



TECHNISCHE HOCHSCHULE MITTELHESSEN

THM

**CAMPUS
FRIEDBERG**

IEM

Informationstechnik-
Elektrotechnik-Mechatronik

Automatische Konvertierung von Xfrog Shader-Netzwerken

Studiengang Medieninformatik

Masterarbeit

vorgelegt von

Ben Wilde

geb. in Frankfurt am Main

durchgeführt an der
Technischen Hochschule Mittelhessen (THM), Friedberg

Referent der Arbeit: Prof. Dr. Cornelius Malerczyk
Korreferent der Arbeit: M. Sc. Hans Christian Arlt

Friedberg, 2021

Für meine wundervolle Tochter Lilly

Danksagung

Vielen Dank an alle, die mich in irgendeiner Weise bei dieser Arbeit unterstützt haben.

Als erstes möchte ich meinem Schwiegervater für seine Unterstützung während des Studiums danken. Ohne dich wäre das alles nicht möglich gewesen.

Vielen Dank an meine Schwiegermama, die leider nicht mehr bei uns ist. Deine Schultüte war der Startschuss für meine akademische Laufbahn.

Einen riesigen Dank an meine wundervolle Frau und meine wundervolle Tochter, die mich immer unterstützen und mir immer wieder zeigen, was im Leben wirklich wichtig ist. Tut mir leid für die vielen Momente, in denen ich körperlich oder geistig abwesend war.

Danke an meine Mama für das fleißige Korrekturlesen, auch wenn du schon vom „Rendern“ geträumt hast.

Prof. Dr. Cornelius Malerczyk möchte ich für die Betreuung meiner Arbeit und für die stets fantastischen Ideen und Denkanstöße danken. Ich habe unglaublich viel gelernt und es hat (fast) immer richtig viel Spaß gemacht.

Ein besonderer Dank geht auch an meine Kommilitonen und Kommilitoninnen Julius Hilbig, Thamar Vainstain und Olga Zimmermann, die mich als alten Quereinsteiger an die Hand genommen und mir viele Dinge verständlich gemacht haben.

Und natürlich danke an alle, die an meinen Umfragen teilgenommen und sich die unzähligen Bilder von Pflanzenblätter angeschaut haben.

Selbstständigkeitserklärung

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

Friedberg, Oktober 2021

Ben Wilde

Inhaltsverzeichnis

Danksagung	i
Selbstständigkeitserklärung	iii
Inhaltsverzeichnis	v
Abbildungsverzeichnis	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Problemstellung	3
1.3 Zielsetzung	3
1.4 Aufbau der Arbeit	4
1.5 Zusammenfassung der Ergebnisse	5
2 Stand der Technik	7
2.1 Einleitung	7
2.2 Open Shading Language (OSL)	7
2.3 Nvidia Material Definition Language(MDL)	9
2.4 MaterialX	12
2.5 Vergleich von Anbietern von 3D-Pflanzenmodellen	14
2.5.1 Xfrog	14
2.5.2 Laubwerk	16
2.5.3 Quixel Megascans	19
2.6 Zusammenfassende Betrachtung	21
3 Grundlagen	23
3.1 Einleitung	23
3.2 Der Aufbau computergenerierter Pflanzen	23
3.2.1 Algorithmische Erzeugung von Pflanzenmodellen	23
Prozedurale Modellierung	24
Regelbasierte Modellierung	25
Kombiniertes Verfahren von Xfrog - Regelbasierte Objekterzeugung	26
3.2.2 Erzeugung von Pflanzentexturen per Scan	26

3.2.3	Erzeugung von Pflanzentexturen anhand eines Fotos	27
	Materialize (Bounding Box Software)	28
	ShaderMap	29
	Substance 3D Sampler (ehemals Substance Alchemist)	29
3.3	V-Ray in Maya	30
3.4	Arnold in Maya	31
3.5	Subsurface Scattering und Translucency	31
3.5.1	SSS und Translucency in Chaosgroup V-Ray	32
3.5.2	SSS und Translucency in Autodesk Arnold	34
3.6	Bidirektionale Reflexionsverteilungsfunktion(BRDF)	35
3.7	Shader-Netzwerke Chaosgroup V-Ray	37
3.7.1	VRayMtl	38
3.7.2	VRayBlendMtl	38
3.7.3	VRayMtl2Sided	38
3.7.4	VRayBumpMtl	38
3.8	Shader-Netzwerke Autodesk Arnold	39
3.8.1	aiStandardSurface	39
3.9	Maya Embedded Language (MEL)	39
4	Analyse der V-Ray Shader-Netzwerke und deren Parameter	41
4.1	Das V-Ray Blatt Shader-Netzwerk	41
4.1.1	Allgemeine Struktur	43
4.1.2	Texture-Maps	44
4.1.3	Das Bump Material	46
4.1.4	Blend Material (Coat)	46
4.1.5	Coat Material	47
4.1.6	Zweiseitiges Material	48
4.1.7	Front Material	49
4.1.8	Back Material	51
4.2	Das V-Ray Bark Shader-Netzwerk	51
4.2.1	Allgemeine Struktur	51
4.2.2	Texture-Maps	51
4.2.3	Bump Material	52
4.2.4	Bark Material	52
4.3	Zusammenfassende Betrachtung	53
5	Analyse der Arnold Shader-Netzwerke und deren Parameter	55
5.1	Das Arnold Blatt Shader-Netzwerk	55
5.1.1	Allgemeine Struktur	55
5.1.2	Texture-Maps	56
5.1.3	Das Blatt-Material (aiStandardSurface)	58
5.2	Das Arnold Bark Shader-Netzwerk	60
5.2.1	Texture-Maps	60
5.2.2	Bark-Material	60

5.3 Zusammenfassende Betrachtung	60
6 Konzept und Entwicklung eines Plugins zur automatischen Konvertierung der Xfrog Shader-Netzwerke	63
6.1 Konzept	63
6.2 Das Plugin	64
6.3 Konvertierbare Materialien und getestete Software-Versionen	66
6.4 Die Konvertierung am Beispiel des V-Ray Blatt Shader-Netzwerks	67
6.5 Zusammenfassung	70
7 Evaluation	73
7.1 Aufbau der Evaluation	73
7.2 Umfang und Zielgruppe	73
7.3 Statistische Auswertung	75
7.4 Vergleich zwischen V-Ray Renderings mit V-Ray Materialien und Maya Materialien	75
7.5 Vergleich zwischen V-Ray Renderings mit V-Ray Materialien und konvertierten Maya Materialien mit automatisch generierten Texture-Maps	77
7.6 Vergleich unter Verwendung extern generierter Texture-Maps (Shadermap)	79
7.7 Vergleich unter Verwendung extern generierter Texture-Maps (Substance 3D Sampler)	81
7.8 Vergleiche von Renderings mit dem Renderer Arnold und unkonvertierten sowie konvertierten Maya Materialien	83
7.9 Vergleich von Fotografien mit V-Ray Renderings mit V-Ray Materialien und Maya Materialien	84
7.9.1 Vergleich 1	84
7.9.2 Vergleich 2	85
7.10 Zusammenfassung	85
8 Zusammenfassung und Ausblick	89
8.1 Zusammenfassung	89
8.2 Ausblick	90
A Bilderpaare von Umfrage 1 unter Verwendung des Renderers Arnold	93
Literaturverzeichnis	99

Abbildungsverzeichnis

1.1	Beispiel-Rendering einer Szene mit <i>Xfrog</i> -Pflanzenmodellen	2
1.2	Vergleich von Blättern eines V-Ray Renderings	4
2.1	Beispiel-Shader der Open Shading Language	9
2.2	Vergleich von Renderings mit MDL-Materialien	10
2.3	Kombination verschiedener MDL Shader-Bestandteile in einem einfachen Beispiel-Shader	11
2.4	Schematische Übersicht der Shader-Generierung mittels <i>MaterialX ShaderGen</i>	13
2.5	Darstellung des <i>MaterialX</i> -Viewers mit einem Kupfer-Material	15
2.6	V-Ray Rendering von sieben verschiedenen Varianten des Modells <i>Butea Monosperma</i> von <i>Xfrog</i>	16
2.7	Darstellung von 10 der 360 Pflanzen des <i>Plants Kit 1</i> von <i>Laubwerk</i>	16
2.8	Darstellung des Plant Library Browser von <i>Laubwerk</i> in Autodesk Maya	17
2.9	Übersicht der Einstellmöglichkeiten des Modells <i>Field Maple</i> von <i>Laubwerk</i> in <i>Maya</i>	18
2.10	Testrendering einer Basilikumpflanze von <i>Quixel</i> . Rendering mit <i>V-Ray für Maya</i>	20
3.1	Zellulärer Automat mit einfachem Muster und Darstellung der Generationen	24
3.2	Ersetzungssystem am Beispiel der Von-Koch-Kurve	25
3.3	Aus einem Foto und einem Scan erstellte Texturen zum Rendern von Pflanzenblättern	27
3.4	Screenshot des Tools <i>Materialize</i> zur Erstellung von Texture-Maps	28
3.5	Screenshot des Tools <i>ShaderMap</i> zur Erstellung von Texture-Maps	29
3.6	Screenshot des Programms <i>Substance 3D Sampler</i> zur Erstellung von Texture-Maps	30
3.7	V-Ray Rendering eines <i>VRayAiSurface</i> -Materials	32
3.8	V-Ray Rendering eines <i>VRayMtl</i> -Materials mit aktivierter SSS-Option	33
3.9	V-Ray Rendering eines <i>VRayFastSSS2</i> -Materials	33
3.10	V-Ray Rendering eines Blattes mit einem <i>VRayMtl2Sided</i> -Material	34
3.11	Beispiel von SSS mit dem <i>aiStandardSurface</i> -Shader	35
3.12	Vergleich zwischen einem opaken und einem transluzenten Material in Arnold	35
3.13	Darstellung der BRDF-Variablen in einem Kugelkoordinatensystem	36
3.14	Schematische Darstellung der Lichtstreuungen von BRDF und BSSRDF	37
3.15	Schematische Darstellung der Komponenten des <i>aiStandardSurface</i> -Shaders	40

3.16	Erzeugung von Polygon-Objekten in <i>Maya</i> mit der Skriptsprache <i>MEL</i>	40
4.1	Darstellung des Blatt-Shader-Netzwerkes von <i>Xfrog</i> für <i>V-Ray</i>	42
4.2	Grundstruktur des <i>V-Ray</i> Blatt-Shader-Netzwerks	43
4.3	Die sechs Texture-Maps für Blatt 1 des Modells <i>Golden Bamboo</i> von <i>Xfrog</i> . .	44
4.4	Einstellungen für alle im <i>V-Ray</i> Blatt Shader-Netzwerk verwendeten Texturen, mit Ausnahme der Normal-Map	45
4.5	Einstellungen für das Bump Material im <i>V-Ray</i> Blatt Shader-Netzwerk	46
4.6	Einstellungen für das Bump Material im <i>V-Ray</i> Leaf Shader-Netzwerk	47
4.7	Input für die Parameter <i>Blend Amount</i> und <i>Coat Material</i> im <i>V-Ray</i> Blatt Shader-Netzwerk	47
4.8	Parameter des Coat-Materials im <i>V-Ray</i> Blatt Shader-Netzwerk	48
4.9	Grundlegende Struktur des zweiseitigen Materials im <i>V-Ray</i> Blatt Shader-Netzwerk	49
4.10	Grundlegende Struktur des Front Materials im <i>V-Ray</i> Blatt Shader-Netzwerk .	50
4.11	Parameter des Front-Materials im <i>V-Ray</i> Blatt Shader-Netzwerk	50
4.12	Grundstruktur des <i>V-Ray</i> Bark Shader-Netzwerks des Modells <i>Golden Bamboo</i> von <i>Xfrog</i>	52
4.13	Die zwei Texture-Maps für die Baumrinde des Modells <i>Golden Bamboo</i> von <i>Xfrog</i>	52
5.1	Darstellung der grundlegenden Struktur des Arnold Leaf Shader-Netzwerks des Modells <i>Golden Bamboo</i> von <i>Xfrog</i>	56
5.2	Einstellungen für alle im <i>Arnold</i> Blatt Shader-Netzwerk verwendeten Texturen, mit Ausnahme der Normal-Map	57
5.3	Einstellungen der Parameter des <i>Arnold</i> Blatt-Materials	58
6.1	Darstellung des Hauptmenüs des <i>Xfrog</i> -Converter Plugins	64
6.2	Darstellung des Texture-Map Selectors des <i>Xfrog</i> -Converter Plugins	66
6.3	Darstellung des <i>V-Ray</i> Blatt Shader-Netzwerkes des Modells <i>Golden Bamboo</i> von <i>Xfrog</i>	68
6.4	Darstellung des <i>Maya</i> Blatt Shader-Netzwerkes des Modells <i>Golden Bamboo</i> von <i>Xfrog</i>	69
6.5	Darstellung des zu <i>V-Ray</i> konvertierten <i>Maya</i> Blatt Shader-Netzwerkes des Mo- dells <i>Golden Bamboo</i> von <i>Xfrog</i>	69
6.6	Darstellung des zu <i>V-Ray</i> konvertierten <i>Maya</i> Blatt Shader-Netzwerkes des Mo- dells <i>Golden Bamboo</i> von <i>Xfrog</i> mit Texture-Maps aus <i>ShaderMap</i>	70
7.1	Bilderpaare 1, 9 und 17 von Umfrage 1	76
7.2	Bilderpaare 2, 10 und 18 von Umfrage 1	78
7.3	Bilderpaare 3, 11 und 19 von Umfrage 1	80
7.4	Bilderpaare 4, 12 und 20 von Umfrage 1	82
7.5	Darstellung der im ersten Teil von Umfrage 2 verwendeten Blätter	84
7.6	Darstellung der im zweiten Teil von Umfrage 2 verwendeten Blätter	85
A.1	Bilderpaare 5, 13 und 21 von Umfrage 1	94
A.2	Bilderpaare 6, 14 und 22 von Umfrage 1	95

A.3 Bilderpaare 7, 15 und 23 von Umfrage 1	96
A.4 Bilderpaare 8, 16 und 24 von Umfrage 1	97

Kapitel 1

Einleitung

In dieser Arbeit wird die Machbarkeit einer automatischen Konvertierung von Materialien der 3D-Pflanzenmodelle des Anbieters *Xfrog*¹ untersucht. Die Konvertierung wird in Form eines Plug-ins in Autodesk Maya implementiert und soll die Umwandlung der von *Xfrog* angebotenen Materialien für die Renderer *V-Ray*² von *Chaosgroup* und *Arnold*³ von *Autodesk* realisieren. Bei erfolgreicher Konvertierung werden die Ergebnisse hinsichtlich Kompatibilität, Flexibilität, Performance und schließlich der Renderqualität untersucht.

1.1 Motivation

Der große technische Fortschritt im Bereich des 3D-Renderings erlaubt das Erzeugen von fotorealistischen Bildern, die meist kaum oder gar nicht von der Realität unterscheidbar sind. Unzählige Hollywoodfilme enthalten computergenerierte 3D-Modelle, welche der Kreativität nahezu unendliche Möglichkeiten bieten. Diese Techniken werden allerdings nicht nur von der Filmindustrie genutzt, sondern sie sind mittlerweile in nahezu jedem medialen Erzeugnis anzutreffen. Während dies vor einigen Jahren noch großen Unternehmen mit der nötigen Rechenleistung vorenthalten war, sind heute auch kleine Unternehmen und sogar Privatleute mit geringen finanziellen Mitteln in der Lage, 3D-Szenen in Studioqualität zu rendern. Computergenerierte Bilder ermöglichen nicht nur die Darstellung von Objekten und Sachverhalten, die mit herkömmlichen Methoden nicht darstellbar wären, sondern sie bedeuten zudem meist auch eine enorme Zeit- und Kostenersparnis.

Während in der technischen Produktvisualisierung häufig bereits 3D-Modelle aus der Produktentwicklung vorliegen und somit selbst geschaffene Objekte dargestellt werden sollen, müssen Pflanzen möglichst detailgetreu der Natur nachempfunden werden. Aus diesem Grund und nicht zuletzt aufgrund der enormen Vielfalt der Pflanzenwelt, stellt die realistische Darstellung von Vegetation eine besondere Herausforderung in der 3D-Visualisierung dar. Damit der Nutzer nicht jede benötigte Pflanze modellieren und texturieren muss, bieten Unternehmen wie *Xfrog* eine große Auswahl an 3D-Modellen der Vegetation zum Kauf an.

¹<http://xfrog.com/>

²<https://www.chaosgroup.com/vray/maya>

³<https://www.arnoldrenderer.com/arnold/arnold-for-maya/>



Abbildung 1.1: Beispiel-Rendering einer Szene mit Xfrog-Pflanzenmodellen.

Quelle: <http://xfrog.com/gallery/Image%20of%20the%20Day/inside-massi-san.jpg.php>
- Zuletzt geprüft: 07.05.2021

Die Modelle werden üblicherweise in mehreren Varianten angeboten, so dass jede Pflanze in verschiedenen Wachstumsstadien ihres Lebenszyklus verwendet werden kann. Es sind unzählige Modelle real existierender Pflanzen, aus allen Bereichen der Erde, als digitaler Download erhältlich. Diese Modelle werden algorithmisch erzeugt und die Texturen werden per Scan erstellt. Aus den Texturen werden wiederum Materialien erstellt, die von den verschiedenen am Markt erhältlichen Renderern interpretiert und schließlich gerendert werden können. Hierbei enthält jedes Modell meist mehrere Materialien, so dass Stamm, Blätter, Blüten und weitere Bestandteile ihr jeweils charakteristisches Erscheinungsbild erhalten. Zudem können die enthaltenen Materialien in verschiedenen Qualitätsabstufungen verwendet werden. So kann für sehr detailreiche Nahaufnahmen eine komplexe Variante des jeweiligen Materials verwendet werden, während für ein weit entferntes Objekt eine sehr einfache Variante des Materials zum Einsatz kommen kann. Auf diese Weise kann effektiv Rechenleistung eingespart und für wichtige Bereiche des Bildes eingesetzt werden. Abbildung 1.1 zeigt beispielhaft ein Rendering mit Xfrog-Pflanzenmodellen.

Das Verwenden von 3D-Modellen der Vegetation ist für eine Vielzahl von Anwendern interessant: Spieleentwickler, Architekten, Printmedien und viele weitere Bereiche können ihre Produkte auf diese Weise aufwerten. Üblicherweise werden hierbei allerdings branchenspezifische Softwarelösungen zur Erstellung und zum Rendern ihres digitalen Produkts verwendet. Jede Software stellt hierbei ihre individuellen Anforderungen an die Formate von 3D-Modellen und deren Materialien. Dies stellt die Anbieter von 3D-Modellen vor die schwierige Aufgabe, ihre Modelle für möglichst viele Softwarelösungen kompatibel zu gestalten.

1.2 Problemstellung

Aufgrund der großen und stetig wachsenden Anzahl an verfügbarer Render-Software, ist es Unternehmen wie *Xfrog* nahezu unmöglich, kompatible Materialien eines jeden Pflanzenmodells für jede Render-Software anzubieten. Jede Render-Software verwendet ihre eigene Struktur der Shader-Netzwerke, um die Materialien möglichst realitätsgetreu abzubilden. Zusätzlich zum individuellen Aufbau der Shader-Netzwerke, enthalten diese zudem eine Vielzahl an Attributen, welche meist frei gewählt werden können. Der Nutzer hat in dem Fall, in dem das jeweilige Modell nicht für die gewünschte Render-Software verfügbar ist, zwei Möglichkeiten: Er muss ein anderes, kompatibles Modell wählen oder er muss die Materialien manuell der jeweiligen Render-Software anpassen. Die manuelle Anpassung ist sehr zeitaufwändig und fehleranfällig, zumal jedes Modell meist mehrere Materialien enthält. Um letztendlich ein fotorealistisches Ergebnis zu erhalten, ist es notwendig, die Attribute der Materialien so zu wählen, dass sie das Erscheinungsbild der realen Pflanze möglichst detailgetreu nachbilden. Auch hierzu verwenden verschiedene Renderer unterschiedliche Werte, welche nicht immer durch reale Parameter mit genormten Einheiten repräsentiert werden. Somit existieren in den meisten Fällen keine einheitlichen Umrechnungsfaktoren oder Gleichungen, die diesen Prozess universell gültig machen.

Der Renderer *V-Ray* von *Chaosgroup* unterstützt nativ das Rendern der Standard *Maya*-Materialien. Die *Maya*-Materialien bieten allerdings bei weitem nicht den gleichen Funktionsumfang der *V-Ray* Materialien. Somit lässt sich ein Pflanzenmodell, welches für den *Maya*-Renderer angeboten wird, zwar mit *V-Ray* ohne Probleme rendern, allerdings fehlen den entstehenden *V-Ray* Materialien wichtige Informationen, so dass das Rendering nicht dem gewünschten Ergebnis entspricht. In Abbildung 1.2 sind zur Verdeutlichung dieses Umstandes die Blätter zweier Bäume dargestellt. Das Bild wurde mit *V-Ray* gerendert und es wurde auf der linken Seite ein *Maya*-Modell und auf der rechten Seite ein *V-Ray* Modell des gleichen Baumes verwendet. Zwar mussten die Materialien des *Maya*-Modells nicht konvertiert werden, um sie rendern zu können, doch es ist deutlich zu erkennen, dass die Blätter des rechten Baumes realistischer dargestellt werden. Während das *Maya*-Material lediglich ein Bild eines echten Blattes für die Darstellung aller Blätter verwendet, enthält das *V-Ray* Material unterschiedliche Texturen für Vorder- und Rückseite der Blätter. Zudem verwendet das *V-Ray* Modell ein zweiseitiges Material (*V-RayMtl2Sided*) für die Blätter, welches das Durchdringen von Licht simuliert. Es wird deutlich, dass es eine allumfassende automatische Konvertierung für jedes erdenkliche Material nicht geben kann, da die Materialien immer den gegebenen Umständen angepasst werden müssen. Ein Blatt eines Baumes benötigt eine vollkommen andere Shader-Struktur als die menschliche Haut oder technische Materialien wie Metalle.

1.3 Zielsetzung

Ziel dieser Arbeit ist es, zu erforschen, ob eine automatische Materialkonvertierung der angebotenen *Xfrog*-Modelle möglich ist, ohne dem Nutzer die nötige Flexibilität zu nehmen, gewünschte Parameter individuell zu gestalten. Des Weiteren soll untersucht werden, ob für



Abbildung 1.2: Vergleich von Blättern eines V-Ray Renderings. Links ist das Maya-Modell des Baumes zu sehen. Auf der rechten Seite ist das V-Ray Modell des Baumes zu sehen. Es ist deutlich zu erkennen, dass die Blätter des linken Baumes weniger realistisch dargestellt werden. Die Blätter des rechten Baumes sind auf der Rückseite heller, um das Durchdringen von Licht zu simulieren

die jeweilige Rendssoftware bereits verfügbare Materialien, im Hinblick auf Fotorealismus sowie Renderdauer, auf diese Weise optimiert werden können.

Die automatische Konvertierung wird mit *Autodesk Maya* umgesetzt, daher wird prototypisch ein Maya-Plugin zur automatischen Konvertierung für die Renderer *V-Ray* von *Chaosgroup* und *Arnold* von *Autodesk* entwickelt und anschließend eine detaillierte Evaluation bezüglich Kompatibilität, Flexibilität, Performance und schließlich der Renderqualität durchgeführt.

Zudem ist diese Untersuchung für die *Technische Hochschule Mittelhessen* interessant, da die Studierenden der Medieninformatik mit Modellen von *Xfrog* arbeiten und es wünschenswert wäre, dass sie ihren Arbeitsaufwand für das kreative Gestalten ihrer 3D-Szenen aufbringen können, anstatt den Großteil ihrer Zeit mit dem manuellen Konvertieren der Materialien zu verschwenden. Auf diese Weise sollen die Untersuchungsergebnisse dieser Arbeit aktiv in den Lehrbetrieb der Hochschule einfließen und auch im Anschluss an diese Arbeit gepflegt, optimiert und verwaltet werden.

1.4 Aufbau der Arbeit

Die Arbeit ist in acht Kapitel gegliedert. Das erste Kapitel, die Einleitung, zeigt auf, wie wichtig und allgegenwärtig fotorealistische Renderings in unserem Alltag sind. Zudem wird

gezeigt, welche Probleme bei der Verwendung von 3D-Pflanzenmodellen auftreten und wie diese Arbeit dazu dienen soll, diese Probleme zu lösen.

Das zweite Kapitel, Stand der Technik, beschreibt den gegenwärtigen Stand der Forschung zu dem Versuch, eine einheitliche Beschreibung von Materialien von 3D-Modellen zu erschaffen, sodass 3D-Szenen mit unterschiedlicher Software gerendert werden können und dennoch ein nahezu identisches Ergebnis liefern.

Im dritten Kapitel, Grundlagen, wird zunächst die Erzeugung von 3D-Pflanzenmodellen beschrieben. Anschließend werden die grundlegenden Funktionen der in dieser Arbeit verwendeten Renderer, sowie die Skriptsprache Maya Embedded Language (MEL) von *Autodesk Maya*, die zur Entwicklung des Plugins verwendet wurde, vorgestellt.

In Kapitel vier und fünf werden die Shader-Netzwerke der von *Xfrog* angelieferten Materialien für die Renderer *V-Ray* und *Arnold* analysiert und deren Parameter hinsichtlich Plausibilität, in Bezug auf fotorealistisches Rendering, untersucht.

Kapitel sechs beschreibt das Konzept und die Entwicklung des Plugins zur automatischen Konvertierung der *Xfrog* Shader-Netzwerke. Es werden alle Funktionen detailliert beschrieben, sowie die grundlegende Funktionsweise erläutert.

In Kapitel sieben erfolgt die Auswertung der Umfragen, in welchen die Befragten verschiedene Renderings vergleichen sollen. Nach der Erläuterung des Aufbaus der Evaluation werden die erhaltenen Antworten ausgewertet.

Im letzten Kapitel, Fazit und Ausblick, wird die Arbeit und deren Ergebnisse zusammengefasst und es wird ein Ausblick gegeben, ob und wie die automatische Konvertierung von Shader-Netzwerken in künftigen Projekten eingesetzt werden kann.

1.5 Zusammenfassung der Ergebnisse

Die Benutzung von auf dem Markt erhältlichen 3D-Modellen der Pflanzenwelt ist, trotz neuer Technologien, welche eine anwendungsübergreifende Verwendung von Materialien ermöglichen sollen, weiterhin an den Gebrauch eines dem Material der Modelle entsprechenden Renderer gebunden. Möchte der Nutzer ein bestimmtes Modell nutzen, welches nicht für den vorgesehenen Renderer verfügbar ist, so müssen die Materialien manuell an den zu verwendenden Renderer angepasst werden. Die meisten Renderer, wie zum Beispiel *V-Ray* von *Chaosgroup*, sind in der Lage, *Maya* Materialien vor dem Rendervorgang in verwendbare Materialien zu konvertieren, allerdings gehen hierbei wichtige Eigenschaften des Materials, wie die Transluzenz, verloren. Für eine benutzerfreundliche, einfache und schnelle Konvertierung wurde der *Xfrog-Converter* entwickelt, welcher die vorhandenen *Maya* Materialien in korrekt interpretierbare *V-Ray* und *Arnold* Materialien konvertiert.

Das aus 12 Skripten bestehende *Maya* Plugin wurde verwendet, um Materialien zu konvertieren, welche im Anschluss in verschiedenen Renderings im Rahmen zweier Umfragen verglichen wurden. Es konnte gezeigt werden, dass das Plugin qualitativ vergleichbare Ergebnisse erzielen kann und somit für eine schnelle Konvertierung von Materialien, welche nicht für *V-Ray* oder *Arnold* verfügbar sind, geeignet ist.

Kapitel 2

Stand der Technik

2.1 Einleitung

Anhand aktueller Bemühungen in der Welt der Computergrafik, einheitliche und softwareübergreifende Beschreibungen von Materialien zu definieren, wird deutlich, dass die Industrie nach gemeinsamen Standards sucht, um das Rendern von 3D-Szenen mittels verschiedener Software, aber mit gleichem Ergebnis, zu realisieren. Zu diesem Zweck wurden Technologien entwickelt, welche die anwendungsübergreifende Verwendung von Materialien oder ganzen 3D-Szenen ermöglichen. Im Folgenden werden die drei wichtigsten aktuellen Ansätze dieser Idee erläutert und deren Gemeinsamkeiten und Unterschiede hervorgehoben. Zusätzlich werden in diesem Kapitel Anbieter von 3D-Pflanzenmodellen verglichen, um den Umgang mit den Materialien für die verschiedenen Renderer zu beleuchten.

2.2 Open Shading Language (OSL)

Die Open Shading Language (OSL) ist eine von *Sony Pictures Imageworks*¹ entwickelte Shader-Sprache, welche für den Renderer *Arnold* entworfen wurde. *OSL* kam bereits in vielen Filmproduktionen wie „The Falcon and the Winter Soldier“ oder „Toy Story 4“ zum Einsatz. *OSL* wurde als Open Source veröffentlicht und somit anderen Studios und Anbietern von Render-Software zugänglich gemacht. Es ist zur Zeit die de facto Standard Shader-Sprache und die Arbeit an *OSL* wurde 2017 mit dem *Academy Award for Technical Achievement* ausgezeichnet. Der *OSL* Code fällt unter die „BSD 3-Clause“-Lizenz, was es Entwicklern ermöglicht, den Code in eigene Software zu integrieren und nach Belieben zu modifizieren, insofern die Copyright-Anmerkungen eingehalten werden, welche in der Lizenz beschrieben sind [Aca21a].

OSL verwendet die Syntax der Programmiersprache *C* und wurde speziell für fortgeschrittene Renderalgorithmen entworfen. Mit *OSL* kann der Anwender eigene Shader schreiben, die den jeweiligen speziellen Anforderungen genügen, falls die gewünschten Funktionen nicht durch Shader abgebildet werden, welche in der verwendeten Software bereits verfügbar sind.

¹<http://opensource.imageworks.com/osl.html>

2. STAND DER TECHNIK

Allerdings ist nicht jedes am Markt erhältliche Digital-Content-Creation Tool (DCC) für die Verwendung von *OSL* ausgelegt. In Tabelle 2.1 sind die *OSL*-kompatiblen DCC-Tools und Renderer aufgelistet.

Unternehmen	Software
Sony Pictures Imageworks	in-house „Arnold“ renderer
Chaos Group	V-Ray
Pixar	Photorealistic RenderMan RIS
Isotropix	Clarisse
Autodesk	Beast
Open Source	Appleseed
Animal Logic	Glimpse renderer
Image Engine	Gaffer(for expressions and deformer)
DNA Research	3Delight
Autodesk/SolidAngle	Arnold
Autodesk	3DS Max 2019

Tabelle 2.1: Auflistung aller DCC-Tools und Renderer, welche die Verwendung der Open Shading Language unterstützen

Quelle: <https://github.com/AcademySoftwareFoundation/OpenShadingLanguage> - Zuletzt geprüft: 14.06.2021

Zudem variieren die verfügbaren *OSL*-Funktionen zwischen den unterschiedlichen Anwendungen. *Blender* schließt die Verwendung von *OSL*-Nodes für das Rendern mittels Grafikkarte (GPU-Rendering) aus², während *Autodesk Arnold* die Verwendung von *OSL* für GPU-Rendering implementiert hat, wenn auch mit einigen Einschränkungen³.

Ein *OSL*-Shader besitzt Inputs und Outputs, welche als Shader-Parameter bezeichnet werden. Die Shader-Parameter erhalten zunächst immer Standardwerte, sie können vom Renderer allerdings zur Laufzeit überschrieben werden. Der Beispiel-Shader in Abbildung 2.1 erhält als Input einen Farbwert (Cin) und gibt einen gammakorrigierten Farbwert als Output (Cout) wieder aus. *OSL* weist zu anderen Shader-Sprachen einige Unterschiede auf, von welchen im Folgenden die wichtigsten erläutert werden [GSKC10]:

Die Surface- und Volumeshader von *OSL* berechnen nicht die Farben von Oberflächen, sondern speichern die berechneten Informationen in sogenannten *Closures* (zu Deutsch Verschluss oder Schließung). Herkömmliche Shader berechnen die Oberflächenfarben auf direkte Weise und dies zudem nur aus einer bestimmten Richtung. Der Renderer kann in diesem Fall lediglich die Berechnung ausführen, um schließlich diesen einen Wert zu erhalten. *OSL* Shader hingegen berechnen symbolische Beschreibungen der Art und Weise, wie eine Oberfläche Licht streut oder emittiert. Die Berechnungen liefern physikalische Werte mit Einheiten der Strahlung. Auf diese Weise kann der entsprechende Renderer die Berechnung zu einem beliebigen Zeitpunkt durchführen und zudem aktiv eine Richtung bestimmen, die für die Be-

²https://docs.blender.org/manual/en/latest/render/shader_nodes/osl.html

³<https://docs.arnoldrenderer.com/display/A5ARP/Supported+Features+and+Known+Limitations>

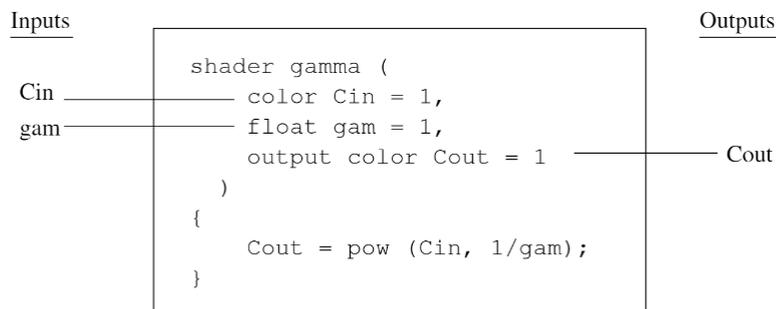


Abbildung 2.1: Beispiel eines *OSL*-Shaders mit Inputs und Outputs. Der Shader erhält als Input einen Farbwert (*Cin*) und gibt einen gammakorrigierten Farbwert als Output (*Cout*) wieder aus

Quelle: Academy Software Foundation aus [Aca21b]

rechnung sinnvoll erscheint. Diese Herangehensweise eignet sich ideal für physically-based rendering (PBR). Ein weiterer Vorteil der Verwendung von *Closures* ist es, dass der Renderer ein Strahlenkontingent ermitteln kann, um das Sampling der Oberfläche zu optimieren und somit Rechenleistung einzusparen.

Im Gegensatz zu anderen Shader-Sprachen, ist es bei Verwendung von *OSL* zudem nicht notwendig, einzelne Renderbestandteile, sogenannte AOV's (arbitrary output variables), per Code im Shader zu definieren. AOV's werden üblicherweise verwendet, um die verschiedenen Lichtberechnungen des Renderers einzeln auszugeben und im Anschluss im Compositing wieder zusammenzufügen. Somit ist es beispielsweise möglich, auch nach dem Rendervorgang die Intensität der diffusen Reflexionen (oder weitere Parameter, welche als AOV exportiert wurden) zu modifizieren. *OSL* benötigt hierzu keine explizite Renderanweisung, sondern stellt die einzelnen Bestandteile über eine Regular-Expression basierte Notation bereit.

Abschließend ist zudem zu erwähnen, dass *OSL* einen LLVM (Low Level Virtual Machine) Compiler verwendet, der die Shader-Netzwerke spontan (just in time oder „JIT“) in Maschinensprache übersetzt und dabei stark optimiert. Der Compiler hat hierbei globales Wissen über die Shader-Parameter, die er nicht hätte, wenn der Sourcecode des Shaders kompiliert werden würde. Auf diese Weise kann die Ausführung der Shader-Netzwerke bis zu 25% schneller als herkömmlich erfolgen [Aca21a].

2.3 Nvidia Material Definition Language(MDL)

Die *Nvidia Material Definition Language*, entwickelt von der *Nvidia Corporation*, dient der Definition von Materialien, welche anwendungsübergreifend verwendet werden können. Hierbei beschreiben *MDL*-Materialien das Verhalten des Lichts auf einer hohen Abstraktionsebene, was sie von einer Shader-Sprache, wie der *Open Shading Language* (*OSL*, siehe Kapitel 2.2), unterscheidet [Cor21]. Wie das Bild letztendlich gerendert wird, ist ausschließlich dem verwendeten Renderer überlassen. Für die anwendungsübergreifende Verwendung wird allerdings zusätzliche Software benötigt und die Anzahl der kompatiblen Programme ist relativ

2. STAND DER TECHNIK

klein. In Tabelle 2.2 sind die entsprechenden Unternehmen und die jeweilige kompatible Software aufgelistet. *Nvidia* bietet zudem ein Software Development Kit (SDK) für Entwickler an, um *MDL*-Kompatibilität in die eigene Software zu integrieren.

Unternehmen	Software
Adobe (Allegorithmic)	Substance 3D Designer
Chaos Group	V-Ray for 3ds Max/Maya
Dassault Systems	Solidworks Visualize
DAZ 3D	DAZ Studio
Epic	Unreal Studio
NVIDIA	Iray for 3ds Max/Rhino/Maya
Siemens	NX

Tabelle 2.2: Auflistung aller Softwareanwendungen, welche die Verwendung von *MDL*-Materialien unterstützen

Quelle: <https://developer.nvidia.com/vmaterials> - Zuletzt geprüft: 15.06.2021

Die abstrakte Beschreibung eines *MDL*-Materials ermöglicht eine nahezu identische optische Darstellung in verschiedenen Render-Anwendungen. Auf diese Weise kann der Nutzer beispielsweise zwischen schnellem Ray-Tracing und Global Illumination wechseln, ohne die Szene anpassen zu müssen. Dies ist Beispielhaft in Abbildung 2.2 dargestellt.



Abbildung 2.2: Vergleich von Renderings mit *MDL*-Materialien aus verschiedenen Render-Anwendungen

Quelle: NVIDIA aus [Cor21]

Nvidia bietet die kostenlose MDL-Sammlung *vMaterials*⁴ zum Download an. Sie beinhaltet eine Vielzahl an Materialien und Material-Variationen, welche in kompatibler Software eingesetzt werden können. *V-Ray für Maya* beinhaltet ebenfalls bereits einige MDL-Materialien, welche nach der Installation von *V-Ray* sofort nutzbar sind.

Mit *Substance 3D Designer* von Adobe können eigene MDL-Materialien erstellt und weiterführend beispielsweise in *Substance 3D Painter* verwendet oder mit einem der kompatiblen Renderer gerendert werden. Hier können auch alle konfigurierbaren Parameter des Materials von außen sichtbar gemacht werden (exposed parameters), sodass diese in der zu verwendenden 3D-Anwendung nach Belieben angepasst werden können. Zudem können zuvor erstellte Materialien und Texturen als Input für die einzelnen Bausteine des MDL-Materials verwendet werden. Sehr interessant ist auch die Möglichkeit, Scanergebnisse von Materialscannern (siehe Kapitel 3.2.2), wie die der Geräte von *X-Rite*⁵, in die Shader mit einzubinden und somit realistische physikalische Parameter der real existierenden Materialien zu verwenden. Die einzelnen Bestandteile der Shader können kombiniert werden, um das Verhalten der Shader nach eigenen Wünschen anpassen zu können. In Abbildung 2.3 ist ein einfacher Shader dargestellt, der die Kombination der einzelner Shader-Bestandteile realisiert.

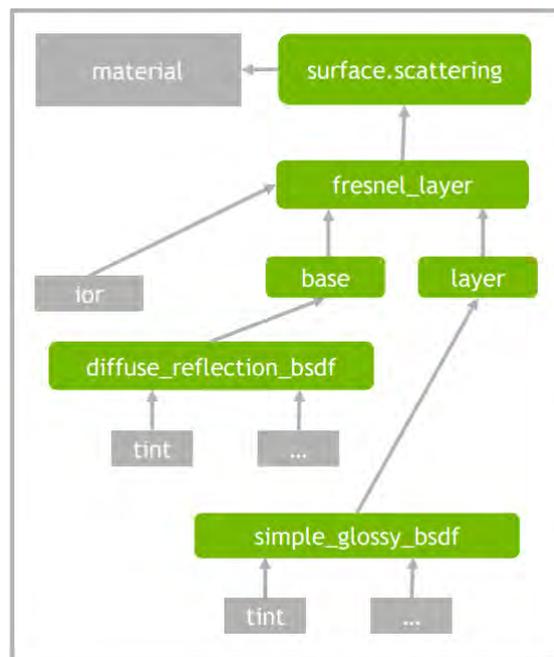


Abbildung 2.3: Kombination verschiedener MDL Shader-Bestandteile in einem einfachen Beispiel-Shader

Quelle: Aus [LE19]

Zu sehen ist die Kombination zweier Reflexionsverteilungsfunktionen, welche letztendlich

⁴<https://developer.nvidia.com/vmaterials>

⁵<https://www.xrite.com/de/>

die Shaderfunktionen definieren. Die Reflexionsverteilungsfunktionen werden über die Nodes *diffuse_reflection_bsdf* und *simple_glossy_bsdf* abgebildet. Diese Nodes stellen Funktionen dar, welche die physikalische Verteilung der Lichtstreuung beschreiben (siehe Kapitel 3.6). Die diffuse Reflexion dient als Basisschicht des darauf folgenden *fresnel_layer*, während die glänzende Reflexion eine auf der diffusen Reflexion aufgetragene Glanzschicht darstellt. Der *fresnel_layer* kombiniert beide Schichten in Abhängigkeit des Betrachtungswinkels und des Refraktionsindex (*ior*, index of refraction). Somit erscheint das Material an den vom Betrachter abgewandten Seiten reflektierender, als bei senkrechter Betrachtung. Auf diese Weise lassen sich Shader mit einer beliebigen Anzahl an Schichten erstellen. Die Eingabeparameter der Reflexionsverteilungsfunktionen können von außen sichtbar gemacht (*exposed parameters*) und später in einem kompatiblen DCC-Tool beliebig angepasst werden. In diesem Fall könnten beispielsweise die *tint*-Parameter zur späteren Definition der Farben des Materials verwendet werden. Schließlich kann eine eigene Sammlung an Materialien und Material-Variationen aufgebaut werden, welche später in allen MDL-kompatiblen Anwendungen verwendet werden kann.

2.4 MaterialX

MaterialX ist ein offener Standard, der einen Austausch von Material- und Look-Development Inhalten zwischen DCC-Anwendungen und Renderern ermöglicht. Entwickelt wurde das Open Source Projekt von Lucasfilm⁶. Das Projekt wurde 2012 gestartet und zum ersten Mal in *Star Wars: The Force Awakens* in 2015 eingesetzt [med19]. *MaterialX* wird derzeit nur von wenigen Anwendungen unterstützt⁷:

- Substance 3D Designer - MaterialX Plugin
- Autodesk Arnold - MaterialX Operator
- Pixar - USD (Universal Scene Description)/MaterialX

Da *MaterialX* nicht nur Materialien, sondern ganze Looks plattformübergreifend verwendbar macht, ist zunächst eine Definition der Bezeichnung *Look* nötig. Hierzu werden in der MaterialX-Spezifikation vier Datenbeziehungen genannt, die benötigt werden, um einen Look zu spezifizieren:

- Bildverarbeitungs-Netzwerke mit Quellen, Operatoren, Verbindungen und Parameter, welche eine oder mehrere Dateien zu Texturen kombinieren.
- Geometriespezifische Informationen, wie zugeordnete Textur-Dateinamen oder ID's für die verschiedenen Mapping Arten.
- Parameterwerte und Verbindungen zu den Bildverarbeitungs-Netzwerken.
- Eine Verbindung zwischen Geometrien und Materialien in einem Modell.

⁶<https://www.lucasfilm.com/>

⁷<http://www.materialx.org/ThirdPartySupport.html>

Um *MaterialX*-Inhalte lesen und schreiben zu können, wird ein XML-basiertes Dateiformat verwendet. Die *MaterialX*-Beschreibungen enthalten keinerlei Details über die Implementation der jeweiligen Materialien. Damit eine Anwendung diese Beschreibungen nutzen kann, werden mit Hilfe eines Shader-Generators entsprechende Shader generiert. *MaterialX* enthält jeweils einen Shader-Generator für jede unterstützte Shader-Sprache. Das Generieren der verschiedenen Shader wird mit Hilfe von *MaterialX ShaderGen*⁸ realisiert. Eine schematische Funktionsübersicht ist in Abbildung 2.4 dargestellt.

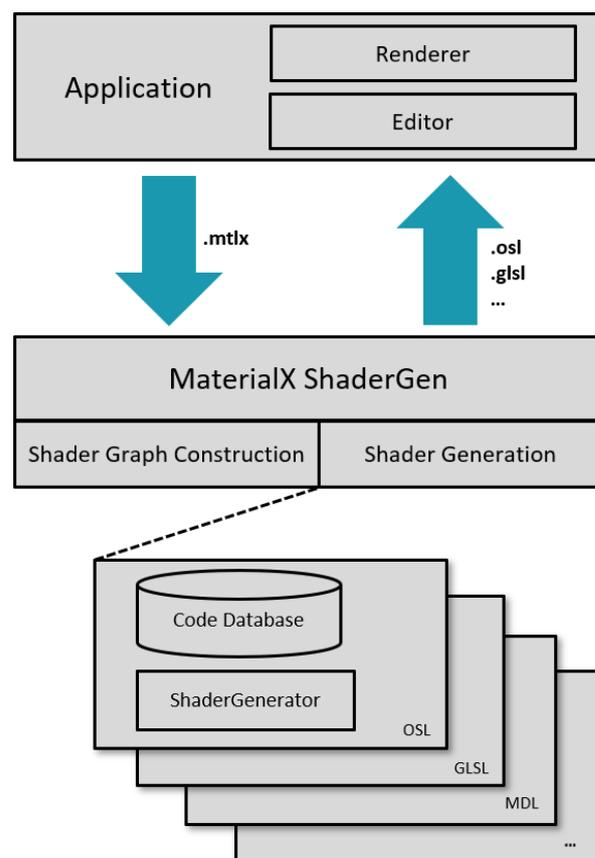


Abbildung 2.4: Schematische Übersicht der Shader-Generierung mittels *MaterialX ShaderGen*

Quelle: MaterialX ShaderGen aus [KS21]

ShaderGen erhält die Shaderdefinition von MaterialX in Form einer *.mtlx*-Datei. Anhand der in der Code-Datenbank enthaltenen Definitionen der jeweiligen kompatiblen Shader-Sprachen, wird nun ein entsprechender Shader generiert, welcher von einer kompatiblen Anwendung verwendet werden kann. Der entstehende Sourcecode wird vom Compiler der

⁸<https://github.com/materialx/MaterialX/blob/main/documents/DeveloperGuide/ShaderGeneration.md>

Shader-Sprache kompiliert, bevor dieser vom verwendeten Renderer ausgeführt wird. Zum Zeitpunkt der Erstellung dieser Arbeit umfasst ShaderGen die Generierung von Shadern für die Shader-Sprachen

- Open Shading Language (OSL, siehe Kapitel 2.2)
- OpenGL Shading Language (GLSL)
- NVIDIA Material Definition Language (MDL, siehe Kapitel 2.3)

Zudem können durch Vererbung der vorhandenen Klassen von ShaderGen individuelle Generatoren programmiert werden, um die Liste der kompatiblen Shader-Sprachen nach eigenen Wünschen zu erweitern.

Des Weiteren enthält *MaterialX* den *MaterialX Viewer*, welcher ein schnelles Anzeigen eines *MaterialX*-Materials ermöglicht. In Abbildung 2.5 ist der *MaterialX-Viewer* mit einem Kupfer-Material zu sehen. Das Programm erlaubt die Verwendung eigener 3D-Modelle und HDR-Environment-Maps. Über den *Property Editor* können alle im *MaterialX*-Material definierten Parameter variiert und Texture-Maps ausgewählt werden. Geänderte Parameter können zudem in einem neuen Material gespeichert werden. Gerendert wird das angezeigte Bild per OpenGL.

2.5 Vergleich von Anbietern von 3D-Pflanzenmodellen

Wie in den vorigen Abschnitten gezeigt wurde, ist die Industrie bemüht, offene Standards zu erschaffen, um eine anwendungsübergreifende Verwendung von Materialien von 3D-Modellen zu ermöglichen. Doch so sinnvoll die vorgestellten Technologien für einzelne Workflows und Pipelines sind, bieten sie Herstellern von 3D-Pflanzenmodellen wenig bis keine Vorteile für eine universelle Distribution ihrer Produkte, solange nicht alle relevanten DCC's und Renderer kompatibel für die Frameworks sind. Aus diesem Grund müssen die Anbieter eine Auswahl an Renderern treffen, für welche sie ihre Shader-Netzwerke entwickeln. Da Umfang und Qualität der Produktpaletten zwischen den Anbietern variieren, werden in diesem Abschnitt verschiedene Anbieter von 3D-Pflanzenmodellen verglichen, um Gemeinsamkeiten und Unterschiede im Umgang mit den Materialien für die verschiedenen Renderer aufzuzeigen. Hierbei ist zu beachten, dass der Vergleich keinesfalls alle existierenden Anbieter berücksichtigen kann.

2.5.1 Xfrog

*Xfrog*⁹ ist ein Unternehmen, welches 3D-Modelle der Vegetation zum Kauf anbietet. Ge gründet wurde das Unternehmen in Karlsruhe im Jahre 1996 von *Oliver Deussen* und *Bernd Lintermann*. *Xfrog* bietet zudem Software an, mit welcher sich Pflanzenstrukturen generisch erzeugen lassen (siehe Kapitel 3.2.1). *Xfrog* wurde für den *Scientific Achievement Award 2015* von der *Academy of Motion Pictures (OSCAR)* nominiert und kam in Filmen wie *Avatar*, *Krieg der Welten* und *Polar Express* zum Einsatz. Hinter *Xfrog* steht fundierte

⁹<http://xfrog.com/>

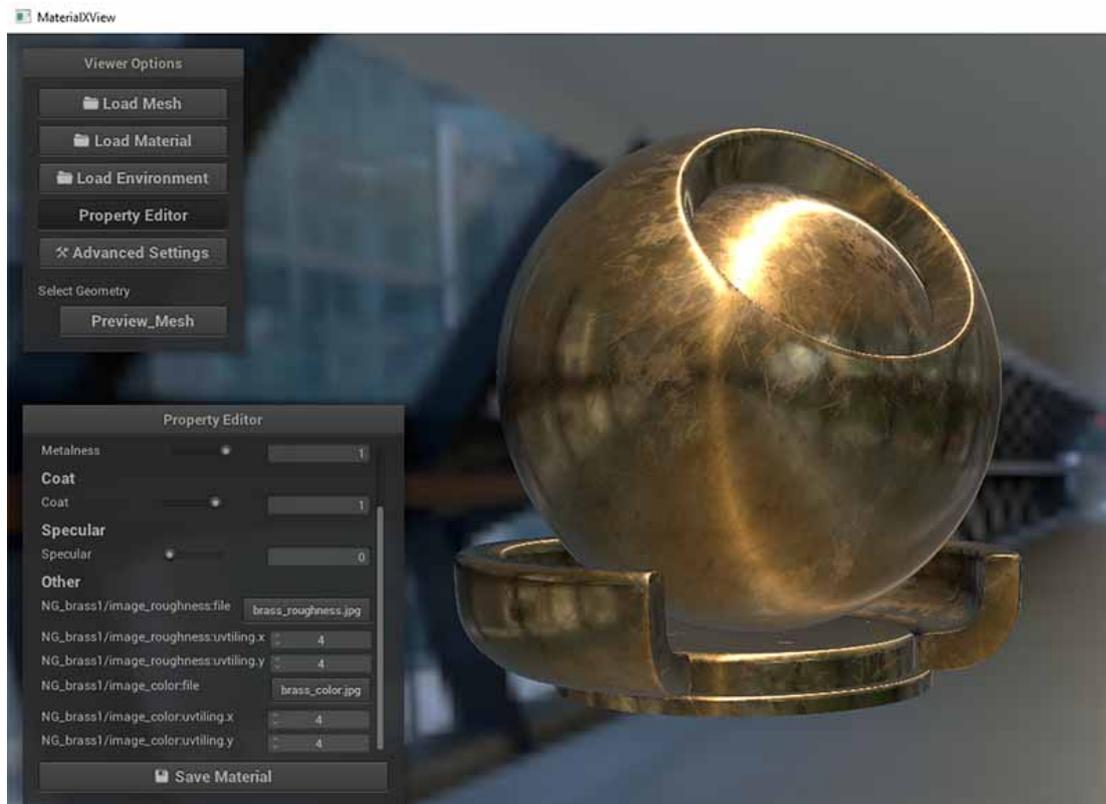


Abbildung 2.5: Darstellung des *MaterialX*-Viewers mit einem Kupfer-Material

Quelle: Eigene Darstellung

wissenschaftliche Forschung in vielen Bereichen, die für eine möglichst gute Approximation des Erscheinungsbildes der Modelle an reale Pflanzen nötig sind¹⁰. Auf der Homepage von *Xfrog* werden unzählige Modelle aus allen Teilen der Welt angeboten, welche auch in Paketen erhältlich sind. Paket 1 enthält 1200 Modelle und Paket 2 enthält 1600 Modelle. Die enthaltenen Modelle sind für verschiedene DCC-Tools und Renderer vorbereitet, allerdings unterscheidet sich diesbezüglich Paket 1 von Paket 2. Gegenstand dieser Arbeit ist die Untersuchung einer Möglichkeit, jene Materialien von *Xfrog* für die Renderer *V-Ray* von *Chaosgroup* und *Arnold* von *Autodesk* zu konvertieren, welche beim Kauf nicht für den Einsatz mit diesen Renderern vorgesehen sind. Die 1200 Modelle aus Paket 1 sind für *Maya* lediglich für den Renderer *Mental Ray* ausgelegt und somit handelt es sich hierbei um jene Modelle, die für diese Arbeit relevant sind. Jedes *Xfrog*-Modell enthält mehrere Varianten des Modells, welche die verschiedenen Stadien in dessen Lebenszyklus darstellen. Abbildung 2.6 zeigt sieben verschiedene Varianten des Modells *Butea Monosperma* von *Xfrog*.

¹⁰<http://xfrog.com/company/research/>



Abbildung 2.6: V-Ray Rendering von sieben verschiedenen Varianten des Modells *Butea Monosperma* von Xfrog. Die Varianten stellen unterschiedliche Zeitpunkte im Lebenszyklus der Pflanze dar

Quelle: Eigene Darstellung

2.5.2 Laubwerk

Die *Laubwerk GmbH*¹¹ ist ein Unternehmen aus Berlin, welches, zum Zeitpunkt der Erstellung dieser Arbeit, 16 Pflanzen-Kits anbietet. Zudem ist auf der Homepage ein kostenloses Probepaket verfügbar, damit sich Interessierte einen ersten Einblick in die Funktionsweise der Produkte verschaffen können. Abbildung 2.7 zeigt 10 der 360 im *Plants Kit 1 - Temperate Deciduous Trees* enthaltenen Elemente. Der Download des Probepakets beinhaltet lediglich



Abbildung 2.7: Darstellung von 10 der 360 Pflanzen des *Plants Kit 1* von *Laubwerk*

Quelle: <https://www.laubwerk.com/store/plants-kit-1/>

eine Installationsdatei, welche die benötigten Plugins und den Inhalt des Pflanzen-Kits für das gewünschte DCC-Tool installiert. Im Test wurde die automatische Installation gewählt, welche das *Laubwerk*-Plugin für *Autodesk Maya 2020* sowie *Maxon Cinema4D S24* instal-

¹¹<https://www.laubwerk.com/>

liert hat. Nach der Installation ist der Library-Browser von *Laubwerk* in *Maya* über einen Shelfbutton oder das Menü verfügbar. Abbildung 2.8 zeigt den Browser mit den im Probepaket enthaltenen Inhalten, sowie der Auswahlmöglichkeit eines Zielrenderers. Die Thumbnails

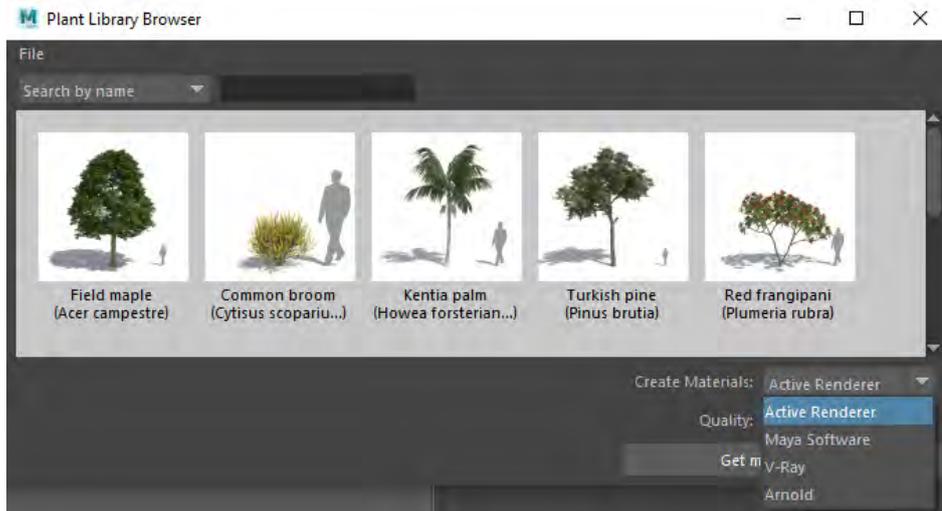


Abbildung 2.8: Darstellung des Plant Library Browser von *Laubwerk* in *Autodesk Maya*
Quelle: Eigene Darstellung

enthalten praktische Menschen-Silhouetten, um einen direkten Eindruck für die Maße des Objekts zu erhalten. Die Pflanzenmodelle können mit dem *Maya Software-Renderer*, *V-Ray* oder *Arnold* verwendet werden. Alternativ kann das Plugin den aktuell verwendeten Renderer ermitteln und somit die Auswahl selbsttätig treffen, insofern der verwendete Renderer unterstützt wird. Zusätzlich kann zwischen den Qualitätsstufen Low, Medium und High gewählt werden. Die Eigenschaften der Modelle in den verschiedenen Qualitätsstufen zeigen sich unter Verwendung von *V-Ray* wie nachfolgend beschrieben. Das Verhalten der Materialien unter Verwendung von *Arnold* kann analog betrachtet werden. Einzelheiten zu den Materialien der verschiedenen Renderer werden in den Kapiteln 3.7 und 3.8 behandelt.

- **Qualitätsstufe Low:**

Sämtliche Materialien bestehen lediglich aus einem einzigen Standard *V-Ray*-Material. Dieses Material erhält keinerlei Input in Form von Texture-Maps. Dies resultiert darin, dass beispielsweise Blätter weder Struktur noch Form besitzen und lediglich eine Farbe und Reflektionen liefern. Diese Qualitätsstufe sollte daher für Objekte verwendet werden, welche so weit im Hintergrund der Szene platziert sind, dass die fehlenden Details nicht auffallen.

- **Qualitätsstufe Medium:**

Auf dieser Qualitätsstufe bestehen die Materialien des Modells ebenfalls aus nur einem Standard *V-Ray* Material. Allerdings enthalten diese nun Texture-Maps, welche für Farbe, Deckkraft und Bump-Map verwendet werden. Somit erhalten beispielsweise

Blätter eine Form, eine Farbe und eine Struktur. Die Materialien der Blätter werden gleichermaßen für Vorder- und Rückseite der Blätter verwendet.

- **Qualitätsstufe High:**

Diese Einstellung resultiert in einem detaillierten Erscheinungsbild der Materialien. Die Standard *V-Ray*-Materialien sind analog der Qualitätsstufe Medium aufgebaut, allerdings existieren für Blätter unterschiedliche Materialien für Vorder- und Rückseite, welche in einem zweiseitigen Material (siehe Kapitel 3.7.3) kombiniert werden. Für die Gewichtung von Vorder- und Rückseite wird ein Graustufenbild des Blattes verwendet. Auf diese Weise wirken die Blätter im Rendering realistischer, da das Licht, welches auf die von der Kamera abgewandten Seite auftrifft, teilweise auf der Kamera zugewandten Seite zu sehen ist (siehe Kapitel 3.5.1).

Nachdem ein Modell ausgewählt und per Doppelklick in die *Maya*-Szene geladen wurde, können weitere Einstellungen getroffen werden, wie in Abbildung 2.9 zu sehen ist. Zunächst

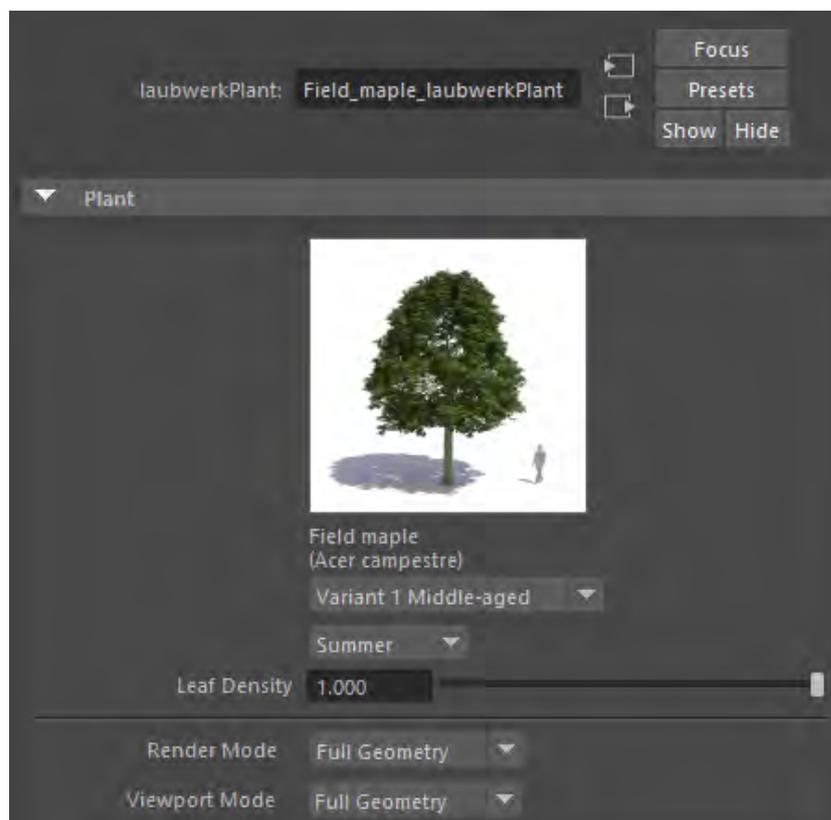


Abbildung 2.9: Übersicht der Einstellmöglichkeiten des Modells *Field Maple* von *Laubwerk* in *Maya*

Quelle: Eigene Darstellung

kann zwischen den verschiedenen Varianten des Modells mit Hilfe eines Dropdown-Menüs ausgewählt werden. Hierbei enthält jede Variante drei verschiedene Variationen bezüglich

des Alters der Pflanze. Es kann zwischen Young (jung), Middle-aged (mittleres Alter) und Fully-grown (voll ausgewachsen) ausgewählt werden. Die Auswahl des Alters und der Variante der Pflanze resultiert daher in einer Änderung des verwendeten Meshes. Eine weitere Einstellmöglichkeit ist die Jahreszeit. Hier kann eine der vier Jahreszeiten gewählt werden, was in einer entsprechenden Auswahl der verwendeten Texturen resultiert. Zudem können die Anzeigemodi der Meshes für den Viewport und das Rendering eingestellt werden, um so ein flüssiges Arbeiten im Viewport zu ermöglichen, da die Meshes aus sehr vielen Polygonen bestehen. Es existieren noch weitere Einstellungen, welche hier allerdings nicht näher betrachtet werden.

2.5.3 Quixel Megascans

*Quixel*¹² bietet mit *Megascans*¹³ eine Bibliothek mit tausenden sehr detaillierten 2D und 3D Assets an, welche kostenlos mit der *Unreal Engine*¹⁴ verwendet werden können. Für die Verwendung der Assets mit einem anderen DCC-Tool muss ein entsprechendes kostenpflichtiges Abonnement abgeschlossen werden. Mit der *Quixel Bridge*¹⁵ können die erhältlichen oder bereits erworbenen Assets schnell und einfach verwaltet, exportiert oder direkt an die gewünschte 3D-Anwendung gesendet werden. Die angebotenen Inhalte eignen sich sehr gut für die Gaming-Industrie und die Architektur-Visualisierung und liegen in der Regel in verschiedenen Qualitätsstufen, Texturauflösungen und LOD's (Level of Detail) vor. Die Bibliothek enthält unter anderem Assets aus der Pflanzenwelt, allerdings beschränkt sich die Auswahl zur Zeit auf Pflanzen, Bäume sind nicht erhältlich. In folgende DCC-Tools können die ausgewählten Assets exportiert werden:

- Unreal Engine
- 3ds Max
- Maya
- Unity
- Blender
- Houdini
- Cinema 4D
- Clarisse
- Marmoset
- Quixel Mixer

¹²<https://quixel.com/>

¹³<https://quixel.com/megascans>

¹⁴<https://www.unrealengine.com/en-US/>

¹⁵<https://quixel.com/bridge>

Im Falle von *Maya* kann in den Exporteinstellungen der *Quixel Bridge Maya* ausgewählt und im Anschluss das erforderliche Plugin installiert werden. Hierbei wird sowohl das benötigte Plugin für die *Quixel Bridge*, als auch das Plugin für *Autodesk Maya* automatisch installiert. Das *Maya*-Plugin wird nach dem nächsten Start von *Maya* automatisch aktiviert und reagiert je nach ausgewähltem Renderer unterschiedlich. Wird im *Maya*-Projekt beispielsweise der *V-Ray* Renderer gewählt, so werden bei der Übertragung der *Quixel*-Assets zu *Maya* automatisch die entsprechenden *V-Ray* Materialien der gewählten Assets übertragen. Der Funktionsumfang kann in der offiziellen Dokumentation des jeweiligen Plugins eingesehen werden¹⁶. Das *Maya*-Plugin ist kompatibel mit *Maya* 2018, 2019 und 2020 und den Renderern *Arnold*, *Redshift* und *V-Ray*, jeweils in der neuesten Version.

Die enthaltenen *V-Ray* Materialien sind einfach aufgebaut. Es werden einige Texturen mitgeliefert, welche mit einem Standard *V-Ray* Material (siehe Kapitel 3.7.1) verwendet werden. Die Pflanzen bestehen aus einem einzigen Mesh, sodass jegliche Änderungen an diesem dazu führen, dass die Materialien nicht mehr korrekt dargestellt werden (UV-mapping). Bei den getesteten Assets werden sämtliche Texturen der Materialien im korrekten Farbraum verwendet und das Pflanzenobjekt erhält zudem alle nötigen *V-Ray* Zusatzattribute, die für eine korrekte Anzeige des Displacements im Rendering von *V-Ray* benötigt werden. Abbildung 2.10 zeigt ein Testrendering einer Basilikumpflanze (Basil), exportiert aus *Quixel Bridge* und gerendert mit *V-Ray für Maya*.



Abbildung 2.10: Testrendering einer Basilikumpflanze (Basil) von *Quixel*. Rendering mit *V-Ray für Maya*

Quelle: Eigene Darstellung

Die Verwendung der *Quixel*-Assets ist zusammengefasst sehr benutzerfreundlich, sehr schnell und kaum fehleranfällig. Allerdings sollten die Meshes nach dem Import nicht mehr geändert werden, um fehlerhaft gemappte Texturen zu vermeiden. Die Auswahl beschränkt sich auf Bodenpflanzen, doch diese besitzen sehr komfortable Einstellungsmöglichkeiten (vor Export) und enthalten zudem immer mehrere Varianten der gleichen Pflanze. Alternativ zu ei-

¹⁶<https://help.quixel.com/hc/en-us>

nem anwendungsspezifischen Export, können die Assets auch unabhängig in einen beliebigen Ordner exportiert werden, wobei eine Vorauswahl an gewünschten Texture-Maps getroffen werden kann. Hierbei handelt es sich allerdings um eine Art Wunschliste. Exportiert werden nur jene Texturen der Auswahl, die auch tatsächlich verfügbar sind.

2.6 Zusammenfassende Betrachtung

Anhand der obigen Betrachtungen werden Probleme deutlich, welche im Rahmen dieser Arbeit genauer untersucht werden. Zum einen werden aktuell große Bemühungen angestellt, 3D-Modelle und Materialien einheitlich und auf hoher Abstraktionsebene zu beschreiben, so dass diese in einer möglichst großen Bandbreite an 3D-Software verwendet werden können. Mit *OSL*(Kapitel 2.2), *MDL*(Kapitel 2.3) und *MaterialX*(Kapitel 2.4) wurden Systeme geschaffen, die diese Idee umsetzen sollen. Während jedes dieser Systeme seine Daseinsberechtigung hat und unterschiedliche Ansätze verfolgt, kann jedoch keines eine universelle Lösung anbieten, sondern lediglich eine Teillösung für darauf ausgerichtete Pipelines und Workflows darstellen. Solange nicht alle relevanten Softwaresysteme eine einheitliche Implementierung der oben genannten Technologien gewährleisten, müssen Anbieter von 3D-Modellen Präferenzen setzen und somit die für sie am relevantesten erscheinenden Programme und Renderer bedienen (Kapitel 2.5).

Dies führt zu dem Problem, dass verschiedene Anbieter von 3D-Modellen ihre Modelle und Materialien für unterschiedliche Softwarelösungen konzipieren und somit die Modelle mehrerer Anbieter nicht ohne Weiteres in einem Projekt Anwendung finden können. Hier bleibt dem Nutzer die Wahl, Modelle anderer Anbieter zu verwenden oder nicht kompatible Modelle aufwändig und fehleranfällig auf manuelle Weise der verwendeten Software anzupassen. Die Idee, diese Konvertierung zu automatisieren, wird in dieser Arbeit untersucht und wurde prototypisch in Form eines *Maya*-Plugins für die Renderer *V-Ray* und *Arnold* umgesetzt. Um die Einzelheiten der Konvertierung verstehen zu können, werden im folgenden Kapitel die notwendigen Grundlagen erläutert.

Kapitel 3

Grundlagen

3.1 Einleitung

In diesem Kapitel werden die Grundlagen erläutert, die zum Verständnis der folgenden Kapitel und der Funktionsweise des entwickelten *Maya*-Plugins notwendig sind. Hierbei werden zunächst die algorithmische Erzeugung von Pflanzenmodellen und anschließend Konzepte erläutert, welche das fotorealistische Rendering betreffen. Im Anschluss wird die verwendete Software, sowie benötigte Plugins genauer beschrieben, welche bei der Entwicklung der automatischen Materialkonvertierung zum Einsatz kamen.

3.2 Der Aufbau computergenerierter Pflanzen

3D-Pflanzenmodelle, welche von Unternehmen wie *Xfrog* erworben werden können, ermöglichen die einfache Integration von Bestandteilen der Pflanzenwelt in eine 3D-Szene. Die Komplexität und Einzigartigkeit von Pflanzen legt nahe, dass es nicht möglich und auch nicht sinnvoll ist, all diese Strukturen manuell am Computer zu erzeugen. Die Simulation des Pflanzenwachstums ist ein Gebiet, welches Wissenschaftler schon seit langer Zeit beschäftigt. Aber nicht allein die Struktur einer Pflanze oder eines Baumes ist für ein fotorealistentes Rendering wichtig, sondern auch die verwendeten Texturen und somit die physikalische Interaktion von Stamm, Blättern, Blüten, usw. mit der in der jeweiligen 3D-Szene verwendeten Beleuchtung.

3.2.1 Algorithmische Erzeugung von Pflanzenmodellen

Die Idee der computergestützten Generierung von Pflanzenstrukturen geht zurück bis in das Jahr 1966. Zu dieser Zeit wurde das erste Verfahren zur Simulation einer Verzweigungsstruktur mit dem Computer vorgestellt [Deu03]. Seitdem näherten sich die Ergebnisse der Simulationen, nicht zuletzt aufgrund der rasanten Entwicklung der Computertechnologie, immer mehr dem Punkt, an welchem sie kaum mehr von echten Fotografien unterscheidbar waren. Somit wurden diese Techniken auch für Architekten, die Spieleindustrie, Landschaftsplaner und viele weitere Bereiche interessant. Da der Fokus dieser Arbeit auf der Konver-

tierung der Materialien von Pflanzenmodellen liegt und die Thematik der algorithmischen Erzeugung von Pflanzenstrukturen sehr komplex und umfangreich ist, wird hier lediglich ein grober Überblick über grundlegende Methoden zur Erzeugung von Pflanzenmodellen gegeben. Oliver Deussen, Mitbegründer von *Xfrog*, unterteilt die Methoden zur Modellierung von Pflanzen in zwei Verfahren:

Prozedurale Modellierung

Dieses Verfahren startete mit sogenannten zellulären Automaten ab 1966, mit welchen erste Verzweigungsstrukturen am Computer hergestellt wurden [Deu03, S. 45ff]. Zelluläre Automaten teilen den Raum in Zellen gleicher Form auf. Jede dieser Zellen besitzt eine feste Anzahl an Nachbarn und einen definierten Zustand, der im Laufe der Simulation geändert werden kann. Mit zuvor festgelegten Wachstumsvorschriften können nun automatische Abläufe gestartet werden. Beispielsweise können von einer Anfangszelle ausgehend alle Nachbarzellen sichtbar geschaltet werden, mit Ausnahme jener Zellen, welche von zwei oder mehr bereits sichtbar geschalteten Zellen berührt werden. Auf diese Weise entstehen interessante Strukturen, wie sie in Abbildung 3.1 zu sehen ist. Im Laufe der Jahre, insbesondere vorangetrieben

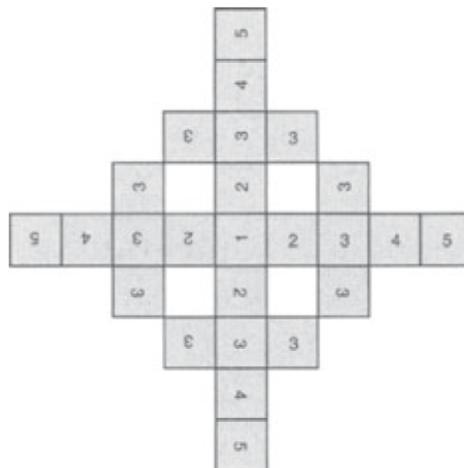


Abbildung 3.1: Zellulärer Automat mit einfachem Muster und Darstellung der Generationen
Quelle: Zellulärer Automat aus [Deu03, S. 46]

durch die stetige Weiterentwicklung der Computertechnologie, wurden viele weitere Verfahren entwickelt und die Systeme fanden immer breiteres Interesse. Die prozeduralen Methoden sind parametrisierbare Algorithmen, mit deren Hilfe bestimmte Pflanzenarten erzeugt werden können, meist sogar nur eine einzige Spezies.

Regelbasierte Modellierung

Aristid Lindenmayer verwendete im Jahre 1968 erstmals Textersetzungssysteme zur Beschreibung zellulärer Interaktion [Lin68]. Diese verwenden Regeln, um Zeichen eines Wortes durch andere Zeichen zu ersetzen und dabei häufig sogar die Länge des Wortes zu vergrößern. Solch ein regelbasiertes System wird genutzt, um einen komplexen Endzustand aus einem Initialzustand heraus zu erreichen [Deu03]. Im Laufe der Jahre wurden viele weitere regelbasierte Verfahren entwickelt. Ein optisch leicht nachvollziehbares Ersetzungssystem ist die Von-Koch-Kurve in Abbildung 3.2. Hierbei wird eine Gerade in drei gleichlange Teilstücke unterteilt und das mittlere Teilstück entfernt. An der Stelle des entfernten Teilstückes werden nun zwei Geraden eingesetzt, die in einem Winkel von 60° zueinander stehen und jeweils die Länge des entfernten Teilstückes besitzen. In jeder Iteration wird das gleiche Verfahren auf jedes Teilstück erneut angewandt.

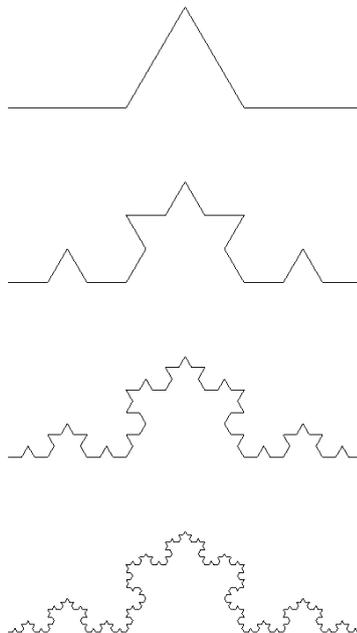


Abbildung 3.2: Ersetzungssystem am Beispiel der Von-Koch-Kurve. Mit jeder Iteration (vertikaler Ablauf) wird das mittlere Drittel eines Teilstückes durch zwei im 60° -Winkel zueinander stehenden Geraden, mit jeweils der Länge eines Teilstückes, ersetzt

Quelle: Eigene Darstellung

Mit den Arbeiten von Aristid Lindenmayer konnten erstmals ansatzweise natürlich aussehende Pflanzen generiert werden. Mit Hilfe der regelbasierten Systeme und vor allem der Lindenmayer-Systeme (L-Systeme) können Pflanzen in sehr kompakter Form formuliert werden. Mit diesen kann eine Vielzahl verschiedener Pflanzen hergestellt werden, während die prozeduralen Systeme für zuvor definierte Strukturen entwickelt werden. Die Arbeitsweise mit L-Systemen ist allerdings wenig intuitiv und es ist sehr aufwändig damit spezifische Pflan-

zen zu erzeugen. Angestrebte Pflanzengeometrien können nur durch Ausprobieren generiert werden, was für eine nutzerfreundliche Anwendung nicht praktikabel ist [DL97].

Kombiniertes Verfahren von Xfrog - Regelbasierte Objekterzeugung

Die Idee der regelbasierten Objekterzeugung ist eine Kombination aus der Intuitivität des prozeduralen Modellierens und der Mächtigkeit regelbasierter Verfahren [Deu03, S. 93ff]. Da das Generieren einer gewünschten Geometrie, auch unter Verfügung einiger Erfahrung, nur durch Ausprobieren erfolgen kann, findet ein kombiniertes Verfahren unter ständigem optischen Feedback Anwendung. *Xfrog* hat diesbezüglich ein Baukastensystem entwickelt, mit welchem gewünschte Pflanzengeometrien zusammengesetzt werden können¹. Der Baukasten besteht aus parametrisierbaren Objekten, wie Blätter und Äste, aber auch aus Modifikatoren wie Multiplikatoren, um beispielsweise Objekte zu vervielfältigen [DL01, S. 64]. Auf diese Weise kann der Nutzer beliebige Pflanzen von Grund auf erstellen, ohne die wenig intuitive Natur der L-Systeme verstehen zu müssen. Das Modellierungssystem ist als Standalone-Version für *Windows*, sowie in Form von Plugins für *Autodesk Maya* und *Maxon Cinema4D* erhältlich. In den Anwendungen können zudem verschiedene Parameter des Baukastensystems per Keyframes zeitlich interpoliert werden, sodass beispielsweise das Wachstum einer Pflanze animiert werden kann.

3.2.2 Erzeugung von Pflanzentexturen per Scan

Die Texturen von 3D-Pflanzenmodellen werden üblicherweise durch das Einbinden von Texture Maps in das jeweilige Material visualisiert. Das Generieren dieser Texture-Maps wird von jedem Anbieter auf eigene Art und Weise realisiert, allerdings sollen an dieser Stelle die grundlegenden Konzepte und Ideen vorgestellt werden.

Eine Möglichkeit ist die automatische Erzeugung der benötigten Maps durch moderne Software anhand selbst erstellter Fotografien. Hierzu ist in der Regel nur ein einziges Foto des nachzubildenden Materials notwendig. Diese Prozedur wird in Abschnitt 3.2.3 genauer erläutert.

Eine weitere Möglichkeit ist ein manueller Prozess, in welchem die einzelnen Maps auf verschiedene Art und Weise erstellt, angepasst und schließlich zusammengeführt werden. Oliver Franzke und Oliver Deussen haben auf einem Symposium in Beijing in 2003 ein System vorgestellt, welches realistische grafische Darstellungen von Pflanzenblättern ermöglicht [FD03]. Für die Erzeugung der Texture-Maps verwenden sie sowohl Fotografien als auch Scans. Die Fotografien dienen als Texture-Maps für die diffusen Reflexionen von Vorder- und Rückseite des Blattes. Aufgrund der Unebenheit der Blätter und damit einhergehend einer minderwertigen Texturqualität durch einen Scan, fotografieren sie die Blätter in einem Wasserbad. Auf diese Weise, und durch zusätzliche Beleuchtung mit einem Flächenlicht, können störende Glanzlichter und Schattierungen umgangen werden. Für alle weiteren Texture-Maps wird ein handelsüblicher Scanner mit einem nachgerüsteten Durchlichtaufsatz verwendet. Diese dienen üblicherweise dem Scannen von lichtdurchlässigen Objekten wie Dias und Negativen. Das Ergebnis des Scans per Durchlichtaufsatz kann als Texture-Map für

¹<http://xfrog.com/category/XF.html>

die Transluzenz (siehe Kapitel 3.5) verwendet werden, da das Bild aus dem transmittierten Licht des Blattes entstanden ist. Anschließend können weitere Texture-Maps aus dem Scan, wie eine Alpha-Map, abgeleitet werden, wie in Abbildung 3.3 dargestellt ist. Hierbei ist zu beachten, dass, je nach verwendetem Renderer und Material, unterschiedliche Arten von Texture-Maps benötigt werden. Eine weitere, wesentlich kostspieligere, Möglichkeit

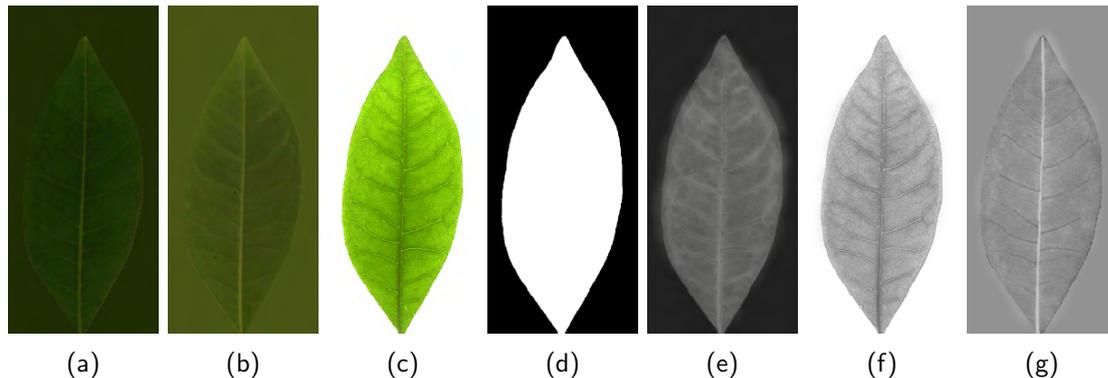


Abbildung 3.3: Aus einem Foto und einem Scan erstellte Texturen zum Rendern von Pflanzenblättern: (a) Adaxiale diffuse Reflexion, (b) Abaxiale diffuse Reflexion, (c) Transluzenz, (d) Alpha Map, (e) Dicke, (f) Extinktion, (g) Bump Map;

Quelle: Texturen zum Rendern von Pflanzenblättern. Aus [FD03]

Materialien und deren Eigenschaften per Scan zu erfassen, ist die Verwendung von professionellen Material-Scannern wie dem *X-Rite Pantone TAC7*². Solche Systeme sind für die Digitalisierung von physikalischen Materialeigenschaften ausgelegt und ermöglichen mittels zusätzlicher Nachbearbeitungssoftware sehr interessante Möglichkeiten zur Visualisierung von realen Materialien. Auch interessant ist die Möglichkeit einer direkten Einbindung der mit solchen Systemen erhaltenen Ergebnisse in *MDL*-Materialien (siehe Kapitel 2.3).

3.2.3 Erzeugung von Pflanzentexturen anhand eines Fotos

Wie im vorigen Abschnitt angedeutet, können mit Hilfe moderner Software und wenigen Mausklicks Texture-Maps aus Fotografien hergestellt werden. Software zum Erstellen und Editieren von Texture-Maps ist sowohl als Open-Source-Software als auch in Form kostenpflichtiger Programme und Plugins für DCC-Tools erhältlich. Generell ist bezüglich des Realismus der Materialien die softwaregestützte Generierung von Texture-Maps gegenüber Scan-Techniken im Nachteil, da sämtliche gewonnenen Informationen über Struktur und Eigenschaften des Materials Ergebnisse von Algorithmen sind, welche lediglich eine bestmögliche Approximation an die realen Eigenschaften des Materials sein können. Zudem ist die Qualität der generierten Maps stark von der Qualität der Fotografie und somit auch vom verwendeten Equipment, wie Kamera und Beleuchtung, aber auch von den Einstellungen der Kamera, des Lichts, usw. abhängig. Manche Software, wie ShaderMap, unterstützt die Verwendung von mehreren Aufnahmen mit Beleuchtung aus unterschiedlichen Winkeln, um

²<https://www.xrite.com/categories/appearance/total-appearance-capture-ecosystem/tac7>

3. GRUNDLAGEN

auf diese Weise die Genauigkeit der entstehenden Texture-Maps zu verbessern. Das bessere Ergebnis bedingt allerdings erhöhten Aufwand bzw. Kosten. Im Folgenden wird eine kleine Auswahl an aktuell verfügbarer Software zur Erstellung von Texture-Maps aus Fotografien vorgestellt.

Materialize (Bounding Box Software)

Materialize³ ist ein Standalone-Tool zur Erstellung und Editierung von Texture-Maps. Das Tool ist Open-Source-Software und kostenlos über Github erhältlich⁴. Abbildung 3.4 zeigt beispielhaft die Generierung von Texture-Maps eines Blattes. Nach dem Laden eines Fotos

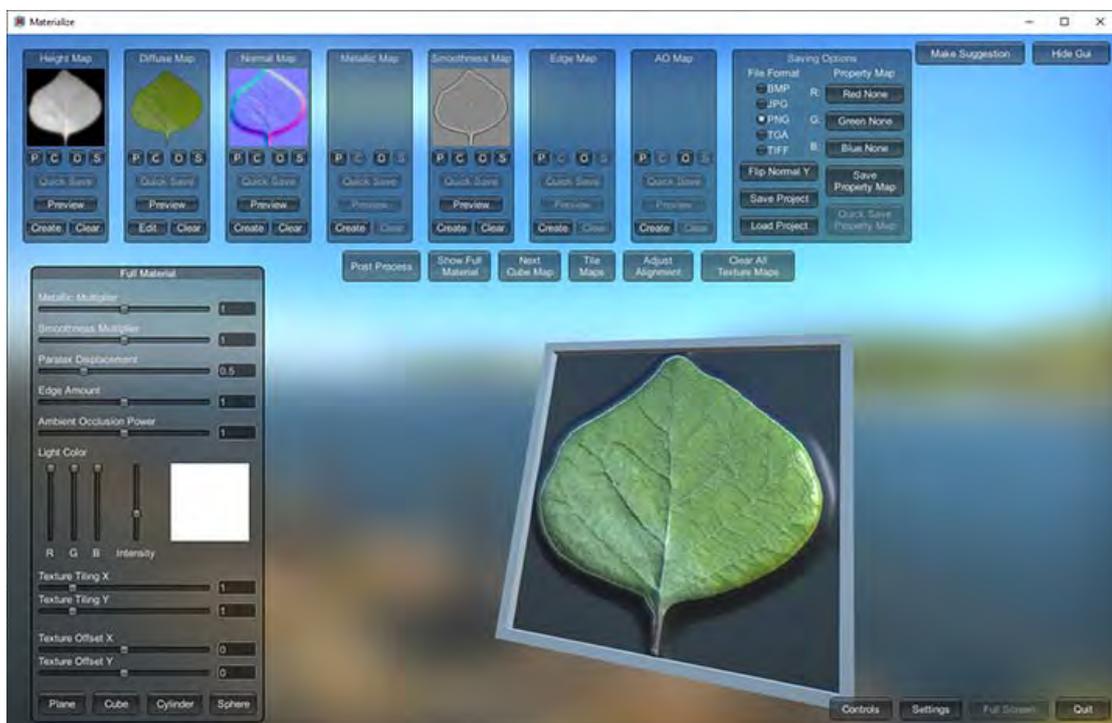


Abbildung 3.4: Screenshot des Tools *Materialize* von Bounding Box Software zur Erstellung von Texture-Maps am Beispiel eines Blatt-Materials

Quelle: Eigene Darstellung

oder eines Bildes in den Diffuse-Kanal können die verschiedenen benötigten Maps erstellt und angepasst werden. Das aus den Maps zusammengeführte Material kann parallel in einem OpenGL-Viewer betrachtet werden. Auf diese Weise können sämtliche verfügbaren Parameter interaktiv angepasst und schließlich die generierten Maps exportiert werden.

³<http://boundingboxsoftware.com/materialize/index.php>

⁴<https://github.com/BoundingBoxSoftware/Materialize>

ShaderMap

*ShaderMap*⁵ ist ebenfalls ein Standalone-Programm und sowohl in einer kostenlosen als auch in einer kostenpflichtigen Pro-Version verfügbar. Nach dem Laden eines Fotos oder eines Bildes werden unter den Standardeinstellungen automatisch die Texture-Maps Normal, Displacement, Specular und Ambient Occlusion erstellt. Auch dieses Tool ermöglicht eine Vielzahl an Anpassungsmöglichkeiten und bietet eine Live-Ansicht des aus den Texture-Maps resultierenden Materials. Abbildung 3.5 zeigt diesen Prozess für ein Blatt-Material. *ShaderMap* wird im Rahmen dieser Arbeit genutzt, um die nötigen Texture-Maps der *Xfrog*-Modelle zu erstellen und in das entsprechende *V-Ray* oder *Arnold* Material einzubinden.

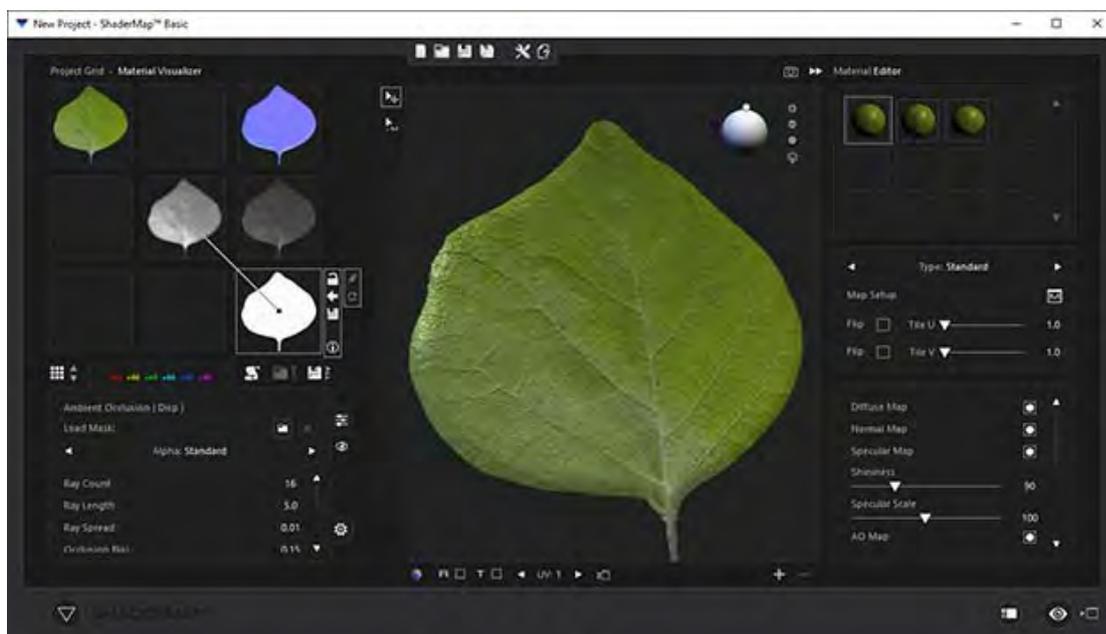


Abbildung 3.5: Screenshot des Tools *ShaderMap* zur Erstellung von Texture-Maps am Beispiel eines Blatt-Materials

Quelle: Eigene Darstellung

Substance 3D Sampler (ehemals Substance Alchemist)

*Substance 3D Sampler*⁶ von Adobe, ehemals *Substance Alchemist*, ist ein Standalone-Tool, welches üblicherweise parallel zu *Substance 3D Painter*⁷ und/oder *Substance 3D Designer*⁸ Verwendung findet. Das Tool ermöglicht das Anpassen und Zusammenführen existierender Materialien und bietet diesbezüglich eine Vielzahl an Filtern an, welche beispielsweise das Material in Form von Mustern darstellen oder anhand verschiedener Parameter mit anderen

⁵<https://shadermap.com/home/>

⁶<https://www.adobe.com/de/products/substance3d-sampler.html>

⁷<https://www.adobe.com/de/products/substance3d-painter.html>

⁸<https://www.adobe.com/de/products/substance3d-designer.html>

3. GRUNDLAGEN

Materialien mischen. Der Filter „Image to Material“ generiert, ähnlich wie oben genannte Tools, Texture-Maps aus einem Foto oder einem Bild. In den neueren Versionen von *Substance 3D Sampler* existiert dieser Filter in einer weiteren Form, dem KI-gestützten „Image to Material (AI Powered)“-Filter⁹. Dieser nutzt Machine Learning um Formen und Objekte zu erkennen und anhand dieser Informationen akkurate Height- und Normal-Maps zu generieren. Zusätzlich entfernt der Algorithmus selbstständig Schatten und Glanzlichter des Bildmaterials, was die Qualität der Ergebnisse weiterhin verbessert und bei Nutzung mit den meisten anderen Tools im Vorfeld manuell durchgeführt werden muss. Zum Zeitpunkt der Erstellung dieser Arbeit wurde das neurale Netzwerk ausschließlich mit Daten von Materialien im Außenbereich trainiert und es wird darauf hingewiesen, dass andere Materialtypen unerwartete Ergebnisse produzieren können. Abbildung 3.6 zeigt den Prozess der automatischen Texture-Map Generierung mit *Substance 3D Sampler* am Beispiel eines Blatt-Materials. *Substance 3D Sampler* wird im Rahmen dieser Arbeit genutzt, um die nötigen Texture-Maps der *Xfrog*-Modelle zu erstellen und in das entsprechende *V-Ray* oder *Arnold* Material einzubinden.

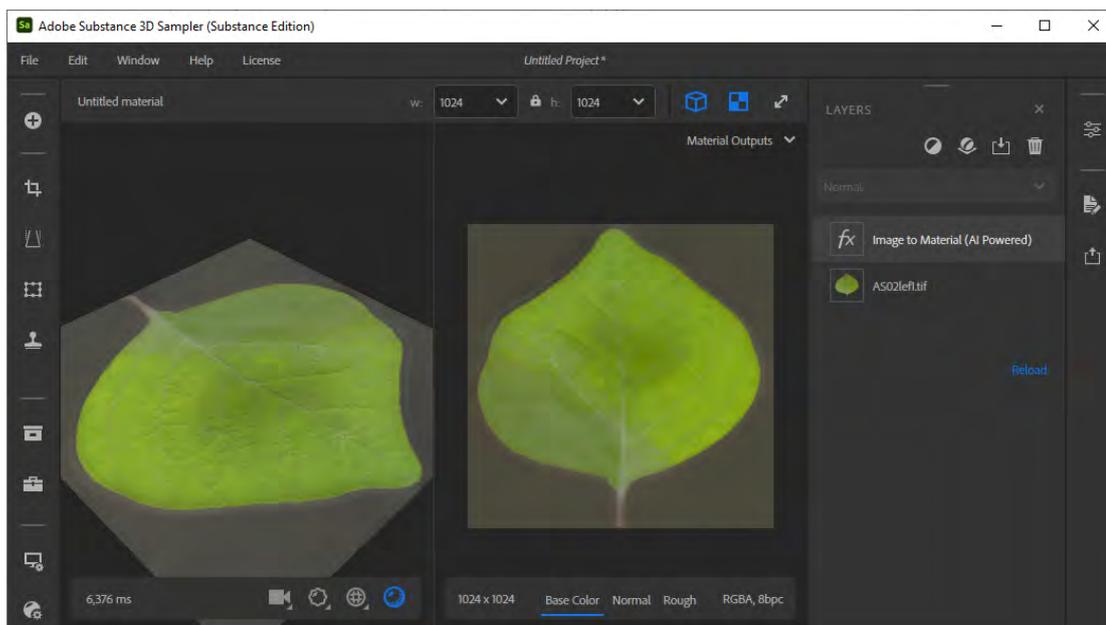


Abbildung 3.6: Screenshot des Tools *Substance 3D Sampler* zur Erstellung von Texture-Maps am Beispiel eines Blatt-Materials

Quelle: Eigene Darstellung

3.3 V-Ray in Maya

V-Ray ist ein Offline Raytracing-Renderer, welcher in vielen Bereichen der Computergrafik eingesetzt wird und für Filme und Serien wie *Doctor Strange* und *Game of Thrones* Ver-

⁹<https://substance3d.adobe.com/documentation/sadoc/image-to-material-215286041.html>

wendung fand. Der Renderer stammt vom Unternehmen *Chaosgroup*¹⁰ aus Bulgarien, ist in Form von Plugins für viele DCC's, darunter auch *Autodesk Maya*, erhältlich und unterstützt sowohl Multicore CPU-Rendering als auch GPU-Rendering. Das Rendern per Grafikkarte unterliegt allerdings einigen Einschränkungen und sollte daher entsprechend bei der Planung eines Projekts berücksichtigt werden¹¹. Das Plugin enthält viele nützliche Features wie Interactive Production Rendering (IPR), Distributed Rendering (Paralleles Rendern auf mehreren Computern) und Batch Rendering (Rendern über die Kommandozeile). Materialien können wie gewohnt mit Hilfe von *Hypershade* erstellt und kombiniert werden. Zudem können *Maya*-Objekte mit speziellen *V-Ray* Attributen versehen werden, welche die Grundfunktionalität der Objekte um *V-Ray* spezifische Funktionen erweitert. So können beispielsweise Subdivision, Displacement oder abgerundete Kanten zur Renderzeit verwendet werden. *Maya*-Kameras können mit dem *V-Ray* Attribut *Physical cam* versehen werden, um physikalisch korrekte Werte und Einheiten für Blende, Verschlusszeit und ISO verwenden zu können. Im Rahmen dieser Arbeit wird die Version *V-Ray 5 for Maya, hotfix 2* verwendet.

3.4 Arnold in Maya

Arnold for Maya (MtoA)¹² ist ein Plugin für *Autodesk Maya* zur Integration des *Arnold*-Renderers und ist standardmäßig Bestandteil der *Maya*-Installation. *Arnold* wurde ursprünglich von *Autodesk* und *Sony Pictures Imageworks* entwickelt und wird weltweit in über 300 Studios verwendet¹³. Es handelt sich um einen Monte Carlo Ray-Tracing Renderer und er wurde in Filmen wie *Pacific Rim* und *Gravity* verwendet. *Arnold for Maya* beinhaltet Interactive Production Rendering (IPR), unterstützt *Maya*-Systeme wie *Maya Hair* und Volume Rendering mit *Maya Fluids*, und ist durch Plugins erweiterbar. Materialien können mit Hilfe von *Hypershade* erstellt und kombiniert werden und es kann per CPU oder GPU gerendert werden. Bei Verwendung des GPU-Renderers sind einige Einschränkungen zu berücksichtigen¹⁴.

3.5 Subsurface Scattering und Translucency

Subsurface Scattering (SSS), zu Deutsch Volumenstreuung, beschreibt die Streuung des Lichts in Materie. Viele Materialien, die in der Computergrafik zum Einsatz kommen, reflektieren das eintreffende Licht lediglich an der Oberfläche des Objekts. Manche Substanzen können mit diesen Materialien allerdings nicht realistisch dargestellt werden. Prominente Beispiele sind Haut, Wachs, Milch oder Jade. In solche Substanzen kann das Licht eindringen und, nach einer gewissen Anzahl an Reflexionen innerhalb des Materials, an anderer Stelle wieder austreten. In der Computergrafik wurden für solche Zwecke Algorithmen entwickelt, die diesen Prozess approximieren können (siehe BSSRDF, Kapitel 3.6).

¹⁰<https://www.chaosgroup.com/>

¹¹<https://docs.chaosgroup.com/display/VMAYA/Supported+Features>

¹²<https://www.arnoldrenderer.com/arnold/arnold-for-maya/>

¹³<https://www.arnoldrenderer.com/arnold/>

¹⁴<https://docs.arnoldrenderer.com/display/A5ARP/Supported+Features+and+Known+Limitations>

Die Verwendung von Subsurface Scattering ist allerdings nur sinnvoll, wenn das zu rendernde Objekt ein Volumen, beziehungsweise eine geschlossene Hülle besitzt. Im Rahmen dieser Arbeit und den für das Konvertierungs-Plugin verwendeten 3D-Pflanzenmodellen sind Pflanzen- und Blütenblätter die einzigen Elemente, die Licht nicht ausschließlich an der Oberfläche reflektieren und somit ein Einsatz von SSS sinnvoll erscheint. Allerdings werden in den verwendeten Modellen ausschließlich Pflanzen- und Blütenblätter ohne Hülle, also Geometrien ohne Volumen, verwendet. Dies schließt die Verwendung von SSS aus und führt dazu, dass das Durchdringen von Licht auf andere Weise simuliert werden muss.

Da der Schwerpunkt dieser Arbeit auf der Konvertierung der verfügbaren *Xfrog*-Materialien in Materialien für *V-Ray* und *Arnold* liegt, werden im Folgenden kurz die Funktionsweisen von SSS und Translucency der beiden Renderer dargestellt. Eine detaillierte Analyse der Shader-Netzwerke, einschließlich der entsprechenden Parameter, erfolgt in Kapitel 4 und 5.

3.5.1 SSS und Translucency in Chaosgroup V-Ray

V-Ray for Maya bietet für das Rendering von SSS mehrere Möglichkeiten an. Das Material *VRayAISurface* dient hauptsächlich der Visualisierung von Haut und basiert auf dem *alShader* von Anders Langland¹⁵. Abbildung 3.7 zeigt ein *V-Ray* Rendering unter Verwendung des *VRayAISurface*-Materials. Des weiteren beinhaltet das *V-Ray* Standard-Material *VRayMtl*



Abbildung 3.7: *V-Ray* Rendering eines *VRayAISurface*-Materials
Quelle: Eigene Darstellung

die Möglichkeit, SSS optional zu aktivieren. Dies funktioniert allerdings nur, wenn Refraction aktiviert ist. Zudem werden sämtliche SSS-Optionen deaktiviert, sobald in den Render-Einstellungen das Rendern per GPU aktiviert wird. Abbildung 3.8 zeigt ein *V-Ray* Rendering

¹⁵<https://www.chaosgroup.com/blog/v-rays-implementation-of-the-anders-langlands-alsurface-shader>

mit einem Standard *VRayMtl* mit aktivierter SSS-Option. Zusätzlich ist die Verwendung



Abbildung 3.8: V-Ray Rendering eines *VRayMtl*-Materials mit aktivierter SSS-Option

Quelle: Eigene Darstellung

des Materials *VRayFastSSS2* möglich, welches eine physikalisch weniger genaue Variante des SSS darstellt, im Gegenzug allerdings schneller zu rendern ist. Abbildung 3.9 zeigt ein V-Ray Rendering eines *VRayFastSSS2*-Materials. Allerdings handelt es sich, wie oben be-



Abbildung 3.9: V-Ray Rendering eines *VRayFastSSS2*-Materials

Quelle: Eigene Darstellung

reits angedeutet, bei den in dieser Arbeit relevanten Geometrien um Objekte ohne Volumen

und die Verwendung von SSS ist somit nicht zielführend. Um dünne und lichtdurchlässige Materialien zu simulieren, bietet *V-Ray* das Material *VRayMtl2Sided* zur Verwendung an. Dieses Material erlaubt es, das Licht hinter dünnen Objekten sehen zu können. Auf diese Weise können Oberflächen wie Papier, Stoffe, Pflanzenblätter, etc. simuliert werden (siehe Kapitel 3.7.3). Abbildung 3.10 zeigt zur Verdeutlichung ein *V-Ray* Rendering eines Blattes, hinter welchem sich eine kugelförmige Lichtquelle befindet. Es ist zu erkennen, wie die Textur der Rückseite des Blattes, in Abhängigkeit des Lichtes, auf der Vorderseite des Blattes zu sehen ist.



Abbildung 3.10: *V-Ray* Rendering eines Blattes mit einem *VRayMtl2Sided*-Material. Die Textur der Rückseite des Blattes ist auf der Vorderseite des Blattes, in Abhängigkeit der kugelförmigen Lichtquelle hinter dem Blatt, sichtbar

Quelle: Eigene Darstellung

3.5.2 SSS und Translucency in Autodesk Arnold

Für die Verwendung von SSS in *Arnold for Maya* kommt das Material *aiStandardSurface* (siehe Kapitel 3.8.1) zum Einsatz. Der Parameter *Weight* (Gewichtung) für SSS in den Materialeinstellungen beträgt standardmäßig 0 und SSS ist somit nicht aktiv. Werte größer 0 aktivieren SSS und somit auch die zusätzlichen Einstellungen. Abbildung 3.11 zeigt ein Rendering eines Gesichts unter Verwendung von SSS des *aiStandardSurface*-Shaders. Es ist deutlich zu erkennen, dass das Licht in die Haut eindringt und dort gestreut wird.

Unter dem Reiter *Geometry* (Geometrie) der Materialeinstellungen kann zudem die Option *Thin Walled* (dünnwandig) aktiviert werden. Diese Option kann für dünne und lichtdurchlässige Objekte verwendet werden, ähnlich wie das *VRayMtl2Sided*-Material von *V-Ray* im vorigen Abschnitt. Durch das Anwählen dieser Option wird der Term zur Berechnung von SSS im Shader eliminiert und es ergibt sich eine diffuse Transmission [GPA⁺]. Dies repräsentiert eine unendlich dünne Hülle wie beispielsweise ein idealisiertes Blatt Papier. Dieser Workflow kommt ihm Rahmen dieser Arbeit für transluzente Objekte, wie Blätter, zum Ein-

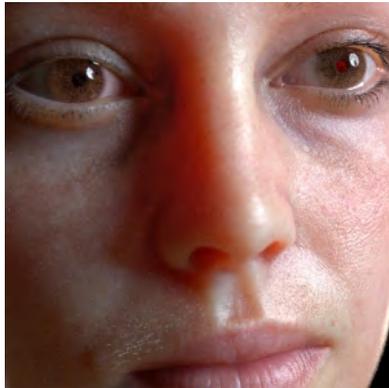


Abbildung 3.11: Beispiel von SSS mit dem *aiStandardSurface*-Shader. An der Nase ist deutlich zu erkennen, dass das Licht in die Haut eindringt und dort gestreut wird

Quelle: Aus [GPA⁺]

satz. Abbildung 3.12 zeigt den Unterschied zwischen einem opaken und einem transluzenten Material unter Verwendung des *Thin Walled*-Attributs.

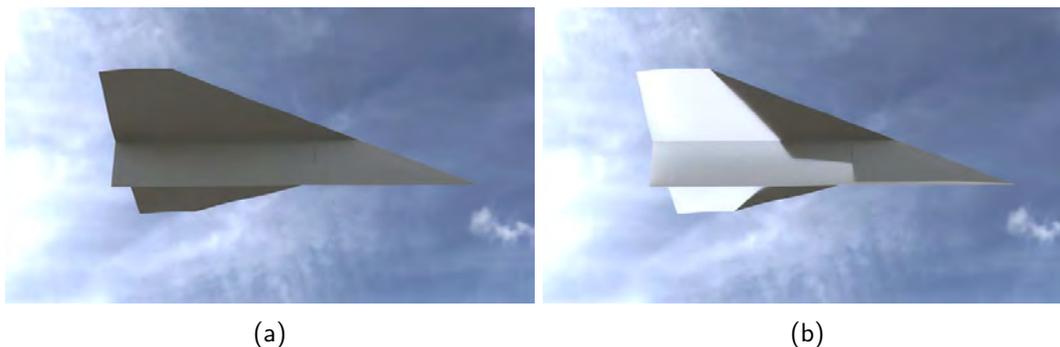


Abbildung 3.12: Vergleich zwischen einem opaken (a) und einem transluzenten (b) Material in Arnold unter Verwendung des *Thin Walled*-Attributs

Quelle: Aus [GPA⁺]

3.6 Bidirektionale Reflexionsverteilungsfunktion(BRDF)

Eine bidirektionale Reflexionsverteilungsfunktion dient der Berechnung des Reflexionsverhaltens eines Materials unter beliebigen Einfallswinkeln. Sie beschreibt das Verhältnis aus der differentiellen Leuchtdichte dL_r in Betrachtungsrichtung und der differentiellen Bestrahlungsstärke dE_i , die aus der Beleuchtungsrichtung auf die Oberfläche wirkt, für eine gegebene Wellenlänge [Geb03]:

$$BRDF(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{dL_r(\theta_r, \phi_r)}{dE_i(\theta_i, \phi_i)} \quad (3.1)$$

In Gleichung 3.1 ist θ_i der Winkel zwischen der Projektion des Vektors des einfallenden Lichtstrahls auf die xy -Ebene und der positiven x -Achse. ϕ_i ist der Winkel zwischen dem Vektor des einfallenden Lichtstrahls und der z -Achse (Normalenvektor). θ_r und ϕ_r sind analog, allerdings in Bezug auf den Vektor der Betrachtungsrichtung, definiert [Pap16, S. 124]. Die geometrischen Zusammenhänge des Kugelkoordinatensystems sind in Abbildung 3.13 verdeutlicht. In der Abbildung stellt L den Richtungsvektor des einfallendes Lichtes und V den Vektor der Betrachtungsrichtung dar.

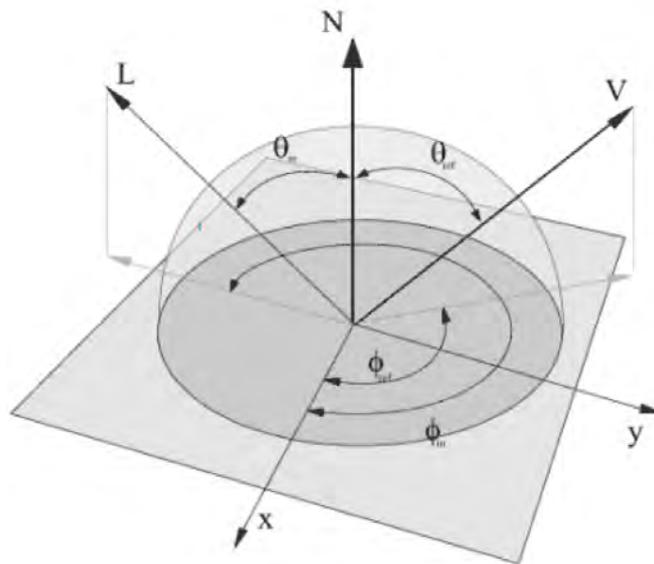


Abbildung 3.13: Darstellung der BRDF-Variablen in einem Kugelkoordinatensystem
Quelle: Aus [Geb03]

Die Leuchtdichte ist eine raumwinkelabhängige Größe und ist somit abhängig von der Richtung der Beleuchtung sowie der Richtung des Betrachters. Die Bestrahlungsstärke hingegen zählt zu den raumwinkelunabhängigen Größen und nimmt gemäß dem Abstandsgesetz proportional mit dem Quadrat des Abstands ab. Berücksichtigt man nun den Einfallswinkel und somit eine verminderte Lichtintensität der beleuchteten Fläche im Gegensatz zu einem senkrechten Lichteinfall, kann Gleichung 3.1 wie folgt formuliert werden:

$$BRDF(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{dL_r(\theta_r, \phi_r)}{L_i(\theta_i, \phi_i) \cdot \cos\theta_i \cdot d\omega_i} \quad (3.2)$$

Um physikalisch korrekte Ergebnisse zu liefern, muss eine BRDF-Funktion zwei Kriterien erfüllen:

- Helmholtz-Reziprozität:
 Dieses Kriterium definiert die Bedingung, dass das Ergebnis der Berechnung nicht von dem Ergebnis nach einem Austausch von Einfalls- und Ausfallwinkel abweichen darf.

- **Energieerhaltung:**
Die Summe der reflektierten Energie darf nicht größer sein als die Summe der einstrahlenden Energie.

BRDF ist eine vereinfachte Variante von BSSRDF (Bidirectional scattering-surface reflectance distribution function) und setzt implizit voraus, dass das Licht an dem selben Punkt auf der Oberfläche ein- und austritt. BSSRDF hingegen berücksichtigt zusätzlich die Streuung in der Materie und ermöglicht somit beispielsweise Berechnungen, die auch als Subsurface Scattering (SSS, s. Kapitel 3.5) bekannt sind [JMLH01]. Der Unterschied der beiden Funktionen ist in Abbildung 3.14 schematisch dargestellt. Im Laufe der Jahre wurden viele

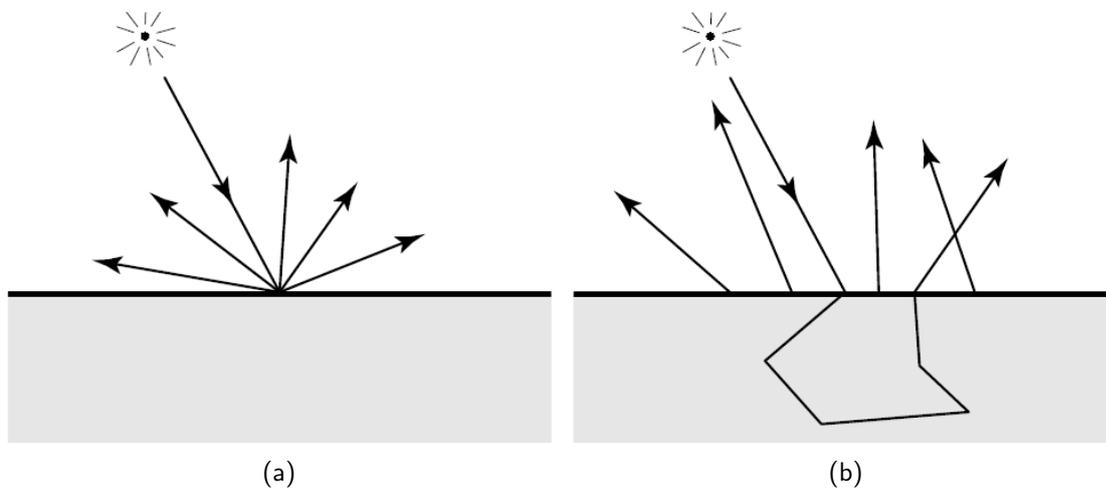


Abbildung 3.14: Schematische Darstellung der Lichtstreuungen von (a) BRDF und (b) BSSRDF

Quelle: Aus [JMLH01]

BRDF-Modelle entwickelt, welche für unterschiedliche Materialtypen verwendet werden. Im Rahmen dieser Arbeit und in der Umsetzung des Konvertierungs-Plugins wird konsequent das BRDF-Modell GGX verwendet, welches auch in der *V-Ray* Dokumentation zur Verwendung empfohlen wird¹⁶. Da Pflanzenblätter in der Regel in Form von texturierten Polygonen ohne Hülle dargestellt werden, ist eine Berechnung der Lichtstreuung im Blatt per SSS nicht sinnvoll und es muss somit mit anderen Techniken gearbeitet werden. Ein positiver Nebeneffekt ist die eingesparte Rechenleistung, da SSS sehr aufwändig zu berechnen ist.

3.7 Shader-Netzwerke Chaosgroup V-Ray

In diesem Kapitel werden die *V-Ray* Nodes in *Maya* vorgestellt, welche von *Xfrog* für die Gestaltung der ausgelieferten Materialien verwendet werden. Die konzeptuelle Verwendung

¹⁶<https://docs.chaosgroup.com/display/VMAYA/VRayMtl+Reflection>

der Nodes und somit die hierarchische Struktur der Shader-Netzwerke wird in Kapitel 4 detailliert untersucht. Aufgrund der Vielzahl an verfügbaren Nodes wird an dieser Stelle auf eine Betrachtung jener Nodes verzichtet, die in dieser Arbeit nicht zur Verwendung kommen.

3.7.1 VRayMtl

Der *VRayMtl*-Node¹⁷ stellt das Standard-Material von *V-Ray* in *Maya* dar. Er ist für eine Vielzahl an Anwendungsszenarien ausgelegt und enthält einige Presets, welche eine solide Basis für universelle Einsätze bieten. Das Material und die Presets verwenden Einstellungen, die im Hinblick auf Rendergeschwindigkeit und Realismus erfahrungsgemäß gut funktionieren. Für spezifischere Darstellungen, wie etwa durchsichtige Objekte im Fokus der Darstellung, müssen die Werte allerdings gegebenenfalls angepasst und optimiert werden. Das Material unterstützt die Verwendung von Texture-Maps und die Darstellung von Oberflächenstrukturen via Bump-Maps, Normal Maps oder Displacement-Maps. Zudem kann eine Reflexionsverteilungsfunktion gewählt werden, um die Interaktion des Materials mit der in der Szene vorhandenen Beleuchtung anzupassen.

3.7.2 VRayBlendMtl

Das *VRayBlendMtl*¹⁸ dient der Schichtung von kompatiblen Materialien, um komplexe, mit einem Standard-Material nicht erzeugbare, Darstellungen zu ermöglichen. Auf diese Weise kann beispielsweise eine Glanzschicht über ein Metall geschichtet werden, um Autolack zu simulieren. Das *VRayBlendMtl* nutzt ein Basismaterial als Grundschicht und legt eine beliebige Anzahl an Schichten darüber. Die überlagernden Schichten können mit numerischen Werten oder mit zugeführten Texturen auf die jeweils darunterliegende Schicht überblendet werden.

3.7.3 VRayMtl2Sided

Das *VRayMtl2Sided*¹⁹ ist ein sogenanntes *Utility*-Material und erlaubt es, das Licht auf der Rückseite eines Objektes sehen zu können. Auf diese Weise können dünne Oberflächen wie Papier, Kleidung oder Pflanzenblätter simuliert werden. Der Node benötigt jeweils ein Material für die Vorder- und die Rückseite des Objekts. Mit numerischen Werten oder unter Verwendung einer Textur kann zudem bestimmt werden, wie groß der sichtbare Anteil von jeweils Vorder- und Rückseite relativ zur Kameraposition ist. Das Material ist dafür ausgelegt, mit Objekten ohne Volumen verwendet zu werden.

3.7.4 VRayBumpMtl

Das *VRayBumpMtl*²⁰ ermöglicht die Verwendung von Bump- oder Normal-Maps mit einem beliebigen kompatiblen Material. Auf diese Weise können komplexe Shader-Netzwerke ab-

¹⁷<https://docs.chaosgroup.com/display/VMAYA/VRayMtl>

¹⁸<https://docs.chaosgroup.com/display/VMAYA/VRayBlendMtl>

¹⁹<https://docs.chaosgroup.com/display/VMAYA/VRayMtl2Sided>

²⁰<https://docs.chaosgroup.com/display/VMAYA/VRayBumpMtl>

schließlich mit einer Oberflächenstruktur versehen werden. Das *VRayBumpMtl* verwendet ein Basismaterial und überlagert dies mit der Oberflächenstruktur. Zudem können mehrere dieser Nodes iterativ verwendet werden, um die Oberfläche mit komplexen Strukturen aus mehreren Texture-Maps zu optimieren.

3.8 Shader-Netzwerke Autodesk Arnold

Dieses Kapitel dient der Übersicht über die Shader, die von *Xfrog* für die Materialien der Pflanzenmodelle verwendet werden, welche für den Renderer *Arnold für Maya* zur Verfügung gestellt werden. Hierbei ist zu beachten, dass die vorliegenden Modelle mit *Arnold* Materialien einer inoffiziellen Testversion entstammen. Zum Zeitpunkt der Erstellung dieser Arbeit werden keine Modelle mit *Arnold* Materialien von *Xfrog* zum Kauf angeboten. Im Gegensatz zu den Materialien für den Renderer *V-Ray* aus dem vorigen Abschnitt, kommt für die Materialien für *Arnold* lediglich ein einziger Shader zum Einsatz, der *aiStandardSurface*-Shader. Eine detaillierte Analyse der *Arnold*-Shadernetzwerke für verschiedene Pflanzenbestandteile wird in Kapitel 5 durchgeführt.

3.8.1 aiStandardSurface

Der *aiStandardSurface*-Shader²¹ ist ein universeller Shader, welcher viele Komponenten vereint und somit für eine Vielzahl an Materialien einsetzbar ist. Er berücksichtigt die Energieerhaltung und es kann somit ausgeschlossen werden, dass mehr Licht das Material verlässt, als eingestrahlt wurde. Wird beispielsweise der *metallic*-Parameter erhöht, nimmt der Beitrag des Glanzlicht-Anteils zu und gleichzeitig automatisch der Beitrag des diffusen Anteils ab. Der Shader vereint zehn Komponenten (Abbildung 3.15), welche hierarchisch geschichtet und gemischt werden. Hierbei werden horizontal angeordnete Komponenten statistisch überblendet und vertikal angeordnete Komponenten geschichtet (*layering*). Die mit einem Stern markierten Komponenten sind optional.

3.9 Maya Embedded Language (MEL)

Die Benutzeroberfläche von *Autodesk Maya* wird größtenteils mit Hilfe der Skriptsprache *MEL*²² erstellt und alle Aktionen, welche in der Benutzeroberfläche ausführbar sind, können alternativ durch die Verwendung von *MEL*-Befehlen ausgeführt werden [Tea03, S. 13ff]. Neben dem Ausführen von Aktionen, können mit Hilfe von *MEL* Skripte zur Automatisierung von individuellen Workflows erstellt oder fortgeschrittene Aktionen ausgeführt werden. *MEL* stammt von der UNIX-Skriptsprache ab und basiert daher eher auf der Ausführung von Befehlen als auf der Manipulation von Datenstrukturen oder der Verwendung von objekt-orientierten Methoden. Abbildung 3.16 zeigt beispielhaft die Erzeugung und gleichzeitige

²¹<https://docs.arnoldrenderer.com/display/A5AFMUG/Standard+Surface>

²²<https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Maya-Scripting/files/GUID-60178D44-9990-45B4-8B43-9429D54DF70E-htm.html>

3. GRUNDLAGEN

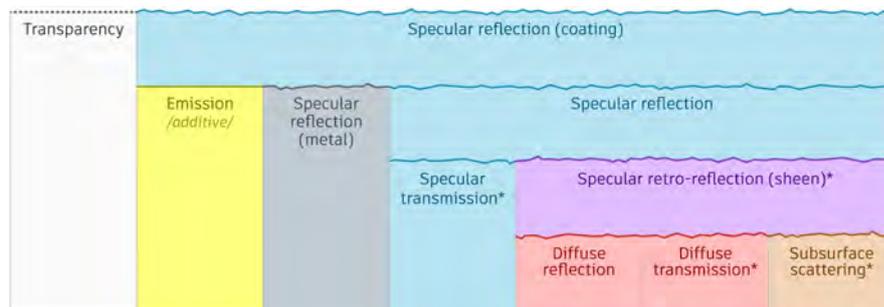


Abbildung 3.15: Schematische Darstellung der Komponenten des *aiStandardSurface*-Shaders. Horizontal angeordnete Komponenten werden statistisch überblendet und vertikal angeordnete Komponenten werden geschichtet (layering). Die mit einem Stern markierten Komponenten sind optional

Quelle: Aus <https://docs.arnoldrenderer.com/display/A5AFMUG/Standard+Surface> - Zuletzt geprüft: 08.07.2021

Verschiebung von Polygon-Objekten in Maya mit *MEL*. In diesem Beispiel wurde der Maya-interne Skripteditor verwendet. Es existieren allerdings auch Code-Erweiterungen für externe Editoren wie beispielsweise *Microsoft Visual Studio Code*. Hier kann die Erweiterung namens *MayaCode* über den Marktplatz heruntergeladen und aktiviert werden. Die Erweiterung enthält praktische Features wie Syntax Highlighting und Auto-Completion und kann, insofern der entsprechende Port für den Empfang von *Maya*-Befehlen in *Maya* geöffnet wurde, den geschriebenen Code direkt zur Ausführung an *Maya* senden.

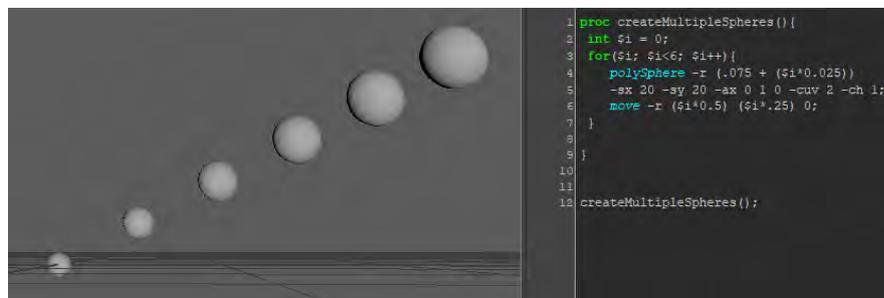


Abbildung 3.16: Erzeugung von Polygon-Objekten in Maya mit der Skriptsprache *MEL* im *Maya*-Skripteditor

Quelle: Eigene Darstellung

Das entwickelte Plugin zur Konvertierung der *Xfrog*-Materialien basiert auf mehreren *MEL*-Skripten, welche über den internen *Maya* Skript-Editor aufgerufen werden können. Zusätzlich zur Funktionalität, wurde, ebenfalls in Form eines *MEL*-Skripts, eine Benutzeroberfläche zur erleichterten Bedienung des Plugins entwickelt (siehe Kapitel 6).

Kapitel 4

Analyse der V-Ray Shader-Netzwerke und deren Parameter

In diesem Kapitel werden die von *Xfrog* für den Renderer *V-Ray* ausgelieferten Materialien hinsichtlich der grundlegenden Struktur untersucht. Alle zum Zeitpunkt der Erstellung dieser Arbeit vorliegenden *Xfrog*-Modelle und deren Materialien besitzen die gleiche Struktur, daher können die Ergebnisse der folgenden Untersuchung voraussichtlich auf alle verfügbaren *Xfrog*-Modelle bezogen werden. *Xfrog* hat die Shader-Netzwerke in Kooperation mit *Chaosgroup*, dem Hersteller des Renderers *V-Ray*, entwickelt, um eine möglichst optimale Darstellung ihrer Modelle mit dem Renderer zu ermöglichen¹.

Da das entwickelte Konvertierungs-Plugin zu diesem Zeitpunkt nur die Konvertierung der Materialien von Blättern, Ästen, und Stämmen unterstützt, werden im Folgenden entsprechend nur solche Materialien untersucht.

4.1 Das V-Ray Blatt Shader-Netzwerk

Das *V-Ray* Leaf Material (Blatt-Material) stellt das komplexeste Material aller untersuchten *V-Ray* Materialien dar. Die Shader-Netzwerke der Blätter sind dreistufig aufgebaut, so dass der Nutzer, entsprechend dem jeweiligen Einsatz in der 3D-Szene, das Material mit geringer, mittlerer oder hoher Detailstufe verwenden kann. Dies ist in Abbildung 4.1 mit den Buchstaben A, B, und C gekennzeichnet. Stufe A bezeichnet die niedrigste Qualitätstufe und beinhaltet lediglich das Material für die Vorderseite des Blattes, ohne Strukturdetails. Der Nutzer kann dieses Material für weit entfernte oder verdeckte Objekte in einer 3D-Szene verwenden. Stufe B stellt die mittlere Qualitätsstufe dar, welche aus einem zweiseitigen Material (*V-RayMtl2Sided*, s. Kapitel 3.7.3) besteht, ebenfalls ohne Strukturdetails. Es werden also verschiedene Materialien für Vorder- und Rückseite des Blattes verwendet. Blattstrukturen werden mittels einer Normal-Map in Stufe C hinzugefügt. Das *V-Ray Bump Material*

¹<http://xfrog.com/support/rendering/vray/>

4. ANALYSE DER V-RAY SHADER-NETZWERKE UND DEREN PARAMETER



Abbildung 4.1: Darstellung des Blatt-Shader-Netzwerkes von *Xfrog* für *V-Ray*
Quelle: Aus <http://xfrog.com/support/rendering/vray/> - Zuletzt geprüft: 03.08.2021

(*V-RayBumpMtl*, s. Kapitel 3.7.4), sollte demnach für Objekte im Kamerafokus oder in unmittelbarer Umgebung der Kameraposition verwendet werden. Im Rahmen dieser Arbeit werden ausschließlich die vollständigen Shader-Netzwerke (Stufe C) verwendet. Die grundlegenden Funktionen der in den Shader-Netzwerken enthaltenen Nodes werden in Kapitel 3.7 beschrieben. Die automatische Erstellung dieser Shader-Netzwerke mit Hilfe des im Vorfeld dieser Arbeit entwickelten Plug-ins wird in Kapitel 6 detailliert beschrieben.

Im Folgenden wird das *V-Ray* Blatt-Shader-Netzwerk exemplarisch am Modell *Golden Bamboo* (*Bambusa Vulgaris*) von *Xfrog* betrachtet. Das Modell ist im *Asia*-Paket von *Xfrog* enthalten und beinhaltet Shader-Netzwerke für mehrere Blatt-Variationen, welche allerdings alle identisch aufgebaut sind. Es werden für die verschiedenen Blatt-Materialien lediglich unterschiedliche Texturen verwendet. Aufgrund der hohen Anzahl an Einstellmöglichkeiten

werden ausschließlich jene Parameter der verschiedenen Nodes betrachtet, welche von den Standardeinstellungen abweichen.

4.1.1 Allgemeine Struktur

Das Blatt-Shader-Netzwerk ist, wie oben beschrieben, dreistufig aufgebaut. Abbildung 4.2 zeigt die Material-Nodes des Shader-Netzwerks. Aus Gründen der Übersichtlichkeit zeigt die Abbildung das Shader-Netzwerk ohne die dazugehörigen Texturen und Utility-Nodes.

Zu sehen ist (von rechts nach links) das *VRayBumpMtl*, welches die Darstellung der Blattstruktur durch die Verwendung einer Normalmap realisiert. Es stellt das letzte Material im Shader-Netzwerk dar und kann letztendlich dem gewünschten Szenen-Objekt zugewiesen werden. Das Material verwendet als Basismaterial das *VRayBlendMtl*, welches wiederum das *VRayMtl2Sided* als Basismaterial verwendet. Auf dieses wird im *BlendMtl* eine weitere Schicht aufgetragen, welche aus einem Standard *VRayMtl* besteht (Coat). Das *VRayMtl2Sided* vereint die beiden Materialien für Vorder- und Rückseite des Blattes, welche ebenfalls jeweils aus einem Standard *VRayMtl* bestehen. Die in Abbildung 4.2 nicht dargestellten Nodes und deren Integration in das Shader-Netzwerk werden in Kapitel 6 genauer erläutert.

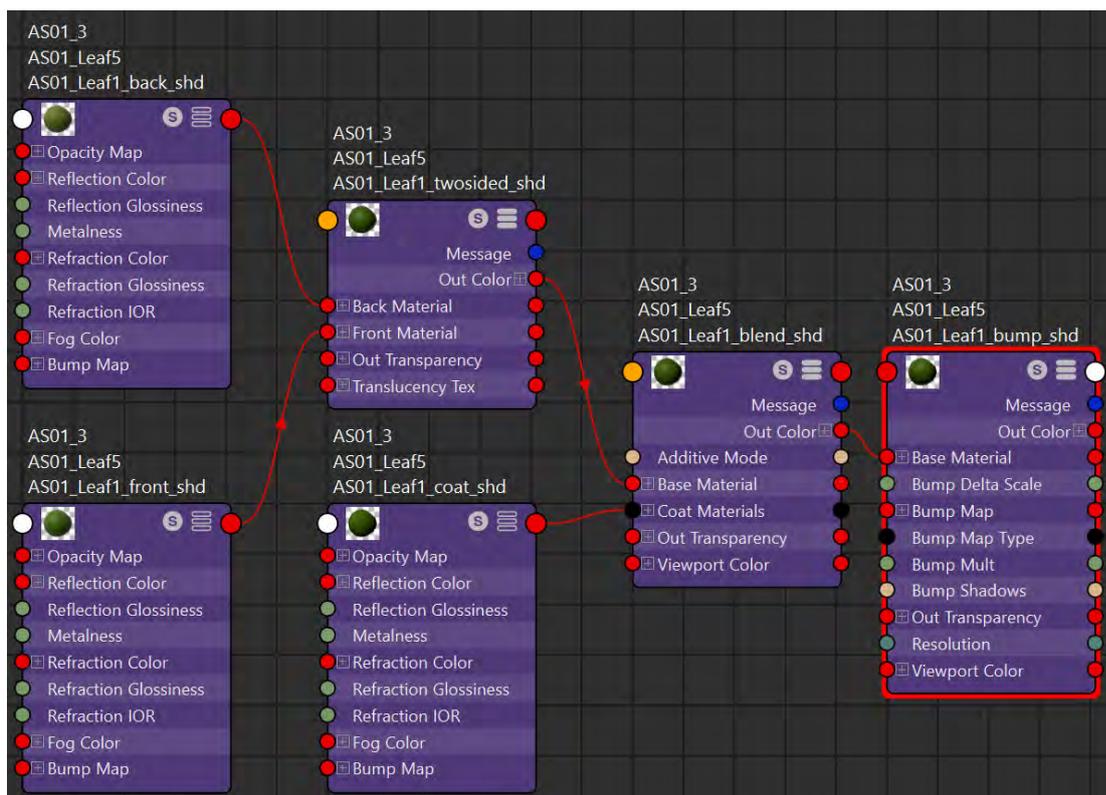


Abbildung 4.2: Grundstruktur des V-Ray Blatt-Shader-Netzwerks des Modells *Golden Bam-boo* von *Xfrog*

Quelle: Eigene Darstellung

4.1.2 Texture-Maps

Das Shader-Netzwerk für das Blatt enthält, wie alle Blatt Shader-Netzwerke von *Xfrog*, die für *V-Ray* ausgelegt sind, sechs Texture-Maps (siehe Abbildung 4.3).

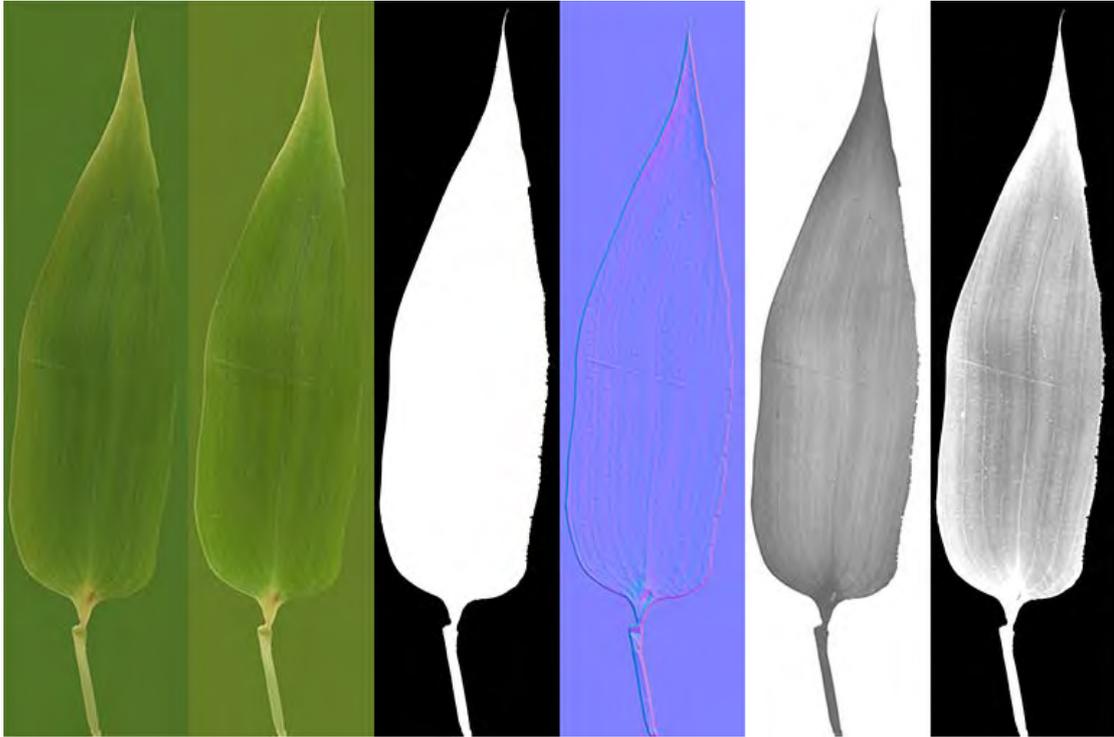


Abbildung 4.3: Die sechs Texture-Maps für Blatt 1 des Modells *Golden Bamboo* von *Xfrog*. Zu sehen sind die Texturen (von links nach rechts):

Leaf Front, Leaf Back, Alpha, Normal, Coat, Reflection

Quelle: Eigene Darstellung, Texturen aus Modell *Golden Bamboo* von *Xfrog*

Diese werden jeweils mittels eines Standard *Maya* File-Nodes eingebunden. Die Einstellungen der File-Nodes sind, mit Ausnahme der Normal-Map, für alle Texturen gleich (siehe Abbildung 4.4). Der Filter ist deaktiviert, für den Farbraum wurde *sRGB* gewählt und die File-Nodes enthalten das *V-Ray* Zusatzattribut *Texture Input Gamma*, zu sehen in der Abbildung unter *Extra Attributes* (die letzten drei Parameter). Der File-Node für die Normal-Map verwendet ebenfalls das Zusatzattribut, der einzige Unterschied zu den File-Nodes der anderen Texturen ist die Verwendung eines quadratischen Filters. Das Zusatzattribut beinhaltet drei Parameter, welche im Folgenden kurz beschrieben werden:

- **Vray File Gamma Enable:**
Wenn aktiviert, werden die eingehenden Bilddaten in Abhängigkeit der folgenden Einstellungen modifiziert.
- **Vray File Color Space:**

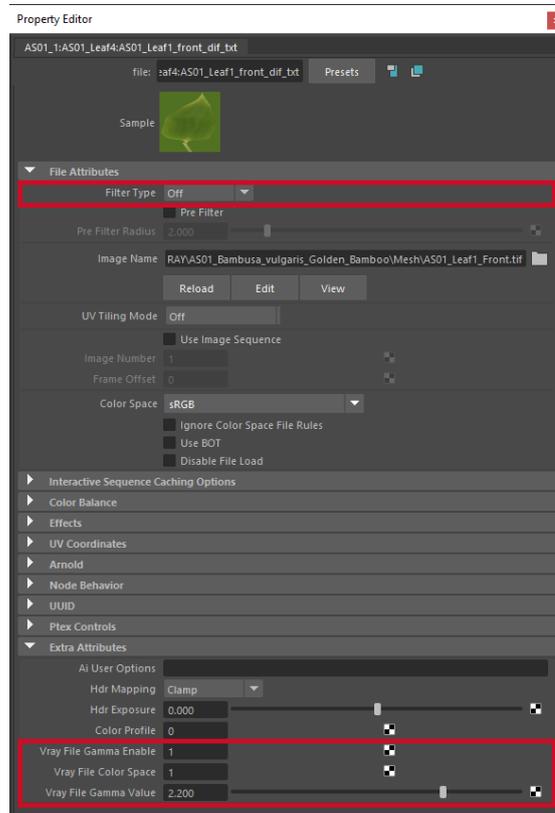


Abbildung 4.4: Einstellungen für alle im V-Ray Blatt Shader-Netzwerk verwendeten Texturen, mit Ausnahme der Normal-Map. Die rot markierten Parameter weichen von den Standardeinstellungen ab

Quelle: Eigene Darstellung

- 0, Linear - Die Bilddaten existieren bereits im linearen Farbraum. Es sind keine Anpassungen nötig.
 - 1, Gamma - Die Bilddaten sind mit dem Wert von **Vray File Gamma Value** gammakorrigiert. V-Ray entfernt die Gammakorrektur vor dem Rendern.
 - 2, sRGB - Die Bilddaten existieren im sRGB-Farbraum. V-Ray konvertiert die Bilddaten vor dem Rendern in den linearen Farbraum.
- **Vray File Gamma Value:**
Das Entfernen der Gammakorrektur vor dem Rendern wird auf Basis dieses Wertes durchgeführt, falls unter **Vray File Color Space** die Option **Gamma** ausgewählt wurde.

Diese Parameter besitzen für alle verwendeten Texturen die gleichen Werte. Wie in Abbildung 4.4 zu sehen ist, ist die Gammakorrektur für eingehende Bilddaten aktiviert und sie werden als gammakorrigierte Bilder mit dem Korrekturwert 2,2 interpretiert. Dies führt zu

korrekten Lichtberechnungen im Rendervorgang, allerdings werden auf diese Weise zusätzliche Warnungen von *Maya* ausgegeben, die darauf hinweisen, dass die Texturen im falschen Farbraum für das Normal-Mapping verwendet werden, da der Parameter *Color Space* auf *sRGB* eingestellt ist.

4.1.3 Das Bump Material

Das Bump Material (*VRayBumpMtl*) ist das letzte Material im Blatt Shader-Netzwerk, welches letztendlich einem Objekt in der 3D-Szene zugewiesen wird. Die Einstellungen des Materials sind in Abbildung 4.5 abgebildet. Für die Darstellung der Oberflächenstruktur des Blattes wird hier die Normal-Map mit dem Input *Bump Map* verbunden. Der Parameter *Bump Mult* dient der Einstellung der Intensität und wird auf 0,35 festgelegt. Unter *Map Type* ist *Normal map in tangent space* gewählt.

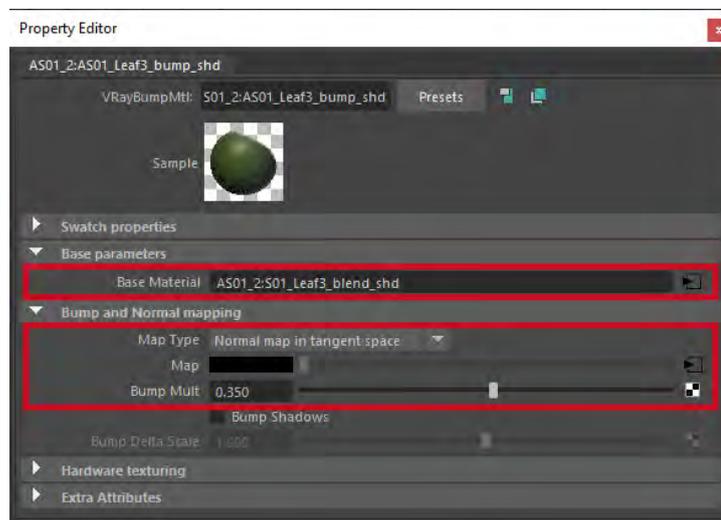


Abbildung 4.5: Einstellungen für das Bump Material im V-Ray Blatt Shader-Netzwerk. Die relevanten Einstellungen sind rot markiert

Quelle: Eigene Darstellung

Als Basismaterial kommt das Blend-Material zum Einsatz und wird dementsprechend mit dem Parameter *Base Material* verbunden. Das Blend-Material wird im nächsten Abschnitt beschrieben.

4.1.4 Blend Material (Coat)

Das Blend-Material, hier auch als *Coat* bezeichnet, besteht aus einem Basismaterial und einer darauf aufgetragenen Schicht. Diese Schicht besteht wiederum aus dem Coat Material und dem Parameter *Blend Amount*, welcher dem Blend-Material als Überblendungsfaktor dient. Die Einstellungen für das Blend-Material sind in Abbildung 4.6 dargestellt. Der Überblendungsfaktor besteht aus der Texture-Map für den Coat (siehe Abbildung 4.3) und einem *remapHsv*-Node (siehe Abbildung 4.7, rote Markierung). Der *remapHsv*-Node dient der

nachträglichen Feinjustierung der Coat-Map und dunkelt diese mit einem *Value*-Wert von 0.9, für die Farbe Weiß, leicht ab.

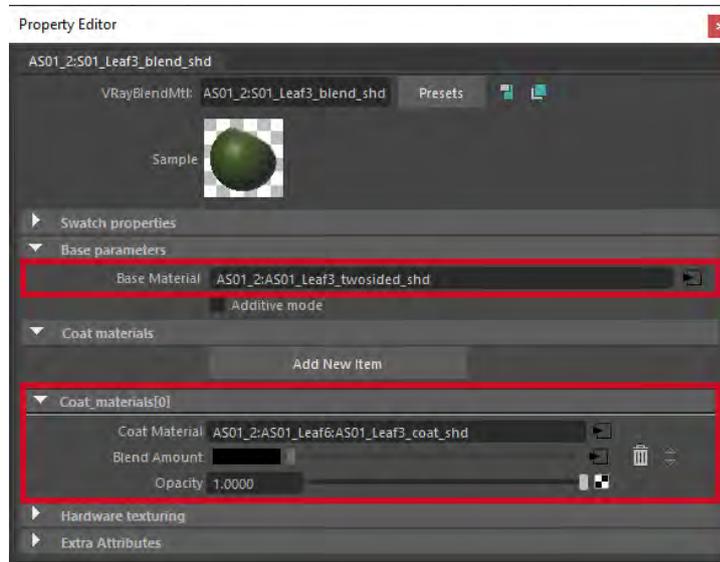


Abbildung 4.6: Einstellungen für das Blend Material im V-Ray Leaf Shader-Netzwerk. Die relevanten Einstellungen sind rot markiert

Quelle: Eigene Darstellung

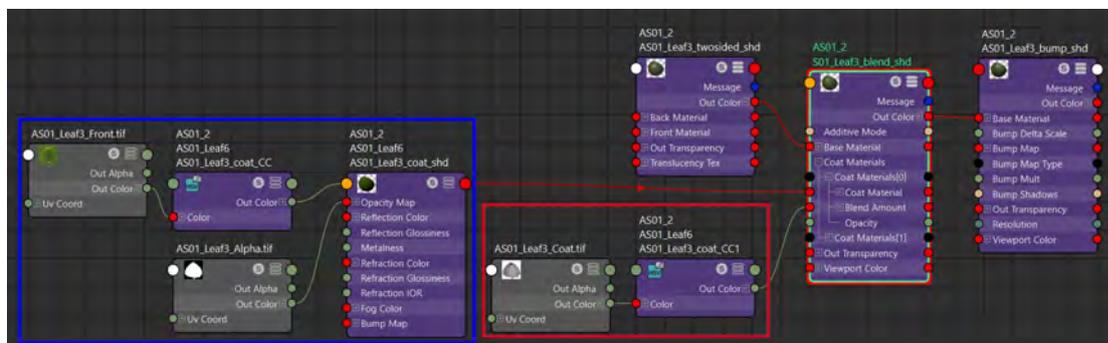


Abbildung 4.7: Input für die Parameter *Blend Amount* und *Coat Material* im V-Ray Blatt Shader-Netzwerk

Quelle: Eigene Darstellung

4.1.5 Coat Material

Das Coat-Material erhält den Input für den *Color*-Parameter von dem File-Node mit der Texture-Map *Leaf Front* und einem zwischengeschalteten *remapHsv*-Node (siehe Abbildung 4.7, blaue Markierung). Für den Parameter *Opacity Map* wird der File-Node mit der Alpha-Map verwendet, sodass nur der sichtbare Teil des Blattes und nicht das vollständige Mesh

4. ANALYSE DER V-RAY SHADER-NETZWERKE UND DEREN PARAMETER

des Blattes gerendert wird. Der *remapHsv*-Node dunkelt die Textur mit einem *Value*-Wert von 0,72, für die Farbe Weiß, leicht ab. Dies entspricht dem allgemeinen Erscheinungsbild von Pflanzen, auf der Vorderseite des Blattes eine etwas dunklere, wachsartige Glanzschicht zu besitzen. Die von den Standardwerten abweichenden Parameter des Coat-Materials sind in Abbildung 4.8 abgebildet.

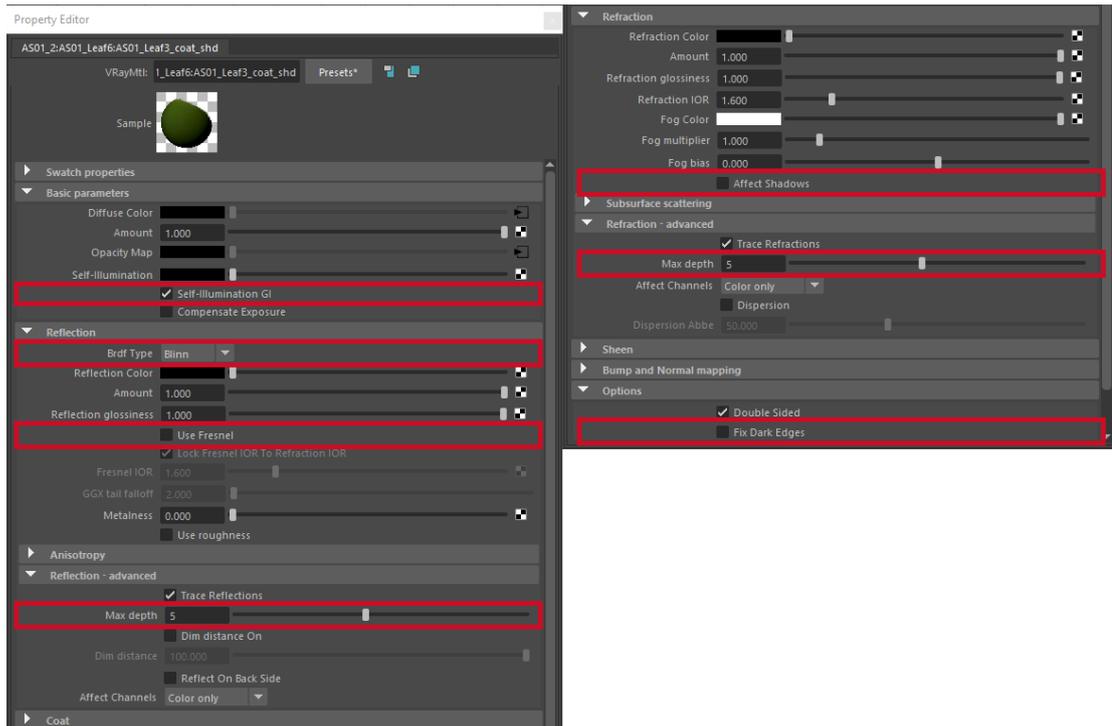


Abbildung 4.8: Parameter des Coat-Materials im V-Ray Blatt Shader-Netzwerk
Quelle: Eigene Darstellung

4.1.6 Zweiseitiges Material

Das *VRayMtl2Sided* dient als Basismaterial für das Blend-Material (siehe Kapitel 4.1.4) und vereint die Materialien für Vorder- und Rückseite des Blattes, in Abhängigkeit des Parameters *Translucency Tex*, welcher als Überblendungsfaktor dient. Die grundlegende Struktur des zweiseitigen Materials ist in Abbildung 4.9 dargestellt. Je nach Wert des Parameters *Translucency Tex* und räumlicher Ausrichtung des Blattes relativ zur Kameraposition, wird ein prozentualer Anteil von Vorder- oder Rückseite des Blattes in Betrachtungsrichtung gerendert. Dieser Wert wird in den Shader-Netzwerken mit Hilfe eines Graustufen-Gradienten (*ramp*-Node) festgelegt. Der Gradient enthält zwei Knoten, deren Werte in Tabelle 4.1 aufgelistet sind.

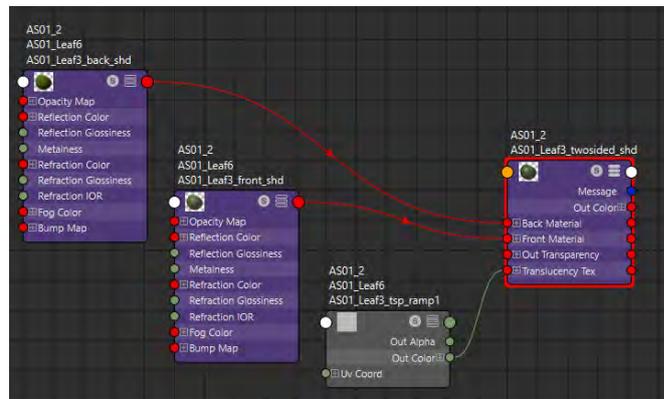


Abbildung 4.9: Grundlegende Struktur des zweiseitigen Materials im V-Ray Blatt Shader-Netzwerk

Quelle: Eigene Darstellung

Knoten	Position	Hue	Saturation	Value
1	0,47	0	0	0,5
2	1	0	0	0

Tabelle 4.1: Auflistung der Knoten des *ramp*-Nodes für den Parameter *Translucency Tex* und deren Werte

4.1.7 Front Material

Das Front Material bestimmt die Darstellung der Vorderseite des Blattes und dient dementsprechend als Input für den Parameter *Front Material* des zweiseitigen Materials. Die grundlegende Struktur des Front Materials ist in Abbildung 4.10 dargestellt. Als Input für den Parameter *Color* des Front Materials dient der File-Node mit der Textur *Front Leaf* mit einem zwischengeschalteten *remapHsv*-Node. Letzterer dient der nachträglichen Feinjustierung der Textur, schleift in diesem Fall allerdings lediglich die Textur durch, ohne Änderungen an den Bilddaten vorzunehmen. Für den Parameter *Opacity Map* wird der File-Node mit der Alpha-Map verwendet. Der Parameter *Reflection Color* erhält den Input von dem File-Node mit der Reflection-Textur, ebenfalls mit einem zwischengeschalteten *remapHsv*-Node. Auch dieser schleift die Textur lediglich durch und ändert an dieser nichts. Abschließend erhält der Parameter *Reflection Glossiness* einen Graustufen-Gradienten als Input. Der Gradient beinhaltet zwei Knoten, deren Werte in Tabelle 4.2 abgebildet sind. Die von den Standard-

Knoten	Position	Hue	Saturation	Value
1	0	0	0	0,89
2	0,785	0	0	0,65

Tabelle 4.2: Auflistung der Knoten des *ramp*-Nodes für den Parameter *Reflection Glossiness* des Front Materials und deren Werte

4. ANALYSE DER V-RAY SHADER-NETZWERKE UND DEREN PARAMETER

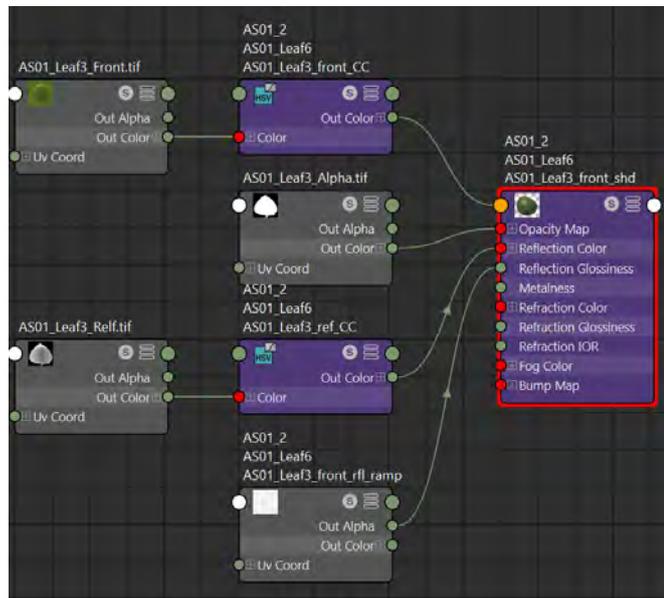


Abbildung 4.10: Grundlegende Struktur des Front Materials im V-Ray Blatt Shader-Netzwerk

Quelle: Eigene Darstellung

einstellungen abweichenden Parameter des Front-Materials sind in Abbildung 4.11 dargestellt und rot markiert.

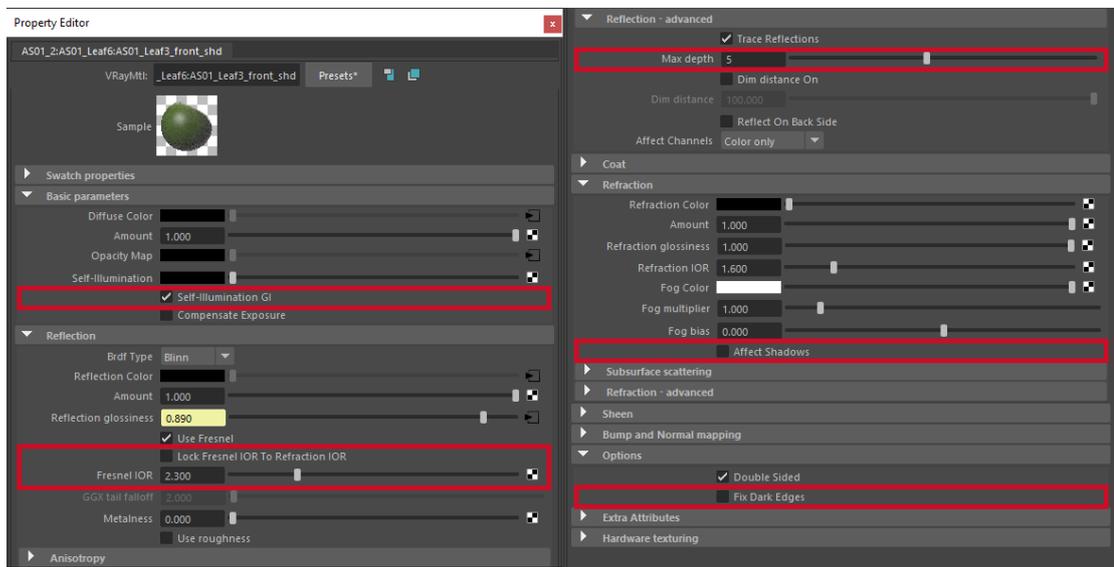


Abbildung 4.11: Parameter des Front-Materials im V-Ray Blatt Shader-Netzwerk

Quelle: Eigene Darstellung

4.1.8 Back Material

Das Back Material ist analog zum Front Material aufgebaut. Als Input erhält das Material den File-Node mit der Textur *Leaf Back* und zwischengeschaltetem *remapHsv*-Node. Auch hier werden die Bildinformationen nicht angepasst und somit unverändert weitergeleitet. Die Knoten des *ramp*-Nodes für den Parameter *Reflection Glossiness* und deren Werte sind in Tabelle 4.3 aufgelistet.

Knoten	Position	Hue	Saturation	Value
1	0	0	0	0.80
2	0.89	0	0	0.538

Tabelle 4.3: Auflistung der Knoten des *ramp*-Nodes für den Parameter *Reflection Glossiness* des Back Materials und deren Werte

4.2 Das V-Ray Bark Shader-Netzwerk

Das Bark Shader-Netzwerk (Baumrinde) ist im Vergleich zum Blatt Shader-Netzwerk sehr viel einfacher aufgebaut und wird von *Xfrog* nicht ausschließlich für Baumrinde, sondern auch für Zweige (Twigs), Äste (Branches), Wurzeln (Roots) und Stämme (Trunks) verwendet. Die folgende Analyse des Bark Shader-Netzwerks kann demnach für die anderen Typen analog betrachtet werden.

4.2.1 Allgemeine Struktur

Das Bark Shader-Netzwerk ist nicht mehrstufig aufgebaut und wird immer in vollem Umfang genutzt. Der grundlegende Aufbau des Shader-Netzwerks ist in Abbildung 4.12 dargestellt. Es besteht aus einem Bump-Material (*VRayBumpMtl*) und einem Standard *V-Ray*-Material (*VrayMtl*). Aus Gründen der Übersichtlichkeit zeigt die Abbildung das Shader-Netzwerk ohne die dazugehörigen Texturen und Utility-Nodes.

4.2.2 Texture-Maps

Für das Bark Shader-Netzwerk kommen zwei Texture-Maps zum Einsatz: Eine Normal-Map und eine Baumrinden-Textur (siehe Abbildung 4.13). Im Gegensatz zu den File-Nodes der Blatt Shader-Netzwerke, werden in Bark Shader-Netzwerken keine Filter für die Normal-Maps und zudem keine *V-Ray* Zusatzattribute verwendet. Dennoch ist der Farbraum in den File-Node Einstellungen der Normal-Maps auf *sRGB* eingestellt, was letztendlich im Rendering zu einer nicht korrekten Lichtberechnung führt. Der File-Node der Baumrinden-Textur hingegen besitzt die *V-Ray* Zusatzattribute und wird somit von *V-Ray* wie ein Bild im linearen Farbraum behandelt, was ebenfalls zu nicht korrekten Berechnungen führt.

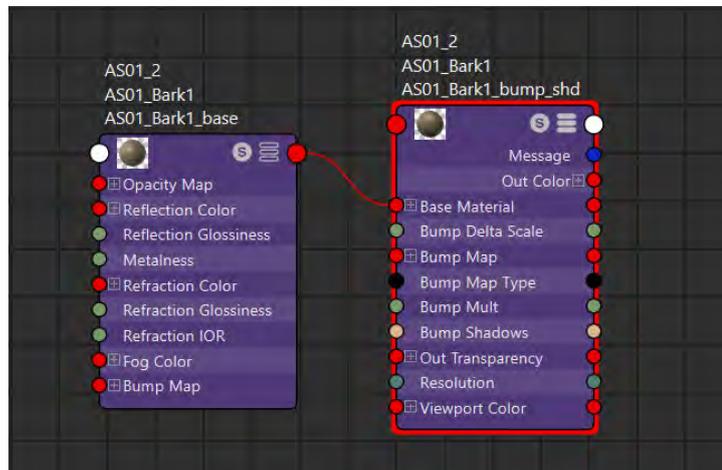


Abbildung 4.12: Grundstruktur des V-Ray Bark Shader-Netzwerks des Modells *Golden Bamboo* von *Xfrog*

Quelle: Eigene Darstellung

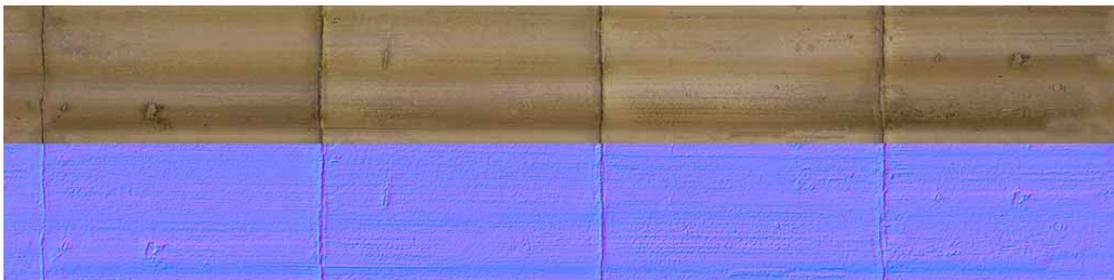


Abbildung 4.13: Die zwei Texture-Maps für die Baumrinde des Modells *Golden Bamboo* von *Xfrog*. Oben ist die Bark-Textur und darunter die entsprechende Normal-Map zu sehen. Die Texturen wurden aus Gründen der Übersichtlichkeit um 90° gedreht

Quelle: Eigene Darstellung, Texturen aus Modell *Golden Bamboo* von *Xfrog*

4.2.3 Bump Material

Das Bump Material des Bark Shader-Netzwerks verwendet als Basismaterial das *VRayMtl* und den File-Node mit der Normal-Map als Input für den Parameter *Bump Map*. Die Intensität *Bump Mult* hat den Wert 0.6 und der Parameter *Map Type* hat den Wert *Normal map in tangent space*.

4.2.4 Bark Material

Das Bark Material (*VRayMtl*) erhält als Input für den Parameter *color* die Texture-Map des File-Nodes für die Bark-Textur mit einem zwischengeschaltetem *remapHsv*-Node. Dieser dient auch hier nur der nachträglichen Feinjustierung der Farbgebung des Bildes und ändert auch in diesem Fall nichts am Bildmaterial.

4.3 Zusammenfassende Betrachtung

Die betrachteten Shader-Netzwerke folgen klaren Strukturen und werden für eine Vielzahl an 3D-Objekten eingesetzt. Das Blatt Shader-Netzwerk wird für alle Arten von Blättern verwendet. Das Bark Shader-Netzwerk ist im Gegensatz zum Blatt Shader-Netzwerk recht einfach aufgebaut und wird für viele verschiedene Teile der Pflanzen, wie Äste, Stamm und Wurzeln verwendet. Die Shader-Netzwerke für Blütenblätter wurden in diesem Kapitel nicht betrachtet, da sie zum derzeitigen Stand des Konvertierungs-Plugins noch nicht konvertiert werden können. Dennoch ist anzumerken, dass die Struktur der Blütenblätter Shader-Netzwerke identisch zur Struktur der Blatt Shader-Netzwerke ist und sich lediglich einige wenige Parameter in ihren Werten unterscheiden. Die verwendeten Utility-Nodes, wie die *remapHsv*-Nodes, wurden so in die Shader-Netzwerke integriert, dass eine potentielle individuelle Anpassung der wichtigsten Parameter jederzeit leicht umzusetzen ist.

Die klaren und konsequent verwendeten Strukturen der Shader-Netzwerke machen eine automatische Konvertierung grundsätzlich möglich. Die den Modellen beiliegenden Texture-Maps sind im Falle der Verwendung eines für *Maya* ausgelegten Modells nicht vorhanden und müssen für eine Konvertierung entweder mittels externer Software oder in möglichst guter Approximation durch kombinierte Verwendung von Utility-Nodes (*remapHsv*, *reverse*, etc.) generiert werden. Dies wird in Kapitel 6 genauer erklärt und in Kapitel 7 werden die verschiedenen Methoden hinsichtlich des Renderergebnisses untersucht.

Unklar ist die unsachgemäße Verwendung von strukturgebenden Texture-Maps wie den Normal-Maps, welche nicht immer im korrekten Farbraum verwendet werden und somit zu einer nicht physikalisch korrekten Berechnung im Renderprozess und teilweise sogar zu Warnungen in *Maya* führen. Auch sind teilweise farbgebende Texturen mit den *V-Ray* Zusatzattributen bestückt, welche die sRGB-Bilder letztendlich wie Bilder im linearen Farbraum behandeln. Besonders im Vergleich zwischen den Baumrinden- und Blatt Shader-Netzwerken ist bei der Verwendung der *V-Ray* Zusatzattribute und der nativen Farbeinstellungen der File-Nodes keine einheitliche Vorgehensweise erkennbar und es ist davon auszugehen, dass diese nicht korrekt verwendeten Texturen zu falschen Lichtberechnungen beim Rendervorgang führen.

Im folgenden Kapitel werden die Shader-Netzwerke von *Xfrog* für den Renderer *Arnold* detailliert untersucht.

Kapitel 5

Analyse der Arnold Shader-Netzwerke und deren Parameter

In diesem Kapitel werden die Shader-Netzwerke von *Xfrog* für den Renderer *Arnold* untersucht. Hierbei werden, wie im vorigen Kapitel, lediglich die Shader-Netzwerke für Blätter und für Baumrinde untersucht, da dies die einzigen Shader-Netzwerke sind, die derzeit vom Konvertierungs-Plugin unterstützt werden. An dieser Stelle muss erwähnt werden, dass die vorliegenden *Xfrog* Modelle mit *Arnold* Materialien aus einer inoffiziellen Testversion stammen, welche zum Zeitpunkt der Erstellung dieser Arbeit von *Xfrog* nicht kommerziell zum Kauf angeboten werden.

5.1 Das Arnold Blatt Shader-Netzwerk

Das Arnold Blatt Shader-Netzwerk besteht insgesamt aus nur einem Material, dem *aiStandardSurface*-Material (siehe Kapitel 3.8.1). Es ist somit nicht mehrstufig aufgebaut und kann ausschließlich in vollem Umfang genutzt werden, auch wenn das jeweilige 3D-Objekt in der 3D-Szene weit im Hintergrund platziert ist, so dass die enthaltenen Details nicht sichtbar sind. Im nächsten Abschnitt wird das Arnold Blatt Shader-Netzwerk exemplarisch am Modell *Golden Bamboo* von *Xfrog* detailliert untersucht.

5.1.1 Allgemeine Struktur

Wie im vorigen Abschnitt bereits erwähnt, ist das Arnold Leaf Shader-Netzwerk nicht mehrstufig aufgebaut und besteht lediglich aus einem einzigen Material. Es existiert somit keine komplexe Struktur, die vorhandenen Texture-Maps und Utility-Nodes werden lediglich mit den entsprechenden Inputs des *aiStandardSurface*-Materials verbunden. Abbildung 5.1 zeigt diesen Aufbau anhand des Shader-Netzwerks von Blatt 1 des Modells *Golden Bamboo*.

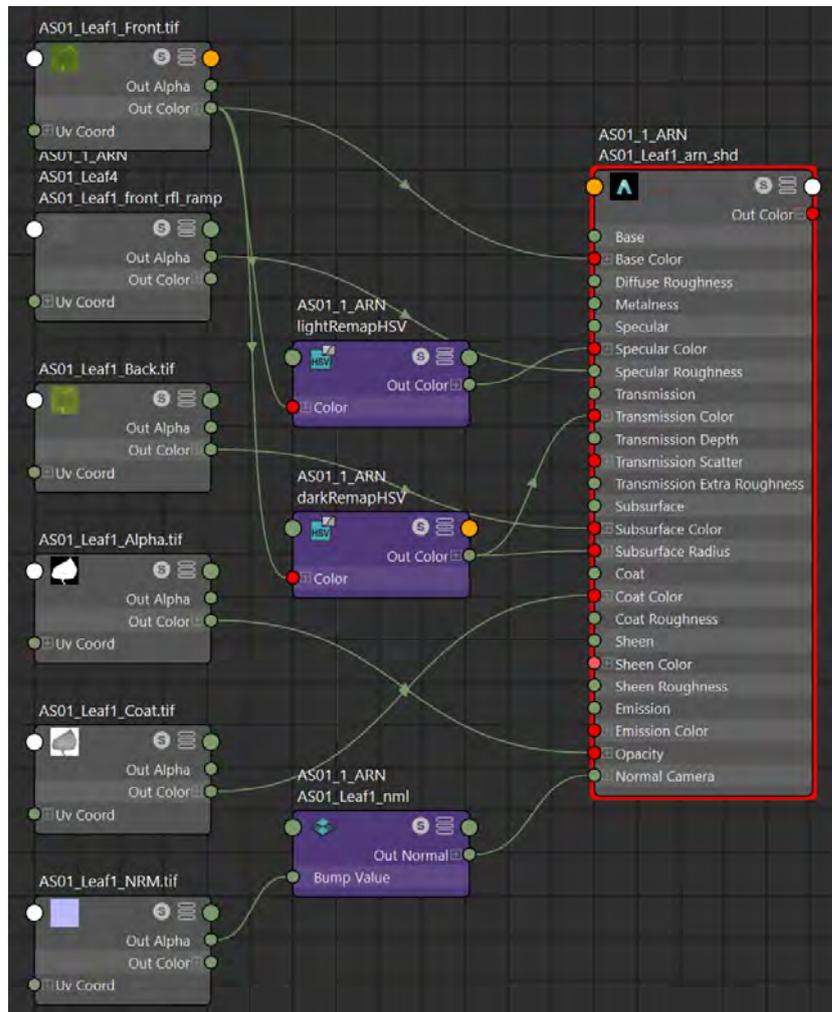


Abbildung 5.1: Darstellung der grundlegenden Struktur des Arnold Leaf Shader-Netzwerks des Modells *Golden Bamboo* von *Xfrog*
Quelle: Eigene Darstellung

5.1.2 Texture-Maps

In den Dateordnern der *Arnold*-Modelle von *Xfrog* finden sich exakt die gleichen Texturen wie bei den Modellen für *V-Ray*. Sie entsprechen also den Texture-Maps aus Abbildung 4.3. Allerdings werden in den Arnold Leaf Shader-Netzwerken nur fünf der sechs Texture-Maps verwendet, die Reflection-Map kommt nicht zum Einsatz. Anstelle des File-Nodes mit der Reflection-Map wird ein Graustufen-Gradient (ramp) verwendet. Für alle File-Nodes, mit Ausnahme des File-Nodes der Normal-Map, werden die gleichen Einstellungen verwendet. Die von den Standardeinstellungen abweichenden Parameter sind in Abbildung 5.2 rot markiert. Die Texture-Maps werden ohne Filter, die Normal-Map hingegen mit einem quadratischen Filter verwendet. Die Farbraum-Einstellung wird bei der Normal-Map komplett

deaktiviert, alle anderen Texturen verwenden den *sRGB*-Farbraum. Alle Texturen, mit Ausnahme der Normal-Map, verwenden zudem die *V-Ray*-Zusatzattribute, welche in den *V-Ray* Blatt Shader-Netzwerken zum Einsatz kommen und eine Auswahl des verwendeten Farbraums zur Renderzeit ermöglichen. Da diese Zusatzattribute *V-Ray* spezifisch sind, muss hier angenommen werden, dass es sich um einen Fehler in der Konstruktion der *Arnold* Shader-Netzwerke handelt.

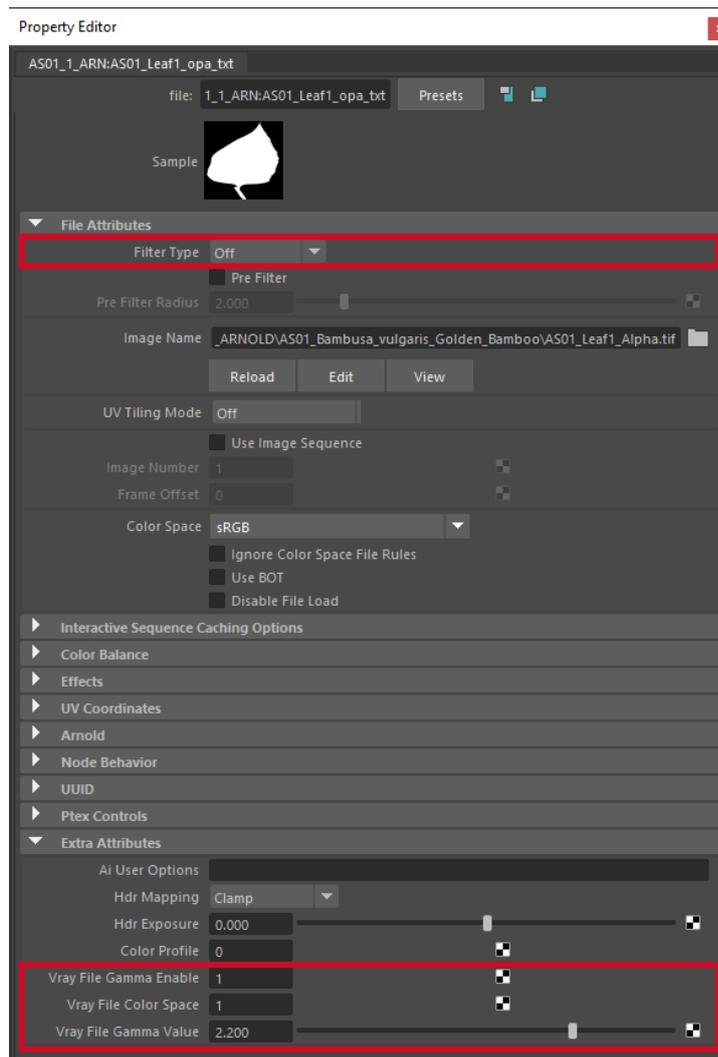


Abbildung 5.2: Einstellungen für alle im *Arnold* Blatt Shader-Netzwerk verwendeten Texturen, mit Ausnahme der Normal-Map. Die rot markierten Parameter weichen von den Standardeinstellungen ab

Quelle: Eigene Darstellung

5.1.3 Das Blatt-Material (aiStandardSurface)

Im Folgenden wird der Aufbau und die Verbindung der verwendeten Nodes, wie in Abbildung 5.1 abgebildet, zum Blatt-Material detailliert beschrieben. Die Parameter des Materials sind in Abbildung 5.3 abgebildet, die von den Standardeinstellungen abweichenden Werte sind rot markiert.

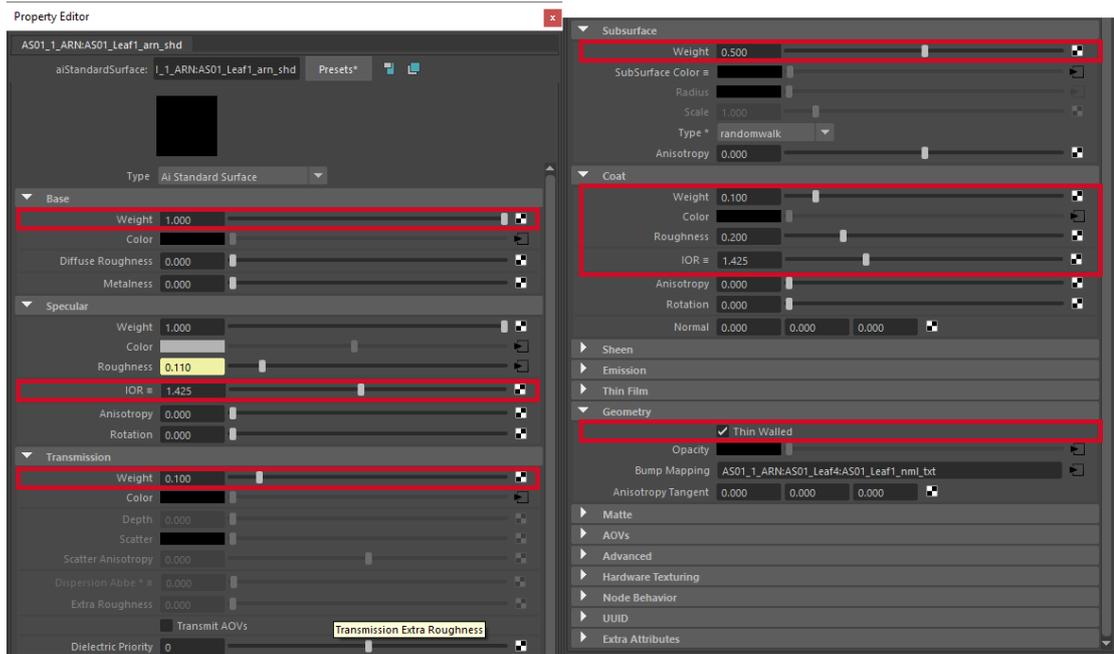


Abbildung 5.3: Einstellungen der Parameter des *Arnold* Blatt-Materials. Die rot markierten Parameter weichen von den Standardeinstellungen ab

Quelle: Eigene Darstellung

Das Blatt-Material erhält seine Farbgebung über den Input *Base Color*, welcher vom File-Node mit der *Front Leaf*-Textur geliefert wird. Dieser wird ohne zusätzliche Nodes direkt über den Output *Out Color* mit dem Input des Materials verbunden. Der Wert des Parameters *Weight* unter dem Reiter *Base* beträgt 1.0.

Die *Leaf Front*-Textur wird zusätzlich für den Input *Specular Color* verwendet, allerdings mit einem zwischengeschalteten *remapHsv*-Node. Die Werte dieses Nodes sind in Tabelle 5.1 dargestellt. Somit wird für den Parameter *Specular Color* eine in den dunklen Bereichen

Knoten	Position	Hue	Saturation	Value
1	0.0	0	0	0.45
2	1	1	1	1

Tabelle 5.1: Auflistung der Knoten des *remapHsv*-Nodes für den Parameter *Specular Color* und deren Werte

leicht aufgehellte Variante der *Front Leaf*-Textur verwendet.

Für den Parameter *Specular Roughness* kommt, wie oben bereits erwähnt, nicht die mitgelieferte Reflection-Textur, sondern ein Graustufen-Gradient zum Einsatz. Dieser wird über den Output *Alpha* mit dem Input *Specular Roughness* des Blatt-Materials verbunden. Die Werte der beiden Knoten des Gradienten sind in Tabelle 5.2 aufgelistet. Zudem ist die Option *Invert* unter dem Reiter *Effects* des *ramp*-Nodes aktiviert, was zu einer fehlerhaften Ausgabe des Alphakanals führt und das gesamte Material stark glänzend dargestellt wird. Hier muss angenommen werden, dass dies so nicht vorgesehen war und es sich um einen Fehler handelt. Der Parameter *IOR* (Index of refraction) unter dem Reiter *Specular* erhält den Wert 1.425.

Knoten	Position	Hue	Saturation	Value
1	0.0	0	0	0.89
2	0.785	0	0	0.65

Tabelle 5.2: Auflistung der Knoten des *ramp*-Nodes für den Parameter *Specular Roughness* des *Arnold* Blatt-Materials und deren Werte

Für den Parameter *Transmission Color* wird eine stark abgedunkelte Variante der *Leaf Front*-Textur verwendet. Diese wird ebenfalls mit einem *remapHsv*-Node realisiert. Die Werte des *remapHsv*-Nodes sind in Tabelle 5.3 aufgelistet. Der Parameter *Weight* unter dem Reiter *Transmission* erhält den Wert 0.1. Es gilt zu beachten, dass für die Verwendung von *Transmission* die Option *Opaque* unter dem Reiter *Arnold* des jeweiligen *Shape*-Nodes manuell deaktiviert werden muss.

Knoten	Position	Hue	Saturation	Value
1	0.0	0	0	0.0
2	1	1	1	0.135

Tabelle 5.3: Auflistung der Knoten des *remapHsv*-Nodes für den Parameter *Transmission Color* des *Arnold* Blatt-Materials und deren Werte

Die *Leaf Back*-Textur wird für den Material-Input *Subsurface Color* verwendet. Der Output *Out Color* der Textur wird mit dem Material-Input verbunden. Der Input *Subsurface Radius* erhält denselben Input vom *ramp*-Node wie der Parameter *Specular Roughness*. Der Parameter *Weight* unter dem Reiter *Subsurface* des Materials erhält den Wert 0.5. Wie in Kapitel 3.5.2 beschrieben, ist die Berechnung von SSS (Subsurface Scattering) im *aiStandardSurface*-Material davon abhängig, ob das Objekt, welches mit dem Material gerendert wird, als Objekt mit Volumen oder als unendlich dünnes Objekt angesehen wird. Da in diesem Falle lediglich das Durchdringen von Licht, nicht aber die Streuung des Lichts in der Materie von Interesse ist, muss dies unter dem Reiter *Geometry* des Materials durch aktivieren der Option *Thin Walled* aktiviert werden.

Der Input *Coat Color* des Materials erhält seine Information vom *File*-Node mit der *Coat*-Textur. Dieser wird ohne zusätzliche *Utility*-Nodes über den Output *Out Color* mit dem Material verbunden. Unter dem Reiter *Coat* des Materials werden die Werte *Weight* mit 0.1, *Roughness* mit 0.2 und *IOR* mit 1.425 festgelegt.

Damit nicht das gesamte Mesh des Blattes, sondern nur der sichtbare Teil des Blattes gerendert wird, erhält das Material über den Input *Opacity* die notwendigen Informationen von dem File-Node mit der Alpha-Textur. Dieser wird über den Output *Out Color* mit dem Material verbunden.

Abschließend erhält das Material den Input des File-Nodes mit der Normal-Map zur Darstellung der Oberflächenstruktur. Der Output *Out Alpha* des FileNodes wird allerdings zunächst an den Input *Bump Value* eines *bump2d*-Nodes angeschlossen. Der Output *Out Normal* dieses Nodes wird dann schließlich an den Input *Normal Camera* des *aiStandard-Surface*-Materials angeschlossen. Unter dem Reiter *2d Bump Attributes* des *bump2d*-Nodes werden die Werte *Bump Depth* mit 1.0 und *Use As* mit *Tangent Space Normals* eingestellt.

5.2 Das Arnold Bark Shader-Netzwerk

Das Arnold Bark Shader-Netzwerk besteht, wie das Arnold Leaf Shader-Netzwerk, aus nur einem einzigen Material. Analog zum *V-Ray* Bark-Material, wird das Material nicht nur für Baumrinde, sondern auch für Zweige, Äste und Stämme verwendet. Die Struktur des Shader-Netzwerks ist bei all diesen Typen die Gleiche.

5.2.1 Texture-Maps

Das Shader-Netzwerk benötigt eine Textur für die Farbgebung und eine Textur für die Oberflächenstruktur des Objekts. Die Einstellungen der File-Nodes entsprechen den Einstellungen der File-Nodes des *Arnold* Blatt Shader-Netzwerks, mit dem Unterschied, dass auch für die Normal-Map kein Filter verwendet wird.

5.2.2 Bark-Material

Das Bark-Material erhält die nötigen Informationen zur Farbgebung vom File-Node mit der *Bark*-Textur über den Input *Base Color*. Der Parameter *Weight* unter dem Reiter *Base* des Materials hat den Wert 1.0. Der Parameter *Weight* unter dem Reiter *Specular* hat den Wert 0.0 und wurde somit deaktiviert. Das Material liefert demnach rein diffuse Reflexionen.

Die Normal-Map wird, analog zum *Arnold* Blatt Shader-Netzwerk, über einen *bump2d*-Node an den Input *Normal Camera* angeschlossen. Die Parameter des *bump2d*-Nodes werden für *Bump Depth* mit 1.0 und *Use As* mit *Tangent Space Normals* festgelegt.

5.3 Zusammenfassende Betrachtung

Die Blatt Shader-Netzwerke für *Arnold* sind, wie die *V-Ray* Blatt Shader-Netzwerke, klar strukturiert und besitzen bei allen untersuchten Modellen die gleiche Struktur. Eine automatische Konvertierung von Materialien der *Maya*-Modelle in *Arnold*-Materialien ist somit möglich.

Dennoch erwecken die *Arnold* Shader-Netzwerke den Eindruck, nicht ausgereift zu sein. An dieser Stelle sei erneut angemerkt, dass die vorliegenden *Arnold* Modelle einer inoffiziellen Testversion entstammen und nicht kommerziell zum Kauf angeboten werden. Das

größte Problem stellt die Invertierung des Graustufen-Gradienten für den Parameter *Specular Roughness* dar, welcher zu unrealistisch wirkenden Renderings führt, da das Blatt stark glänzend und somit unnatürlich aussieht. Die Invertierung muss daher vor dem Rendern rückgängig gemacht werden. Dieser Umstand ist insofern verwunderlich, da den Modellen, wie oben beschrieben, eine Texture-Map für die Reflektionen beiliegt und somit für realistischere Renderergebnisse hätte verwendet werden können.

Die zusätzlichen *V-Ray* Attribute, welche allen File-Nodes, außer denen der Normal-Maps, zugewiesen wurden, sind für den *Arnold*-Renderer irrelevant und wahrscheinlich nicht absichtlich hinzugefügt worden.

Diese Probleme wurden im Rahmen dieser Arbeit im Konvertierungs-Plugin berücksichtigt und im Anschluss dieser Analyse entsprechend korrekt implementiert. Im folgenden Kapitel werden diese und weitere Probleme, sowie deren Lösung, genauer beschrieben. In Kapitel 7 wird herausgearbeitet, inwiefern eine automatische Konvertierung die hier beschriebenen Shader-Netzwerke nachbilden und im besten Fall sogar optimieren kann.

Kapitel 6

Konzept und Entwicklung eines Plugins zur automatischen Konvertierung der Xfrog Shader-Netzwerke

Teile des Plugins wurden bereits im Vorfeld dieser Arbeit im vorbereitenden Modul „Entwicklungsprojekt“ entwickelt und im Rahmen dieser Arbeit weiterentwickelt. Das Plugin wird genutzt, um die Materialien von *Maya*-Modellen zu konvertieren und im Anschluss mit Hilfe einer Evaluation eine Aussage zu erhalten, inwiefern die Konvertierung realistische Ergebnisse im Vergleich zu den von *Xfrog* ausgelieferten *V-Ray* und *Arnold* Materialien erzielt und um Vergleiche zwischen den Herstellungsverfahren der Texture-Maps zu erhalten.

Das Plugin stellt keine abschließend entwickelte Software dar, sondern dient zunächst lediglich zum Testen der Machbarkeit einer automatischen Materialkonvertierung. Aufgrund der enormen Vielzahl an variablen Parametern und somit potentiellen Fehlerquellen in der Konvertierung, sind die Ergebnisse der Konvertierung zunächst als Grundlage gedacht, welche die Vor- und Nachteile einer automatischen Konvertierung aufzeigen sollen. Die Probleme der *Xfrog* Shader-Netzwerke, welche in Kapitel 4 und Kapitel 5 herausgearbeitet wurden, sind bereits zum Teil, im Rahmen dieser Arbeit, im Plugin berücksichtigt.

6.1 Konzept

Das Plugin dient in erster Linie der Untersuchung über die Machbarkeit einer automatischen Konvertierung. Des Weiteren soll anhand der konvertierten Materialien untersucht werden, ob deren Qualität mit der Qualität der von *Xfrog* ausgelieferten Materialien vergleichbar ist, oder sie im besten Fall sogar überbieten kann. Eine benutzerfreundliche grafische Benutzeroberfläche und auf die wichtigsten Parameter beschränkte Auswahlmöglichkeiten sollen nicht nur eine schnelle, sondern vor allem eine robuste und qualitativ gleichbleibende Konvertierung ermöglichen.

6.2 Das Plugin

Das Plugin besteht zur Zeit aus zwölf *MEL*-Skripten, welche je nach Anwendungsfall zum Einsatz kommen. Sobald die Skripte korrekt in *Maya* installiert wurden, kann das Plugin mit dem *MEL*-Befehl *xfrogconverter* im *Maya*-internen Skript-Editor gestartet werden. Das Programm startet zunächst die grafische Benutzeroberfläche, welche in Abbildung 6.1 zu sehen ist. Im oberen Feld werden alle Shadinggroups des derzeit ausgewählten Objekts angezeigt,

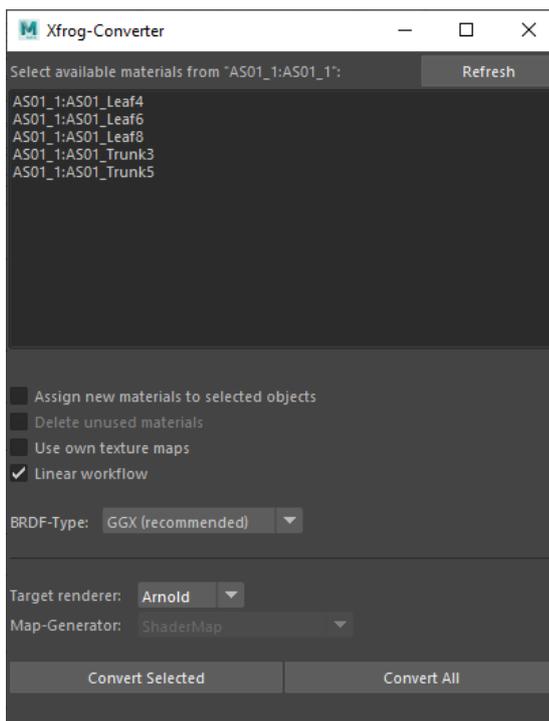


Abbildung 6.1: Darstellung des Hauptmenüs des Xfrog-Converter Plugins

Quelle: Eigene Darstellung

insofern sie nicht mit einem *V-Ray* oder *Arnold* Material verbunden sind. Im mittleren Feld sind die folgenden Optionen wählbar:

- **Assign new materials to selected objects:**
Wenn aktiviert, werden die Materialien nach der Konvertierung automatisch dem ausgewählten Szenen-Objekt zugewiesen.
- **Delete unused materials:**
Ist diese Option aktiviert, werden nach der Konvertierung alle nicht verwendeten Nodes gelöscht. Diese Option ist nur verfügbar, wenn die Option *Assign new materials to selected objects* aktiviert ist.
- **Use own texture maps:**
Diese Option schaltet die optionale Verwendung eigener Texture-Maps frei. Wird die

Konvertierung gestartet, öffnet sich der *Texture-Map Selector* und es können die benötigten Texture-Maps geladen werden. Ist diese Option nicht aktiviert, sucht sich das Programm die im Material vorhandenen Texturen und erstellt durch Kombination verschiedener Utility-Nodes die Texturen selbstständig. Bei diesem Workflow kann allerdings keine Normal-Map erstellt werden und es wird auf eine Bump-Map zurückgegriffen.

- **Linear Workflow:**

Bei Verwendung des linear workflow werden sämtliche strukturgebenden Texturen im *raw*-Farbraum verwendet. Diese Option ist standardmäßig aktiviert und stellt sicher, dass im Rendervorgang physikalisch korrekte Berechnungen durchgeführt werden können.

- **BRDF-Type:**

Mit Hilfe dieses Dropdown-Menüs kann zwischen vier verschiedenen BRDF-Modellen gewählt werden. GGX ist die empfohlene Methode und ist standardmäßig eingestellt.

Unter *Target Renderer* kann zwischen *Arnold* und *V-Ray* gewählt werden. Die Auswahl wird entsprechend dem derzeitig gewählten Renderer in *Maya* automatisch voreingestellt, kann aber nach Belieben geändert werden.

Das Dropdown-Menü *Map-Generator* ermöglicht die Auswahl zwischen der Verwendung von Texture-Maps aus *ShaderMap* oder *Substance 3D Sampler*. Diese Auswahlmöglichkeit war im ursprünglichen Plugin nicht implementiert und wurde im Rahmen dieser Arbeit hinzugefügt. Zuvor verwendete das Plugin die sechs von *Xfrog* ausgelieferten Texture-Maps für *V-Ray/Arnold*, welche in einem realen Anwendungsfall nicht vorhanden sind. Die neu implementierte Verwendung von Texture-Maps aus *ShaderMap* oder *Substance 3D Sampler* erstellt die Shader-Netzwerke auf Basis der mit den externen Tools erzeugten Texture-Maps so, dass die Generierung der Texture-Maps in den Tools mit den Standardeinstellungen und somit ohne zusätzliche Anpassungen erfolgen kann. Dies beschleunigt die Erstellung der benötigten Texture-Maps erheblich und sorgt zudem für reproduzierbare Ergebnisse. Die Auswahl eines *Map-Generators* ist nur möglich, wenn die Option *Use own texture maps* aktiviert ist. Je nach gewähltem Map-Generator, werden die zu ladenden Texture-Maps und deren Integration in die entsprechenden Shader-Netzwerke unterschiedlich behandelt.

Mit *Convert Selected* können die im oberen Fensterbereich ausgewählten Materialien konvertiert werden. Mit *Convert All* wird die obere Auswahl ignoriert und es werden alle verfügbaren Materialien des ausgewählten Objekts konvertiert.

Nach Start der Konvertierung läuft das Programm selbstständig ab, es sei denn es sollen eigene Texture-Maps verwendet werden. In diesem Fall öffnet sich der *Texture-Map Selector* (siehe Abbildung 6.2). Dieser dient dem Laden der benötigten und mit der jeweiligen externen Software generierten Texture-Maps. Jede Zeile stellt ein zu konvertierendes Material dar und durch Klicken der grauen Schaltflächen öffnet sich jeweils ein Fenster zum Suchen und Auswählen der benötigten Map. Die ausgewählten Texture-Maps werden in den grauen Schaltflächen als Thumbnails dargestellt, sodass der Nutzer ein optisches Feedback erhält. Die Schaltflächen sind zudem beschriftet, sodass jederzeit eindeutig definiert ist, welche Texture-Map benötigt wird. Der *Texture-Map Selector* wird, je nach getroffener Auswahl der

6. KONZEPT UND ENTWICKLUNG EINES PLUGINS ZUR AUTOMATISCHEN KONVERTIERUNG DER XFROG SHADER-NETZWERKE

zu konvertierenden Materialien, dynamisch erstellt. Mit dem Button *Convert Materials* am unteren Rand des Fensters kann schließlich die Konvertierung gestartet werden. Sollten nicht alle benötigten Texture-Maps ausgewählt worden sein, wird eine entsprechende Warnung angezeigt, welche es dem Nutzer überlässt, die fehlenden Texture-Maps hinzuzufügen oder die Konvertierung dennoch zu starten. In diesem Fall werden dem Shader-Netzwerk leere *File-Nodes* hinzugefügt, welche im Nachhinein mit Texture-Maps bestückt werden können.

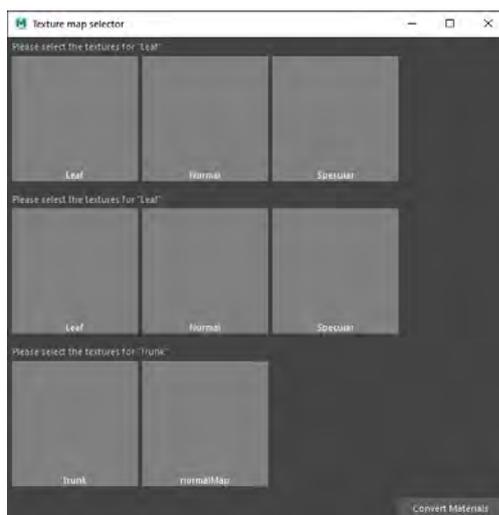


Abbildung 6.2: Darstellung des Texture-Map Selectors des Xfrog-Converter Plugins. Durch Klicken der entsprechenden Buttons können die benötigten Texture-Maps geladen werden
Quelle: Eigene Darstellung

6.3 Konvertierbare Materialien und getestete Software-Versionen

Die Konvertierung der Shader-Netzwerke ist modular aufgebaut, sodass die Programmlogik des Hauptfensters und des *Texture-Map Selectors* von der Logik der Konvertierung der einzelnen Materialien getrennt ist. Auf diese Weise können im Anschluss an diese Arbeit Konvertierungen weiterer Materialtypen, wie Blütenblätter, implementiert werden. Zum Zeitpunkt der Erstellung dieser Arbeit können die Materialien für Blätter (Leaf) und Baumrinde (Bark), jeweils für den Renderer *V-Ray* oder *Arnold*, konvertiert werden. Wie bereits erwähnt, werden von *Xfrog* einige Shader-Netzwerke für mehrere Objekte verwendet, sodass beispielsweise Stämme (Trunk), Zweige (Twig), Wurzeln (Root), und Äste (Branch) mit der Programmlogik der Baumrinden-Konvertierung ebenfalls konvertiert werden können. Somit ergeben sich weitere mögliche Konvertierungen, welche in Tabelle 6.1 aufgelistet sind. Die Konvertierungen, die für die Renderings der Evaluation verwendet werden, werden mit den in Tabelle 6.2 aufgelisteten Software-Versionen durchgeführt. Es ist nicht auszuschließen, dass sich die Funktionalität einzelner verwendeter *MEL*-Funktionen in neueren Software-Versionen ändert.

Materialtyp	V-Ray	Arnold
Leaf	✓	✓
Bark	✓	✓
Trunk	✓	✓
Twig	✓	✓
Roots	✓	✓
Branch	✓	✓

Tabelle 6.1: Auflistung aller zum Zeitpunkt der Erstellung dieser Arbeit mit dem Xfrog-Converter konvertierbaren Materialien

Software	Version	Getestet
Autodesk Maya	2020	✓
V-Ray 5 for Maya	v5.00.22	✓
Arnold for Maya	MtoA 4.1.1	✓

Tabelle 6.2: Auflistung aller zum Zeitpunkt der Erstellung dieser Arbeit verwendeten Software-Versionen

6.4 Die Konvertierung am Beispiel des V-Ray Blatt Shader-Netzwerks

Das Plugin erstellt automatisch Shader-Netzwerke, welche dem Aufbau der entsprechenden *Xfrog* Shader-Netzwerke zum größten Teil entspricht. Je nach verwendeten Einstellungen variieren die erstellten Netzwerke jedoch leicht. Zudem werden bei der Konvertierung entweder automatisch Texture-Maps erstellt oder extern generierte Texture-Maps verwendet. Dies führt ebenfalls zu einer abweichenden Struktur der generierten Shader-Netzwerke im Vergleich zu den Shader-Netzwerken von *Xfrog*.

Es ist zu beachten, dass das Plugin zur Zeit einen Prototypen darstellt, mit welchem die Machbarkeit und die Qualität der Konvertierungen untersucht werden soll. Anhand der Ergebnisse dieser Arbeit können im Anschluss an diese Arbeit Funktionen, Parameter und Einstellungen optimiert werden. Das im Folgenden zur Demonstration der Konvertierungen verwendete *Xfrog*-Modell *Golden Bamboo* liegt für *Autodesk Maya*, *V-Ray* und *Arnold* vor, und eignet sich somit für einen Vergleich der verschiedenen Shader-Netzwerke und den jeweiligen Ergebnissen der Konvertierung. Die Vergleiche werden für das Material von Blatt 1 des Modells durchgeführt, da die Blatt-Materialien die komplexesten Shader-Netzwerke darstellen. Die Betrachtungen gelten analog für die anderen Materialien und den Renderer *Arnold*, auch wenn die Struktur sich zwischen den beiden Renderern deutlich unterscheidet. In allen folgenden Darstellungen der Shader-Netzwerke wurden die zugehörigen Shadinggroups und *place2dTexture-Nodes* aus Gründen der Übersichtlichkeit ausgeblendet.

Es wird beispielhaft die Konvertierung eines *Maya*-Materials in ein *V-Ray* Material be-

6. KONZEPT UND ENTWICKLUNG EINES PLUGINS ZUR AUTOMATISCHEN KONVERTIERUNG DER XFROG SHADER-NETZWERKE

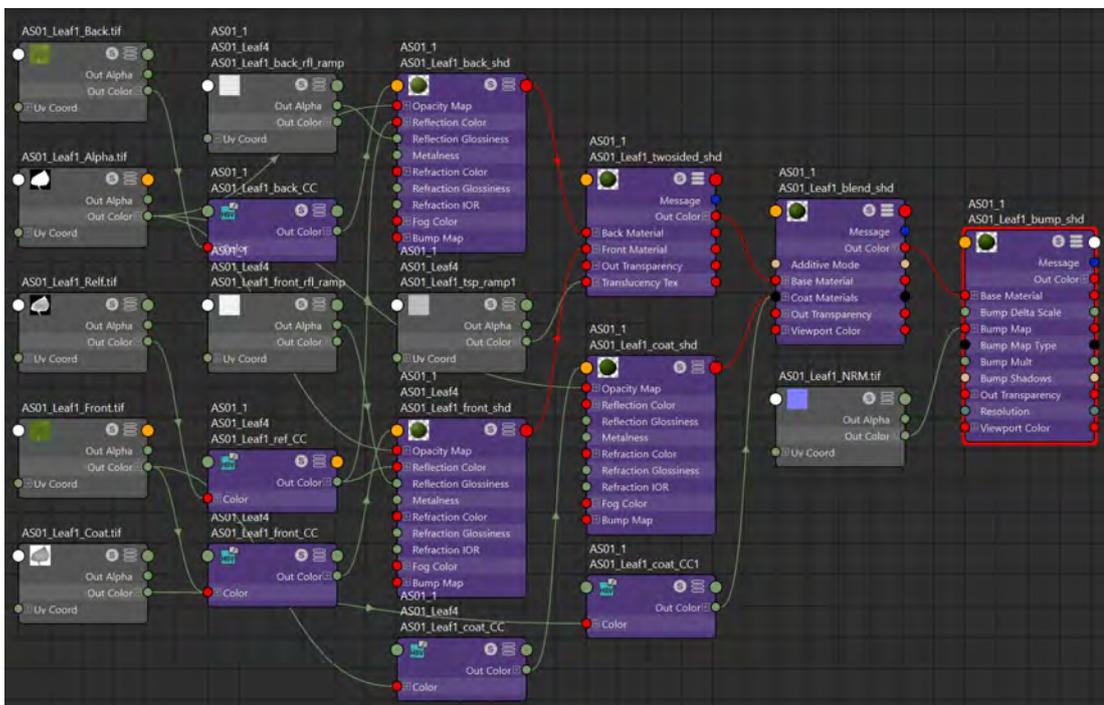


Abbildung 6.3: Darstellung des V-Ray Blatt Shader-Netzwerkes des Modells *Golden Bam-boo* von *Xfrog*

Quelle: Eigene Darstellung

trachtet. Das von *Xfrog* ausgelieferte V-Ray Material ist in Abbildung 6.3 dargestellt. Zum Vergleich ist das *Maya* Material des gleichen Modells in Abbildung 6.4 dargestellt. Werden für die Konvertierung keine extern generierten Texture-Maps verwendet (*Use own texture maps* deaktiviert), erstellt das Plugin anstelle der entsprechenden *File-Nodes* einige *Utility-Nodes*, welche das Aussehen der Maps auf Grundlage des im *Maya*-Material vorhandenen Bildes des Blattes bestmöglich nachbilden. Das Ergebnis dieser Konvertierung ist in Abbildung 6.5 zu sehen. Im Gegensatz zum V-Ray Material von *Xfrog* ist zu erkennen, dass anstelle der fünf Texture-Maps auf der linken Seite und der Normal-Map rechts, lediglich ein einziges Bild des Blattes in Form eines *File-Nodes* verwendet wird. Diese Datei wird bei der Konvertierung rekursiv im *Maya* Material gesucht und extrahiert. Die Suche läuft über alle enthaltenen Nodes, ausgehend vom letzten Node im Shader-Netzwerk. Die fehlenden Texture-Maps werden mit Hilfe der *Utility-Nodes* (Reverse, RemapHSV, ColorCorrect) nachgebildet. Dies stellt die einfachste und schnellste Konvertierung dar, da keine Texture-Maps extern generiert werden müssen. Allerdings kann auf diese Weise keine Normal-Map erzeugt werden, weshalb ein Graustufenbild der vorhandenen Blatt-Textur als Bump-Map verwendet wird. Dieses wird mit Hilfe eines remapHSV-Nodes erzeugt, indem die Sättigung des Bildes auf 0 gesetzt wird.

Sollen für die Konvertierung eigene Texture-Maps verwendet werden, müssen die erforderlichen Maps im *Texture-Map Selector* ausgewählt werden. Das mit Texture-Maps aus

6.4. Die Konvertierung am Beispiel des V-Ray Blatt Shader-Netzwerks

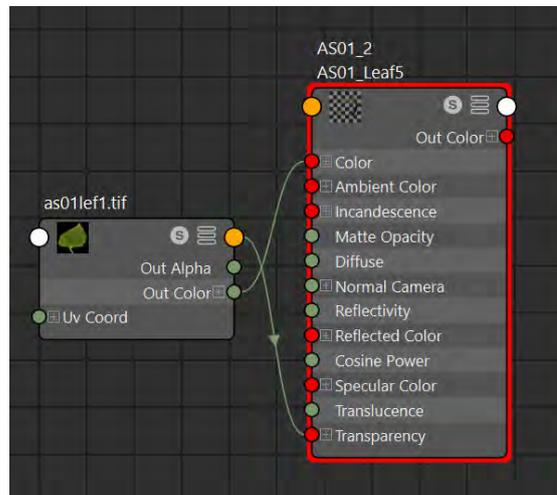


Abbildung 6.4: Darstellung des *Maya* Blatt Shader-Netzwerkes des Modells *Golden Bamboo* von *Xfrog*

Quelle: Eigene Darstellung

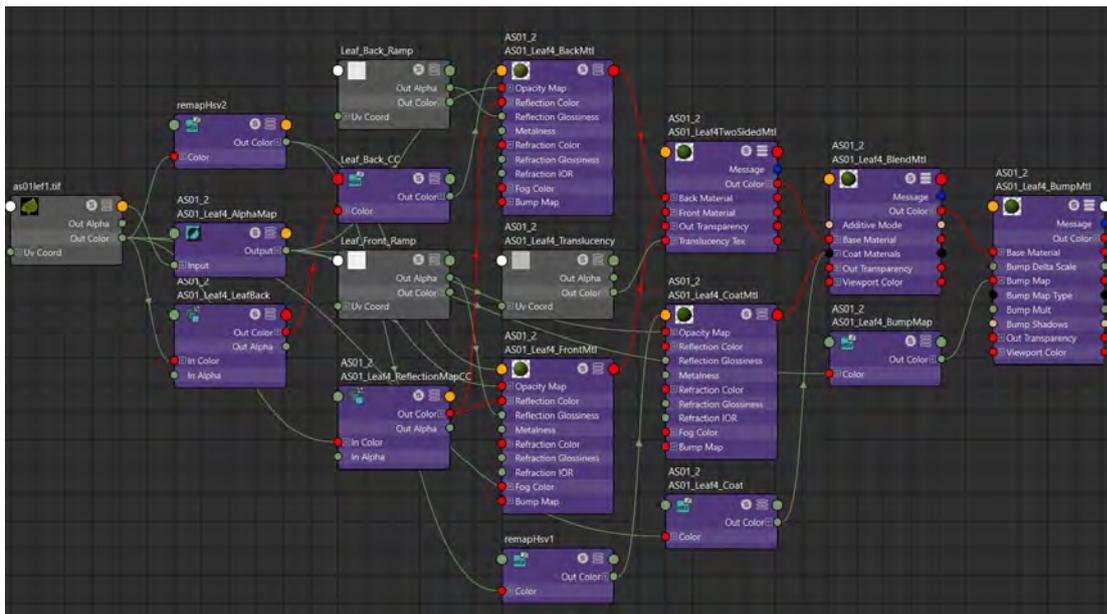


Abbildung 6.5: Darstellung des zu *V-Ray* konvertierten *Maya* Blatt Shader-Netzwerkes des Modells *Golden Bamboo* von *Xfrog*. Bei dieser Konvertierung wurden keine extern generierten Texture-Maps verwendet

Quelle: Eigene Darstellung

ShaderMap konvertierte Blatt-Material ist in Abbildung 6.6 abgebildet. Hier sind die drei mit *ShaderMap* generierten Texturen zu sehen (ganz links im Bild und in der zweiten Spalte von rechts). Die Texturen für *LeafBack*, *Reflection* und die Alphamap werden mit Hilfe von

6. KONZEPT UND ENTWICKLUNG EINES PLUGINS ZUR AUTOMATISCHEN KONVERTIERUNG DER XFROG SHADER-NETZWERKE

Utility-Nodes aus den importierten Maps abgeleitet. Auf diese Weise können die Standardeinstellungen in *ShaderMap* verwendet werden und es müssen keine zusätzlichen Texturen generiert werden. Das Shader-Netzwerk ist bei Verwendung von Texture-Maps aus *Substance 3D Sampler* analog zu diesem Shader-Netzwerk aufgebaut, mit Ausnahme einiger Parameter der *Utility-Nodes*. Da *ShaderMap* zusätzlich zur Normal-Map eine Specular-Map, *Substance 3D Sampler* hingegen eine Roughness-Map erzeugt, müssen die Texturen jeweils unterschiedlich korrigiert werden, um den von *Xfrog* erstellten Texture-Maps zu entsprechen.

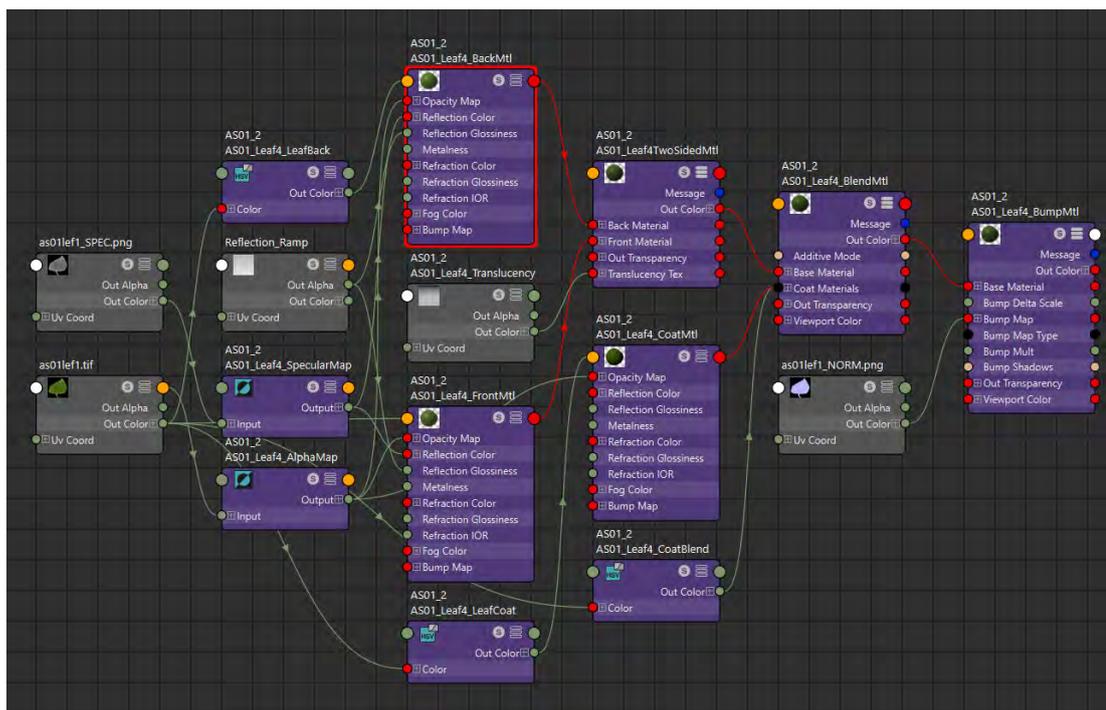


Abbildung 6.6: Darstellung des zu V-Ray konvertierten Maya Blatt Shader-Netzwerkes des Modells *Golden Bamboo* von *Xfrog* mit Texture-Maps aus *ShaderMap*
Quelle: Eigene Darstellung

6.5 Zusammenfassung

Die automatische Konvertierung der *Xfrog* Shader-Netzwerke ist aufgrund der klaren Strukturen grundsätzlich möglich. Allerdings ist es nicht möglich, sämtliche Parameter für jedes individuelle Modell zu berücksichtigen. Somit müssen für die meisten Parameter Mittel- oder Erfahrungswerte verwendet werden und die Konvertierung kann somit nur als Grundlage gesehen werden, welche im Einzelfall manuell angepasst werden muss. Dennoch sind die erstellten Shader-Netzwerke vollständig kompatibel mit dem entsprechenden Renderer und können von diesem korrekt interpretiert werden.

Die nicht vorhandenen Texture-Maps der *Maya*-Modelle müssen extern generiert oder möglichst gut mit Hilfe von *Utility-Nodes* angenähert werden. Dennoch ist von einer deutli-

chen Zeitersparnis gegenüber der manuellen Erzeugung der Shader-Netzwerke auszugehen.

Aufgrund der Komplexität und des Umfangs der Converter-Skripte wird an dieser Stelle auf eine Darstellung des Quellcodes verzichtet.

Im folgenden Kapitel werden Renderings mit den konvertierten Materialien anhand einer Evaluation im Hinblick auf Qualität und Renderdauer untersucht. Hierbei wird neben der Verwendung der *Xfrog*-Materialien und der mit dem Plugin generierten Materialien auch die Verwendung von unkonvertierten *Maya*-Materialien mit den Renderern *V-Ray* und *Arnold* untersucht.

Kapitel 7

Evaluation

7.1 Aufbau der Evaluation

Die Evaluation wird anhand von zwei Umfragen durchgeführt. Die erste Umfrage dient dem Vergleich zwischen Renderings mit unterschiedlichen Materialien und besteht aus 24 zu vergleichenden Bildpaaren. Diese zeigen verschiedene Renderings, welche eine allgemeine Betrachtung bezüglich des Fotorealismus in Alltagssituationen unter homogener Beleuchtung ermöglichen sollen. Die Antworten dieser Umfrage sollen Aufschluss darüber geben, inwiefern sich die Verwendung der Renderer-spezifischen Materialien im Vergleich zu Standard *Maya*-Materialien auf die Qualität auswirkt. Des Weiteren soll mit der Umfrage untersucht werden, ob die Qualität der mit dem entwickelten Plugin konvertierten Materialien mit den Renderer-spezifischen Materialien vergleichbar ist. Die Bilderpaare der ersten Umfrage und deren Parameter sind in Tabelle 7.1 aufgelistet.

Die zweite Umfrage soll anhand von Vergleichen zwischen realen Fotografien und Renderings den substantiellen Unterschied zwischen den Renderer-spezifischen Materialien und den Standard *Maya* Materialien, die Fähigkeit Transluzenz zu simulieren, näher beleuchten. Hierzu wurden zwei Fotografien von Pflanzenblättern mit jeweils zwei Renderings verglichen. Die Fotografien zeigen die Blätter vor einer homogenen Hintergrundbeleuchtung, sodass das Licht, in Richtung des Betrachters, durch die Blätter hindurchdringt. Zu jeder Fotografie wurde zum Vergleich ein *V-Ray* Rendering mit *V-Ray* Materialien und ein *V-Ray* Rendering mit *Maya* Materialien gezeigt. Die Befragten sollten nun entscheiden, welches der beiden Renderings am ehesten dem Erscheinungsbild der Fotografie entspricht.

7.2 Umfang und Zielgruppe

Bei beiden Umfragen fand keine gezielte Auswahl der Befragten statt und es wurden keine personenbezogenen Daten erhoben. An der ersten Befragung nahmen insgesamt 40 Personen, an der zweiten Umfrage 43 Personen teil. Die Umfragen wurden mit *Google Forms* erstellt und die erhaltenen Antworten mit *Microsoft Excel* ausgewertet.

#	Renderer	Mat 1	Mat 2	Maps	Δt 1 [s]	Δt 2 [s]
1	<i>V-Ray</i>	<i>V-Ray</i>	<i>Maya</i>	-	125	79
2	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	-	125	140
3	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	SM	125	132
4	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	SS	125	133
5	<i>Arnold</i>	<i>Arnold</i>	<i>Maya</i>	-	327	37
6	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	-	327	339
7	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	SM	327	304
8	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	SS	327	319
9	<i>V-Ray</i>	<i>V-Ray</i>	<i>Maya</i>	-	250	126
10	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	-	250	227
11	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	SM	250	231
12	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	SS	250	218
13	<i>Arnold</i>	<i>Arnold</i>	<i>Maya</i>	-	669	36
14	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	-	669	637
15	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	SM	669	583
16	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	SS	669	574
17	<i>V-Ray</i>	<i>V-Ray</i>	<i>Maya</i>	-	199	94
18	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	-	199	241
19	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	SM	199	231
20	<i>V-Ray</i>	<i>V-Ray</i>	<i>Converted</i>	SS	199	223
21	<i>Arnold</i>	<i>Arnold</i>	<i>Maya</i>	-	1145	65
22	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	-	1145	1152
23	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	SM	1145	1091
24	<i>Arnold</i>	<i>Arnold</i>	<i>Converted</i>	SS	1145	1086

Tabelle 7.1: Auflistung aller Bildpaare der ersten Umfrage. Die Spalten zeigen (von links nach rechts) die Nummer des Bildpaares, den verwendeten Renderer, die Materialien des jeweils ersten und zweiten Bildes (Mat 1 und Mat 2), die Anwendung, mit welcher die benötigten Texture-Maps erstellt wurden (SM-ShaderMap, SS-Substance 3D Sampler), und die Renderzeiten des jeweils ersten und zweiten Bildes (Δt_1 und Δt_2)

7.3 Statistische Auswertung

Da es sich bei den Fragestellungen der Umfragen ausschließlich um Vergleiche zwischen zwei Bildern handelt, werden die Antworten mittels eines Vergleichs zweier Häufigkeiten ausgewertet [HS18, S. 682]. Um die Nullhypothese H_0 (Gleichheit beider Häufigkeiten) zu bestätigen oder zu widerlegen, wird ein Vertrauensbereich mit Gleichung 7.1 berechnet.

$$\frac{b}{(a+1) \cdot F_1} \leq \frac{\beta}{\alpha} \leq \frac{(b+1) \cdot F_2}{a} \quad (7.1)$$

In der Gleichung beschreibt b den jeweils größeren Wert (höhere Anzahl an Antworten) und a den jeweils niedrigeren Wert (geringere Anzahl an Antworten) der beiden Häufigkeiten. β/α ist das Verhältnis der beiden Häufigkeiten, welche mit b/a geschätzt wird. Die Werte für $F_1 = F_{2a+2;2b}$ und $F_2 = F_{2b+2;2a}$ können einer Tabelle der F-Verteilung entnommen werden. Liegt der Wert 1 nicht innerhalb des Vertrauensbereichs, so muss die Nullhypothese H_0 abgelehnt werden. Im Rahmen dieser Auswertung wurden alle Vertrauensbereiche am 5%-Signifikanzniveau berechnet.

7.4 Vergleich zwischen V-Ray Renderings mit V-Ray Materialien und Maya Materialien

In diesem Kapitel werden die Antworten von Umfrage 1 ausgewertet, welche die *V-Ray* Renderings betreffen. Zunächst werden die Antworten bezüglich jener Bilderpaare betrachtet, die Renderings mit *V-Ray* Materialien und Renderings mit unkonvertierten *Maya*-Materialien zeigen. Diese Betrachtung soll Aufschluss darüber geben, ob, und wenn ja, in welchen Fällen, eine Konvertierung nötig oder sinnvoll ist. Es werden die Antworten der Bilderpaare 1, 9 und 17 ausgewertet. Die entsprechenden Bilderpaare sind in Abbildung 7.1 abgebildet. Die Ergebnisse der Umfrage bezüglich dieser Bilderpaare sind in Tabelle 7.2 dargestellt.

	Bilderpaar	V-Ray bevorzugt	Maya bevorzugt	keine Präferenz
	1	16	9	15
	9	15	13	12
	17	17	18	5
Summe		48	40	32

Tabelle 7.2: Umfrageergebnisse der Bilderpaare 1,9 und 17 von Umfrage 1

48 der Befragten sagten aus, die mit *V-Ray* Materialien gerenderten Bilder sähen realistischer aus, als die mit *Maya* Materialien gerenderten Bilder. 72 der Befragten empfanden die mit *Maya* Materialien gerenderten Bilder als realistischer oder empfanden beide Bilder als ähnlich realistisch.

Hypothese: Die mit *V-Ray* Materialien gerenderten Bilder sehen realistischer aus, als die mit *Maya* Materialien gerenderten Bilder.

7. EVALUATION

Die berechneten Werte ergeben sich zu $1,03 \leq \beta/\alpha \leq 2,21$. Die Nullhypothese H_0 (Gleichheit der Häufigkeiten) wird abgelehnt. Es kann somit, mit einer Irrtumswahrscheinlichkeit von 5%, ein statistisch signifikanter Unterschied zwischen den Antwortgruppen nachgewiesen werden. Die Hypothese, die mit *V-Ray* Materialien gerenderten Bilder sähen realistischer aus, als die mit *Maya* Materialien gerenderten Bilder, wird daher abgelehnt.



(a) Bilderspaar 1



(b) Bilderspaar 9



(c) Bilderspaar 17

Abbildung 7.1: Bilderpaare 1, 9 und 17 von Umfrage 1. In diesen Bilderpaaren werden *V-Ray* Renderings mit *V-Ray* Materialien (jeweils das linke Bild) und *V-Ray* Renderings mit nicht konvertierten *Maya*-Materialien miteinander verglichen (jeweils das rechte Bild)

7.5 Vergleich zwischen *V-Ray* Renderings mit *V-Ray* Materialien und konvertierten *Maya* Materialien mit automatisch generierten Texture-Maps

In diesem Abschnitt werden die Umfrageergebnisse von Umfrage 1 bezüglich der automatisch konvertierten *Maya* Materialien ausgewertet. Die zu untersuchenden Bildpaare entsprechen denen aus Abschnitt 7.4, doch wurden hier die *Maya* Materialien mit dem entwickelten Plugin konvertiert. Hierbei wurden keine extern generierten Texture-Maps verwendet, sie wurden automatisch von dem Plugin generiert. Die relevanten Bildpaare sind in Abbildung 7.2 abgebildet. Die Ergebnisse der Umfrage sind in Tabelle 7.3 aufgelistet.

	Bilderpaar	<i>V-Ray</i> bevorzugt	<i>Maya</i> bevorzugt	keine Präferenz
	2	11	3	26
	10	6	12	22
	18	9	3	28
Summe		26	18	76

Tabelle 7.3: Umfrageergebnisse der Bilderpaare 2, 10 und 18 von Umfrage 1

26 der Befragten sagten aus, die mit *V-Ray* Materialien gerenderten Bilder sähen realistischer aus, als die mit konvertierten *Maya* Materialien gerenderten Bilder. 94 der Befragten empfanden die mit konvertierten *Maya* Materialien gerenderten Bilder als realistischer oder empfanden beide Bilder als ähnlich realistisch.

Hypothese: Die mit konvertierten *Maya* Materialien gerenderten Bilder sehen realistischer oder ähnlich realistisch aus, wie die mit *V-Ray* Materialien gerenderten Bilder.

Die berechneten Werte ergeben sich zu $1,5 \leq \beta/\alpha \leq 1,6$. Die Nullhypothese H_0 (Gleichheit der Häufigkeiten) wird abgelehnt. Es kann somit, mit einer Irrtumswahrscheinlichkeit von 5%, ein statistisch signifikanter Unterschied zwischen den Antwortgruppen nachgewiesen werden. Die Hypothese, die mit konvertierten *Maya* Materialien gerenderten Bilder sähen realistischer oder ähnlich realistisch wie die mit *V-Ray* Materialien gerenderten Bilder aus, wird daher nicht abgelehnt.



(a) Bilderpaar 2



(b) Bilderpaar 10



(c) Bilderpaar 18

Abbildung 7.2: Bilderpaare 2, 10 und 18 von Umfrage 1. In diesen Bilderpaaren werden *V-Ray* Renderings mit *V-Ray*-Materialien (jeweils das linke Bild) und *V-Ray* Renderings mit konvertierten *Maya*-Materialien, ohne extern generierte *Texture-Maps*, miteinander verglichen (jeweils das rechte Bild)

7.6 Vergleich unter Verwendung extern generierter Texture-Maps (Shadermap)

In diesem Abschnitt werden die Umfrageergebnisse von Umfrage 1 bezüglich der automatisch konvertierten *Maya* Materialien ausgewertet. Die zu untersuchenden Bildpaare entsprechen denen aus Abschnitt 7.4, doch wurden hier die *Maya* Materialien mit dem entwickelten Plugin konvertiert. Hierbei wurden die benötigten Texture-Maps mit dem Programm *ShaderMap* generiert. Die relevanten Bildpaare sind in Abbildung 7.3 abgebildet. Die Ergebnisse der Umfrage sind in Tabelle 7.4 aufgelistet.

	Bilderpaar	V-Ray bevorzugt	Maya bevorzugt	keine Präferenz
	3	14	9	17
	11	17	9	14
	19	29	4	7
Summe		60	22	38

Tabelle 7.4: Umfrageergebnisse der Bilderpaare 3, 11 und 19 von Umfrage 1

60 der Befragten sagten aus, die mit *V-Ray* Materialien gerenderten Bilder sähen realistischer aus, als die mit konvertierten *Maya* Materialien gerenderten Bilder. 60 der Befragten empfanden die mit konvertierten *Maya* Materialien gerenderten Bilder als realistischer oder empfanden beide Bilder als ähnlich realistisch.

Hypothese: Die mit konvertierten *Maya* Materialien gerenderten Bilder sehen realistischer oder ähnlich realistisch aus, wie die mit *V-Ray* Materialien gerenderten Bilder.

Die berechneten Werte ergeben sich zu $0,69 \leq \beta/\alpha \leq 1,45$. Die Nullhypothese H_0 (Gleichheit der Häufigkeiten) kann nicht abgelehnt werden. Es kann somit kein statistisch signifikanter Unterschied zwischen den Antwortgruppen nachgewiesen werden.



(a) Bilderpaar 3



(b) Bilderpaar 11



(c) Bilderpaar 19

Abbildung 7.3: Bilderpaare 3, 11 und 19 von Umfrage 1. In diesen Bilderpaaren werden *V-Ray* Renderings mit *V-Ray*-Materialien (jeweils das linke Bild) und *V-Ray* Renderings mit konvertierten *Maya*-Materialien, mit extern generierten Texture-Maps (*ShaderMap*), miteinander verglichen (jeweils das rechte Bild)

7.7 Vergleich unter Verwendung extern generierter Texture-Maps (Substance 3D Sampler)

In diesem Abschnitt werden, analog zum vorigen Abschnitt, die Umfrageergebnisse bezüglich der konvertierten *Maya* Materialien untersucht. Hierbei wurden die Materialien, unter Verwendung von Texture-Maps, welche mit *Substance 3D Sampler* generiert wurden, mit dem Plugin konvertiert. Die relevanten Bildpaare sind in Abbildung 7.4 abgebildet. Die Ergebnisse der Umfrage sind in Tabelle 7.5 aufgelistet.

	Bilderpaar	V-Ray bevorzugt	Maya bevorzugt	keine Präferenz
	4	14	5	21
	12	12	14	14
	20	32	4	4
Summe		58	23	39

Tabelle 7.5: Umfrageergebnisse der Bilderpaare 4, 12 und 20 von Umfrage 1

58 der Befragten sagten aus, die mit *V-Ray* Materialien gerenderten Bilder sähen realistischer aus, als die mit konvertierten *Maya* Materialien gerenderten Bilder. 62 der Befragten empfanden die mit konvertierten *Maya* Materialien gerenderten Bilder als realistischer oder empfanden beide Bilder als ähnlich realistisch.

Hypothese: Die mit konvertierten *Maya* Materialien gerenderten Bilder sehen realistischer oder ähnlich realistisch aus, wie die mit *V-Ray* Materialien gerenderten Bilder.

Die berechneten Werte ergeben sich zu $0,74 \leq \beta/\alpha \leq 1,43$. Die Nullhypothese H_0 (Gleichheit der Häufigkeiten) kann nicht abgelehnt werden. Es kann somit kein statistisch signifikanter Unterschied zwischen den Antwortgruppen nachgewiesen werden.



(a) Bilderpaar 4



(b) Bilderpaar 12



(c) Bilderpaar 20

Abbildung 7.4: Bilderpaare 4, 12 und 20 von Umfrage 1. In diesen Bilderpaaren werden *V-Ray* Renderings mit *V-Ray*-Materialien (jeweils das linke Bild) und *V-Ray* Renderings mit konvertierten *Maya*-Materialien, mit extern generierten Texture-Maps (*Substance 3D Sampler*), miteinander verglichen (jeweils das rechte Bild)

7.8 Vergleiche von Renderings mit dem Renderer *Arnold* und unkonvertierten sowie konvertierten *Maya* Materialien

Die zuvor durchgeführten Untersuchungen wurden zusätzlich analog mit Renderings mit dem Renderer *Arnold* und entsprechend nicht konvertierten oder konvertierten Materialien durchgeführt. Die Ergebnisse werden in Tabelle 7.6 in verkürzter Form dargestellt. Die entsprechenden Bilderpaare sind im Anhang A einzusehen.

Bilderpaare	<i>Arnold</i> bevorzugt	<i>Arnold</i> nicht bevorzugt/keine Präferenz
<i>Arnold</i> / <i>Maya</i>	101	19
<i>Arnold</i> / <i>Konvertiert</i>	41	79
<i>Arnold</i> / <i>Konvertiert SM</i>	32	88
<i>Arnold</i> / <i>Konvertiert SS</i>	31	89

Tabelle 7.6: Umfrageergebnisse bezüglich der Bilderpaare, welche mit dem Renderer *Arnold* gerendert wurden

Für alle Umfrageergebnisse bezüglich des *Arnold*-Renderers konnte mit einer Irrtumswahrscheinlichkeit von 5% ein statistisch signifikanter Unterschied zwischen den Antwortgruppen festgestellt werden. Somit können folgende Hypothesen nicht abgelehnt werden:

- Die mit *Arnold* Materialien von *Xfrog* durchgeführten Renderings sehen realistischer aus, als die mit *Maya* Materialien gerenderten Bilder.
- Konvertierte *Maya* Materialien ohne extern generierte Texture-Maps sehen im Rendering realistischer aus, als Renderings mit *Arnold* Materialien von *Xfrog*.
- Konvertierte *Maya* Materialien mit Texture-Maps aus *ShaderMap* sehen im Rendering realistischer aus, als Renderings mit *Arnold* Materialien von *Xfrog*.
- Konvertierte *Maya* Materialien mit Texture-Maps aus *Substance 3D Sampler* sehen im Rendering realistischer aus, als Renderings mit *Arnold* Materialien von *Xfrog*.

7.9 Vergleich von Fotografien mit V-Ray Renderings mit V-Ray Materialien und Maya Materialien

In diesem Abschnitt werden die Umfrageergebnisse von Umfrage 2 ausgewertet. Den Befragten wurden Fotografien von Pflanzenblättern vor einer homogenen Hintergrundbeleuchtung gezeigt, welche sie mit *V-Ray* Renderings mit *V-Ray* Materialien und mit *V-Ray* Renderings mit *Maya* Materialien vergleichen sollten.

7.9.1 Vergleich 1

Die zu bewertenden Bilder des ersten Vergleichs von Umfrage 2 sind in Abbildung 7.5 abgebildet.

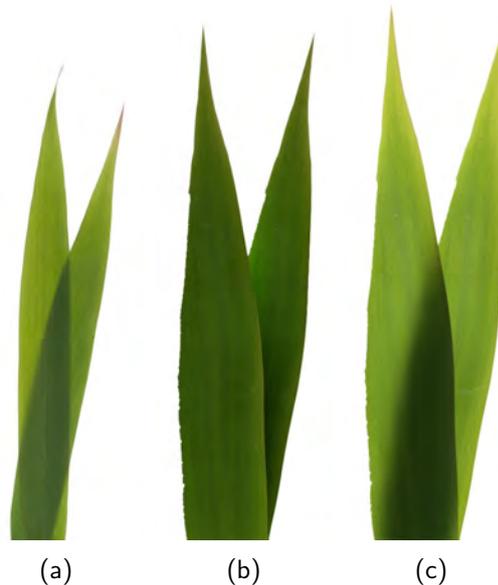


Abbildung 7.5: Darstellung der im ersten Teil von Umfrage 2 verwendeten Blätter. Die Abbildung zeigt eine Fotografie (a), ein *V-Ray* Rendering mit *Maya* Material (b) und ein *V-Ray* Rendering mit *V-Ray* Material (c)

Von den 43 Befragten gaben 36 Personen an, dass das Rendering mit *V-Ray* Materialien der Fotografie am ehesten entspricht. 7 Personen gaben an, das Rendering mit *Maya* Materialien entspräche am ehesten der Fotografie.

Hypothese: Das mit *V-Ray* Materialien gerenderte Bild entspricht am ehesten der Fotografie.

Die berechneten Werte ergeben sich zu $2,26 \leq \beta/\alpha \leq 13,69$. Die Nullhypothese H_0 (Gleichheit der Häufigkeiten) wird abgelehnt. Es besteht mit einer Irrtumswahrscheinlichkeit von

5% ein statistisch signifikanter Unterschied zwischen den beiden Antwortgruppen. Die Hypothese, das mit *V-Ray* Materialien gerenderte Bild entspräche am ehesten der Fotografie, wird nicht abgelehnt.

7.9.2 Vergleich 2

Die zu bewertenden Bilder des zweiten Vergleichs von Umfrage 2 sind in Abbildung 7.6 abgebildet.

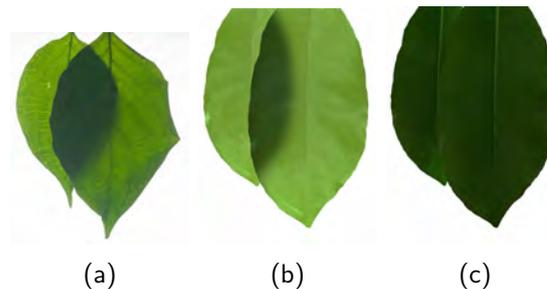


Abbildung 7.6: Darstellung der im zweiten Teil von Umfrage 2 verwendeten Blätter. Die Abbildung zeigt eine Fotografie (a), ein *V-Ray* Rendering mit *V-Ray* Material (b) und ein *V-Ray* Rendering mit *Maya* Material (c)

Von den 43 Befragten gaben 32 Personen an, dass das Rendering mit *V-Ray* Materialien der Fotografie am ehesten entspricht. 11 Personen gaben an, das Rendering mit *Maya* Materialien entspräche am ehesten der Fotografie.

Hypothese: Das mit *V-Ray* Materialien gerenderte Bild entspricht am ehesten der Fotografie.

Die berechneten Werte ergeben sich zu $1,43 \leq \beta/\alpha \leq 6,4$. Die Nullhypothese H_0 (Gleichheit der Häufigkeiten) wird abgelehnt. Es besteht mit einer Irrtumswahrscheinlichkeit von 5% ein statistisch signifikanter Unterschied zwischen den beiden Antwortgruppen. Die Hypothese, das mit *V-Ray* Materialien gerenderte Bild entspräche am ehesten der Fotografie, wird nicht abgelehnt.

7.10 Zusammenfassung

In Abschnitt 7.4 wurde untersucht, wie die Befragten *V-Ray* Renderings mit *V-Ray* Materialien im Vergleich zu *V-Ray* Renderings mit *Maya* Materialien bezüglich des Fotorealismus bewerten. Es konnte gezeigt werden, dass die Befragten die Renderings mit *Maya* Materialien für realistischer halten. Dies ist ein interessantes Ergebnis, da die *Maya* Shader-Netzwerke, im Gegensatz zu den *V-Ray* Shader-Netzwerken, nicht in der Lage sind, das Durchdringen von Licht durch die Blätter zu simulieren. Es ist zu vermuten, dass die homogene Beleuchtung der gewählten Szenen den Effekt der Transluzenz minimiert und somit weniger auffällig

erscheinen lässt. Zudem scheint den Befragten der Fakt, dass Pflanzenblätter zu einem gewissen Teil lichtdurchlässig sind, nicht bewusst zu sein.

In Abschnitt 7.9 wurde aus diesem Grund dieser Aspekt mit Hilfe von Umfrage 2 gezielt untersucht. Hierbei sollten die Befragten *V-Ray* Renderings mit *V-Ray* Materialien und *Maya* Materialien mit vorgegebenen Fotografien, welche sehr deutlich die Transluzenz der Blätter zeigen, vergleichen. Hierbei konnte gezeigt werden, dass die Befragten sich recht eindeutig für die Renderings mit *V-Ray* Materialien entschieden haben. Die Befragten konnten zudem in Form eines Freitextes Gründe angeben, die ihre Entscheidung beeinflusst haben. Hierbei wurde von fast allen Probanden angegeben, dass die fehlende Lichtdurchlässigkeit der ausschlaggebende Punkt für ihre Entscheidung war. Das Fehlen der Transluzenz unter Verwendung von *Maya* Materialien scheint im direkten Vergleich mit Fotografien also sehr deutlich aufzufallen. Somit ist, wie erwartet, die Verwendung von *V-Ray* Materialien bei Verwendung des *V-Ray* Renderers eindeutig vorzuziehen.

In Abschnitt 7.5 wurde untersucht, ob die mit dem Plugin automatisch konvertierten *Maya* Materialien im Vergleich zu den *V-Ray* Materialien, in Bezug auf den Fotorealismus der Renderings, vergleichbare Qualität liefern. Hierbei wurden keine extern generierten Texture-Maps verwendet, das Plugin hat diese automatisch generiert. Es konnte gezeigt werden, dass die Befragten keinen qualitativen Unterschied zwischen den verwendeten Materialien ausmachen konnten. Somit eignet sich diese Art der Konvertierung für die Verwendung von *Xfrog* Modellen, für welche keine *V-Ray* Materialien und Texturen vorliegen.

In Abschnitt 7.6 und 7.7 wurden die Umfrageergebnisse bezüglich der konvertierten Materialien mit extern generierten Texture-Maps untersucht. Bei beiden Auswertungen konnte kein statistisch signifikanter Unterschied zwischen den Antwortgruppen festgestellt werden. Eine sichere Aussage über die Qualität der mit den konvertierten Materialien erzeugten Renderings kann somit nicht getätigt werden. Eine Vermutung, warum die Probanden eine solch geteilte Meinung zu den Renderings haben, liegt in der derzeit noch unausgereiften Implementierung der Einbindung von extern generierten Texture-Maps in die automatische Konvertierung. Das Plugin soll daher im Anschluss an diese Arbeit mit den gewonnenen Informationen weiter optimiert werden. Weitere Ausführungen diesbezüglich werden in Kapitel 8 gegeben.

Abschnitt 7.8 zeigt die Auswertungen der Umfrageergebnisse bezüglich des Renderers *Arnold*. Diese sind in diesem Kapitel in verkürzter Form dargestellt, da sie im Vergleich zu den *V-Ray* spezifischen Untersuchungen nur beschränkt aussagekräftig sind. Zum einen wurden die erstellten Renderings ohne zusätzliches Modifizieren der Shader-Netzwerke durchgeführt, um reproduzierbare Ergebnisse zu erhalten. Dies führt in den *Arnold*-Renderings mit *Maya* Materialien zu fehlerhaften Darstellungen bezüglich der Reflexionen und des Alphakanals (s. Kapitel 5.3). Die Polygone der Blätter werden somit außerhalb des Bereichs des Blattes schwarz dargestellt und die Blätter glänzen sehr stark. In der Umfrage führte dies erwartungsgemäß zu einem deutlich unrealistischeren Eindruck. Zum anderen stammen die *Xfrog*-Modelle mit *Arnold* Materialien aus einer inoffiziellen Testversion. Die Modelle werden zum jetzigen Zeitpunkt nicht kommerziell angeboten. Das heißt letztendlich, dass der Vergleich von *Arnold* Materialien und konvertierten *Maya* Materialien nicht aussagekräftig ist. Dennoch generiert das Plugin verwendbare *Arnold* Shader-Netzwerke, welche ohne das Plugin manuell erstellt werden müssten, da zur Zeit keine Pflanzenmodelle mit *Arnold* Materialien

von *Xfrog* angeboten werden.

Die Untersuchungen bezüglich der Transluzenz der Blätter in Abschnitt 7.9 haben gezeigt, dass diese die Entscheidungen bezüglich des Fotorealismus deutlich beeinträchtigt. Da die *Arnold* Shader-Netzwerke, wie auch die *V-Ray* Shader-Netzwerke, in der Lage sind, das Durchdringen von Licht durch die Blätter zu simulieren, ist zu vermuten, dass das Plugin qualitativ gleichwertige Ergebnisse beim Konvertieren von *Maya* Materialien für den Renderer *Arnold* erzielen kann.

Die Renderzeiten unterscheiden sich zwischen den verwendeten Materialien kaum, mit Ausnahme der unkonvertierten *Maya* Materialien, da diese keine rechenintensiven Effekte, wie Transluzenz, beinhalten. Die Renderzeiten der mit dem Plugin konvertierten Materialien waren im Vergleich zu den *V-Ray* Materialien von *Xfrog* bis zu 6% länger. Die Renderzeiten der zu *Arnold* konvertierten Materialien waren bis zu 8% kürzer, als die *Arnold* Materialien von *Xfrog*.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

In dieser Arbeit wurde die automatische Konvertierung der Materialien von 3D-Pflanzenmodellen des Anbieters *Xfrog* untersucht. Einige der angebotenen Modelle sind für die Verwendung mit dem Renderer *V-Ray* von *Chaosgroup* vorbereitet, allerdings bei Weitem nicht alle angebotenen Modelle. Für den Renderer *Arnold* werden, zum Zeitpunkt der Erstellung dieser Arbeit, keine entsprechend vorbereiteten Modelle von *Xfrog* angeboten. Möchte der Nutzer allerdings ein Pflanzenmodell mit *V-Ray* oder *Arnold* verwenden, für welches keine entsprechend vorbereiteten Materialien vorliegen, so muss er die Shader-Netzwerke manuell erstellen, was nicht nur zeitaufwändig sondern auch fehleranfällig ist. Das entwickelte Plugin *XfrogConverter* dient der einfachen Konvertierung dieser Shader-Netzwerke und soll eine Zeitersparnis sowie reproduzierbare Ergebnisse ermöglichen. In dieser Arbeit wurde die Machbarkeit der automatischen Konvertierung, sowie die Qualität der Konvertierungen untersucht.

Zunächst wurden die Shader-Netzwerke und die Parameter der *V-Ray* und *Arnold* Materialien von *Xfrog* detailliert untersucht. Einige der Erkenntnisse dieser Untersuchungen wurden im Rahmen dieser Arbeit bereits zur Anpassung und Optimierung des Plugins verwendet. Anschließend wurden Renderings erstellt, welche Materialien von *Xfrog* und (mit dem Plugin) konvertierte Materialien verwendeten. Diese wurden in zwei Umfragen hinsichtlich des Fotorealismus verglichen.

Die Auswertung der Umfrageergebnisse lieferte zunächst ein sehr interessantes Ergebnis. Für die Mehrheit der Befragten erschienen die *V-Ray* Renderings mit Standard *Maya* Materialien realistischer als die *V-Ray* Renderings mit den von *Xfrog* ausgelieferten *V-Ray* Materialien. Hierbei ist zu vermuten, dass den Befragten die Eigenschaft von Pflanzenblättern, zu einem gewissen Grad lichtdurchlässig zu sein, nicht allgemein bewusst ist. In einer weiteren Umfrage wurden daher Renderings mit echten Fotografien verglichen, wobei die Mehrheit der Befragten jene Renderings für am ehesten der Fotografie entsprechend empfanden, welche die Lichtdurchlässigkeit durch die Pflanzenblätter zeigten. Im direkten Vergleich mit Fotografien, war das hauptsächliche Entscheidungskriterium also die Transluzenz der Pflanzenblätter. Da Standard *Maya* Materialien keine Transluzenz abbilden können,

ist die Verwendung von *V-Ray* Materialien mit dem Renderer *V-Ray*, sowie entsprechend die Verwendung von *Arnold* Materialien für den Renderer *Arnold*, sinnvoll und empfehlenswert. Dies ist, wie bereits erwähnt, nur möglich, insofern die entsprechenden Materialien für das gewünschte 3D-Pflanzenmodell verfügbar sind.

In der weiteren Auswertung wurde die Qualität von Renderings mit konvertierten Materialien betrachtet. Hier konnte gezeigt werden, dass die automatische Konvertierung von *Maya* Materialien mit dem *XfrogConverter*, ohne die Verwendung extern generierter Texture-Maps, qualitativ gleichwertige Ergebnisse im Vergleich zur Verwendung der Renderer-spezifischen Materialien ermöglichen kann. Das Rendern der Szenen mit den konvertierten Materialien hat im Vergleich zum Rendern der Szenen mit den Materialien von *Xfrog* lediglich bis zu 6% länger gedauert. Die Verwendung dieser Art der Konvertierung mit Hilfe des Plugins stellt demnach eine Möglichkeit dar, Pflanzenmodelle mit gleichwertiger Qualität zu rendern, welche nicht für die Verwendung mit dem Renderer *V-Ray* vorbereitet sind.

Im Gegensatz zur Konvertierung ohne Verwendung extern generierter Texture-Maps, konnte die Auswertung der Umfrage bezüglich der konvertierten Materialien mit extern generierten Texture-Maps (ShaderMap und Substance 3D Sampler) keine statistisch gesicherten Aussagen liefern. Die Meinungen bezüglich der Qualität der Renderings waren sehr uneindeutig und es kann somit keine Aussage getroffen werden, inwiefern die Qualität der konvertierten Materialien zu bewerten ist.

Die Auswertungen bezüglich der Verwendung des Renderers *Arnold* sind kaum aussagekräftig, da die konvertierten Materialien, wie auch die konvertierten Materialien unter Verwendung von *V-Ray*, ohne weitere manuelle Bearbeitung für das Rendern verwendet wurden. Im Falle der *Arnold* Materialien entstehen auf diese Weise allerdings einige Probleme, welche zu einer eindeutig schlechteren Bewertung der Bildqualität durch die Befragten führten. Allerdings war diese Herangehensweise notwendig, um reproduzierbare und vergleichbare Ergebnisse zu erhalten. Da *Xfrog* zur Zeit ohnehin keine Pflanzenmodelle mit *Arnold* Materialien zum Kauf anbietet, fällt dieser Umstand allerdings wenig ins Gewicht. Die Konvertierung von *Maya* Materialien zu *Arnold* Materialien mit Hilfe des Plugins generiert dennoch vollständige und von *Arnold* korrekt interpretierbare Shader-Netzwerke.

Dies führt zu dem wichtigen Hinweis, dass die vom Plugin erstellten Shader-Netzwerke lediglich als funktionale Grundlagen gesehen werden können. Die vielseitigen Parameter der einzelnen Bestandteile wurden anhand von Mittelwerten und Erfahrungswerten ausgelegt und können somit unmöglich für alle existierenden Pflanzenmodelle von *Xfrog* korrekte Ergebnisse liefern. Dennoch ist von einer deutlichen Zeitersparnis im Gegensatz zum manuellen Erzeugen der Shader-Netzwerke auszugehen.

Zusammenfassend konnte gezeigt werden, dass eine automatische Konvertierung der komplexen Shader-Netzwerke möglich ist und die konvertierten Materialien zudem qualitativ gleichwertige Ergebnisse erzeugen können.

8.2 Ausblick

In dieser Arbeit konnte gezeigt werden, dass eine automatische Konvertierung von *Xfrog* Shader-Netzwerken möglich ist und diese qualitativ vergleichbare Ergebnisse liefern kann. Dennoch

sind im Rahmen dieser Arbeit einige Faktoren deutlich geworden, die eine weitere Optimierung des Plugins notwendig machen.

Die automatische Konvertierung, unter Verwendung von extern generierten Texture-Maps, liefert zum jetzigen Zeitpunkt keine ausreichende Qualität, wie die Umfrage zeigen konnte. Wie bereits erwähnt, wurde diese Funktion im Rahmen dieser Arbeit im Plugin implementiert und zudem so ausgelegt, dass kompatible Drittsoftware (*ShaderMap* und *Substance 3D Sampler*) ohne Anpassung von Parametern, durch einfaches Einfügen der vorhandenen Texture-Map, generiert werden können. Dieser Ansatz dient zunächst der einfachen Handhabung und der Möglichkeit, reproduzierbare Ergebnisse im Sinne einer Vergleichbarkeit von Renderings erhalten zu können. Hier ist eine Optimierung, sowohl im Hinblick auf Software-spezifische Einstellungen, als auch in Bezug auf die Einbindung der erhaltenen Texture-Maps in die vom Plugin generierten Shader-Netzwerke, sinnvoll. Allerdings muss auch hier beachtet werden, dass die automatische Konvertierung keinesfalls alle Pflanzenarten und angebotenen Modelle ausreichend berücksichtigen kann. Beispielsweise können die Reflexionseigenschaften von Pflanzenblättern zwischen verschiedenen Pflanzenarten durchaus variieren, was bei Verwendung des Plugins im Nachgang an die Konvertierung manuell angepasst werden muss. Hier wäre eine weitere, benutzerfreundliche Bedienoberfläche denkbar, welche eine Anpassung der wichtigsten Parameter der Shader-Netzwerke nach der Konvertierung erlaubt. Gleiches gilt für die Konvertierung ohne extern generierte Texture-Maps. Zudem wäre eine Funktion denkbar, welche die mühsame Auswahl der benötigten Texture-Maps im Plugin automatisiert. Beispielsweise durch Auswahl eines Ordners, welcher alle relevanten Texture-Maps enthält. Auf diese Weise könnte die Fehleranfälligkeit weiter reduziert und die Benutzerfreundlichkeit, sowie die Zeitersparnis, weiter gesteigert werden.

Allgemein wäre eine aktive Aktualisierung und Optimierung des Plugins durch die Nutzer wünschenswert. Aus diesem Grund wird das Plugin über ein GitLab-Repository frei zugänglich sein, sodass Änderungsvorschläge vorgetragen und eingepflegt werden können. Dies richtet sich primär an die Studierenden der Technischen Hochschule Mittelhessen, die *Xfrog*-Modelle für ihre Projekte verwenden und durch die Verwendung des Plugins die Zeit, die für manuelle Konvertierungen eingespart werden kann, für die kreative Auslegung ihrer Projekte verwenden wollen. Es ist zudem davon auszugehen, dass in Zukunft Aktualisierungen am Plugin bezüglich der verwendeten Software (*Maya*, *V-Ray*, *Arnold*, etc.) nötig sein werden.

Abschließend ist zu sagen, dass diese als Grundlage dienende Arbeit und die daraus gewonnenen Informationen genutzt werden können, um eine weitere Optimierung der automatischen Konvertierung voranzutreiben. Auch könnten weitere Umfragen bezüglich der Qualität der erzeugten Materialien durchgeführt werden, um die hier erhaltenen Informationen mit Daten zu Nutzergruppen, Anwendungsszenarien und persönlichen Erfahrungen zu erweitern. Auf diese Weise könnte das Plugin weiter ausgebaut und besser an individuelle Bedürfnisse angepasst werden.

Anhang A

Bilderpaare von Umfrage 1 unter Verwendung des Renderers Arnold



(a) Bilderpaar 5



(b) Bilderpaar 13



(c) Bilderpaar 21

Abbildung A.1: Bilderpaare 5, 13 und 21 von Umfrage 1. In diesen Bilderpaaren werden *Arnold* Renderings mit *Arnold* Materialien (jeweils das linke Bild) und *Arnold* Renderings mit nicht konvertierten *Maya*-Materialien miteinander verglichen (jeweils das rechte Bild)



(a) Bilderspaar 6



(b) Bilderspaar 14



(c) Bilderspaar 22

Abbildung A.2: Bilderpaare 6, 14 und 22 von Umfrage 1. In diesen Bilderpaaren werden *Arnold* Renderings mit *Arnold* Materialien (jeweils das linke Bild) und *Arnold* Renderings mit konvertierten *Maya*-Materialien, ohne extern generierte Texture-Maps, miteinander verglichen (jeweils das rechte Bild)



(a) Bilderspaar 7



(b) Bilderspaar 15



(c) Bilderspaar 23

Abbildung A.3: Bilderpaare 7, 15 und 23 von Umfrage 1. In diesen Bilderpaaren werden *Arnold* Renderings mit *Arnold* Materialien (jeweils das linke Bild) und *Arnold* Renderings mit konvertierten *Maya*-Materialien, mit extern generierten Texture-Maps (*ShaderMap*), miteinander verglichen (jeweils das rechte Bild)



(a) Bilderspaar 8



(b) Bilderspaar 16



(c) Bilderspaar 24

Abbildung A.4: Bilderpaare 8, 16 und 24 von Umfrage 1. In diesen Bilderpaaren werden *Arnold* Renderings mit *Arnold* Materialien (jeweils das linke Bild) und *Arnold* Renderings mit konvertierten *Maya*-Materialien, mit extern generierten Texture-Maps (*Substance 3D Sampler*), miteinander verglichen (jeweils das rechte Bild)

Literaturverzeichnis

- [Aca21a] ACADEMYSOFTWAREFOUNDATION: *OpenShadingLanguage*. <https://github.com/AcademySoftwareFoundation/OpenShadingLanguage>. Version: Juni 2021. – original-date: 2011-03-10T02:44:05Z
- [Aca21b] ACADEMYSOFTWAREFOUNDATION: *OSL Specification*. <https://github.com/AcademySoftwareFoundation/OpenShadingLanguage/blob/7f8a1a1702ffe3f7cfa11f78c9f4d4e858313d18/src/doc/osl-languagespec.pdf>. Version: Juni 2021. – original-date: 2011-03-10T02:44:05Z
- [Cor21] CORPORATION, NVIDIA: *Material Definition Language from NVIDIA*. <https://www.nvidia.com/en-us/design-visualization/technologies/material-definition-language/>. Version: Mai 2021
- [Deu03] DEUSSEN, Oliver: *Computergenerierte Pflanzen: Technik und Design digitaler Pflanzenwelten*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003 (X.media.press). <http://dx.doi.org/10.1007/978-3-642-55822-1>. <http://dx.doi.org/10.1007/978-3-642-55822-1>. – ISBN 978–3–642–62835–1 978–3–642–55822–1
- [DL97] DEUSSEN, Oliver ; LINTERMANN, Bernd: Erzeugung komplexer botanischer Objekte in der Computergraphik. In: *Informatik-Spektrum* 20 (1997), August, Nr. 4, 208–215. <http://dx.doi.org/10.1007/s002870050067>. – DOI 10.1007/s002870050067. – ISSN 0170–6012, 1432–122X
- [DL01] DEUSSEN, Oliver ; LINTERMANN, Bernd: Computerpflanzen. In: *Spektrum der Wissenschaft* (2001), Nr. 2, S. 58–65
- [FD03] FRANZKE, Oliver ; DEUSSEN, Oliver: Accurate graphical representation of plant leaves. In: *Plant Growth Modeling, Simulation, Visualization, and their Applications (PMA)*, 2003, 9
- [Geb03] GEBHARDT, Nikolaus: *Einige BRDF Modelle*. <http://www.irrlicht3d.org/papers/BrdfModelle.pdf>. Version: 2003
- [GPA⁺] GEORGIEV, Iliyan ; PORTSMOUTH, Jamie ; ANDERSSON, Zap ; HERUBEL, Adrien ; KING, Alan ; OGAKI, Shinji ; SERVANT, Frederic: *Autodesk Standard Surface*. <https://autodesk.github.io/standard-surface/>

- [GSKC10] GRITZ, Larry ; STEIN, Clifford ; KULLA, Chris ; CONTY, Alejandro: Open Shading Language. In: *ACM SIGGRAPH 2010 Talks on - SIGGRAPH '10*. Los Angeles, California : ACM Press, 2010. – ISBN 978–1–4503–0394–1, 1
- [HS18] HEDDERICH, Jürgen ; SACHS, Lothar: *Angewandte Statistik: Methodensammlung mit R*. 16., überarbeitete und erweiterte Auflage. Berlin : Springer Spektrum, 2018. <http://dx.doi.org/10.1007/978-3-662-56657-2>. <http://dx.doi.org/10.1007/978-3-662-56657-2>. – ISBN 978–3–662–56656–5
- [JMLH01] JENSEN, Henrik W. ; MARSCHNER, Stephen R. ; LEVOY, Marc ; HANRAHAN, Pat: A practical model for subsurface light transport. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*, ACM Press, 2001. – ISBN 978–1–58113–374–5, 511–518
- [KS21] KWOK, Bernard ; STONE, Jonathan: *MaterialX - ShaderGeneration*. <https://github.com/materialx/MaterialX/blob/main/documents/DeveloperGuide/ShaderGeneration.md>. Version: Juni 2021. – original-date: 2017-07-05T22:18:25Z
- [LE19] LLAMAS, Ignacio ; EISENSTEIN, Maksim: REAL-TIME RAY TRACING WITH MDL. (2019), März, 42. <https://developer.download.nvidia.com/video/gputechconf/gtc/2019/presentation/s9908-real-time-ray-tracing-with-mdl-materials-v2.pdf>
- [Lin68] LINDENMAYER, Aristid: Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. In: *Journal of Theoretical Biology* 18 (1968), Nr. 3, 280–299. [http://dx.doi.org/https://doi.org/10.1016/0022-5193\(68\)90079-9](http://dx.doi.org/https://doi.org/10.1016/0022-5193(68)90079-9). – DOI [https://doi.org/10.1016/0022-5193\(68\)90079-9](https://doi.org/10.1016/0022-5193(68)90079-9). – ISSN 0022–5193
- [med19] *Open Source at Autodesk: MaterialX (featuring Lucasfilm and Allegorithmic)*. <https://vimeo.com/355165002>. Version: August 2019
- [Pap16] PAPULA, Lothar: *Mathematik für Ingenieure und Naturwissenschaftler. Band 3: Vektoranalysis, Wahrscheinlichkeitsrechnung, mathematische Statistik, Fehler- und Ausgleichsrechnung: mit 550 Abbildungen, zahlreichen Beispielen aus Naturwissenschaft und Technik sowie 295 Übungsaufgaben mit ausführlichen Lösungen*. 7., überarbeitete und erweiterte Auflage. 2016 (Lehrbuch). – ISBN 978–3–658–11923–2
- [Tea03] TEAM, Maya D.: Mel and Expressions. (2003), 312. https://courses.cs.washington.edu/courses/cse459/06wi/help/mayaguide/Complete/MEL_Expressions.pdf