

IDM-3D. Geometry

Ekkehard Beier (Editor)

Version

Spezifikation: IDM-3D.Geometry 1.2

Dokumentversion: Finale Version, Update 1

Spezifikation

IDM-3D.Geometry

IDM-3D.Geometry ist ein Bestandteil der IDM-3D-Spezifikation und beschreibt den Austausch von Geometriedaten auf Basis von 3D-Meshes¹. Hierbei steht die Stammdatenebene im Vordergrund. Die Bewegungsdatenebene kann davon abweichen und den Fokus auf effiziente Formate z.B. aus IDM-3D.Geometry legen oder andere Formate (ggf. auch mit anderer Granularitaet) erzeugen, z.B. glTF oder USD[z].

Ein IDM-3D.Geometry-Datensatz ist vorzugsweise eine vollumfaengliche Zusammenstellung der Geometrien eines Projekts, wobei die Zuordnung eines Projekts zu einem Hersteller, einer Produktserie, einer Teilproduktserie, mehreren Produktserien [auf kaufmaennischer Ebene] nicht definiert ist. So kann es beispielsweise sinnvoll sein, herstellerruebergreifend Geometrien in einem Projekt zusammenzufassen.

IDM-3D.Geometry erlaubt unter anderem folgende Einsatzszenarien:

- Bereitstellung der Geometrien des Dienstleisters fuer einen anderen Dienstleister oder Anwender als Ausgangspunkt fuer den kaufmaennisch parametrisierten Zusammenbau der Produkte

¹ Aktuell besteht der exklusive Fokus auf Mesh-Geometrien. Zu spaeterem Zeitpunkt kann dies in Richtung parametrische Volumenkoerper (CAD/BIM) oder anderweitig erweitert werden.

- Bereitstellung der Geometrien vom Anwender an einen Dienstleister oder Vertriebspartner zur Weiterverwendung (z.B. CGI-Generierung)

Die Entsprechung von IDM-3D.Geometry auf der Materialebene wird IDM-3D.Material sein. IDM-3D.Geometry und IDM-3D.Material werden Bausteine fuer weitere IDM-3D-Spezifikationen sein.

Geometrie

Prinzipiell ist eine Geometrie eine atomare Entitaet aus Sicht einer strukturierten, parametrischen 3D-Produktrepraesentation, deren Existenz durch folgende Kriterien bestimmt wird:

- Die Geometrie kann eine direkte Materialzuweisung empfangen.
- Die Geometrie kann ueber die kaufmaennische Konfiguration gesteuert werden (Sichtbarkeit, Position, ggf. Skalierung, etc.).
- Die Geometrie wird im Rahmen einer Funktionsdarstellung angesprochen.

Die [parametrische] Strukturdefinition sowie die Laufzeitanimationssteuerung sind ggf. Gegenstand weiterer IDM-3D-Standards, welche auf IDM-3D.Geometry aufsetzen. Dies gilt auch fuer die Deformationsproblematik, d.h. die Moeglichkeit, Geometrien auf Vertex-Ebene zu manipulieren. Die Deformations-relevanten Formate werden dann in einer zukuenftigen Version von IDM-3D.Geometry definiert.

Da die Geometrie per Definition eine atomare Entitaet ist, kann ihr auch nur genau ein Material zu gewiesen werden.

Es gibt keine kanonische Abbildung eines realen Produkts auf eine Menge von Geometrien. Die oben genannten Kriterien sollten aber prinzipiell beachtet werden. Generell sollte eine hohe strukturelle Uebereinstimmung mit realen Bauteilen angestrebt werden.

Unter einer Geometrie wird also ein atomarer Baustein fuer visuelle Produktrepraesentationen verstanden, welcher durch alternative Basis- oder native Repraesentationen realisiert werden kann. Prinzipiell ergeben sich dabei zwei Dimensionen:

- Fuer eine Geometrie koennen native Repraesentationen in verschiedenen Formaten [einundderselben Guete] hinterlegt werden.
- Fuer eine Geometrie koennen native Repraesentationen in verschiedenen Gueten [einunddesselben Formats] hinterlegt werden.

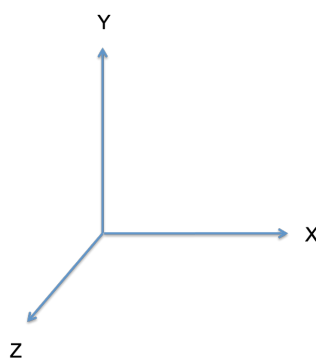
Der Begriff Basisgeometrie beschreibt eine native Repraesentation welchen Bestandteil einer abstrakten Geometrieentitaet, wie oben definiert, ist. Eine Geometrie kann ohne Basisgeometrien existieren - z.B., wenn sie nur Parameter definiert; eine Basisgeometrie kann jedoch in diesem Kontext nicht ohne Geometrie existieren.

Im Bereich der Formate wird unterschieden nach

- Mesh-basierten Geometrieformaten,
- parametrischen Geometrieformaten,
- sonstigen Formaten (Maps, Deformationen, etc.)

Eine durch ein Mesh-Format beschriebene Basisgeometrie beinhaltet mindestens Vertexkoordinaten sowie Dreiecke/Polygone. Weitere Informationen koennen zum Beispiel sein: Texturkoordinaten und Vertexnormalen. Eine Geometrie wird durch ein Material aus `IDM-3D.Material` parametrisiert und beinhaltet daher keine Materialeigenschaften. Insofern sind Materialdateien, welche im Kontext der nachfolgend definierten, nativen Geometriedaten gegebenenfalls existieren, zu ignorieren. Ebenfalls sind nichtgeometrische Informationen, wie Lichtquellen, Kameradefinitionen, Animationen, etc. zu ignorieren - die das native Format gegebenenfalls erlaubt.

Basisgeometrien haben die Maszeinheit 1 Meter. Das heisst: 0.5 korrespondiert zu 0,5 Meter. Dies gilt sowohl fuer Vertexkoordinaten, sowie auch fuer nicht-objektbezogene Texturkoordinaten. Das zugrundegelegte Koordinatensystem ist wie folgt definiert:



- x-Achse: nach rechts [aufsteigende Werte]
- y-Achse: nach oben
- z-Achse: nach vorn

Die Koordinaten einer Basisgeometrie muessen erhalten bleiben. Automatische Ausrichtungen sind somit nicht erlaubt. Weiterhin muessen Basisgeometrien verschiedener Formate/Gueten identisch ausgerichtet sein.

Eine Geometrie hat einen Bezeichner, welcher im Rahmen eines Projekts eindeutig sein muss. Der vollständige Bezeichner einer Geometrie ergibt sich durch einen 3-Ebenen-Namen, wobei die erste Ebene einen globalen Identifier beinhaltet und die zweite Ebene das Projekt eindeutig im Kontext der ersten Ebene beschreibt. Für den Geometriebezeichner, also die dritte Ebene, gilt:

- Als Zeichen sind die Buchstaben A - z, die Ziffern 0 - 9, _ und - erlaubt.
- Innerhalb eines Projekts darf es keine zwei Geometrien geben, die per case-tolerantem Vergleich identisch sind

Anmerkung: Die Vorgaben für die erste und zweite Bezeichnungsebene sind an anderer Stelle definiert und prinzipiell artverwandt. Allerdings darf das erste Zeichen weder eine Ziffer noch ein Bindestrich sein.

Geometriebezeichner sollten möglichst lesbar (z.B. durch Anwendung von Camel-Case oder Unterstrichen) und mit Bezug zur Problemdomäne gewählt werden.

Mesh-Formate

Es werden die nachfolgenden alternativen Mesh-Formate definiert, welche sich bezüglich funktionaler Mächtigkeit, Umfänglichkeit und Verwendbarkeit (Reverse Engineering) deutlich unterscheiden. Somit kann individuell entschieden werden, welche(s) Format(e) im konkreten Fall distribuiert werden sollen.

Wavefront OBJ

OBJ [OBJ] ist ein polygonales² Mesh-Format mit einem optionalen UV-Set.

Hierbei sind die Vertex- und Polygoninformationen zu interpretieren, d.h. Vertexkoordinaten, Normalenvektoren, Texturkoordinaten, sowie die Definition der Polygone. Weiterhin sind Smoothing Groups zu unterstützen. Die Anzahl der Punkte je Polygon ist nicht auf 3 begrenzt. Weiterhin sollen die Polygone konvex und planar sein.

Weitere Informationen werden nicht verwendet. Dies gilt insbesondere für

- Materialbibliotheken und -namen,
- Spline-Kurven und -flächen, Merging Groups
- Trimmkurven
- Level-of-Detail-Informationen
- Raytracing- und Shadow-Casting-Informationen

² zumindest ist das die übliche und hier genutzte Verwendungsform

Falls das UV-Set vorhanden ist, ist dieses fuer die Zuweisung des Materials zu verwenden.

Insofern eine Geometrie Mesh-Formate beinhaltet, muss mindestens das Format OBJ vorhanden sein.

Die Dateierweiterung ist *obj*.

OpenCTM

OpenCTM [OPENCTM] ist ein verlustfrei komprimierendes Dreiecksformat, dass beliebige Vertexinformationen beinhalten kann. Im Kontext dieser Spezifikation sollen diese Informationen wie folgt definiert werden:

- Das optionale Texturkoordinaten-Set 'Material' (alternativ: 'Diffuse') beschreibt ein sich endlos wiederholendes Gitter fuer die Zuweisung des Materials.
- Das optionale Texturkoordinaten-Set 'Object' beschreibt die Abbildung der Dreiecke auf einen sich nichtwiederholenden zweidimensionalen, normierten Wertebereich, der entweder einer Geometrie oder mehreren Geometrien zugeordnet ist.

Die Dateierweiterung ist *ctm*. Es wird ausschliesslich die Version 1.0.3 verwendet.

FBX

FBX [FBX] ist ein komplexes Format, dass in diesem Kontext jedoch nur zur Beschreibung von Meshes verwendet werden darf. FBX kann multiple Texturkoordinaten-Sets beinhalten. Diesbezuglich soll gelten:

- Das erste Texturkoordinaten-Set (falls vorhanden) beschreibt ein sich endlos wiederholendes Gitter fuer die Zuweisung des Materials.
- Das zweite Texturkoordinaten-Set (falls vorhanden) beschreibt die Abbildung der Dreiecke auf einen sich nichtwiederholenden zweidimensionalen, normierten Wertebereich der entweder einer Geometrie oder mehreren Geometrien zugeordnet ist.

Im Kontext von IDM-3D.Geometry sollen bezueglich FBX die folgenden Festlegungen bzw. Einschraenkungen gelten:

- Es wird Version 7 (oder spaeter) in der Binaervariante verwendet.
- Materialien und Animationen werden, falls enthalten, ignoriert.

Die Dateierweiterung ist *fbx*.

Deformationen

Deformationen sind Datenstrukturen, die zur Laufzeit auf Geometrien angewendet werden und insbesondere nützlich sind, um Verformungen effizient, dynamisch und parametrisch zu beschreiben. Je nach Deformationstyp ist auch eine Erweiterung in Richtung Animationen möglich.

Die aktuell unterstützten Ansätze für Deformationen sind:

- Vereinfachte Ersatzgeometrien, z.B. Hüllquader (nachfolgend: Gitter), deren Deformation explizit beschrieben, aber indirekt auf die echte Geometrie angewendet wird.
- Explizite Deformationsversionen der Geometrie

Beim expliziten Ansatz besteht eine Abhängigkeit zur Geometrie, d.h. parallel zur Geometrie (Mesh) sind auch die Deformationsdaten geometrieabhängig zu hinterlegen. Im Kontext der IDM-3D.Geometry-Spezifikation sollen alternative Deformationsansätze definiert werden, auch um dem weiteren technischen Fortschritt Rechnung tragen zu können.

Generell sollten für IDM-3D-Geometrien im Fall dass die Laufzeitumgebung Deformationen nicht oder nur partiell unterstützt, als Fallback die entsprechenden Geometrie angezeigt werden. Konsequenterweise sollten IDM-3D-Daten einen undeformierten Zustand priorisieren, d.h. nicht das Vorhandensein der Deformationsimplementierung voraussetzen, um einen korrekten Initialzustand zu erreichen. Dies kann aus Sicht der Modellierung nachteilig sein.

Die Deformation wird primär über eine JSON-Datei beschrieben, welche die folgenden Informationen bereitstellen muss:

Algorithm: "IG1" | "Blender1" | "Keys1"

Der Parameter legt den Algorithmus und eventuell weitere Dateien fest.

- "IG1" – Es erfolgt eine Gitterdeformation basierend auf den Mean-Value-Koordinaten [FFD].
- "Blender1" – Es erfolgt eine Gitterdeformation entsprechend dem Algorithmus der Modellierungssoftware blender.
- "Keys1" – Es erfolgt eine Deformation über explizite Versionen, welche in einer zusätzlichen Datei beschrieben sind.

Animation: "First" | "All" = "First"

Im Fall des "Keys1"-Algorithmus legt dieser Parameter fest, ob nur die erste oder gegebenenfalls alle enthaltenen Animationen interpretiert werden sollen.

Type: "XYZ" | "XY" | "YZ" | "Z" = "XYZ"

Der Parameter definiert fuer die Algorithmen "IG1" und "Blender1", welche Koordinaten fuer die Beschreibung des Gitters bereitgestellt werden. Falls der Wert nicht "XYZ" ist, muss zur Laufzeit die fehlende Dimension automatisch gesetzt werden, so dass sie die undeformierte Geometrie vollstaendig umfasst incl. eines Offsets, um Rundungsartefakte zu vermeiden.

Dimensions: {"X": Integer, "Y": Integer, "Z": Integer}

Der Parameter gibt fuer die Algorithmen "IG1" und "Blender1" fuer jede Dimension die Anzahl der Punkte an. Falls der Type nicht "XYZ" ist, wird die entsprechende Dimensionsangabe ignoriert und kann somit entfallen.

Vertices: {<t>: {"X": Number, "Y": Number, "Z": Number}[]}

Der Parameter gibt fuer die Algorithmen "IG1" und "Blender1" die Gitterpunkte an. Hierbei steht <t> fuer einen Zeitwert im Bereich 0.0 bis 1.0, der jedoch als String angegeben werden musz.

Die Anzahl der Arrayelemente ergibt sich aus der Multiplikation der Einzelwerte von Dimensions, wobei eine lt. Type nicht angegebene Dimension mit 1 angenommen wird.

Die Vertices werden in der Reihenfolge Z (von hinten nach vorn) – Y (von oben nach unten) – X (von links nach rechts) interpretiert.

Die Deformationsbeschreibungsdatei hat den Namen *deformation.json*. Die alternative protobuf-Variante hat den Namen *deformation.ffd*.

Im Bereich der expliziten Geometriedeformationen (Algorithmus: "Blender1") wird eine parallele Geometriedatei im Format glTF bzw. GLB [GLTF] erwartet. Diese enthaelt dann die Geometrie (mit bis zu 2 Texturkoordinaten-Sets) sowie eine oder mehrere Animationen, die auf den Bereich 0 – 1 abgebildet werden. Dabei sind alle glTF-Animationsarten zu unterstuetzen: Key-Frames, Bones und Verschiebungen. Alle weiteren glTF-Inhalte (wie z.B. Materialien, Lichtquellen, Kameras) sind dagegen zu ignorieren.

Sonstige Formate

Normal-Maps

Eine Normal-Map beschreibt eine Manipulation des sich aus der Geometrie an einer bestimmten Stelle (ermittelt durch die entsprechenden Texturkoordinaten) ergebenden Normalenvektors. Dies erfolgt durch eine Bilddatei, deren Dimensionen Zweierpotenzen sind. Analog zu den Mesh-Formaten kann es auch hier verschiedene Hinterlegungen geben (z.B. verschiedene Gueten).

Die Normal-Maps werden als Tangenten-Normal-Maps festgelegt (hellblau entspricht senkrecht).

Die Dateierweiterung ist *jpeg/jpg* (JPEG) oder *png* (PNG).

Geometrie-Gueten/-modi

Es werden die folgenden Gueten bzw. Modi definiert.

Standard

Die Guete Standard ist fuer die mobile Echtzeitanwendung konzipiert. Hierbei werden schnelle Ladezeit und geringer Speicherbedarf gegenueber der Darstellungsqualitaet priorisiert. Dies kann durch manuelle oder automatische Mesh-Reduktion und/oder den Einsatz von Normal-Maps erreicht werden.

Der Dateiname³ ist *standard* fuer Mesh-Dateien und *normals_standard* fuer Normal-Maps.

Als Richtwerte⁴ fuer Mesh-Daten mit der Standardguete sei definiert (Bezug: unkomprimiertes OBJ, incl. Texturkoordinaten):

- groeszere Objekte (z.B. ganze Bettrahmen, ganze Bettkopfteile): 500 KB bzw. 20.000 Dreiecke
- mittlere Teile (z.B. Armlehnen, Sitzkissen): 200 KB bzw. 10.000 Dreiecke
- kleine Teile (z.B. Griffe, Bettfuesze): 50 KB bzw. 200 Dreiecke

Fuer Normal-Maps betraegt die Pixelgroesze 1024 bzw. 2048.

High Resolution

Die Guete High Resolution beschreibt die minimale vollumfaengliche geometrische Repraesentation. Der primaere Verwendungszweck ist die hochqualitative Bilderzeugung ueber Raytracing-/Radiosity-Verfahren.

³ nachfolgend immer zuzueglich der Dateierweiterung zu interpretieren

⁴ Von harten Limits soll abgesehen werden. Allerdings ist eine statistische Einhaltung der Vorgabe wichtig fuer einheitliche Verarbeitungskriterien (Speicherbedarf, Download-/Ladezeiten).

Hierbei wird im allgemeinen auf geometrienersetzende Datenstrukturen, wie z.B. Normal-Maps, verzichtet. Allerdings sollten Daten dieser Guete minimal sein, d.h. nicht unangemessen umfangreich. Dies gilt z.B. fuer Daten die von einem 3D-Scanner kommen.

Als Richtwert wird Faktor 20 auf die Groesze der korrespondieren Datei der Guete Standard vorgegeben und ein absolutes Limit von 20 MB je Datei definiert.

Es ist legitim, fuer diese Guete eine Normal-Map zu hinterlegen. Allerdings ist im Gegensatz zum Mesh kein Fallback auf eine Normal-Map mit geringerer Aufloesung bei Nichtvorhandensein vorgesehen.

Der Dateiname ist *highres*.

Low Resolution

Die Guete Low Resolution ist eine optionale, stark reduzierte Repraesentation der Aufloesung Standard und der primaere Verwendungszweck ist die schnelle Voransicht. Ueblicherweise wird die Low Resolution-Guete dann durch die Standard-Guete ersetzt. Insofern sind deutlich sichtbare Abweichungen/ Vereinfachungen legitim. Auf Texturkoordinaten kann verzichtet werden.

Als Richtwert fuer die Guete Low Resolution wird eine Reduktion der Guete Standard um 80% vorgegeben.

Der Dateiname ist *lowres*.

Collider

Die Guete Collider ist eine optionale, extrem reduzierte Repraesentation der Geometrie und der primaere Verwendungszweck ist die interne Verwendung fuer die schnelle Kollisionsermittlung. Bezueglich der Simplifizierung ist bei konkaven Geometrien zu beachten, dass diese hinreichend wiedergegeben werden, um Artefakte durch falsche Kollisionserkennung zu vermeiden.

Der Dateiname ist *collider*.

Source

Der Modus Source kennzeichnet eine Source-Datei und kann verwendet werden, um gegebenenfalls die konsistente Ablage der Source-Dateien mit den abgeleiteten Formaten zu ermoeglichen.

Der Dateiname ist *source*. Eine Enumeration der legalen Source-Formate erfolgt nicht.

Geometrieparameter

Optional kann eine Geometrie Parameter definieren, welche ihre Interpretation im verarbeitenden Kontext steuern.

Complexity: int = 100

Der Parameter gibt die Komplexitaet des Bauteils an, um eine Basis fuer quantitative Tests zu haben. Diese Tests sollen zur Bewertung von Datenstaenden Anwendung finden, um die Lauffaehigkeit der Daten in mobilen Szenarien zu bewerten.

Der Default-Wert entspricht mittleren Bauteilen wie oben definiert. Je geringer (hoeher) die raeumliche Groesze eines Bauteils, desto geringer (hoeher) dessen Komplexitaet. Je einfacher (filigraner) die Form eines Bauteils, desto geringer (hoeher) dessen Komplexitaet.

NormalMap: string

Alternativ zu einer Normal-Map als inkludierter Bestandteil einer Geometrie, kann eine Normal-Map autonom existieren und von mehreren Geometrien verwendet werden. Ueber den Parameter NormalMap kann dies umgesetzt werden. Die Normal-Map kann dabei voll- oder nichtqualifiziert angegeben werden. Die nichtqualifizierte Variante ist zu bevorzugen, da der Datensatz dann flexibler verwendbar ist. Die Normal-Map wird ohne konkrete Dateibezeichnung angegeben, da die tatsaechliche Aufloesung erst zur Laufzeit erfolgt.

Existieren fuer eine Geometrie sowohl lokale Normal-Map-Dateien und eine per Geometrieparameter definierte Map, so hat letztere den Vorrang.

NormalMapStrength: float = 1

Ueber diesen Parameter ist die Steuerung der Staerke einer Normal-Map moeglich. Der Parameter geht als direkter Faktor in die Berechnung ein.

NormalMapTiling: bool = false

Dieser Parameter steuert die Interpretation der Texturkoordinaten, welche fuer die Normal-Map verwendet werden, im Bereich ausserhalb [0, 1] wie folgt:

- false – Werte groesser 1 (kleiner 0), werden auf 1 (0) abgebildet („clamp“).
- true – Werte ausserhalb [0, 1], werden in den Bereich [0, 1] abgebildet („repeat“).

PatchSize: float

Der Parameter gibt die Groesze des Texturkoordinatennetzes, welches fuer die Materialzuweisung verwendet wird, in Meter an. Der Parameter kann verwendet werden um eine automatische Anpassung von Materialien, die ihrerseits eine entsprechende Information bereitstellen, zu ermoeglichen.

Informationsdatei

Die Informationsdatei beinhaltet zusaetzliche Informationen zu einer Geometrie.

Die jeweils optionalen Eintraege sind:

- *Name* – Der Name der Geometrie kann optional angegeben werden, um ungewollte Veraenderungen (z.B. Case-Abweichungen) zu vermeiden. Falls ein Name angegeben ist, so hat dieser Prioritaet gegenueber dem Namen, welcher sich aus dem Verzeichnisnamen ableitet. Ist der Name nicht angegeben, so wird er aus dem Verzeichnisnamen abgeleitet. (siehe unten)
- *Description* – Dieser Eintrag beinhaltet eine informative, einzeilige Beschreibung der Geometrie.
- *Parameters* – Dieser Eintrag beinhaltet die Parameter der Geometrie. Die erlaubten Keys des JSON-Dictionaries und die jeweiligen Werte ergeben sich aus den oben definierten Geometrieparametern. Jeder Parameter vom Typ int, float oder bool kann auch als String angegeben werden. Im Fall eines Floats muss der String in wissenschaftlicher Notation vorliegen. Im Fall eines Booleans sind die Werte True/true und False/false erlaubt.

Die Informationsdatei hat den Dateinamen *info.json*. Die Kodierung der Datei muss in ASCII und UTF-8 (ohne BOM) identisch sein. Folglich sollte die Beschreibung keine Umlaute beinhalten.

Indexdatei

Die optionale Indexdatei befindet sich ausserhalb der Geometrieverzeichnisse.

Der primäre Zweck der Indexdatei besteht darin, die für ein Projekt benötigten Geometrien eindeutig zu beschreiben, so dass ein Dienstleister eine entsprechende Ausgangsbasis hat. Falls in einem Projekt sowohl Indexdatei wie auch Geometrie-Informationsdateien existieren, so haben letztere den Vorrang.

Die Datei beinhaltet ein JSON-Dictionary, dessen Keys die Geometrienamen sind. Je Wert wird die Struktur angegeben, wie sie für die Informationsdatei angegeben ist. Zusätzlich gibt es den folgenden optionalen Eintrag:

- *Alignment* – Der Eintrag beschreibt informativ die Ausrichtung der Geometrie. Falls kein Eintrag angegeben ist, gilt “links-unten-hinten” in bezug auf den lokalen Ursprung der Geometrie relativ zum achsenorthogonalen Begrenzungsvolumen aller Vertices. Diese Ausrichtung ist in den allermeisten Fällen praktisch für den späteren parametrischen Zusammenbau von Produkten. Allerdings kann es für achsen- oder rotationssymmetrische Bauteile sinnvoll sein, eine andere Ausrichtung zu wählen, z.B. ‘mittig-unten’ und diese entsprechend per Alignment anzugeben.

Die Index-Datei hat den Dateinamen *index.json*. Die Kodierung der Datei muss in ASCII und UTF-8 (ohne BOM) identisch sein. Folglich sollte die Beschreibung keine Umlaute beinhalten.

Physische Repraesentation

Index

Die optionale Indexdatei befindet sich parallel zu den Geometrieverzeichnissen auf oberster Ebene einer Distribution (z.B. ZIP-File).

Geometriedateien

Pro Geometrie existiert ein Verzeichnis. Der Name des Verzeichnis ergibt sich aus dem Namen der Geometrie an welchen die Erweiterung *geo* angehängen wird. Insofern es sich jedoch um eine geteilte Normal-Map handelt, ist die Erweiterung *tex*.

Innerhalb des Verzeichnis befinden sich die oben definierten Dateien, wobei sich deren Name jeweils aus dem genannten Praefix und der Formatspezifischen Erweiterung ergibt.

Optional kann ein eindeutiger Content-Hash je Datei bereitgestellt werden, was zu Optimierungszwecken verwendet werden kann. Vorgesehene Hashing-Varianten sind:

- MD5 (Dateierweiterung: md5)

Der Name der Hash-Datei ergibt sich aus dem Namen der entsprechenden Datei mit der Erweiterung⁵ *md5*. Der Inhalt der Datei ist der Hash-Wert ohne Zeilenumbruch.

Anmerkung: Der Hash muss nicht notwendigerweise zum Dateiinhalt korrespondieren. So kann der Hash von auflösungsreduzierten Dateien sich auf die Ausgangsauflösung beziehen.

Referenzen

[FBX] <https://docs.fileformat.com/3d/fbx/>

[FFD] <https://www.mn.uio.no/math/english/people/aca/michaelf/papers/mv3d.pdf>

[GLTF] <https://github.com/KhronosGroup/glTF>

[OBJ] <https://docs.fileformat.com/3d/obj/>

<https://www.fileformat.info/format/wavefrontobj/egff.htm>

[OPENCTM] <http://openctm.sourceforge.net>

Historie

IDM-3D.Geometry 1.2 (Update 1) – 10.05.2023

- Layoutanpassung an DCC-Style. Keine inhaltlichen Änderungen.

IDM-3D.Geometry 1.2 – 21.03.2023

- Groeszenangabe fuer High-Resolution-Geometrien
- Definition von Deformationen

IDM-3D.Geometry 1.1 – 10.01.2023

⁵ Die Erweiterung wird also angehängen, z.B. *standard.obj.md5*



- Festlegung zu den Geometrieparametertypen float, int und bool
- Neue Geometrieparameter: Complexity, PatchSize, NormalMapStrength und NormalMapTiling

IDM-3D.Geometry 1.0 – 01.10.2020

Kontakt

Daten Competence Center e. V.

Goebenstraße 4-10

32052 Herford

Info: www.dcc-moebel.org

Dr.-Ing. Olaf Plümer

E-Mail: pluemer@dcc-moebel.org

Tel.: +49 52 21 / 12 65-37

PD Dr. Ing. habil Ekkehard Beier

E-Mail: eb@intelligentgraphics.ag

Tel.: +49 36 77 / 66 73 520