

Spezifikation

OMATS

OFML-kompatible Materialien*

Version 2.0

Status: Release

Thomas Gerth, EasternGraphics GmbH (Editor)

24. Juni 2019

Inhaltsverzeichnis

1	Einleitung	2
2	Die Materialmodelle	2
2.1	Übersicht	2
2.2	Verwendete Datentypen	3
2.3	Materialtypen	4
2.4	Die Materialparameter	5
3	Textur-Mapping-Verfahren	8
3.1	Ebenen-Mapping	8
3.2	Quader-Mapping	9
3.3	Texturkoordinaten	9
4	OFML-Datenformat für Materialien	10
A	Einführung in das physikalische Rendering (PBR)	13
B	Konvertierung von alten Materialien auf das neue Modell	14
C	Historie	15

Literatur

[jfff] JPEG File Interchange Format, Version 1.02
World Wide Web Consortium (W3C)
(www.w3.org/Graphics/JPEG/jfif3.pdf)

[odb] ODB – OFML-Datenbank (OFML Part I), Version 2.1, 1. überarbeitete Auflage.
Industrieverband Büro und Arbeitswelt e.V. (IBA)

[ofml] OFML – Standardisiertes Datenbeschreibungsformat der Büromöbelindustrie, Version 2.0, 3. überarbeitete Auflage
Industrieverband Büro und Arbeitswelt e.V. (IBA)

[png] Portable Network Graphics (PNG) Specification, Version 1.2
PNG Development Group
(www.libpng.org/pub/png/spec/1.2/png-1.2-pdg.html)

1 Einleitung

Diese Spezifikation definiert zwei Modelle zur Beschreibung von Materialien, welche in OFML-basierten Applikationen zur Darstellung von Objektoberflächen (Materialien) zur Anwendung kommen, sowohl im Echtzeit- als auch im photorealistischen Bereich.

Desweiteren werden in dieser Spezifikation die unterstützten Textur-Mapping-Verfahren sowie die Abbildung der abstrakten Modellparameter in OFML-Materialdefinitionsdateien beschrieben.

Die beiden Materialmodelle werden als OMATS1 und OMATS2 referenziert. OMATS1 ist dabei das ältere Modell. In neueren OFML-Applikationen wird es durch das neue Modell OMATS2 abgelöst, welches auf dem Konzept des *physikalischen Renderings* (PBR) basiert. Dieses ermöglicht bei einer kompakteren Materialbeschreibung eine realistischere und ansprechendere Darstellung im Echtzeitmodus (mehr Informationen dazu s. Anh. A). Zudem bieten Materialeditoren, die auf diesem Modell basieren, eine bessere Benutzerfreundlichkeit, da weniger Parameter eingestellt werden müssen und weniger Abhängigkeiten zwischen den Parametern existieren.

Zur Gewährleistung einer abwärtskompatiblen Verarbeitung von Materialien gelten folgende Bestimmungen für Applikationen, die das neue Modell verwenden:

- Materialien, die auf der Basis von OMATS1 angelegt worden sind, werden automatisch auf das neue Modell konvertiert (s.a. Anh. B).
- Beim Export einer OFML-Materialdefinitionsdatei werden die fehlenden Parameter für die Verarbeitung nach OMATS1 — umgekehrt zum vorigen Punkt — aus den Parametern für OMATS2 abgeleitet (und mit exportiert).

2 Die Materialmodelle

2.1 Übersicht

Jedes Model definiert ein Set von Parametern zur Beschreibung von spezifischen Eigenschaften eines Materials.

Eine besondere Rolle spielt dabei der Parameter *Material_Type* (s. 2.3): in Abhängigkeit vom gewählten Materialtyp werden beim Rendering nur bestimmte Materialparameter verwendet.

Die folgende Tabelle gibt einen Überblick über die definierten Materialparameter (in alphabetischer Reihenfolge ihrer Bezeichner) und deren Zuordnung zu den Materialtypen und den beiden Modellen.

Alle Parameter sind optional, d.h. müssen in einer Materialbeschreibung nicht zwingend angeführt werden¹.

Fehlt eine Parameterangabe, wird bei Parametern für das Modell OMATS1 ein vordefinierter Wert verwendet. Dieser wird unten bei der Beschreibung der einzelnen Parameter spezifiziert.

Für Parameter, die nur für das Modell OMATS2 definiert sind, ist kein Wert vordefiniert. Vielmehr wird bei fehlender Parameterangabe der Wert aus OMATS1-Parametern (bzw. deren Default-Werten) abgeleitet².

¹ Theoretisch sind damit auch leere Materialbeschreibungen möglich.

² Das Verfahren für diese Ableitung ist nicht definiert und kann also von Anwendung zu Anwendung variieren.

Parameter	Materialtyp			Modell	
	Common	Glass	Illuminant	OMATS1	OMATS2
Base_Color	X	X		X	X
Base_Color_Map	X	X		X	X
Emissive_Color			X	X	X
Luminance			X	X	X
Metallness	X				X
Metallness_Map	X				X
Normal_Map	X	X		X	X
Refractive_Index		X		X	X
Roughness	X	X			X
Roughness_Map	X	X			X
Shininess	X			X	
Sound_Absorption	X	X	X	X	X
Specular_Color	X			X	
Specular_Factor	X			X	
Transparency	X	X	X	X	X

2.2 Verwendete Datentypen

Bei der Beschreibung der Parameter werden folgende Datentypen verwendet:

PI positive Ganzzahl

FP allgemeine Gleitkommazahl

RGB Vektor aus drei Farbwerten zur Repräsentation der Basisfarben Rot, Grün und Blau
Jeder Farbwert C muss im Bereich $0.0 \leq C \leq 1.0$ liegen.

RGB-IMAGE Bilddatei

Dieser Datentyp beschreibt zweidimensionale Bilddateien, bestehend aus RGB-Farbwerten.

Erlaubt sind folgende Formate: PNG, JPEG

Die Dimensionen der Bilddateien sollen Zweierpotenzen sein. Je nach Beschaffenheit des Materials werden die folgenden Größen vorgeschlagen:

- 1.024 x 1.024 – sehr feine, hoch-strukturierte Materialien
- 512 x 512 – „normale“ Materialien
- 256 x 256 – einfache, gering-strukturierte Materialien

Die Bilddateien sind normalerweise so anzulegen, dass eine Wiederholung in beiden Dimensionen optisch ansprechend möglich ist.

Die Namensvergabe der Bilddateien ist beliebig.

RGBA-IMAGE Bilddatei mit Transparenz

Dieser Datentyp stellt eine Erweiterung des Typs **RGB-IMAGE** dar und beinhaltet einen zusätzlichen Transparenzwert. Dieser kann entweder ein skalarer Wert sein oder ein expliziter Farbwert, welcher digital die Transparenz steuert, d.h. Texel³ mit diesem Farbwert werden transparent dargestellt.

Erlaubt sind folgende Formate: PNG

GRAYSCALE-IMAGE

Graustufen-Bilddatei

Im Gegensatz zu **RGB-IMAGE** umfasst dieser Typ nur einen Wert pro Pixel⁴.

Erlaubt sind folgende Formate: PNG, JPEG

SYMBOL symbolischer Bezeichner

Anmerkung zu den Datentypen **RGB**, **RGB-IMAGE** und **RGBA-IMAGE**:
Für **RGB**-Farbwerte wird der **sRGB**-Farbraum angenommen.

Anmerkungen zu den Bilddatei-Formaten PNG und JPEG:

Bilder im PNG-Format müssen gemäß der „PNG (Portable Network Graphics) Specification“ [png] angelegt werden:

- müssen sequenziell (nicht interlaced/progressive) strukturiert sein
- müssen im Fall von **RGB-IMAGE** das **RGB**-Farbmodell nutzen
- müssen 8 Bit pro (Farb-)Kanal nutzen
- dürfen nicht animiert sein

Bilder im JPEG-Format müssen gemäß der Spezifikation für das „JPEG File Interchange Format“ [jff] angelegt werden:

- müssen sequenziell (nicht interlaced/progressive) strukturiert sein
- müssen *Huffman*-Kodierung (nicht arithmetische Kodierung) nutzen
- müssen das **YCbCr**-Farbmodell nutzen⁵
- müssen 8 Bit pro Farbkanal nutzen

2.3 Materialtypen

Der Materialtyp (Parameter *Material Type* vom Typ **SYMBOL**) dient der Auswahl eines passenden Shaders⁶. Weiterhin können auf Basis des Materialtyps die verfügbaren Parameter in einem Materialeditor eingeschränkt werden.

Folgende Typen sind definiert:

- *Common*

Dies ist der empfohlene Default-Typ, wenn dem Material keiner der u.g. spezifischen Materialtypen zugeordnet werden kann oder soll.

- *Glass*

Dieser Typ sollte allen Glas-Materialien zugeordnet werden. Wird stattdessen der Typ *Common* verwendet, kann es z.B. dazu kommen, dass das Material wie transparenter Kunststoff wirkt.

³ in der 3D-Computergrafik ein Pixel einer Textur

⁴ Liegt eine Bilddatei mit 3 Farbkanälen (RGB) vor, wird der Grauwert der Farbe verwendet.

⁵ Beim Import findet eine Konvertierung in das **RGB**-Farbmodell statt.

⁶ *Shader* sind Programme zur Berechnung von Rendering-Effekten, z.B. für die räumliche Wahrnehmung von 3D-Modellen.

- *Illuminant*

Materialtyp für selbstleuchtende Objekte. Sinnvoll in Verbindung mit dem Parameter *Luminance*.

Fehlt der Parameter, so ermittelt die Applikation anhand von Heuristiken aus den anderen angegebenen Parametern selbständig einen passenden Shader⁷ (welcher dann u.U. nicht die gewünschten Ergebnisse bringt).

2.4 Die Materialparameter

Vorbemerkungen:

Die Parameter sind in alphabetischer Reihenfolge angeführt. Hinter dem Bezeichner des Parameters werden in eckigen Klammern jeweils der Datentyp sowie in geschweiften Klammern die zugehörigen Materialtypen und Modelle angegeben.

- *Base_Color* [RGB] { *Common*, *Glass*, OMATS1, OMATS2 }

Die Basisfarbe dient zur Simulation der diffusen Reflexionseigenschaften der Objektoberfläche. Im Modell OMATS2 dient der Parameter bei Metallen auch zur Bestimmung von Farbe und Intensität der spiegelnden Reflexion.

Der vordefinierte Wert ist 1.0,1.0,1.0 (weiss).

- *Base_Color_Map* [RGB-IMAGE, RGBA-IMAGE, GRAYSCALE-IMAGE] { *Common*, *Glass*, OMATS1, OMATS2 }

Die hierdurch referenzierte Bilddatei dient zur ersetzenden Beschreibung des Parameters *Base_Color*.

Im Fall eines RGBA-IMAGE dienen die Transparenzwerte, welche sich aus dem Alpha-Kanal ergeben, zusätzlich zur ersetzenden Beschreibung des Parameters *Transparency*.

Im Fall eines GRAYSCALE-IMAGE wird für alle 3 Farbkanäle der gleiche Wert verwendet.

Es ist keine Bilddatei diesbezüglich vordefiniert.

Zu den Textur-Mapping-Verfahren siehe Abschn. 3.

- *Emissive_Color* [RGB] { *Illuminant*, OMATS1, OMATS2 }

Definiert die Farbe des emittierten Lichts einer geometrie-basierten Lichtquelle.

Der vordefinierte Wert ist 0.0,0.0,0.0 (schwarz).

- *Luminance* [FP] { *Illuminant*, OMATS1, OMATS2 }

Spezifiziert die Leuchtdichte einer geometrie-basierten Lichtquelle in cd/m^2 . (*Candela* — *cd* — ist die SI-Einheit der Basisgröße *Lichtstärke*.)

Der vordefinierte Wert ist 0.0.

- *Metallness* [FP] { *Common*, OMATS2 }

In der realen Umwelt sind Materialien in Metalle und Nicht-Metalle einteilbar. Deshalb sollte dieser Wert bei den meisten Materialien 0.0 oder 1.0 sein. Zwischenwerte dienen zur Darstellung von Halbmetallen oder verunreinigten Metallen.

⁷ Wenn z.B. der Luminance-Wert des Materials über 0 liegt und das Material nicht texturiert ist, wird der Typ *Illuminant* angenommen.

- *Metalness_Map* [GRAYSCALE-IMAGE] {Common, OMATS2}

Die hierdurch referenzierte Bilddatei dient zur ersetzenden Beschreibung des Parameters *Metalness*: Helle Bildbereiche erhalten Metalleigenschaften, dunkle werden als Nicht-Metall interpretiert.

Es ist keine Bilddatei diesbezüglich vordefiniert.

Zu den Textur-Mapping-Verfahren siehe Abschn. 3.

- *Normal_Map* [RGB-IMAGE] {Common, Glass, OMATS1, OMATS2}

Eine Normal-Map modifiziert die Normalenvektoren der Oberfläche, so dass damit die Beleuchtung von Erhebungen und Vertiefungen simuliert werden kann, welche in der Objektgeometrie nicht vorhanden sind.

Die Werte in der referenzierten Bilddatei werden als normierte Normalenvektoren interpretiert.

Es ist keine Bilddatei diesbezüglich vordefiniert.

Zu den Textur-Mapping-Verfahren siehe Abschn. 3.

- *Refractive_Index* [FP] {Glass, OMATS1, OMATS2}

Der skalare Wert gibt im Fall transparenter Materialien die Brechung des Lichts an. Unter dem Brechungsindex versteht man das Verhältnis der Phasengeschwindigkeit des Lichts im Vakuum zu der im jeweiligen Material.

Ausgewählte Werte sind:

- Wasser: 1.33
- Glas: 1.5 ... 1.9

Der vordefinierte Wert ist 1.0 und entspricht dem Brechungsindex von Vakuum.

- *Roughness* [FP] {Common, Glass, OMATS2}

Der Grad der Rauheit bestimmt, wie glatt oder rau eine Oberfläche ist. Je nach Rauheitsgrad wird das reflektierte Licht an der Oberfläche schwächer oder stärker gestreut.

Die Werte liegen im Bereich von 0.0 bis 1.0.

- *Roughness_Map* [GRAYSCALE-IMAGE] {Common, Glass, OMATS2}

Die hierdurch referenzierte Bilddatei dient zur ersetzenden Beschreibung des Parameters *Roughness*: Helle Bereiche des Bildes erscheinen matt, dunkle Bildbereiche glänzend.

Es ist keine Bilddatei diesbezüglich vordefiniert.

Zu den Textur-Mapping-Verfahren siehe Abschn. 3.

- *Shininess* [FP] {Common, OMATS1}

Der skalare Wert gibt den Glanz für glänzende Oberflächen an. Hierbei handelt es sich um den ganzzahligen Exponenten des *cos-Terms* entsprechend dem Beleuchtungsmodell nach *Phong*.

Als Faustregel gilt: Je größer dieser Wert, desto kleiner der die Spiegelung der Lichtquelle simulierende Glanzeffekt.

Der vordefinierte Wert ist 30.

- *Sound_Absorption* [PI {PI FP}*] {Common, Glass, Illuminant, OMATS1, OMATS2}

Der Parameter wird als eine Menge von Wertepaaren dargestellt, welche für unterschiedliche Frequenzen (1. Wert) den Schallabsorptionsgrad (2. Wert) angeben. Vor den Wertepaaren wird die Anzahl der Wertepaare spezifiziert.

Üblicherweise wird der Schallabsorptionsgrad für folgende Frequenzen angegeben: 125Hz, 250Hz, 500Hz, 1000Hz, 2000Hz, 4000Hz.

Der Schallabsorptionsgrad ist eine nicht-negative Gleitkommazahl. Normalerweise liegt der Wert zwischen 0.0 (keine Absorption) und 1.0 (vollständige Absorption). Es können aber auch Werte vorkommen, die leicht über 1.0 liegen. Das ist dann möglich, wenn die tatsächlich wirksame Fläche eines schallabsorbierenden Objekts größer ist als die geometrische Fläche, die für die Akustik-Berechnung herangezogen wird.

Fehlt die Angabe in der Materialbeschreibung, wird das entsprechende Objekt bei der Akustikberechnung nicht berücksichtigt.

Beispiel: 6 125 0.1 250 0.3 500 0.2 1000 0.1 2000 0.5 4000 0.4

- *Specular_Color* [RGB] {Common, OMATS1}

Die spekulare Farbe dient zur Simulation der spekularen⁸ Reflexionseigenschaften der Objektoberfläche und legt gleichzeitig die Farbe für Glanzlichter (Phong-Modell) fest.

Der vordefinierte Wert ist 0.0,0.0,0.0 (schwarz).

- *Specular_Factor* [FP] {Common, OMATS1}

Die Gewichtung der spekularen Farbe dient zur Steuerung der Intensität der spekularen Reflektion der Objektoberfläche. Glanzlichter (Phong-Modell) werden davon nicht beeinflusst.

Die Werte liegen üblicherweise im Bereich von 0.0 bis 1.0.

Der vordefinierte Wert ist 1.0.

- *Transparency* [FP] {Common, Glass, Illuminant, OMATS1, OMATS2}

Die Transparenz dient zur Simulation transparenter Eigenschaften des Materials. Es handelt sich um einen frequenzunabhängigen skalaren Wert.

Die Werte liegen im Bereich von 0.0 bis 1.0.

Der vordefinierte Wert ist 0.0, d.h. es liegt keine Transparenz vor.

⁸spiegelnden

3 Textur-Mapping-Verfahren

Im Rahmen von OMATS werden die in diesem Abschnitt beschriebenen Textur-Mapping-Verfahren unterstützt. Diese beziehen sich auf die Materialparameter *Base_Color_Map*, *Metalness_Map*, *Normal_Map* und *Roughness_Map*.

Ausgangspunkt sind dabei die Datentypen RGB-IMAGE, RGBA-IMAGE bzw. GRAYSCALE-IMAGE wie in 2.2 definiert. Diese werden, wie in Abb. 1 dargestellt, auf den normierten U-V-Koordinatenraum projiziert, auf den alle weiteren Ausführungen in diesem Abschnitt Bezug nehmen.

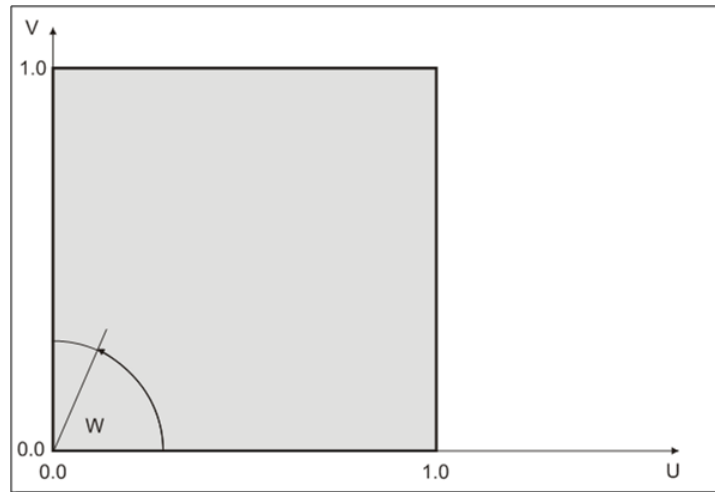


Abbildung 1: U-V-Koordinatenraum

Die folgenden Textur-Transformationen werden unterstützt (in dieser Reihenfolge):

1. Rotation um den Winkel W
2. Verschiebung um einen U-V-Offset
3. Skalierung im U-V-Raum

Für Normal-Maps können separate Transformationsparameter angegeben werden (in Bezug auf die anderen Maps). Ist kein spezifischer Transformationsparameter für die Normal-Map angegeben, wird der entsprechende Parameter für die anderen Maps verwendet (wenn vorhanden).

3.1 Ebenen-Mapping

Hierbei handelt es sich um eine planare Abbildung in eine vorgegebene Projektionsebene. Diese beschreibt die Lage des U-V-Raums und kann folgendermaßen gewählt werden:

- YZ-Ebene
- XZ-Ebene
- XY-Ebene
- Definition durch normierten Normalenvektor

Zusätzlich kann die Projektionsebene beliebig um alle Koordinatenachsen rotiert werden.

Verschiebung und Skalierung im U-V-Raum wird ebenfalls unterstützt.

3.2 Quader-Mapping

Beim Quader-Mapping handelt es sich um die automatische Abbildung der Modellkoordinaten in die Begrenzungsflächen eines achsenparallel ausgerichteten Quaders. Entlang jeder Koordinatenachse definiert der Quader einen eigenen U-V-Raum. Entsprechen die Basisvektoren U und V jeweils einem kanonischen Basisvektor, lassen sich 8 Basisvarianten angeben, wie in Abb. 2 dargestellt.

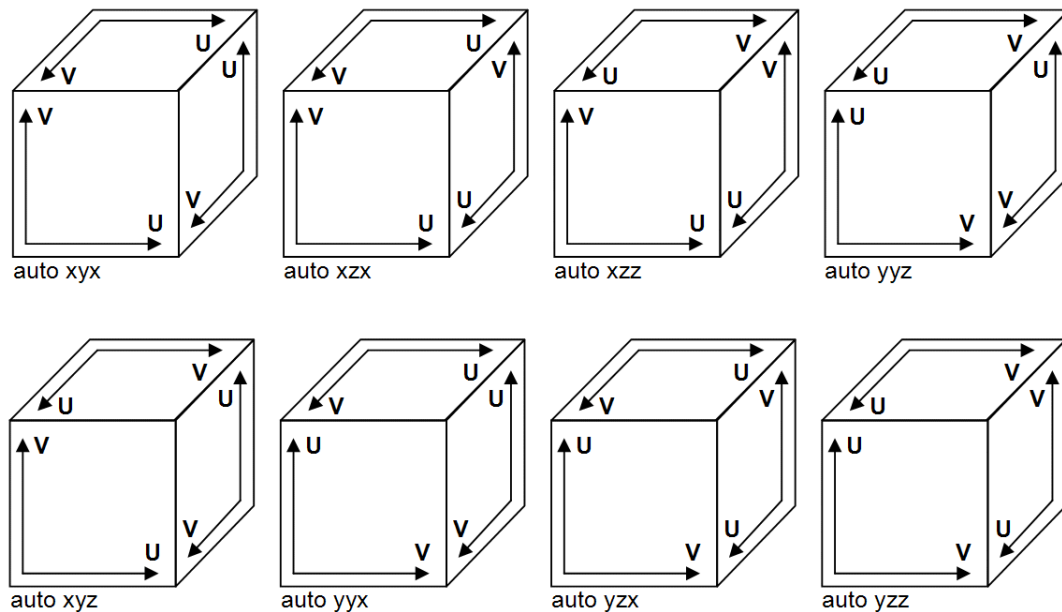


Abbildung 2: Quader-Mapping

Im Tripel hinter dem Schlüsselwort `auto` ist für jede Seite des Quaders die Richtung des U-Vektors kodiert, und zwar in der Reihenfolge vorne, rechts, oben. (Die Ausrichtungen hinten, links und unten sind analog.)

Spiegelungen der Quader an den Koordinatenebenen führen zu entsprechenden Spiegelungen im U-V-Raum, und damit zu weiteren Varianten. Translation, Skalierung und Rotation im U-V-Raum werden ebenfalls unterstützt. Die Zuordnung eines Vertices zu einer Seitenfläche des Quaders erfolgt anhand der betragsgrößten Koordinate des Normalenvektors.

3.3 Texturkoordinaten

Nicht immer läßt sich das gewünschte Ergebnis mittels allgemeiner Mapping-Verfahren beschreiben. Manchmal ist eine explizite Angabe der U-V-Koordinaten erforderlich. Diese werden dann nicht am Material, sondern an der Geometrie selbst hinterlegt. Damit entfällt die Berechnung der Projektion aus dem Modellbereich in den U-V-Raum. Skalierung, Offset und Rotation werden weiterhin auf die U-V-Koordinaten angewendet.

Dieses Verfahren wird auf alle Arten von Maps angewendet.

Wie die Texturkoordinaten hinterlegt sind, richtet sich nach dem Geometrieformat, d.h. die Angabe von Texturkoordinaten muss dort vorgesehen sein. Dafür können die Formate 3DS und OBJ genutzt werden.

4 OFML–Datenformat für Materialien

Vorbemerkung:

Die Ausführungen in diesem Abschnitt ersetzen bzw. aktualisieren die Ausführungen in Anhang D.2 „Materialien“ aus [\[ofml\]](#)!

Die Definition eines OFML–Materials beinhaltet ein Set von *Parametern*. Ein Parameter besteht aus einem Schlüssel, der die Bedeutung des Parameters definiert, gefolgt von durch Leerzeichen getrennten Argumenten⁹. In den Tabellen unten sind die aktuell unterstützten Schlüssel und zugehörigen Argumente definiert.

Eine Materialdefinition kann in in zwei Formaten repräsentiert werden, die sich im Wesentlichen durch die Form der Trennung der Parameter unterscheiden:

- **Materialdefinitionsdatei**

Die Parameter werden durch ein Zeilenende getrennt.

Der Name einer Materialdefinitionsdatei (Erweiterung `.mat`) ergibt sich aus der letzten Komponente des vollqualifizierten Bezeichners des Materials, unter dem es in den OFML–Daten (z.B. ODB, OFML–Part I [\[odb\]](#)) referenziert wird, wobei für den Dateinamen Kleinschreibweise vorgeschrieben ist¹⁰.

- **Inline–Deklaration**

Die Parameter werden durch ein Semikolon (;) getrennt.

Inline–Deklarationen können bei der OFML–Programmierung gemäß Part III der OFML–Spezifikation [\[ofml\]](#) oder bei der ODB–Datenanlage [\[odb\]](#) verwendet werden.

Inline–Deklarationen sind in zwei Formen möglich:

- *Reine Inline–Deklarationen* beginnen mit dem Dollar–Zeichen (\$) und enthalten eine vollständige Materialdefinition.
- *Material–Modifikatoren* beginnen mit einem voll qualifizierten Materialnamen, der auf ein Material in der OFML–Datenbasis verweist (Basis–Material). Danach folgen, durch Semikolon getrennt, einzelne Parameter, die den entsprechenden Parameter des Basis–Materials überschreiben.

Die bei der Beschreibung der Argumente in den folgenden Tabellen verwendeten syntaktischen und lexikalischen Elemente sind in der Legende am Ende dieses Abschnitts beschrieben.

⁹ Es gibt auch Parameter ohne Argumente.

¹⁰ Der Name eines Materials – ohne den vorangestellten Paket–Namensraum – sollte den Regeln für OFML–Bezeichner folgen, also nur alphanumerische Zeichen (inklusive dem Unterstrich) enthalten und nicht mit einer Ziffer beginnen.

Die folgende Tabelle definiert für alle aktuell unterstützten Modell-Parameter (s. Abschn. 2) die zugehörigen Schlüssel und Argumente:

Parameter	Modell	Schlüssel	Argument(e)
Material_Type	1, 2	type	(common glass illuminant)
Base_Color	1, 2	dif	R[F1] G[F1] B[F1]
Base_Color_Map	1, 2	tex image	FT[FT] FN[FN]
Emissive_Color	1, 2	emission	R[F1] G[F1] B[F1]
Luminance	1, 2	luminance	S[F]
Metallness	2	metallic	S[F1]
Metallness_Map	2	metallic image	FT[FT] FN[FN]
Normal_Map	1, 2	bumps	FT[FT] FN[FN]
Refractive_Index	1, 2	ref	S[F]
Roughness	2	roughness	S[F1]
Roughness_Map	2	roughness image	FT[FT] FN[FN]
Shininess	1	shi	S[F]
Sound_Absorption	1, 2	sndabsorb	N[I] {F[I] C[F]}*
Specular_Color	1	spe	R[F1] G[F1] B[F1]
Specular_Factor	1	reflection	S[F]
Transparency	1, 2	tra	S[F1]

Die folgende Tabelle definiert die Schlüssel und zugehörigen Argumente, die für die Textur-Mapping-Verfahren (s. Abschn. 3) benötigt werden:

Parameter	Schlüssel	Argument(e)
<i>Transformationen^a</i>		
Rotation	rotate, nrotate	0 0 A[F]
Verschiebung	offset, noffset	U[F] V[F] 0
Skalierung	scale, nscale	U[F] V[F] 0
<i>Ebenen-Mapping</i>		
YZ-Ebene	prjx	
XZ-Ebene	prjy	
XY-Ebene	prjz	
Normalenvektor	prj	X[F1] Y[F1] Z[F1]
<i>Quader-Mapping</i>	auto	(xyx xzx xzz yyz xyz yyx yzx yzz)
<i>Texturkoordinaten^b</i>	import	

^a Die Parameter, deren Schlüssel mit dem Buchstaben 'n' beginnen, wirken sich nur auf die Normal-Map aus. Ist einer dieser Parameter nicht angegeben, greift der entsprechende Parameter ohne den Buchstaben 'n' am Anfang.

^b In der Objekt-Geometrie hinterlegte Texturkoordinaten werden nur ausgewertet, wenn der `import`-Parameter vorhanden ist. Wenn der Schlüssel angegeben ist, in der Geometrie aber keine Texturkoordinaten hinterlegt sind, ist das Verhalten undefiniert.

Legende:

- Ein Argument wird entweder durch explizite Aufzählung der möglichen (alternativen) Werte in der Form (Wert1|Wert2|...) beschrieben¹¹, oder in der Form Name[Typ], wobei der Name die Semantik des Arguments bezeichnet.
- Eine sich wiederholende Menge von Argumenten wird in der Form {Arg1 ...}* dargestellt.
- Folgende Bezeichner (Abkürzungen) werden für benannte Argumente verwendet:
 - S – skalarer Wert
 - FT, FN – Dateityp, Dateiname
 - R, G, B – Rot, Grün, Blau
 - U, V, A – U-V-Koordinaten bzw. -Skalierung, Winkel
 - X, Y, Z – X-Y-Z-Koordinaten
 - N, F, C – Anzahl, Frequenz, Absorptionsgrad
- Folgende Bezeichner werden für die Typen der Argumente verwendet:
 - F – Gleitkommazahl
 - F1 – Gleitkommazahl im Bereich 0.0 bis 1.0
 - I – Ganzzahl
 - FT – Dateityp: (png|jpg|any)¹²
 - FN – Dateiname: (ggf. voll qualifizierter) OFML-Name, der auf eine Bilddatei in der OFML-Datenbasis verweist¹³

¹¹ Bei einem einzigen möglichen Wert entfällt die umschließende Klammer.

¹² Bei **any** wird der Typ aus der Dateierweiterung bestimmt.

¹³ Eine Qualifizierung ist notwendig, wenn sich die Bilddatei nicht im Datenverzeichnis der OFML-Serie befindet, in der auch die Materialdefinitionsdatei hinterlegt ist bzw. zu der die OFML-Instanz gehört, auf die eine Inline-Deklaration angewendet wird.

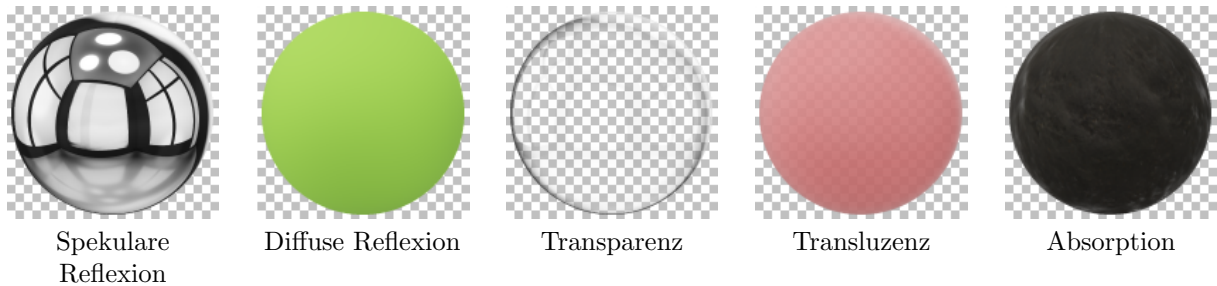
A Einführung in das physikalische Rendering (PBR)

PBR (**Physically Based Rendering**) simuliert, was geschieht, wenn Licht auf die Oberfläche eines Objektes trifft. Ein physikalisch korrekt beschriebenes Material interagiert mit Licht auf unterschiedliche Arten: Es wirft Licht zurück (*Reflexion*), bricht es (*Refraktion*) oder schluckt es (*Absorption*). Unter natürlichen Bedingungen wird Licht nicht zu hundert Prozent absorbiert, reflektiert oder gebrochen – alle Materialien bewegen sich in dem Spektrum zwischen diesen Extremen.

Ein Material ist sichtbar, weil es einfallendes Licht reflektiert. Alternativ nehmen wir auch Materialien wahr, die selbst Licht aussenden.

Die drei oben genannten grundsätzlichen Möglichkeiten der Interaktion von Licht und Material werden u.a. durch die **Materialeigenschaften** beeinflusst:

- Die Materialeigenschaften bestimmen die Art der Reflexion:
Bei der *spekularen*¹⁴ *Reflexion* wird das Licht direkt an der Oberfläche zurückgeworfen.
Diffuse Reflexion entsteht durch Streuung innerhalb des Materials (Lichtstrahlen dringen ein Stück weit in das Material ein und werden in verschiedene Richtungen abgelenkt.)
Metalle reflektieren ausschließlich spekulär, Nichtmetalle überwiegend diffus.
- Je nach Materialbeschaffenheit dringen die Lichtstrahlen tiefer in das Material ein. Sie werden entweder durch das Material hindurchgeleitet (*Transparenz*), innerhalb des Materials zurückgeworfen (*Transluzenz*) oder vom Material geschluckt (*Absorption*).



Neben den Materialeigenschaften (s.o.) werden beim PBR auch die physikalischen **Eigenschaften des Lichts** berücksichtigt:

- Nach dem Grundsatz der Energieerhaltung wird nie mehr Licht reflektiert, als einstrahlt.
Ein – nicht selbstleuchtendes – Material wird der Beleuchtung der Umgebung entsprechend dargestellt.
- Die Menge der reflektierten Lichtstrahlen ist vom Blickwinkel abhängig. Dieser sogenannte *Fresnel-Effekt* bewirkt, dass Oberflächen bei einem flachen Blickwinkel¹⁵ stärker spiegeln als wenn man senkrecht auf die Oberfläche schaut.

Die Eigenschaften des Lichts werden über den Shader der Applikation simuliert und sind nicht direkt durch den Benutzer (Materialdatenanleger) beeinflussbar.

Die Materialeigenschaften hingegen sind die Hebel für die Erstellung physikalisch stimmiger Materialien. Das am PBR orientierte Materialmodell OMATS2 definiert entsprechende Parameter, wobei die Menge der relevanten Materialparameter vom festgelegten *Materialtyp* abhängig ist (s. Abschn. 2).

¹⁴spiegelnden

¹⁵englisch: grazing angle

B Konvertierung von alten Materialien auf das neue Modell

Materialien, die nach dem alten Model OMATS1 angelegt worden sind, werden von einer OFML-Applikation, die das neue Modell verwendet, automatisch auf das neue Modell konvertiert.

Im Normalfall liefert diese Konvertierung eine zufriedenstellende Darstellung. In seltenen Fällen können dennoch Anpassungen erforderlich werden:

- In einigen Fällen glänzen Materialien stärker.
In diesem Fall muss mittels der Angabe des (neuen) Parameters *Roughness* die Rauheit nachjustiert werden.
- Eventuell werden Metalle nicht als solche erkannt (dies kann zum Beispiel bei Chrom der Fall sein).
In diesem Fall muss explizit der neue Parameter *Metallness* (mit dem Wert 1.0) angegeben werden.

C Historie

Die ersten Versionen dieser Spezifikation wurden von Ekkehard Beier (EasternGraphics GmbH) im Auftrag des Arbeitskreises *Industrielle Aspekte der OFML-Normung (IAON)* in Zusammenarbeit mit *wegscheider office solution gmbh* (Deutschland) und *weber office solution gmbh* (Schweiz) erstellt. Beginnend mit Version 1.4 ist die Spezifikation Gegenstand der Normierung durch das OFML-Normungsgremium des IBA.

Version 2.0 (2019-06-19)

- Neues Materialmodell OMATS2
- Neuer Datentyp *GRAYSCALE-IMAGE*
- Datentyp *FP3-IMAGE* entfernt, dafür Beschreibung des Parameters *Normal_Map* angepasst
- Parameter *Diffuse_Color* in *Base_Color* und *Diffuse_Map* in *Base_Color_Map* sowie *Refraction* in *Refractive_Index* umbenannt
- Materialtypen *Glass_Translucent* und *Metal_Polished* sowie Parameter *Ambient_Color* wegen geringer praktischer Relevanz entfernt
- Das Bilddateiformat TGA ist obsolet
- Entfernung der Bezüge auf AutoCAD
- Umstrukturierung des Dokuments

Version 1.5 (2015-02-27)

- Neue, explizite Transformationsparameter *nrotate*, *noffset* und *nscale* für Normal-Maps

Version 1.4, 1. überarbeitete Fassung (2014-01-08)

- Korrektur der Typdeklaration (F anstatt F1) bei einigen Materialparametern in refsec:Datenformat

Version 1.4 (2013-07-17)

- Neue Materialtypen *Glass_Translucent* und *Metal_Polished*
- Restrukturierung, Entfernung nicht (mehr) relevanter Abschnitte

Version 1.3 (2011-12-06)

- Materialtypen
- Selbstleuchtende Materialien
- Schallabsorption

Version 1.2 (2011-12-14)

- Phong-Level
- OFML-Datenformat für Materialien

Version 1.1 (2007-11-07)

- Textur-Transformationen und -Mappings

Version 1.0 (2006-09-25)

- initiale Version