preschern.org

Klaus Preschern

Version 2025 *

Inhaltsverzeichnis

1 Einleitung

preschern.org beschreibt einige vom Autor durchgeführte Softwareprojekte. Eines dieser Projekte hatte zum Inhalt LaTeX in DHTML zu übersetzten, wodurch dieses Dokument auch mit Internet-Browsern dargestellt werden kann (siehe Abschnitt 2.5). 3D Modelle können in aktuellen Internet-Browsern mit Hilfe von WebGL (siehe Abschnitt 2.4) visualisiert werden.

The web site is powered by JaSIL (Abschnitt 2.10).

2 Projektübersicht

2.1 glTF - JPEG of 3D

glTF (Graphics Library Transmission Format oder GL Transmission Format und zuvor bekannt als WebGL Transmissions Format oder WebGL TF) ist ein standardisiertes Dateiformat für 3D Szenen und Modelle. glTF ist ein offener Standard [12] der von der Khronos Group entwickelt wurde. Um glTF sowohl am Windows Desktop, als auch im Browser mit WebGL zu unterstützen, wurde mit Visual Studio C++ und Emscripten ein gltF-Viewer realisiert, der HDRI (high dynamic range imaging) mit glTF 2.0 PBR (physically based rendering) und MaterialX (Abschnitt2.3) unterstützt.

glTF PBR Beispiele: Damaged Helmet, Suzanne, Drachenmodell, Voronoi Diagram auf der Kugel, Schachbrett, Box Animation.

MaterialX Beispiele: Brick Pattern, Copper, Suzanne, BMW-M6.

Längere Ladezeit erforderlich: MaterialX Schachbrett.

2.2 WebAssembly

WebAssembly, abgekürzt Wasm, ist ein sicheres, portables und kompaktes Codeformat mit der Zielsetzung eine möglichst effizienten Ausführung zu erreichen. Das Hauptziel ist die Entwicklung von performanten Web-Anwendungen, wobei WebAssembly plattformunabhängig konzipiert wurde. WebAssembly ist ein offener Standard der von einer W3C Community Group entwickelt wurde [24].

Der Codegenerator von JaSIL (Abschnitt 2.10) wurde erweitert um C/C++-Code zu generieren, der mit Hilfe von Emscripten in WebAssembly übersetzt und ausgeführt wird. Mit folgenden Beispielen kann die Performance von Javascript und WebAssembly verglichen werden: preschern.org, Drachenmodell, CSG-Beispiel, Voronoi Diagram auf der Kugel, WebGL-Demo, MaterialX-Demo, BMW-M6 (das BMW Beispiel verwendet einen einfachen Mark and Sweep Garbage Collector).

2.3 MaterialX mit WebGL

MaterialX ist ein offener Standard für die plattformunabhängige Repräsentation von Material mit hoher Darstellungsqualität im Bereich der Computergraphik. Da auch die OpenGL ES Shading Language (GLSL Version 3.00 ES) unterstützt wird, war es möglich, MaterialX in die Java-WebGL-Schnittstelle (Abschnitt 2.4) zu integrieren.

Über die Links MaterialX-Demo und BMW-M6 können Anwendungsbeispiele aufgerufen werden, wobei allerdings mit längeren Ladezeiten zu rechnen ist. Es wird ein Browser benötigt, der unter anderem die ,,Compression Stream API" unterstützt (im Firefox-Browser erst ab Version 113).

2.4 Java Schnittstelle für WebGL

Mit Hilfe von JaSIL (Abschnitt 2.10) wurde mit Java eine WebGL [27, 28] Schnittstelle implementiert. Damit wurde ein Viewer für 3D Modelle entwickelt, die mit dem Universal 3D (U3D - siehe Abschnitt 2.7) Format gespeichert wurden. Als Anwendungsbeispiele können über folgende Links das Drachenmodell, das CSG-Beispiel oder das Voronoi Diagram auf der Kugel mit dem U3D-Viewer aufgerufen werden.

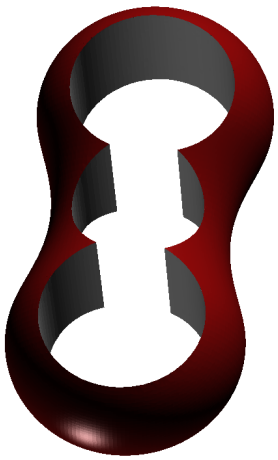
Mit dem Mausrad kann im U3D Viewer "gezoomt"werden und durch "dragging"der Maus rotiert die Kamera um das Modell. Für die Lichtsimulation wird ein Phong Shader verwendet.

2.5 Content Management mit LaTeX

Der Inhalt von preschern.org ist mit LaTeX [18, 19] geschrieben worden und kann in verschiedene Formate übersetzt werden. Inhalts- und Literaturverzeichnis, Verweise und dergleichen können, entsprechend der Beschreibung in LaTeX, von den für die weitere Übersetzung verwendeten Werkzeugen erstellt werden. So kann daraus dieses PDF Dokument erzeugt werden, in das, mit Hilfe der in Abschnitt 2.7 beschriebenen Universal 3D Schnittstelle, 3D Modelle eingebettet worden sind.

Ein mit Java implementierter Compiler übersetzt die, für die Beschreibung des Inhalts, verwendete LaTeX Teilmenge, mit Hilfe von einigen speziellen LaTeX Makros (sogenannten "hints"), in XML. Diese XML Beschreibung wird, mit Hilfe von XML Layout Beschreibungen (parts"), in DHTML übersetzt. Über die Layout Beschreibungen wird die Verbindung zu JavaScript hergestellt, welches mit JaSIL (Abschnitt 2.10) aus Java erzeugt wird.

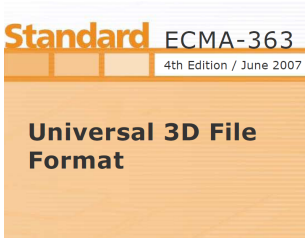
2.6 Modellieren mit impliziten Flächen



Eine Funktion die jedem Punkt $|x,y,z| \in \mathbf{R}^3$ eine Distanz mit positivem Vorzeichen zuordnet, wenn dieser außerhalb der von der Funktion repräsentierten Oberfläche liegt, den Wert 0 wenn der Punkt auf der Fläche liegt und negative Werte für innen liegende Punkte, eignet sich recht gut für die effiziente Implementierung von Mengenoperationen (Constructive Solid Geometry [CSG]) mit Objekten.

Für die Extraktion (Polygonalisierung) der Flächen wurde *Dual Contouring* [15] verwendet, um deren Eigenschaften in Bezug auf Ecken und Kanten, zu bewahren. Folgender WebGL Link zeigt ein 3D Modell das durch die Verknüpfung von impliziten Funktionen erzeugt und mit *Dual Contouring* extrahiert wurde: WebGL.

2.7 Universal 3D Schnittstelle



Visualisierung beinhaltet den Begriff der Pluralität und der Perspektive. Sie ist der Versuch eine, oft in textueller Form (z.B. als mathematische Beschreibung), vorliegende Darstellung, durch eine graphische zu *repräsentieren* [11].

Der Standard ECMA-363 *Universal 3D File Format (U3D)* [7] dient dazu 3D Modelle zu speichern. Für das Einlesen und die Speicherung von 3D Szenen wurde eine U3D Schnittstelle mit Java implementiert, die eine Teilmenge des U3D Standards unterstützt.

Mit Hilfe des LaTeX movie15Package, können multimediale Inhalte, wie Filme, Sound und mit U3D beschriebene Objekte, in PDF Dokumente (Adobe PDF Spezifikation Version 1.6) eingebettet werden. Mit entsprechender Software, wie z.B. dem Acrobat Reader Version ≥ 8.0, können solche Objekte in 3D dargestellt werden. Damit kann eine hochwertige textuelle und graphische Darstellung von Inhalten geboten werden.

2.8 3D Rekonstruktion mit impliziten Flächen

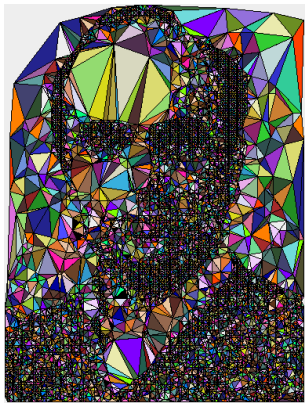
Eine Fläche kann mit einer impliziten Gleichung der Form $f(x,y,z) = f(p) = 0$ beschrieben werden. Ein Punkt p ist Teil der impliziten Fläche wenn das Resultat 0 ist, wenn er in die Gleichung eingesetzt wird.

Ein einfaches Beispiel ist eine Kugel mit Radius 1 (unit sphere), welche mit der impliziten Gleichung $f(x,y,z) = x^2 + y^2 + z^2 - 1$ beschrieben wird. Durch "verschmelzen"von impliziten Flächen können Oberflächen aus nicht orientierten Punktmengen rekonstruiert werden. Die Grundidee ist, einfache Körper, wie etwa Kugeln, miteinander zu verschmelzen. Jede Kugel kann als ein Ätom"gesehen werden und man erhält ein Molekül", indem man seine Atome miteinander vereinigt.

Folgender WebGL Link zeigt ein 3D Modell mit zwei verschmolzenen Kugeln: WebGL.

Mehr im Abschnitt 3.1 . . .

2.9 2D und 3D Delaunay Triangulation



Die Delaunay Triangulation und das Voronoi Diagramm sind duale Strukturen und enthalten die gleiche Information in unterschiedlicher Form. Sie beinhalten die Information über Distanz bzw. Nähe einer Menge von Objekten im Raum.

Diese Strukturen werden in Anwendungen verwendet, wie etwa die Ermittlung der nächstgelegenen Nachbarobjekte, für winkelmaximierte Triangulationen, für die Berechnung größtmöglicher leerer Kreise, den Aufbau minimaler spannender Bäume oder für das Reisendenproblem (traveling salesperson paths).

Mehr im Abschnitt 3.2 . . .

2.10 JavaScript Image Linker

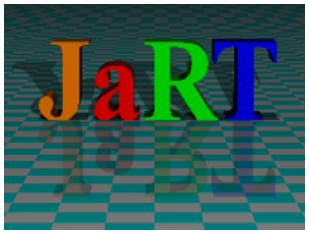
JaSIL ist ein Compiler/Linker der Java Klassen in JavaScript übersetzt und ein optimiertes Programm erzeugt, das in aktellen Web-Browsern ausgeführt werden kann. Als JDK wird derzeit Apache Harmony verwendet. Zudem wurden Java Packages entwickelt, um GUIs für Web-Browser zu erzeugen und um WebGL verwenden zu können. Optional werden preemtive Java Threads unterstützt. Aus den übersetzten Java Klassen wird ein kompaktes optimiertes JavaScript Programm erzeugt, das nur die unbedingt erforderlichen Teile enthält. Für die Speicherverwaltung wird der Garbage Collector von JavaScript verwendet.

Mehr im Abschnitt 3.3 . . .

2.11 WEB-Dock

Das WEB-Dock ermöglicht den schnellen und direkten Zugriff auf wichtige und/oder häufig verwendete Web-Resourcen. Im Vergleich zu ähnlichen Lösungen, wie die Task-List (Task-Dock) des *3D Desktops* (Abschnitt 2.14) oder Apple's Dock, ermöglicht der effiziente bildbasierte Ansatz die Verwendung des WEB-Docks in aktuellen Web-Browsern. Größenänderung der Icons, die verwendet werden kann, um die Icondichte zu erhöhen, und Drag & Drop werden durch den schnellen Austausch einzelner Bildelemente realisiert, ähnlich wie in einem Zeichentrickfilm. Wird der Mauszeiger über einem, auf diese Weise selektierten, Icon stehen gelassen, wird ein kurzer Hinweistext (hover help) angezeigt.

2.12 JaRT - ein Java Ray Tracer

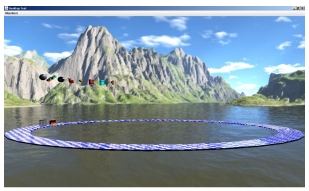


JaRT ist ein Ray Tracer, der mit Java entwickelt wurde, um GUI-Elemente mit 3D Effekten für Webseiten, wie etwa Rahmen mit abgerundeten Ecken und Kanten, zu erzeugen. JaRT unterstützt CSG Operationen für Grundelemente, wie Flächen, Quader, Kugeln, Cylinder, Kegel und für Dreiecksnetze, einschließlich 3D Text. Texturen können mit planaren, spherischen und cylindrischen Projektionen auf Objekte aufgetragen werden. Es werden die Phong/Blinn Lichtgleichungen verwendet, sowie Spiegelungen und Schatten unterstützt. Die Bildqualität kann mit Supersampling verbessert werden.

2.13 3D Desktop Browser

Die Struktur und die Elemente des *3D Desktops* (Abschnitt 2.14) können mit XML definiert werden. Der Desktop Browser verwendet die XML Beschreibung um eine 3D Ansicht zu erzeugen. Callbacks (z.B. für Menüs) können mit JavaScript angegeben werden. Das Document Object Model (DOM) des Desktops bildet eine Schnittstelle für den Zugriff auf Desktop Elemente.

2.14 3D Desktop



Aufbauend auf der Bibliothek für Echtzeitgraphik, die in Abschnitt 2.16 beschrieben wird, wurden Softwarekomponenten für einen 3D Desktop entwickelt. Die Desktop Komponenten unterstützen von Swing Schnittstellen abgeleitetes Drag & Drop und Kontextmenüs. Zudem werden Kreis- und Spirallisten von 3D Objekten mit Selektionen, sowie eine 3D Taskliste (Task-Dock) für aktivierte Funktionen unterstützt. Die abstrakten Listenmodelle entsprechen jenen von Swing. Ein 3D Verzeichnis stellt einen Raum zur Verfügung in dem 3D Objekte rund um ein Zentrum angeordnet werden können.

Screenshots im Abschnitt 3.4 . . .

2.15 Erzeugen von 3D Icons in Echtzeit

Durch die Echtzeitgraphik-Bibliothek (Abschnitt 2.16) wird die Erzeugung von 3D Icons erheblich beschleunigt. Ein komplexes Modell, für das mit dem Ray Tracer mehrere Stunden benötigt werden, kann nun in wenigen Minuten in ein Bild mit reaktiven 3D Elementen verwandelt werden. Die Bildqualität wird durch Antialiasing mittels Supersampling verbessert.

Für die Integration von 3D Modellen und Web-Content werden Layouts mit XML definiert und daraus HTML und JavaScript mit Funktionen für Kamerabewegungen oder Animationen generiert.

2.16 Echtzeitgraphik

Basierend auf JOGL, eine Java/OpenGL [2, 25] Schnittstelle, wurde eine Bibliothek für Echtzeitgraphik entwickelt. Diese Komponenten unterstützen Spiegel- und Schatteneffekte. Shaderkomponenten wurden mit der OpenGL Shading Language (GLSL) [17] realisiert. Mit einem einfachen Textsystem können aus Texten dynamisch 2D Texturen erzeugt werden. Die Objektselektion erfolgt mit Unterstützung der Graphikhardware (hardware accelerated picking). 3D Modelle können aus Grundelementen mit Hilfe von Constructive Solid Geometry (CSG) Operationen konstruiert werden. Zudem können mit Blender erzeugte 3D Objekte verwendet werden.

2.17 Erzeugen von 3D Icons mittels Ray Tracing

Im Rahmen dieses Projekts wurden Werkzeuge für die Erzeugung von Bildern von 3D Modellen mit sensitiven Bereichen (3D icons) entwickelt, die in Web-Designs verwendet werden können. Solche 3D Elemente werden mit der Programmiersprache Java definiert und mit dem Ray Tracer POV-Ray generiert. Reaktive Bereiche werden durch Zuordnung von SSensorenßu 3D Objekten definiert, für die dann automatisch entsprechender HTML code erzeugt wird. Spezielle Panoramakameraabstraktionen werden verwendet, um verzerrungsfreie Weitwinkelsichten zu erzeugen.

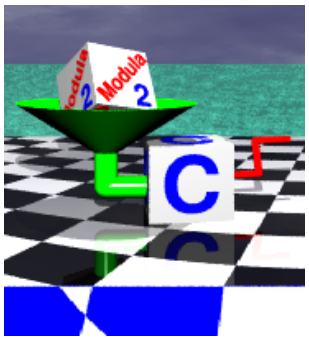
Ein Beispiel generiert mit POV-Ray.

2.18 Modula-3 für OS/2 und MS DOS

Das Modula-3 System (Compiler, Laufzeitsystem, Bibliothek), welches vom Systems Research Center der Firma Digital (DEC-SRC) entwickelt wurde, wurde auf OS/2 (Warp und Merlin) und MS DOS portiert. Carsten Whimster hat einen Artikel über das Modula-3 für OS/2 Projekt im Electronic Development Magazine, OS/2 Ausgabe (EDM/2), veröffentlicht. Die Portierung auf DOS wurde in der Lehre eingesetzt.

- Carsten's Artikel in EDM/2 in Englisch.
- Download Modula-3 für Warp oder Merlin.
- Ankündigungen von Modula-3 für OS/2 und MS DOS.
- Mehr Information über Modula-3 und weitere Literatur [6, 14, 22].

2.19 Modula-2 - C Übersetzer



Im Rahmen dieses Projektes wurde ein Modula-2 - C Übersetzer entwickelt, der in der Lage ist, sich selbst in C zu übersetzen. Der sehr portable Übersetzer ist für MS DOS, UNIX und VMS verfügbar.

- Eine Einführung in die Syntaxanalyse .
- Eine umfangreiche Projektbeschreibung .
- Mehr Information zu Modula-2 und weitere Literatur [4, 30].

3 Details zu einzelnen Projekten

3.1 3D Rekonstruktion mit impliziten Flächen

Eine implizite Fläche wird durch eine implizite Funktion definiert, die eine stetige Funktion über $\mathbf{R}^3$ ist. Die Kugel mit Radius 1 (unit sphere) wird z.B. durch die implizite Funktion $f(\mathbf{p}) = 1 \leq |\mathbf{p}|$ definiert, für alle Punkte $\mathbf{p}$, die Element von $\mathbf{R}^3$ sind. Für Punkte auf der Kugeloberfläche ist $f(p) = 0$. Eine kurze Einführung über implizite Flächen findet sich in [1]. Eine wichtige Klasse von impliziten Flächen sind die sogenannten "blobby surfaces"[5]. Solche Flächen können mathematisch folgendermaßen beschrieben werden:

$$f(\mathbf{p}) = \sum_{i=0}^{n-1} h(r_i)$$

wobei n die Anzahl der Objekte ist und für jedes dieser Objekte eine Distanz r_i berechnet wird, welche zum Beispiel die Distanz von $\mathbf{p}$ zu diesen Objekten sein kann. Die Funktion h beschreibt den Einflußbereich des Objektes i. Für Kugeln kann r_i zum Beispiel die Distanz von $\mathbf{p}$ zum Zentrum der Kugel i sein.

Abbildung 1: Zwei verschmolzene Kugeln.

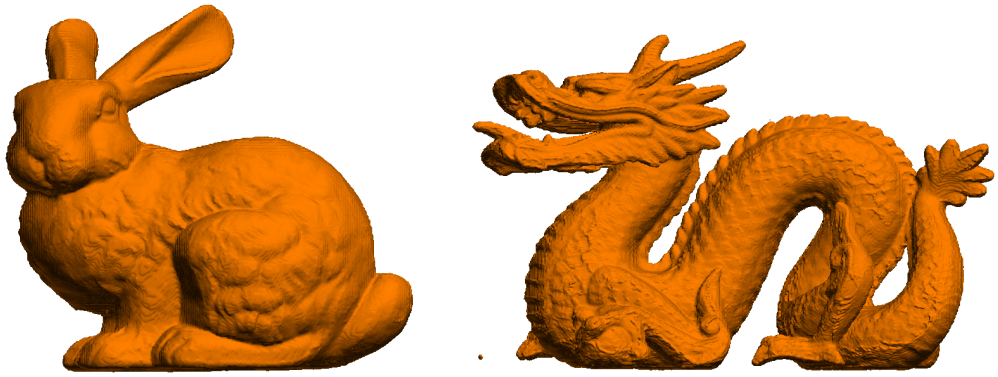
In diesem Fall, beschreibt in der oben angegebenen Gleichung eine einzelne Funktion h_i das Profil einer "blobby sphere", welche eine Gauß Funktion, mit einem bestimmten Zentrum und Standardabweichung, darstellt. Wenn die Zentren von zwei, auf diese Weise definierten, Kugeln nahe genug zusammen liegen, ergibt sich der Eindruck, als ob die beiden Kugeln miteinander verschmolzen sind.

Die Abbildung 1 zeigt zwei verschmolzenen Kugeln. Alle Punkte auf und zwischen den Kugeln, befinden sich im Einflußbereich der beiden Zentren und formen auf diese Weise die Fläche. Folgender WebGL Link zeigt ein entsprechendes 3D Modell: WebGL.

Um die Oberfläche zu rekonstruieren wird ein dreidimensionales Netz über den Bereich gelegt, wo sich die Fläche befindet und der Wert der "Verschmelzungsfunktionän jedem Netzpunkt evaluiert. Eine bekannte

Methode, um Dreiecksnetze aus solchen impliziten Flächen zu erzeugen ist der Marching Cubes [20] Algorithmus.

Die folgenden Rekonstruktionsbeispiele (Abbildung 2) basieren auf den Scandaten des Stanford 3D Scanning Repository, verwendet mit freundlicher Genehmigung des Stanford Computer Graphics Laboratory. Die Rotations- und Positionierungsinformation, die mit diesen Scandaten verfügbar ist, wurde verwendet, um die einzelnen Teile so zu verschieben, daß sie die Punktmenge bilden, die als Ausgangspunkt für die Rekonstruktion diente. Die Information über die Verbindung einzelner Punkte wurde nicht verwendet. Die Punktmenge wurde auf diese Weise als nicht orientierte Punktwolke verwendet.



(a) Hase rekonstruiert aus 362.272 Punkten. (b) Drache rekonstruiert aus 1.235.765 Punkten.

Abbildung 2: Rekonstruktionsbeispiele

Die nachfolgende Abbildung 3 zeigt das Verhalten von verschiedenen Rekonstruktionsmethoden in Bezug auf Löcher in der Oberfläche. Was am Besten ist, hängt von den Anforderungen des Anwendungsgebietes ab. Ein Loch ist entweder beabsichtigt (z.B. gebohrt) oder kann entstehen, wenn der Scanner keine oder zu wenig Punkte geliefert hat. Der Hase wird von unten gezeigt, wo er zwei Löcher und Bereiche mit geringer Datendichte in der Bodenplatte hat.

Das linke Bild 3a wurde mit Zipper erzeugt [26]. Dieses Netz zeigt die erwähnten Löcher und ist nicht geschlossen (wasserdicht).

Der mittlere Hase (Bild 3b) ist jener, der mit der oben beschriebenen, relativ einfachen, Methode aus impliziten Flächen erzeugt wurde. Das Netz dieses Hasen ist wasserdicht und zeigt die Löcher. Er hat eine doppelseitige Hülle.

Für den rechten Hasen (Bild 3c) wurde die Poisson Flächenrekonstruktion verwendet [16]. Diese Methode erzeugt ein wasserdichtes Netz mit ge-

schlossenen Löchern.

(a) Zipper (b) Der Ansatz mit impliziten Flächen. (c) Poisson

Abbildung 3: Rekonstruktionsbeispiele

Die folgenden WebGL Links zeigen 3D Modelle der Rekonstruktionsbeispiele[1]: Zipper WebGL, Ansatz mit impliziten Flächen WebGL und Poisson WebGL.

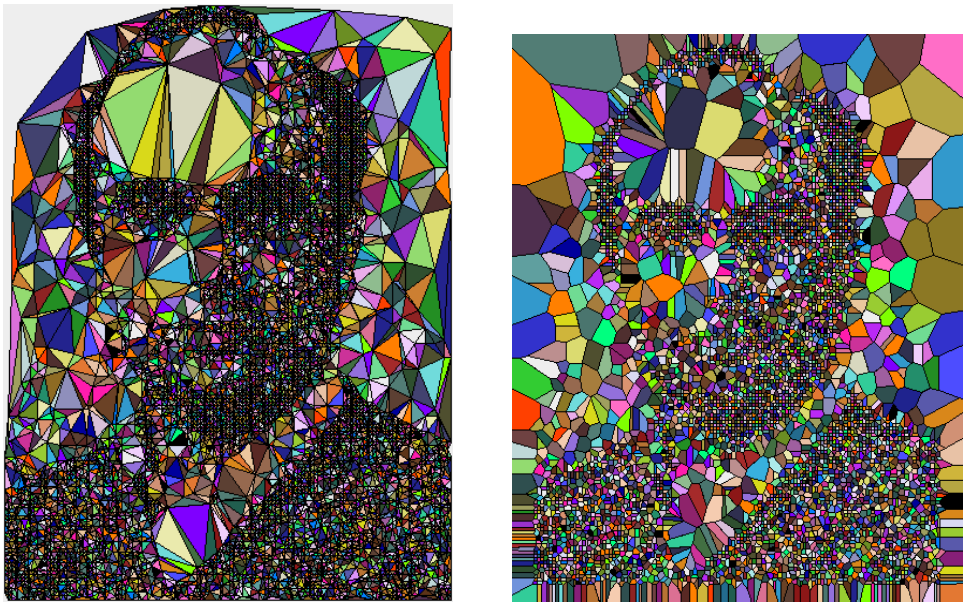
[1] Netzqualität reduziert auf ca. 35K/70K Knoten/Flächen.

3.2 2D und 3D Delaunay Triangulation

Die Delaunay Triangulation und das Voronoi Diagramm sind duale Strukturen und enthalten die gleiche Information in unterschiedlicher Form. Wird eine der beiden Strukturen berechnet, so entsteht automatisch auch die andere. Es besteht ein Zusammenhang zwischen Delaunay Triangulationen und konvexen Hüllen in der jeweils nächsthöheren Dimension. Die Delaunay Triangulation und das Voronoi Diagramm in d Dimensionen können aus einer konvexen Hülle in d+1 Dimensionen konstruiert werden. Die Delaunay Triangulation einer Menge von d-dimensionalen Punkten ist die Projektion der Punkte der Hülle in d+1 Dimensionen. Eine Einführung findet sich in [23].

2D Delaunay Triangulation

Die 2D Delaunay Triangulation wurde in zwei Varianten implementiert. Einerseits als inkrementelle Variante [21] und andererseits rekursiv mit der "divide and conquerSStrategie [13]. Die beiden nachfolgenden Bilder zeigen eine Delaunay Triangulation (Abbildung 4a) und das entsprechende Voronoi Diagramm (Abbildung 4b), erzeugt aus 5964 Punkten.



(a) Delaunay Triangulation (b) Voronoi Diagramm

Abbildung 4: Delaunay Triangulation oder Voronoi Diagramm in 2D

Die Abbildung 5 zeigt beides, die Delaunay Triangulation und das Voronoi Diagramm, in einem Bild, für eine Menge von 10 Punkten.

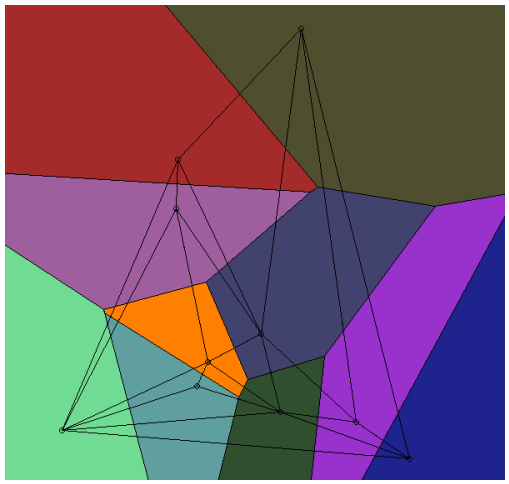


Abbildung 5: Delaunay Triangulation und Voronoi Diagramm in 2D

Voronoi Diagramm auf der Kugel

Das folgende Bild 6 zeigt ein Voronoi Diagramm, das aus den Positionen der Hauptstädte von 228 Ländern (einschließlich Überseeterritorien und ähnliches) erzeugt wurde und überlagert mit der Weltkarte dargestellt wird. Die roten Punkte zeigen die GPS Positionen der Hauptstädte, die mit Hilfe des WGS-84 [29] Ellipsoids in 3D Punkte transformiert wurden. Das Diagramm wurde aus der konvexen Hülle dieser Punkte erzeugt. Folgender WebGL Link zeigt ein 3D Modell mit diesem Voronoi Diagramm: WebGL. Bild 7 zeigt dieses Voronoi Diagramm für die gesamte Welt in 2D.

3D Delaunay Triangulation

Die Implementierung der 3D Delaunay Triangulation beruht auf dem Algorithmus, der in [9] beschrieben ist. Eine ausführlichere Beschreibung findet sich in [3].

Die folgenden beiden Bilder 8a und 8b zeigen den bekannten SStandord Bunny". Das rechte Bild 8b zeigt eine 3D Delaunay Triangulation von 71 gewichteten Punkten seiner Punktmenge.

Die 3D Delaunay Triangulation von Punktmengen mit Symmetrien kann mit Hilfe von *Simulation of Simplicity* [8] erzeugt werden. Die fol-

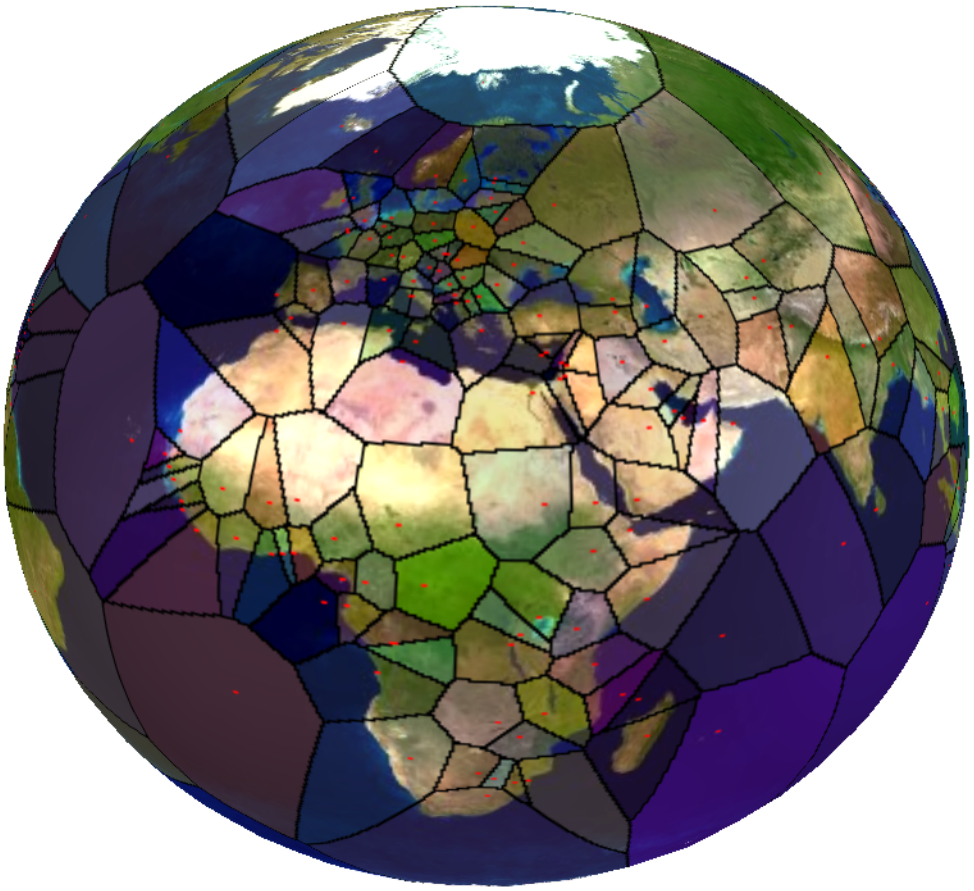


Abbildung 6: Voronoi Diagramm auf der Kugel

genden zwei Bilder 9a und 9b zeigen einen Ikosaeder. Das rechte Bild 9b ist eine mögliche 3D Delaunay Triangulation dieses platonischen Körpers.

Die folgenden WebGL Links zeigen 3D Modelle der Delaunay Triangulationen: Bunny WebGL und Ikosaeder WebGL.

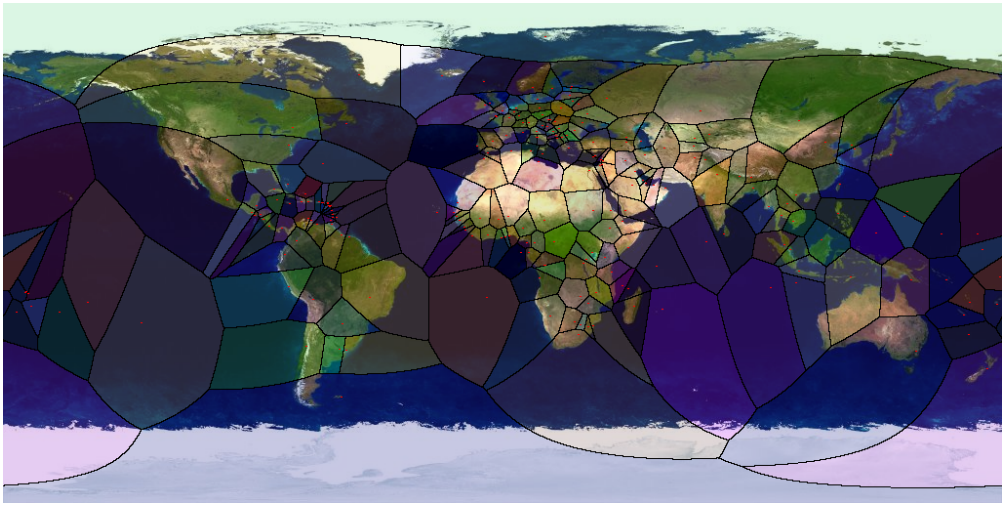
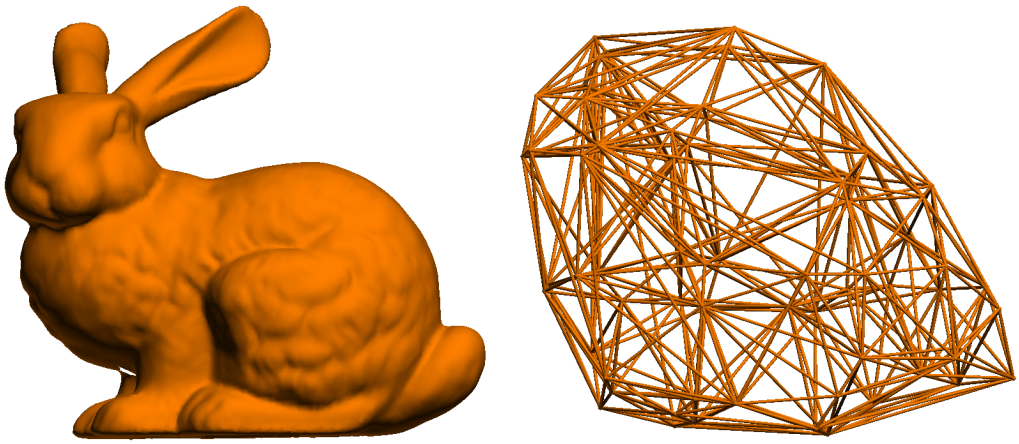
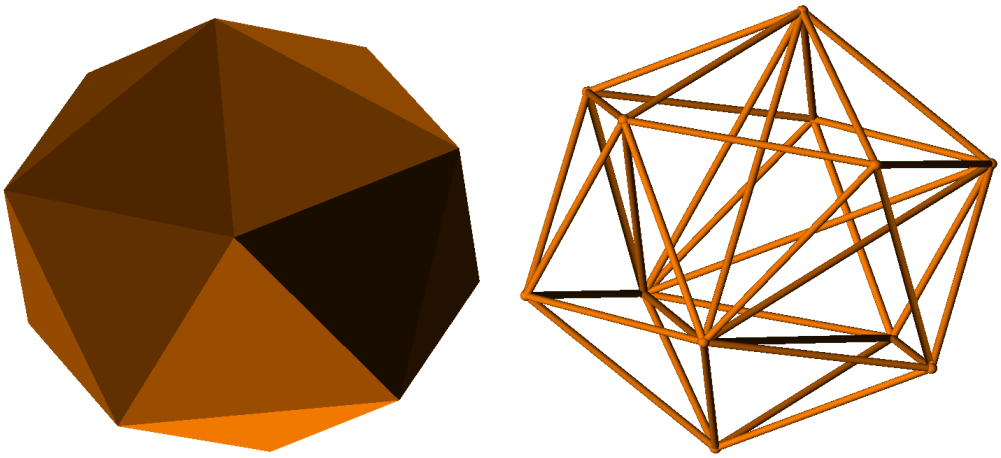


Abbildung 7: Voronoi Diagramm für die gesamte Welt



(a) Stanford Bunny (b) 3D Delaunay Triangulation des Hasen

Abbildung 8: Bunny 3D Delaunay Triangulation



(a) Ikosaeder

(b) 3D Delaunay Triangulation des Ikosaeders

Abbildung 9: Ikosaeder 3D Delaunay Triangulation

3.3 JavaScript Image Linker

Der JaSIL Compiler/Linker übersetzt Java Anwendungen in JavaScript. Als JDK wird derzeit Apache Harmony verwendet. Das vom Linker erzeugte optimierte Programm enthält nur die referenzierten Klassen, Felder und Methoden. Das JaSIL Projekt bietet die Möglichkeit preemitve Java Threads mit JavaScript in Clients von Web-Applikationen zu verwenden.

JaSIL übersetzt Java Bytecode in JavaScript und verwendet daher einen allgemeinerer Ansatz als ähnliche Projekte mit speziellen Java Front-Ends. Das System verwendet den Garbage Collector von JavaScript für die Speicherverwaltung.

Motivation für die Übersetzung von Java in JavaScript

JavaScript ist eine funktionale Sprache, die in allen populären Web-Browsern verfügbar ist. Auf diese Weise ist diese Sprache eine der weltweit meistgenutzten Plattformen. Java ist eine populäre objektorientierte Programmiersprache mit einem starken Typkonzept, die ebenfalls auf vielen verschieden Plattformen verfügbar ist. Sehr gute IDE Unterstützung erlaubt es EntwicklerInnen auf effiziente Weise komplexe Java Anwendungen zu erzeugen und zu warten. Auch wenn Java die Entwicklung von komplexen Web-Anwendungen unterstützt, ist für bestehende Java Programme die Übersetzung auf JavaScript allein in der Regel nicht ausreichend, sondern darüber hinaus eine spezielle Anpassung an die Web-Browser Umgebung erforderlich.

Java/JVM Kompatibilität

JaSIL bietet Optionen für mehr oder weniger Java/JVM Kompatibilität des generierten JavaScript Codes. So ist die Unterstützung für preemtive Java Threads, einschließlich Synchronisierung optional. Derartige Threads sind vollständig durch Software realisiert und bieten kein echtes hardwarebasiertes Multiprocessing. Es ist gegenwärtig nicht beabsichtigt dynamisches Laden von Klassen zu unterstützen und Reflection für Java Klassen ist nur begrenzt möglich. Wenn keine Threads benötigt werden, so kann optional durch die Eliminierung von Sprüngen [10] effizienterer Code generiert werden, wodurch die Laufzeit des Ja-

vaScript Codes um etwa 11 % verbessert wird.

Ein Beispiel mit Threads

Über den Link Wortanzahl Demo wird ein Dialog geöffnet und eine einfache Wortzählfunktion gestartet. Es wird eine Datei vom Server gelesen, die Anzahl der einzelnen Worte in der Datei ermittelt und das Resultat in einer Tabelle angezeigt. Das Schließen des Dialogs beendet die Funktion. Für diese Aufgabe wird ein Thread mit niedriger Priorität verwendet.

Durch die Verwendung von Threads bleibt die GUI reaktiv, der Dialog kann bewegt oder auf das WEB-Dock minimiert werden und die Tabelle wird permanent aktualisiert. Java EntwicklerInnen können die Thread-Konzepte verwenden, die ihnen bereits vertraut sind. Da das System Threads mit Prioritäten unterstützt, kann GUI Aktivität das Worte zählenden Thread bremsen.

Beispiel with WebGL

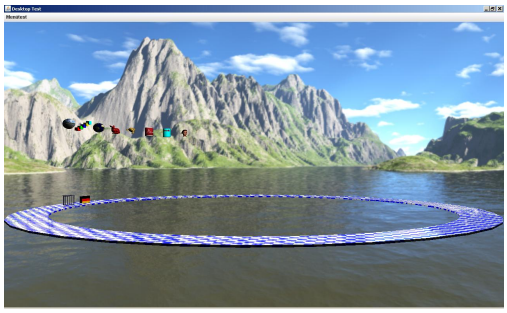
Über den Link WebGL Demo wird ein Dialog geöffnet und eine WebGL Anwendung gestartet. Threads werden in diesem Beispiel keine verwendet, was bedeutet, dass die Anwendung im "GUI Threadläuft.

Ähnliche Projekte

- TeaVM
- Emscripten SDK
- Google Web Toolkit (GWT)
- Java2Script (J2S)
- Links: Web Programming Without Tiers

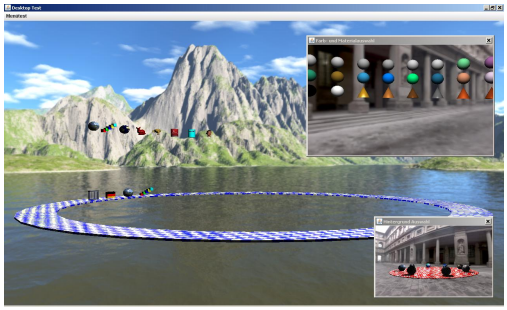
3.4 3D Desktop

Desktop mit 3D Task-List (Task-Dock)



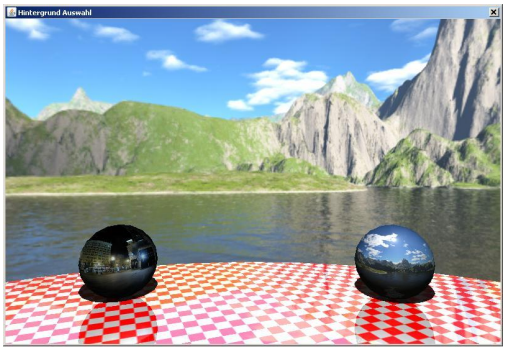
Das Bild zeigt die 3D Task-List mit dem Recycler und der Box für die Sprachauswahl. Über der Task-List befindet sich eine Reihe von 3D Icons. Der Betrachter befindet sich außerhalb des Desktopzentrums.

Desktop mit Hintergrund- und Farb-/Material- Auswahl



Das Bild zeigt den 3D Desktop mit geöffneten 3D Ordnern für die Hintergrund- und Farb-/Material- Auswahl. Die 3D Icons der geöffneten Ordner befinden sich auf der Task-List.

Hintergrundauswahl



Der Ordner für die Hintergrundauswahl bietet eine Auswahl von Cubemaps, die auf Kugeln projiziert werden. Der Hintergrund eines geöffneten 3D Ordners wird geändert, indem eine dieser Kugeln mittels Drag & Drop hineingezogen wird.

Innerhalb des 3D Desktops

Das Bild zeigt den 3D Desktop mit geändertem Hintergrund und einem weiteren geöffneten 3D Ordner. Der geöffnete Ordner ist in der 3D Icon Liste selektiert. Die Betrachterin befindet sich im Desktopzentrum.

4 Über den Autor

Klaus Preschern hat Softwarekomponenten zur Resourcenplanung (Enterprise Resource Planning - ERP), für das Kundenbeziehungsmanagement (Customer Relationship Management - CRM) und die Hostkommunikation entwickelt, die von Banken und deren Rechenzentren verwendet werden. Zudem entwickelt er Software im Bereich Compilerbau, Laufzeitsysteme, Computer Aided Design (CAD) und Visualisierung.

Er war Mitglied von verschiedenen Entwicklungsteams in denen ISO-9001 zertifizierte Software Life Cycles mit entsprechenden Qualitätsprüfungen verwendet wurden. Für die Programmierung werden hauptsächlich objektorientierte Sprachen, wie C/C++, C#, Smalltalk und Java verwendet.

Seine Interessen reichen von Algorithmen und Datenstrukturen, Programmiersprachen, Compilerbau und Laufzeitsystemen, Content-Management und Web-Applikationen, Computergraphik und Computergeometrie, bis hin zu Benutzerschnittstellen und Betriebssystemen.

5 Verzeichnis der PDF Dokumente

preschern.org (Deutsche Version):  .

preschern.org (Englische Version):  .

Beispiel zu CSG Operationen mit impliziten Flächen:  .

3D Modell von zwei verschmolzenen impliziten Flächen:  .

Voronoi Diagramm auf der Kugel:  .

Beispiele zu 3D Delaunay Triangulation:  .

Beispiele von 3D Rekonstruktionen:  .

Einführung in die Syntaxanalyse:  .

Modula-2 - C Übersetzer Report:  .

6 Rechtshinweis

6.1 Inhalt des Onlineangebotes

Der Autor übernimmt keinerlei Gewähr für die Aktualität, Korrektheit, Vollständigkeit oder Qualität der bereitgestellten Information. Haftungsansprüche gegen den Autor, welche sich auf Schäden materieller oder ideeller Art beziehen, die durch die Nutzung oder Nichtnutzung der dargebotenen Information bzw. durch die Nutzung fehlerhafter und unvollständiger Information verursacht wurden, sind grundsätzlich ausgeschlossen, sofern seitens des Autors kein nachweislich vorsätzliches oder grob fahrlässiges Verschulden vorliegt.

Alle Angebote sind freibleibend und unverbindlich. Der Autor behält es sich ausdrücklich vor, Teile der Seiten oder das gesamte Angebot ohne gesonderte Ankündigung zu verändern, zu ergänzen, zu löschen oder die Veröffentlichung zeitweise oder endgültig einzustellen.

6.2 Verweise und Links

Bei direkten oder indirekten Verweisen auf fremde Webseiten ("Hyperlinks"), die außerhalb des Verantwortungsbereiches des Autors liegen, würde eine Haftungsverpflichtung ausschließlich in dem Fall in Kraft treten, in dem der Autor von den Inhalten Kenntnis hat und es ihm technisch möglich und zumutbar wäre, die Nutzung im Falle rechtswidriger Inhalte zu verhindern.

Der Autor erklärt hiermit ausdrücklich, daß für ihn zum Zeitpunkt der Linksetzung keine illegalen Inhalte auf den zu verlinkenden Seiten erkennbar waren. Auf die aktuelle und zukünftige Gestaltung, die Inhalte oder die Urheberschaft der verlinkten/verknüpften Seiten hat der Autor keinerlei Einfluß. Deshalb distanziert er sich hiermit ausdrücklich von allen Inhalten aller verlinkten/verknüpften Seiten, die nach der Linksetzung verändert wurden. Diese Feststellung gilt für alle innerhalb des eigenen Internetangebotes gesetzten Links und Verweise sowie für Fremdeinträge in vom Autor eingerichteten Gästebüchern, Diskussionsforen, Linkverzeichnissen, Mailinglisten und in allen anderen Formen von Datenbanken, auf deren Inhalt externe Schreibzugriffe möglich sind. Für illegale, fehlerhafte oder unvollständige Inhalte und insbesondere für Schäden, die aus der Nutzung oder Nichtnutzung solcherart dargebotener Information entstehen, haftet allein der Anbieter der Seite, auf welche verwiesen wurde, nicht derjenige, der über Links auf die jeweilige Veröffentlichung lediglich verweist.

6.3 Urheber- und Kennzeichenrecht

Der Autor ist bestrebt, in allen Publikationen die Urheberrechte der verwendeten Bilder, Grafiken, Tondokumente, Videosequenzen und Texte zu beachten, von ihm selbst erstellte Bilder, Grafiken, Tondokumente, Videosequenzen und Texte zu nutzen oder auf lizenzfreie Grafiken, Tondokumente, Videosequenzen und Texte zurückzugreifen.

Alle innerhalb des Internetangebotes genannten und ggf. durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichenrechts und den Besitzrechten der jeweiligen eingetragenen Eigentümer. Allein aufgrund der bloßen Nennung ist nicht der Schluß zu ziehen, daß Markenzeichen nicht durch Rechte Dritter geschützt sind!

Alle Rechte für veröffentlichte, vom Autor selbst erstellte Objekte bleiben allein beim Autor der Seiten. Eine Vervielfältigung oder Verwendung solcher Grafiken, Tondokumente, Videosequenzen und Texte in anderen elektronischen oder gedruckten Publikationen ist ohne ausdrückliche Zustimmung des Autors nicht gestattet.

6.4 Datenschutz

Sofern innerhalb des Internetangebotes die Möglichkeit zur Eingabe persönlicher oder geschäftlicher Daten (Emailadressen, Namen, Anschriften) besteht, so erfolgt die Preisgabe dieser Daten seitens des Nutzers auf ausdrücklich freiwilliger Basis. Die Inanspruchnahme und Bezahlung aller angebotenen Dienste ist - soweit technisch möglich und zumutbar - auch ohne Angabe solcher Daten bzw. unter Angabe anonymisierter Daten oder eines Pseudonyms gestattet. Die Nutzung der im Rahmen des Impressums oder vergleichbarer Angaben veröffentlichten Kontaktdaten wie Postanschriften, Telefon- und Faxnummern sowie Emailadressen durch Dritte, zur Übersendung von nicht ausdrücklich angeforderten Informationen, ist nicht gestattet. Rechtliche Schritte gegen die Versender von sogenannten Spam-Mails bei Verstößen gegen dieses Verbot sind ausdrücklich vorbehalten.

6.5 Rechtswirksamkeit dieses Haftungsausschlusses

Dieser Haftungsausschluß ist als Teil des Internetangebotes zu betrachten, von dem aus auf diese Seite verwiesen wurde. Sofern Teile oder einzelne Formulierungen dieses Textes der geltenden Rechtslage

nicht, nicht mehr oder nicht vollständig entsprechen sollten, bleiben die übrigen Teile des Dokumentes in ihrem Inhalt und ihrer Gültigkeit davon unberührt.

7 Literatur

[1] Tomas Akenine-Möller and Eric Haines: *Real-Time Rendering*, 2nd edition, A.K. Peters, 2002, Pages 526-527.

[2] Ken Arnold and James Gosling: *The Java ™ Programming Language*, Addison-Wesley, 1996.

[3] Christoph Baudson und Edgar Klein: *Berechnung und Visualisierung von Voronoi Diagrammen in 3D*, Diplomarbeit an der Rheinischen Friedrich-Wilhelms-Universität Bonn, 2006.

[4] Günther Blaschek, Gustav Pomberger und Franz Ritzinger: *Einführung in die Programmierung mit Modula-2*, Springer Verlag, 1986.

[5] James F. Blinn: *A Generalization of Algebraic Surface Drawing*, ACM Transactions on Graphics, Vol. 1, No. 3, July 1982, Pages 235-256.

[6] Luca Cardelli, James Donahue, Lucille Glassman, Mick Jordan, Bill Kalsow and Greg Nelson: *Modula-3 Report (revised)*, Systems Research Center, Research Report 52, 1989.

[7] Ecma International: *Standard ECMA-363: Universal 3D File Format*, Rue du Rhône 114, CH-1204 Geneva, 4th Edition, June 2007.

[8] Herbert Edelsbrunner and Ernst Peter Mücke: *Simulation of Simplicity: A Technique to Cope with Degenerate Cases in Geometric Algorithms*, ACM Transactions on Graphics, 9(1), 1990, Pages 66-104.

[9] Herbert Edelsbrunner and Nimish R. Shah: *Incremental Topological Flipping Works for Regular Triangulations*, Algorithmica 15, 1996, Pages 223–241, First published in: Proceedings of the 8th Annual ACM Symposium on Computational Geometry, 1992, Pages 43–52.

[10] Ana Maria Erosa: *A Goto-Elimination Method and its Implementation for the McCAT C Compiler*, Master Thesis, School of Computer Science, McGill University, Montreal, May 1995.

[11] Paul Fishwick: *Exploring Multiple Visualization Perspectives with Aesthetic Computing*, International Conference for Visual Languages and Computing, 2003.

[12] The Khronos 3D Formats Working Group: *glTF 2.0 Specification*, Khronos Group.

[13] Leonidas Guibas and Jorge Stolfi: *Primitives for the Manipulation of Generail Subdivisions and the Computation of Voronoi Diagrams*, ACM Transactions on Graphics, Vol. 4, No. 2, Pages 74-123, April 1985.

[14] Samuel P. Harbison: *Modula-3*, Prentice Hall, 1992.

[15] Tao Ju, Frank Losasso, Scott Schaefer and Joe Warren: *Dual contouring of hermite data*, ACM Transactions on Graphics, Volume 21, Number 3, Pages 339–346, Proceedings of ACM SIGGRAPH July 2002.

[16] Michael Kazhdan, Matthew Bolitho and Hugues Hoppe: *Poisson Surface Reconstruction*, Eurographics Symposium on Geometry Processing, 2006.

[17] John Kessenich (Editor): *The OpenGL ® Shading Language (Language Version 4.0)*, The Khronos Group Inc., 2010.

[18] Donald E. Knuth: *The TEXbook*, Addison-Wesley, Reading, Massachusetts, 1986.

[19] Leslie Lamport: *LATEX: A Document Preparation System*, Addison-Wesley, Reading, Massachusetts, 2nd edition, 1994.

[20] William E. Lorensen and Harvey E. Cline: *Marching cubes: A high resolution 3D surface reconstruction algorithm*, SIGGRAPH Computer Graphics, Volume 21, Number 4, Pages 163–169, July 1987.

[21] Dani Lischinski: *Incremental Delaunay Triangulation*, Graphics Gems IV, Editor: Paul S. Heckbert, AP Professional (Academic Press), Boston, 1994.

[22] Greg Nelson (Editor): *Systems Programming with Modula-3*, Prentice Hall Series in Innovative Technology, 1991.

[23] Joseph O'Rourke: *Computational Geometry in C*, 2nd edition, Cambridge University Press, 1998.

[24] Andreas Rossberg (Editor): *WebAssembly Specification*, Release 2.0, WebAssembly Community Group, 2023.

[25] Mark Segal and Kurt Akeley: *The OpenGL ® Graphics System: A Specification*, The Khronos Group Inc., 2010.

[26] Greg Turk and Marc Levoy: *Zippered polygon meshes from range images*, In Proceedings of SIGGRAPH '94 (Orlando, FL, July 24-29, 1994), pages 311–318. ACM Press, July 1994.

[27] Dean Jackson and Jeff Gilbert: *WebGL Specification 1.0 Editor's Draft*, Khronos Group.

[28] Dean Jackson and Jeff Gilbert: *WebGL Specification 2.0 Editor's Draft*, Khronos Group.

[29] National Imagery and Mapping Agency (NIMA): *Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems*, NIMA Technical Report TR8350.2, 3rd edition, Amendment 1, January 2000.

[30] Niklaus Wirth: *Programmieren in Modula-2*, Springer Verlag (Übers. d. 3rd corrected edition), 1985.