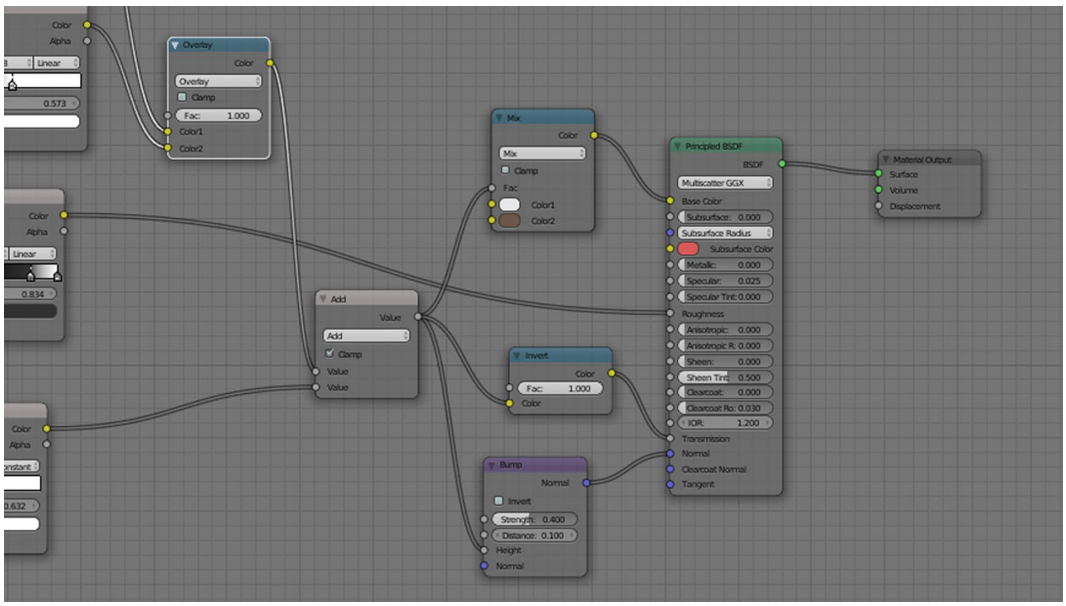

Abbildung 37. Shader Tree des PBR Glas Materials (Close Up) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Abbildung 38. Low Resolution Rendering des Glaszylinders mit dem PBR Glas Shader (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

PBR Gehäuse Laterne Material:

Für das PBR Material des Gehäuses der Laterne wurden die gleichen prozeduralen Noise Texturen wie bei dem NON PBR Shader verwendet. Der Color Input wird von eine Math Node gespeist welche durch ihren Mixfaktor definiert welche Bereiche welche Farbe auf dem 3D Modell haben (siehe Abbildung 39). Der Input des Faktors setzt sich aus mehreren mittels Math Node verbundenen Noise Texturen zusammen.

Der Roughness Input besitzt eine eigens angelegte Noise Textur mit Color Ramp, wohingegen sich der Normal Input wiederrum aus einem Mix zweier Texturen zusammensetzt um die verschiedenen Oberflächen Beschaffenheit des Materials darzustellen.

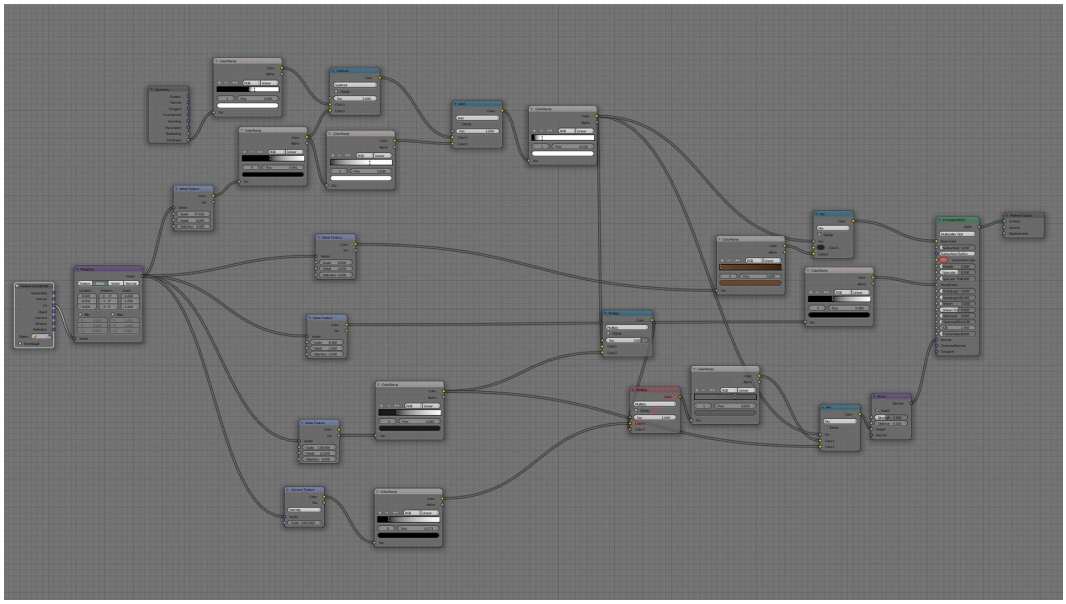

Abbildung 39. Shader Tree des PBR Laternen Materials (gesamt) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

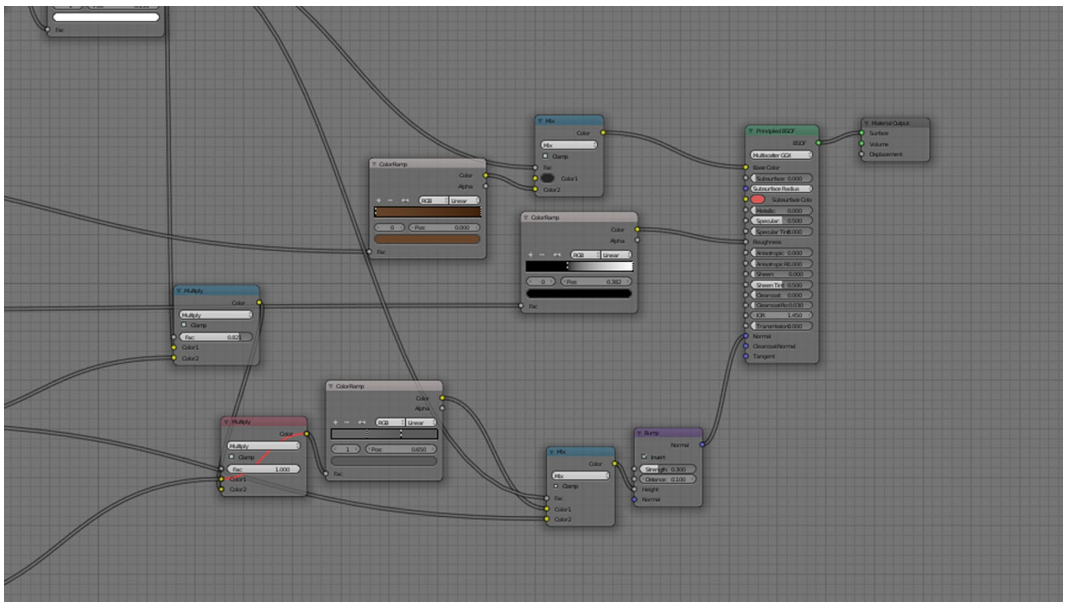

Abbildung 40. Shader Tree des PBR Laternen Materials (Close Up) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

Abbildung 41. Output der Math Node welche den Color Input der Principled Node speist (links). Low Resolution Rendering der Laterne (Mitte). Output der Math Node welche durch die Bump Map Node mit dem Normal Input der Principled Node verbunden wird (rechts) (vom Autor mit Blender 2.79 erstellt und gerendert in Cycles)

6.3.3.3 Shading Laterne PBR Substance Painter

Mittels der Texture Painting Software Substance Painter vom Allegorithmic wurde die Laterne anhand von automatisierten Prozessen, wie etwa mittels Curvature Maps die Kanten mit Rost versehen, aber auch händisch gepainted und so komplett texturiert.

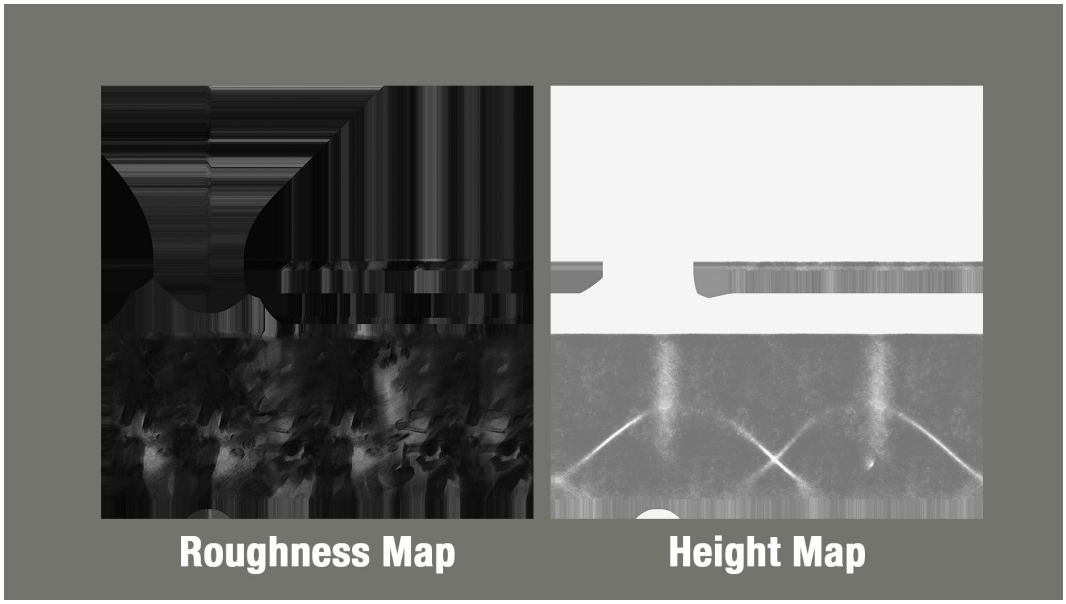

Abbildung 42. Roughness und Height Map des Glaszylinders aus Substance Painter (vom Autor mit Substance Painter erstellt)

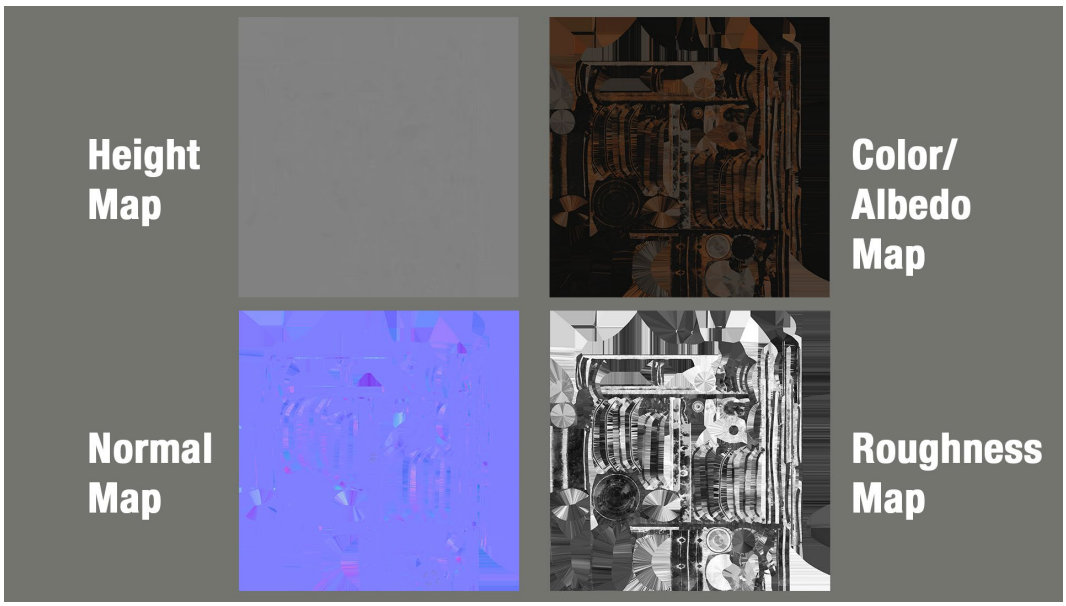

Abbildung 43. Color Map, Normal Map, Roughness Map und Height Map des Glaszylinders aus Substance Painter (vom Autor mit Substance Painter erstellt)

Die exportierten Texturen wurden in Blender 2.79 importiert und anschließend mit den jeweiligen Inputs der Principled Shader Node verbunden. In Blender 2.79 gibt es, wenn man das Node Wrangler Add On aktiviert hat dafür mit Strg+Shift+T einen eigenen Hotkey. Durch die richtige Benennungskonvention beim Export aus

Substance Painter erkennt die Software automatisch welche Textur mit welchem Input versehen werden muss.

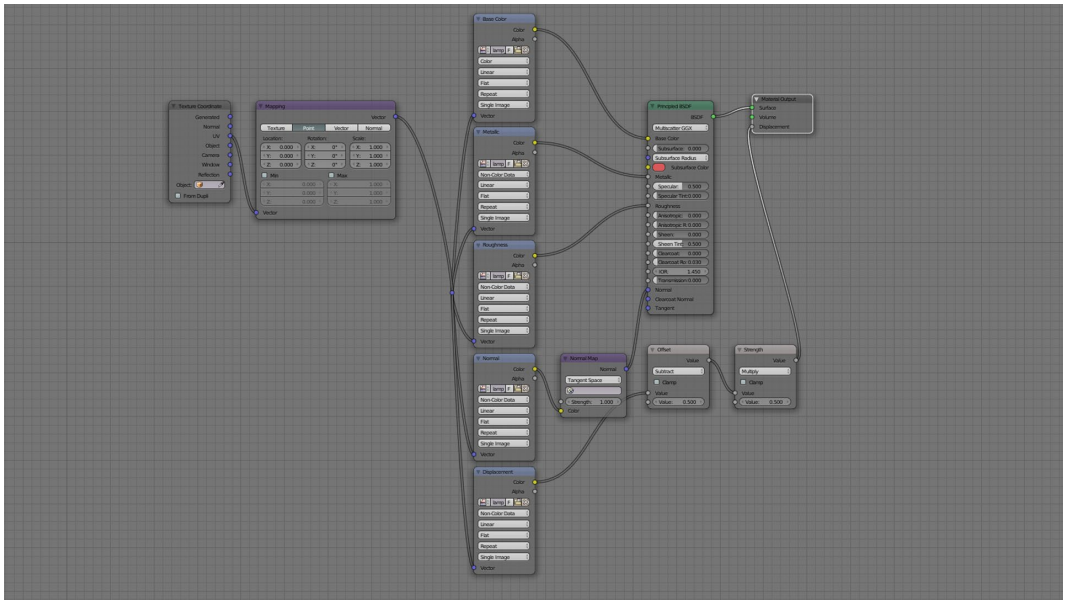

Abbildung 44. Principled Shader Node für das Laternen Gehäuse Material mit den aus Substance Painter exportierten Texturen (vom Autor mit Blender 2.79 erstellt)

Es wurden für das Glas Material und für das Laternen Gehäuse Material jeweils ein Set an Texturen erstellt und aus Substance Painter exportiert.

Für das Glas Material wurden nur zwei Texturen benötigt. Die Roughness Map als Input für den Roughness Wert des Shaders. Die Height Map wurde neben dem Normal Input über die Bump Node auch noch mit einer Mix RGB Node verbunden und dient als Maske die zwischen dem transparenten Glasmaterial und dem diffusen Staub Layer unterscheidet. Zusammengefasst kann man sagen, die Height Map wurde als Transmission Input für die Principled Shader Node verwendet.

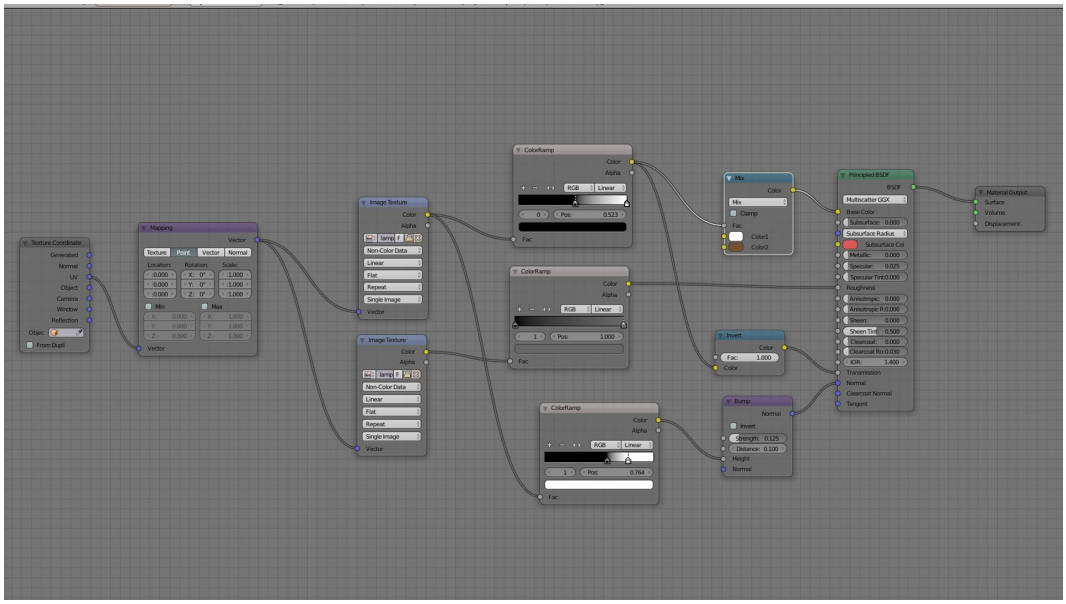

Abbildung 45. Principled Shader Node für das Material des Glaszylinders mit den aus Substance Painter exportierten Texturen (vom Autor mit Blender 2.79 erstellt)

Abbildung 46. Hardsurface Modelle mit den Texturen aus Substance Painter in niedriger Resolution gerendert (vom Autor mit Blender 2.79 erstellt)

6.3.4 Shading Stillleben

Da das Stillleben aus mehreren Modellen besteht, werden natürlich mehrere Shader angelegt. Alle Modelle bestehen ausschließlich aus dielektrischen

Materialien. Einige Modelle besitzen zusätzlich noch die Materialeigenschaft des Subsurface Scatterings.

6.3.4.1 Shading Stillleben NON PBR

Um fotorealistische Ergebnisse, welche anhand der Referenzaufnahmen angefertigt gerendert und verglichen werden, ohne einen PBR Shader Node erzielen, müssen mehrere Shader Nodes mittels etwa einem Mix Shader Node kombiniert werden um die physikalisch akkurate Darstellung garantieren zu können (siehe Abbildung 23). Aufgrund der Ähnlichkeit im Grundaufbau der verschiedenen Shader wird pro Shader hauptsächlich auf dessen Besonderheiten eingegangen. Die Basis bildet jener Shader aus Abbildung 23.

Apfel: Das NON PBR Material des Apfels bedient sich des Grundshaders aus Abbildung 23. Der Color Input der Diffuse Node wird über Noise und Gradient Texturen, welche mittels Math Nodes mit einander verbunden werden und über Color Ramps eingefärbt werden, gesteuert.

Der Roughness Wert wird ebenfalls über Noise Texturen definiert, diese werden allerding über eigene Mapping Nodes anders auf das 3D Modell projiziert als es beim Color Input der Fall war. Auch der Normal Input aller drei Nodes des Grundshaders wird mit setzt sich aus Noise Texturen zusammen

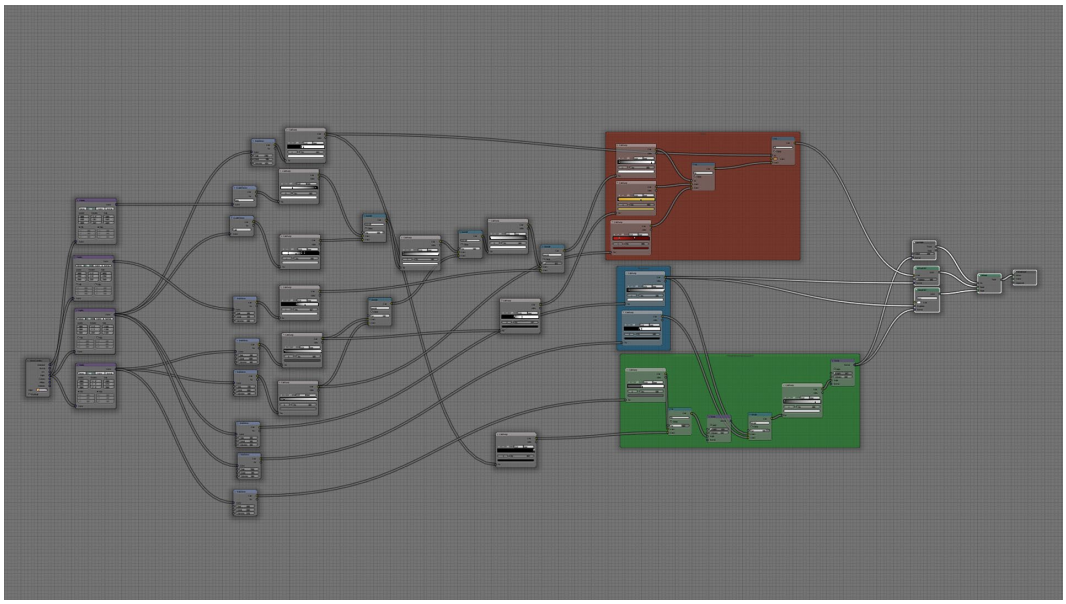

Abbildung 47. Shader Tree des NON PBR Materials für den Apfel (vom Autor mit Blender 2.79 erstellt)

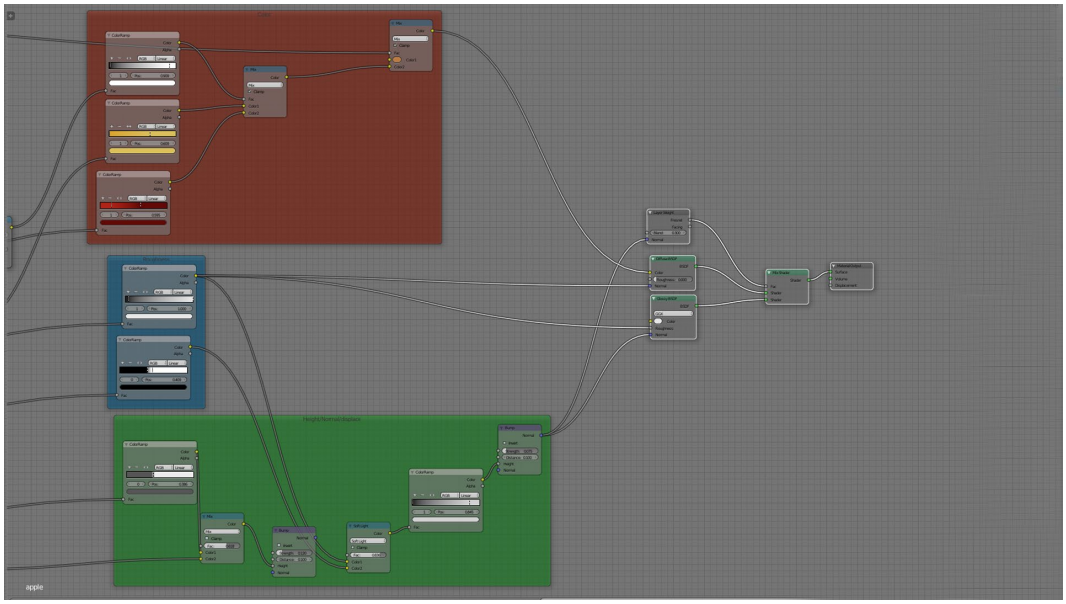

Abbildung 48. Close Up des Shader Tree des NON PBR Materials für den Apfel (vom Autor mit Blender 2.79 erstellt)

Birne: Ähnlich dem NON PBR Shaders des Apfels setzt sich der NON PBR Shader für die Birne aus dem Grundshader zusammen. Auch bei diesem Shader setzen sich die jeweiligen Inputs aus prozeduralen Texturen und Color Ramps zusammen.

Anschließend wird der Grundshader aber noch durch einen weiteren Mix Shader mit im zweiten Input befindlicher Subsurface Scattering Shader Node ergänzt. Der Faktor bei dieser zweiten Mix Node wird über eine Geometry Node und deren Pointiness Output (inklusive einer Color Ramp) gesteuert. Mittels dieser Node kann man schnell Masken anhand der Geometrie des 3D Modelles erstellen.

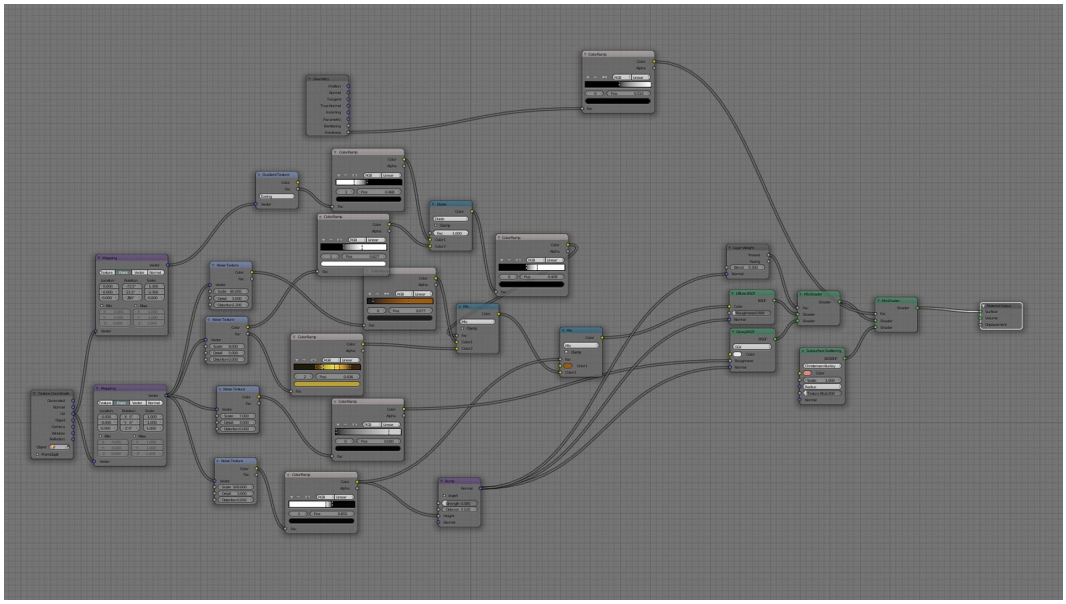

Abbildung 49. Close Up des Shader Tree des NON PBR Materials für die Birne (vom Autor mit Blender 2.79 erstellt)

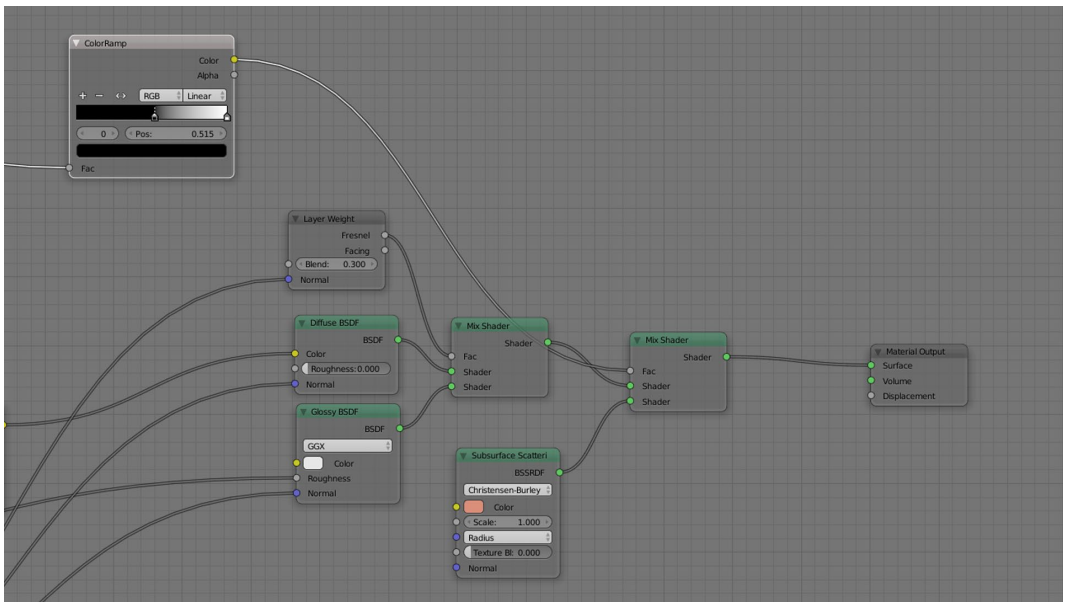

Abbildung 50. Close Up des Shader Tree des NON PBR Materials für die Birne (vom Autor mit Blender 2.79 erstellt)

Orange: Der NON PBR Shader Node Tree für die Orangen, setzt sich ausschließlich aus dem Grundshader aus Abbildung 23 zusammen. Die jeweiligen Inputs werden durch prozedurale Texturen gesteuert wobei es sich beim Color und

Normal Input um eine Voronoi Textur handelt und beim Roughness Input um eine prozedurale Noise Textur.

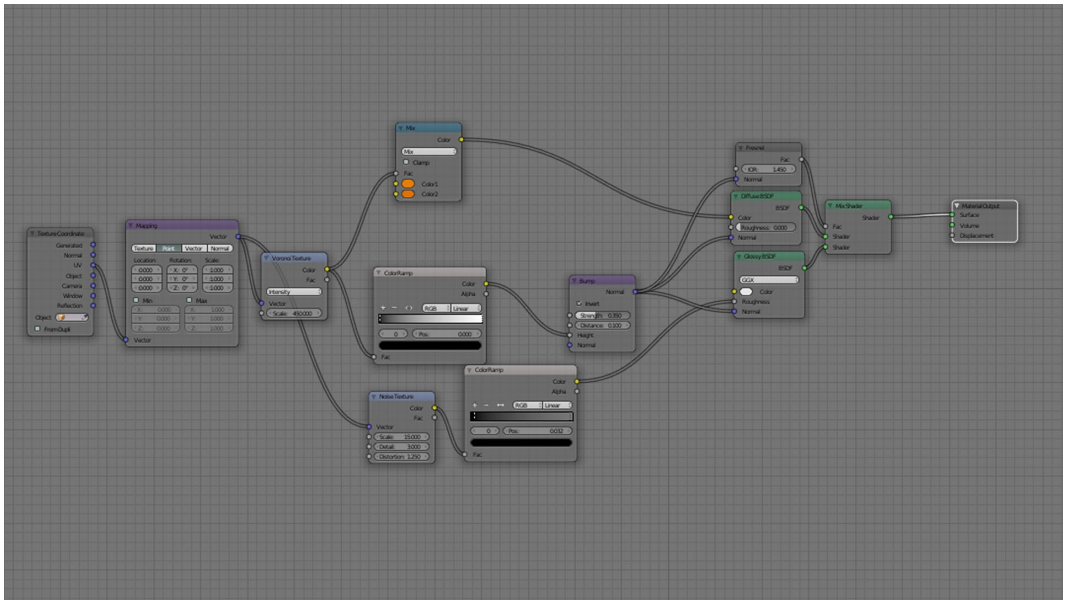

Abbildung 51. Close Up des Shader Tree des NON PBR Materials für die Orange (vom Autor mit Blender 2.79 erstellt)

Das Material der Schnittfläche der halben Orange setzt sich aus den gleichen Komponenten wie der NON PBR Shader der Birne zusammen und kombiniert den Grundshader mittels Mix Shader mit einer Subsurface Scattering Node.

Im Gegensatz zu dem generellen Orangen Material wird die Schnittfläche der Orange nicht über prozedurale sondern über eine Image Textur gesteuert.

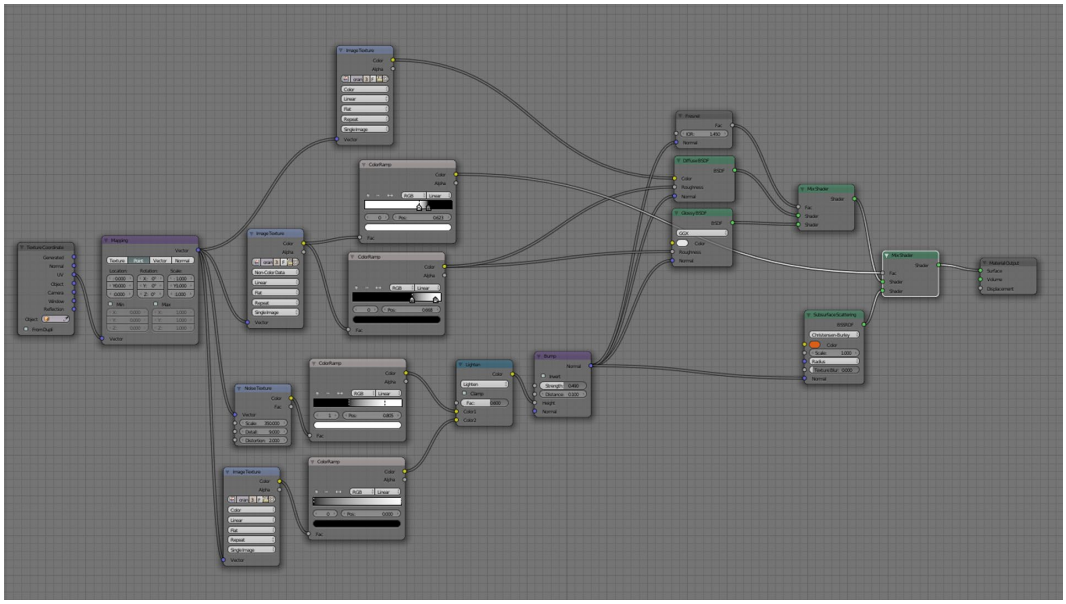

Abbildung 52. Close Up des Shader Tree des NON PBR Materials für die halbe Orange (vom Autor mit Blender 2.79 erstellt)

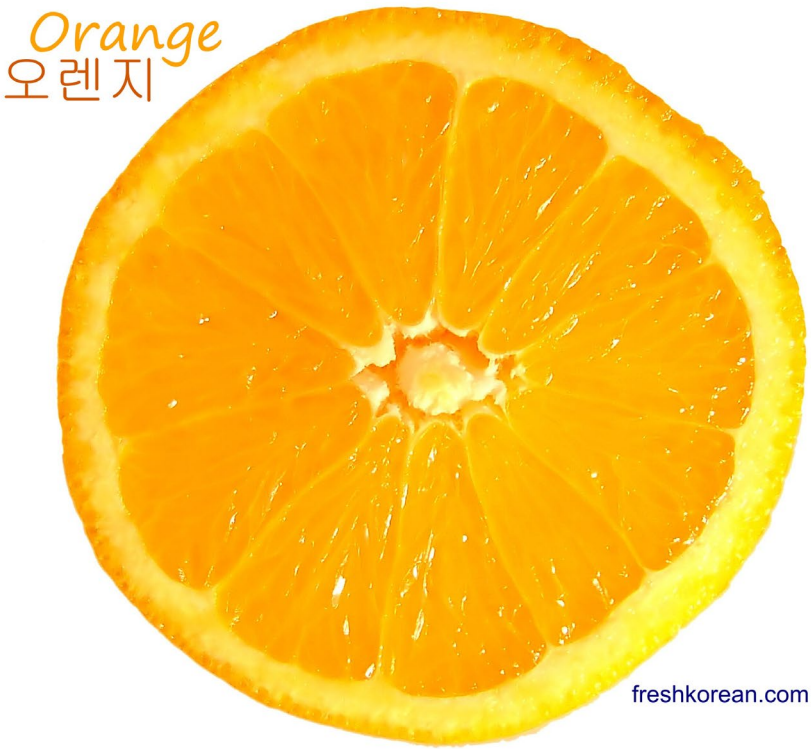

Abbildung 53. Grundtextur der Schnittfläche der Orange (https://freshkorean.com/tag/fruit-in-korean/)

Banane: Der NON PBR Shader der Banane setzt sich aus dem Grundshader zusammen, dessen Inputs mittels prozeduralen Noise Texturen definiert werden. Des Weiteren werden die Längsstreifen der Banane mittels Geometry Node und deren Pointiness Output gesteuert.

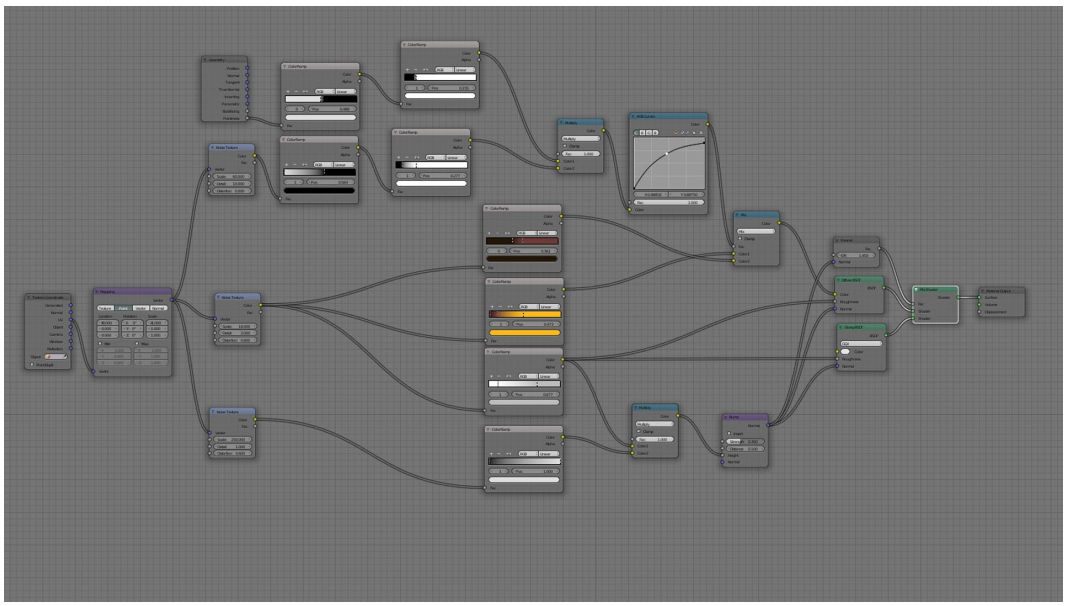

Abbildung 54. Close Up des Shader Tree des NON PBR Materials für die Banane (vom Autor mit Blender 2.79)

6.3.4.2 Shading Stillleben PBR

Als Grund-Shader von jedem organischen Modell steht im Vergleich zu der NON PBR Methode die Principled Shader Node dessen Inputs mit den verschiedenen Maps versehen werden. Es wurden dieselben prozeduralen Texturen wie bei der NON PBR Methode verwendet. Die Node Shader Trees der einzelnen Modelle setzen sich wie folgt zusammen:

Apfel: Mittels der Clearcoat Einstellung und dem extra Input für eine Normal/Bump als auch einer Roughness Map war es möglich dem Material jeweils einen extra Layer an Roughness bzw. Normal/Bump über den bestehenden zu platzieren.

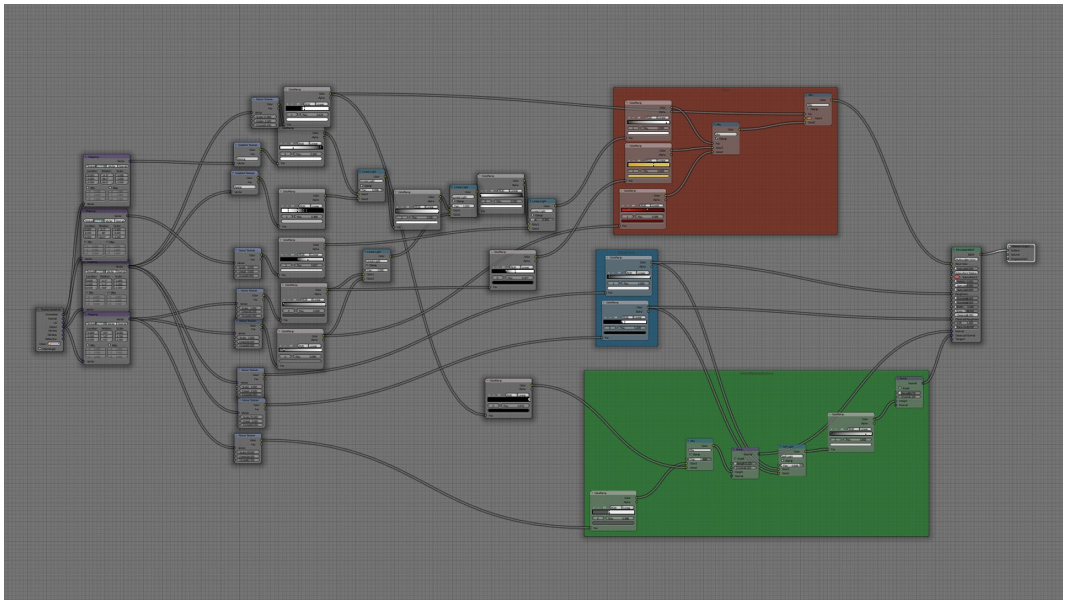

Abbildung 55. Close Up des Shader Tree des PBR Materials für den Apfel (vom Autor mit Blender 2.79)

Birne: Die Darstellung des Subsurface Scattering wird direkt mit der Principled Shader Node gelöst.

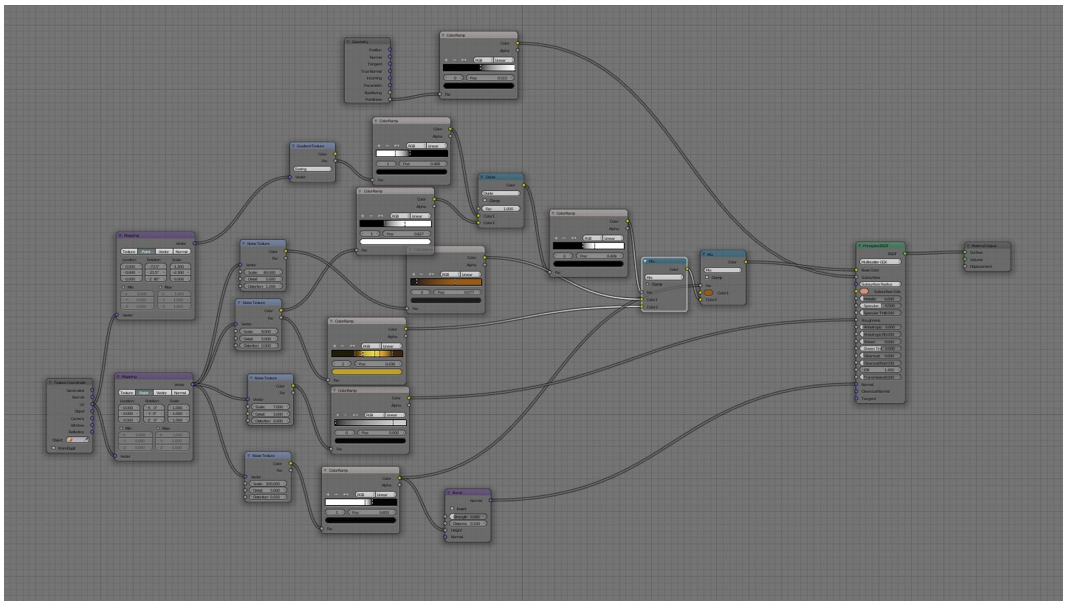

Abbildung 56. Close Up des Shader Tree des PBR Materials für die Birne (vom Autor mit Blender 2.79 erstellt)

Orange:

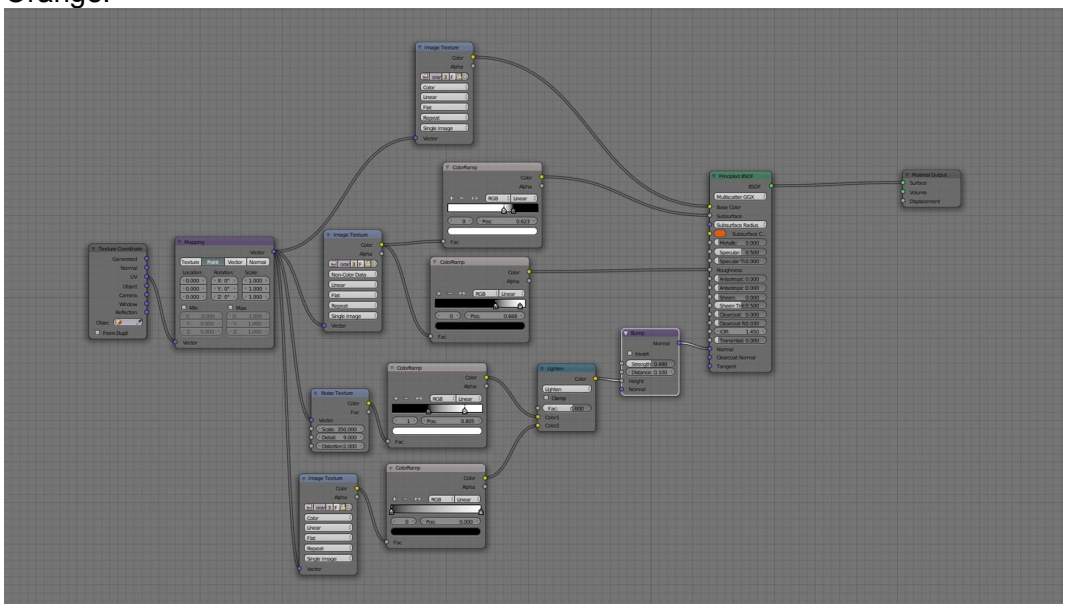

Abbildung 57. Close Up des Shader Tree des PBR Materials für die Orange (vom Autor mit Blender 2.79 erstellt)

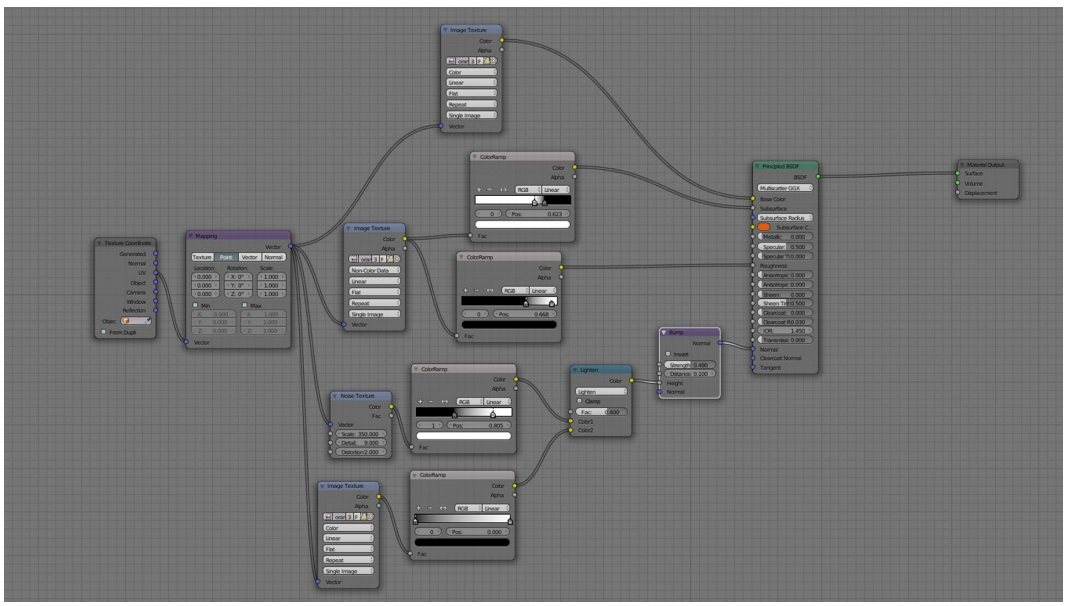

Abbildung 58. Close Up des Shader Tree des PBR Materials für die halbe Orange (vom Autor mit Blender 2.79 erstellt)

Banane:

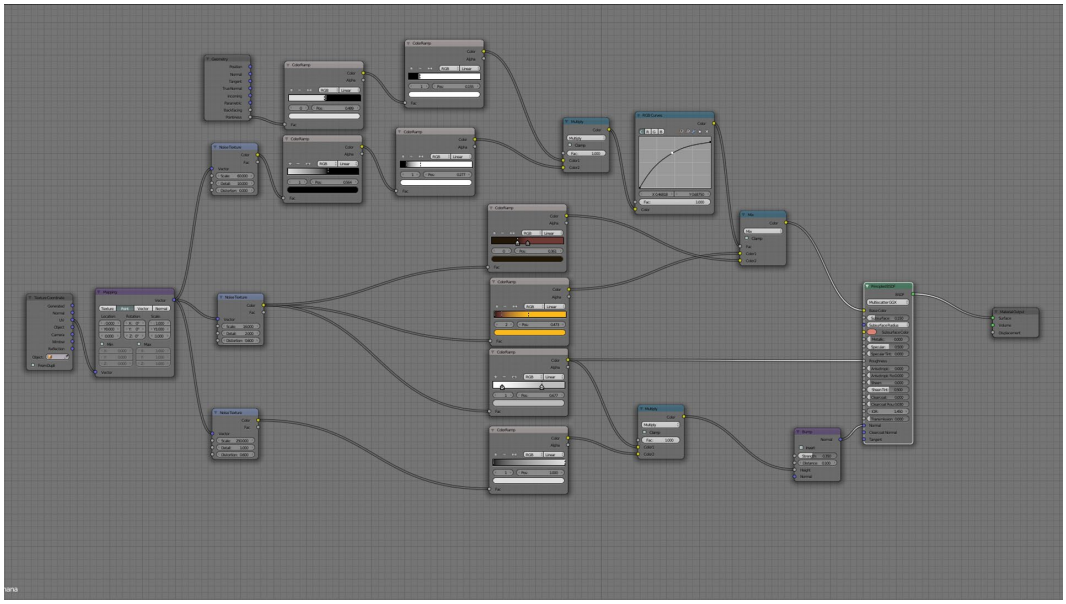

Abbildung 59. Close Up des Shader Tree des PBR Materials für die Banane (vom Autor mit Blender 2.79 erstellt)

6.3.4.3 Shading Stillleben PBR Allogerithmic Substance Painter

Beim Shading mittels der PBR Texture Painting Software Substance Painter von Allogerithmic werden in dem separaten Programm, mittels prozeduralem und händischem, auf einem layerbasierten, Workflow, pro Modell Texture Maps angelegt mit welchen die Materialeigenschaften definiert werden. Danach werden pro Shader jeweils eine Color-, eine Metallic-, eine Height-, eine Normal- und eine Roughness- Map exportiert. In Blender wird dann die Principled Shader Node automatisch mit diesen Maps versehen und gerendert. Der Automatismus mit welchem die Maps mit der Shader Node verbunden werden basiert auf der einheitlichen Benennung der einzelnen Maps bei dem Export aus Substance Painter. Somit ist für jede Texture Map genau ein Input definiert. Daraus ergibt sich natürlich, dass es sich bei diesem Workflow um den zeiteffizientesten handelt. Außerdem kann dieser Workflow auch als am userfreundlichsten beschrieben werden.

Da sich der Node Tree bei jedem der organischen 3D Modellen als (bis auf die verwendetet Texturen) nahezu ident darstellt, wird die Abbildung des Shadertrees für den Shader der Banane als Beispiel für alle 3D Modelle herangezogen.

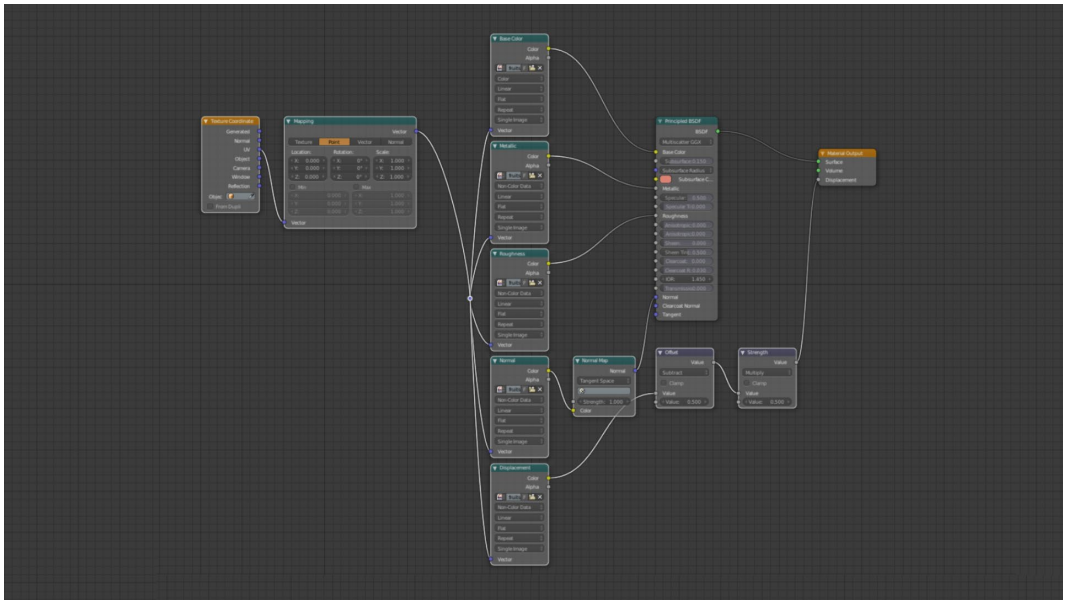

Abbildung 60. Close Up des Shader Tree des PBR Materials für die Banane mit den PBR Texturen aus Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt)

6.4 Messungen und Ergebnisse

Die finalen Renderings wurden anhand der vorher definierten Punkte gemessen und verglichen um eine Aussage bezüglich der Forschungsfragen stellen zu können.

Für die Set Up Auswertungen wird nur der reine Zeitaufwand betrachtet welcher für die Erstellung des Shaders notwendig war. Die Zeit die für die Erstellung der Maps benötigt wurde, findet in dieser Auswertung keine Betrachtung. Falls allerdings Adaptionen notwendig waren um die Maps für den Shader vorzubereiten so wurde dieser Zeitaufwand ebenfalls berücksichtigt.

Die Rendereinstellungen waren bei jeder zu vergleichenden 3D Szene die gleichen. Die generelle Bildauflösung betrug 2048x2048 Pixel. Die Hardsurface Daylight Szenen wurden mit 1024 Samples gerendert, alle anderen Szenen wurden mit 512 Samples gerendert. Die GPU Tile Sizes wurden jeweils mit dem Auto Tile Size Addon automatisch für das Rendering optimiert, hierbei gab es Schwankungen von etwa 208x214 Pixel zu 256x256 Pixel. Die CPU Tile Size blieb bei allen Renderings mit 32x32 Pixel konstant bei einem Wert. Der bei Cycles interne Denoiser wurde mit einem einheitlichen Radius von 8 und einer

einheitlichen Feature Strength bzw. einheitlichen einer Strength von 0,5 bei allen 3D Szenen verwendet.

Als Output Format wurde ein unkomprimiertes 16 Bit PNG mit RGBA Kanälen Format gewählt.

Aufgrund des NON PBR Grundshaders (siehe Abbildung 23.), welcher aufgrund seines Aufbaus mit Mix Shader und Fresnel Node, sich an den Workflow mit der PBR Principled Shader Node anzunähern versucht waren bei der veränderten Lichtsituation keine Änderungen notwendig um die NON PBR geshadeten Modelle anzupassen. Die Ergebnisse sind allerdings trotzdem interessant für die Auswertung, da es doch zu anderen Messergebnissen als bei der Tageslicht Szene kam. Die Tageslicht Szene wurde mit einem 10240x5120 Pixel großem JPEG beleuchtet und die Lowlight Szenen mit einem 8192x4096 Pixel großem HDR Foto.

Das Tageslichtfoto wurde mittels Smartphone und der App Google Street View (https://www.google.com/streetview/) erstellt.

Die für die Lowlight Szenen verwendete Environment Textur wurde von der Website HDRI Haven (https://hdrihaven.com/) bezogen. Die HDRIs dieser Website sind unter der CC0 lizensiert.

Abbildung 61. Reduzierte Darstellung der Environment Texturen (IBL) welche für die Beleuchtung der 3D Szenen verwendet wurden (vom Autor erstellt bzw. von HDRI Haven bezogen)

Die folgenden zwei Tabellen dienen zur Veranschaulichung der gemessenen Werte, welche für die Beantwortung der gestellten Forschungsfragen benötigt wurden.

Die erste Tabelle beinhaltet die gemessenen Werte der Tageslicht Szenen:

Rendering Daylight	*Set Up Zeit*	*Renderzeit GPU in Minuten/Sekundenl*	*Vram*	*Renderzeit CPU in Minuten/Sekundenl*	*RAM (bei CPU Rendering)*
Hardsurface Modell NON PBR (Gehäuse + Glaszylinder)	*~4min.*	*10:08.73*	*234,57mb (Peak: 258,51mb)*	*45:39.68*	*234,57mb (Peak 321,01mb)*
Hardsurface Modell PBR (Gehäuse + Glaszylinder)	*~2min.*	*10:57.32*	*235,91mb (Peak 258,60mb)*	*1:20:28.57*	*235,91mb (Peak 327,02mb)*
Hardsurface Modell PBR Substance Painter (Gehäuse + Glaszylinder)	*30 Sekunden*	*11:36.97*	*1077,30mb (Peak 1101,24mb)*	*51:47.63*	*1077,30mb (Peak 1166,22mb)*
Organisches Modell NON PBR (Stillleben)	*~23min.*	*14:49.36*	*244,89mb (Peak 268,84mb)*	*17:55.66*	*244,89mb (Peak 323,61mb)*
Organisches Modell PBR (Stillleben)	*~14min.*	*8:11.47*	*245,31mb (Peak 269,26mb)*	*19:25.13*	*245,31mb (Peak 335,25mb)*
Organisches Modell PBR Substance Painter (Stillleben)	*~8min.*	*3:39.19*	*3559.80mb (Peak 3559.80mb)*	*21:43.61*	*3559,81mb (Peak 3650,90mb)*

Die zweite Tabelle beinhaltet die gemessenen Werte der Lowlight Szenen:

Rendering Lowlight	*Set Up Zeit*	*Renderzeit GPU in Minuten/Sekundenl*	*Vram*	*Renderzeit CPU in Minuten/Sekundenl*	*RAM (bei CPU Rendering)*
Hardsurface Modell NON PBR (Gehäuse + Glaszylinder) - NIGHT	*0min.*	*9:20.01*	*546,57mb (Peak 570.51mb)*	*40:49.99*	*546,57mb (Peak 648.14mb)*
Hardsurface Modell PBR (Gehäuse + Glaszylinder) - NIGHT	*0min.*	*10:08.05*	*547.91mb (Peak 571.86mb)*	*51:23.93*	*547.91mb (Peak 632.39mb)*
Hardsurface Modell PBR Substance Painter (Gehäuse + Glaszylinder) - NIGHT	*0min.*	*8:31.55*	*1389,30mb (Peak 1413,24mb)*	*39:58.15*	*1389,30mb (Peak 1487.17mb)*
Organisches Modell NON PBR (Stillleben) - NIGHT	*0min.*	*04:34.05*	*556,89mb (Peak 580,84mb)*	*16:28.28*	*556,89mb (Peak 638,06mb)*
Organisches Modell PBR (Stillleben) - NIGHT	*0min.*	*5:16.33*	*577,31mb (Peak 581,26mb)*	*19:34.28*	*557,31mb (Peak 642,22mb)*
Organisches Modell PBR Substance Painter (Stillleben) - NIGHT	*0min.*	*3:49.87*	*4079,83mb (Peak 4103,78mb)*	*16:56.83*	*4079,83mb (Peak 4161.39mb)*

Folgend sind nun die visuellen Ergebnisse des Renderings der Cycles Engine in absteigender Reihenfolge gleich der oberen Tabellen aufgelistet. Die gerenderten Bilder wurden direkt nach dem Rendervorgang aus der Software Blender exportiert und keiner weiteren Bildbearbeitung unterzogen.

Abbildung 62. GPU Rendering Hardsurface Modell NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 63. CPU Rendering Hardsurface Modell NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 64. GPU Rendering Hardsurface Modell PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 65. CPU Rendering Hardsurface Modell PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 66. GPU Rendering Hardsurface Modell PBR mit Texturen von Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

Abbildung 67. CPU Rendering Hardsurface Modell PBR mit Texturen von Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

Abbildung 68. GPU Rendering organisches Modell NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 69. CPU Rendering organisches Modell NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 70. GPU Rendering organisches Modell PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 71. CPU Rendering organisches Modell PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 72. GPU Rendering organisches Modell PBR Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

Abbildung 73. CPU Rendering organisches Modell PBR Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

Abbildung 74. GPU Rendering Hardsurface Modell - Lowlight NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 75. CPU Rendering Hardsurface Modell - Lowlight NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 76. GPU Rendering Hardsurface Modell - Lowlight PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 77.CPU Rendering Hardsurface Modell - Lowlight PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 78. GPU Rendering Hardsurface Modell - Lowlight PBR mit Texturen von Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

Abbildung 79. GPU Rendering Hardsurface Modell - Lowlight PBR mit Texturen von Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

Abbildung 80. GPU Rendering organisches Modell - Lowlight NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 81. CPU Rendering organisches Modell - Lowlight NON PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 82. GPU Rendering organisches Modell - Lowlight PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 83. CPU Rendering organisches Modell - Lowlight PBR (vom Autor mit Blender 2.79 erstellt und in Cycles gerendert)

Abbildung 84. GPU Rendering organisches Modell - Lowlight PBR Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert

Abbildung 85. CPU Rendering organisches Modell - Lowlight PBR Substance Painter (vom Autor mit Blender 2.79 und Substance Painter erstellt und in Cycles gerendert)

6.5 Schlussfolgerung

Anhand der Ergebnisse und Messungen aus dem praktischen Teil der Arbeit kann man folgende Aussagen treffen um die Forschungsfragen zu beantworten.

Bei den allgemeinen Set Up Zeiten der verschiedenen Shader ist ganz klar die Tendenz erkennbar, dass die PBR Workflows deutlich weniger zeitintensiv beim

Aufsetzten sind. Das kann vor allem durch die Vereinfachung der Shadertrees von mindestens 3 Nodes (siehe Abbildung 23) auf eine (Principled Shader) Node erklärt werden, aber auch durch die userfreundlichere UI des Principled Shaders gegenüber mehreren mit Mix Shader verbundenen Nodes. Wird das in Blender 2.79 inkludierte Add On „Node Wrangler“ verwendet, kommt es beim PBR Workflow mit Texturen aus einer externe Software zu weiterer Zeitersparnis, da dieses Add On über einen Shortcut verfügt welcher automatisch alle PBR Texturen mit dem richtigen Input der Principled Shader Node verbindet.

Bei den verwendeten Ressourcen zeigt sich vor allem deutlich, dass die mit extra Software erstellten PBR Texturen (in diesem Fall aus Substance Painter) einen deutlich erhöhten Bedarf an VRAM bzw. RAM hatten. Bei den prozedural geshadeten Szenen kann man von einem minimalen erhöhten Bedarf bei der PBR Methode gegenüber der NON PBR Methode sprechen, so handelt es sich hierbei lediglich um Unterschiede von Megabyte im einstelligen Bereich. Daher gilt zu beachten dass man bei komplexeren Szenen mit mehreren Objekten und Shadern sich innerhalb des von der Hardware zugelassenen Rahmens bewegt. Dies betrifft vor allem das Rendering über die GPU da es bei überschreiten des VRAM zu einer Fehlermeldung und dem Abbruch des Rendervorganges kommt.

Als besonders interessant gestalteten sich die Auswertung der Messwerte der verschiedenen Renderzeiten der 3D Szenen sowohl mit der GPU als auch der CPU. In fast allen Messungen, mit Ausnahme des Organischen Stillebens (Tag), war die prozedural geshadete NON PBR Methode schneller als die prozedural geshadete PBR Methode. In drei Fällen war die NON PBR Methode nicht am schnellsten, sondern die PBR Methode mit den PBR Texturen aus Substance Painter. Wie oben erwähnt, war die prozedural geshadete Methode in lediglich einer Messung schneller als die NON PBR Methode aber niemals die schnellste Methode.

Die prozedural geshadete PBR Methode erwies sich, bis auf eine Ausnahme (Organsich Stillleben PBR SP), bei allen Messungen der Renderzeiten als die am langsamsten, wobei sich die Renderings über die CPU als noch eindeutig langsamer darstellten. Generell war, wie von diesem Hardware Set Up erwartet, die GPU Performance der der CPU eindeutig überlegen.

Die PBR Methode mit den PBR Texturen von Substance Painter, in erster Linie, und die prozedural geshadete PBR Methode in zweiter, waren von den Renderergebnissen definitiv eher mit dem Referenzfoto, in ihrer physikalisch korrekten Darstellung, vergleichbar als die NON PBR Methode, bei der vor allem in den Specular Highlights die nicht physikalisch korrekte Darstellung auffällt. Zusätzlich ist zu erwähnen, dass die von Substance Painter verwendeten Algorythmen für Edge Detection, Dirt Maps, Roughness Maps usw. komplexer sind und daher, vor allem beim Hardsurface modell, zu visuell ansprechenderen Ergebnissen geführt haben.

Für den generellen Einsatz in der Produktion gilt es also zu definieren ob die physikalisch korrekte Darstellung den erhöhten Zeitaufwand beim Rendering rechtfertigt. Vor allem beim Rendering von Animationen wird diese Entscheidung eine tragende Rolle bei der Berechnung der Renderzeiten und in weiterer Folge auch deren Kosten spielen. Wobei dazu erwähnt werden muss, dass bei komplexeren Szenen mit einem Vielfachen an Shadern, umgekehrt die Zeitersparnis beim Set Up eben jener bei den PBR Methoden deutlich kürzer ausfallen dürfte. Mit einem eigenen Texture Painting Programm (wie Substance Painter oder DDO) in der Pipeline werden diese Zeiten nochmals drastisch reduziert, da es auch möglich ist, mehere Texturen für mehrere Objekte gleichzeitig zu exportieren.

Allerdings muss beachtet werden, wenn die PBR Methode mit den PBR Texturen aus einer weiteren Software, zur Anwendung kommt und über die GPU gerendert werden soll, muss diese bzw. müssen diese über genügend VRAM verfügen um die PBR Texturen laden zu können.

Literaturverzeichnis

Das Praxisbuch der Lichttechnik, Rainer Bewer und Kai Steckmann, 2002 GC Carstener Verlag, München

https://en.wikipedia.org/wiki/Refractive_index

http://gamma.cs.unc.edu/courses/graphics-s09/LECTURES/3DModels_SurveyPaper.pdf

https://inst.eecs.berkeley.edu/~cs283/fa10/lectures/283-lecture2.pdf

https://www.cs.utah.edu/~shirley/books/fcg2/rt.pdf

https://www.cycles-renderer.org/about/

https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render/

https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render/light_paths.html

https://users.ph.tum.de/cucke/ftp/lectures/jena96.pdf

Ausgewählte Kapitel der Technik, Viktor Ritter von Niesiolowski-Gawin, Verlag von L.W. Seidel & Sohn, 1908

https://docs.blender.org/manual/en/dev/render/cycles/settings/scene/render/integrator.html#light-paths

http://iopscience.iop.org/article/10.1088/0026-1394/47/3/021/meta

http://www.indigorenderer.com/documentation/manual/indigo-scenes/materials/material-types/oren-nayar

https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/diffuse_shaders.html

http://www.cs.northwestern.edu/~ago820/cs395/Papers/Phong_1975.pdf

https://de.wikipedia.org/wiki/Blinn-Beleuchtungsmodell

http://inst.eecs.berkeley.edu/~cs283/sp13/lectures/cookpaper.pdf

https://docs.blender.org/manual/en/dev/render/blender_render/materials/properties/specular_shaders.html

https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf

https://www.merl.com/brdf/

https://theblog.adobe.com/gaming-technology-changing-design-visualize-products/

https://arnold-rendering.com/2016/02/06/ggx-microfacet-distribution/

https://docs.blender.org/manual/en/latest/render/cycles/nodes/types/shaders/principled.html

http://graphics.pixar.com/library/ApproxBSSRDF/approxbssrdfslides.pdf

http://renderwonk.com/publications/s2010-shading-course/hoffman/s2010_physically_based_shading_hoffman_a_notes.pdf

Abbildungsverzeichnis

Tabellenverzeichnis