



## **Bachelor- /Projektarbeit (Studiengang, max. 2 Zeilen)**

Titel der Arbeit Titel der Arbeit Titel der  
Arbeit Titel der Arbeit Titel der Arbeit Titel  
der Arbeit Titel der Arbeit Titel der Arbeit  
Titel der Arbeit Titel der (max. 4 Zeilen)

<b>Autoren</b>	Vorname Name Vorname Name
<b>Hauptbetreuung</b>	Vorname Name Vorname Name
<b>Nebenbetreuung</b>	Vorname Name Vorname Name
<b>Industriepartner</b>	Firmenname
<b>Externe Betreuung</b>	Vorname Name Vorname Name
<b>Datum</b>	01.01.2012

Bitte füllen Sie das Titelblatt aus und berücksichtigen Sie Folgendes:

- Bitte auf keinen Fall Schriftart und Schriftgrösse ändern. Text soll lediglich überschrieben werden!
- Bitte pro Tabellenzeile max. 4 Textzeilen!
- Vorlage: Haben Sie die richtige Vorlage gewählt? → Logo Institut/Zentrum
- Titel: Fügen Sie Ihren Studiengang direkt nach dem Wort „Bachelorarbeit“ ein (max. 2 Zeilen).
- Titel der Arbeit: Überschreiben Sie den Lauftext mit dem Titel Ihrer Arbeit (max. 4 Zeilen).
- Autoren: Tragen Sie Ihre Vor- und Nachnamen ein (bitte alphabetisch nach Name).
- Betreuer: Tragen Sie Ihren Betreuer / Ihre Betreuer ein (bitte alphabetisch nach Name).
- Nebenbetreuung: Falls Sie keine Nebenbetreuung haben → bitte ganze Tabellenzeile löschen.
- Industriepartner: Falls Sie keinen Industriepartner haben → bitte ganze Tabellenzeile löschen.
- Externe Betreuung: Falls Sie keine ext. Betreuung haben → bitte ganze Tabellenzeile löschen.
- Datum: Bitte aktuelles Datum eintragen.
- Schluss: Am Schluss löschen Sie bitte den ganzen Beschrieb (grau) und speichern das Dokument als pdf. ab.

## Erklärung betreffend das selbständige Verfassen einer Bachelorarbeit an der School of Engineering

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinar massnahmen der Hochschulordnung in Kraft.

Ort, Datum:

.....

Unterschriften:

.....

.....

.....

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Bachelorarbeiten zu Beginn der Dokumentation nach dem Titelblatt mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

## **Zusammenfassung**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# Vorwort

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>6</b>
1.1. Ausgangslage . . . . .	6
1.2. Zielsetzung . . . . .	6
1.2.1. Subsection . . . . .	6
<b>2. Grundlagen</b>	<b>8</b>
2.1. Einführung . . . . .	8
2.1.1. Das Problem . . . . .	8
2.1.2. Lösungsansätze . . . . .	8
2.1.3. Verschiedene Ansätze zur Generierung von LOD Modellen . . . . .	9
<b>3. Vorgehen</b>	<b>11</b>
3.1. Nutzen LOD . . . . .	11
3.1.1. Definition Benchmark . . . . .	11
3.1.2. Profiling . . . . .	11
3.2. Vergleich LOD Systeme . . . . .	12
3.3. Vergleich LOD Algorithmen . . . . .	12
3.4. Pipeline Integration . . . . .	12
3.5. Automatische Generierung von Detail Levels . . . . .	12
3.6. Levelwahl während Laufzeit . . . . .	12
<b>4. Resultate</b>	<b>13</b>
<b>5. Diskussion und Ausblick</b>	<b>14</b>
<b>6. Verzeichnisse</b>	<b>15</b>
<b>A. Anhang</b>	<b>18</b>
A.1. Aufgabenstellung . . . . .	19
A.2. Anhang 2 . . . . .	20

# 1. Einleitung

## 1.1. Ausgangslage

Performance Optimierung ist im 3D Rendering unabdinglich. Die Komplexität der Modelle hat dabei einen signifikanten Einfluss auf die Leistung. Eine Möglichkeit zur Optimierung ist das generieren und verwenden von vereinfachten Modellen. In diversen Rendering Engines gibt es deshalb Möglichkeiten für das verwenden von Level Of Details (LOD) Artefakten. Dabei werden abhängig von Parametern vereinfachte Varianten desselben Modelles verwendet. So kann z.B. ein Objekt in grosser Distanz vereinfacht dargestellt werden, ohne merkbare Auswirkungen auf die Qualität zu haben. Für Engines wie Unreal oder Unity gibt es Möglichkeiten, um den Einsatz von LOD Artefakten zu vereinfachen. Im Web Bereich gibt es zur Zeit keine weit verbreitete Möglichkeit.

## 1.2. Zielsetzung

Ziel der Arbeit ist es, ein Tool zu entwickeln, das den Umgang mit LOD Artefakten im Web vereinfacht.

### 1.2.1. Subsection

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[?]



Abbildung 1.1.: Bildli

### SubSubSection

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Tabelle 1.1.: Eine Tabelle

A	B	C
1	2	3
4	5	6

### Paragraph

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



## 2. Grundlagen

### 2.1. Einführung

Als Level Of Detail (LOD) werden die verschiedenen Detailstufen bei der virtuellen Darstellung bezeichnet. Dies wird verwendet, um die Geschwindigkeit von Anwendungen zu steigern, indem Objekte im Nahbereich detailliert angezeigt werden; wohingegen Elemente im Fernbereich deutlich vereinfacht dargestellt werden.

#### 2.1.1. Das Problem

Geometrische Objekte können zu Komplex werden, um jederzeit performant und interaktiv gerendert zu werden. Gerade wenn viele Objekte zur selben Zeit sichtbar sind, lohnt es sich, zu priorisieren und gewisse Objekte in reduzierter Qualität anzuzeigen. Im Idealfall geschieht dies jedoch, ohne dass der Anwender dies bemerkt.

#### 2.1.2. Lösungsansätze

In diesem Abschnitt werden mögliche Ansätze erklärt, welche helfen sollen, die Render-Perfomanz zu erhöhen. Diese Arbeit konzentriert sich jedoch auf den Ansatz von Level of Detail; die anderen Ansätze werden nur kurz erläutert.

##### *Level of Detail*

Level of Detail auch bekannt als polygonale Simplifizierung, geometrische Simplifizierung oder Mesh Reduzierung basiert darauf, die Komplexität von Objekten zu reduzieren, welche weiter von der Kamera entfernt werden. Es gibt verschiedene Ansätze zur Generierung von LODs, welche in Abschnitt XY im Detail erläutert werden. Zudem braucht es einen berechenbare Methode, die Genauigkeit von Modellen zu definieren, um diese entsprechend anzuwenden. Schlussendlich ist dann noch zu definieren, ab wann welche Artefakte verwendet werden soll, basierend auf der Genauigkeit und der Komplexität.

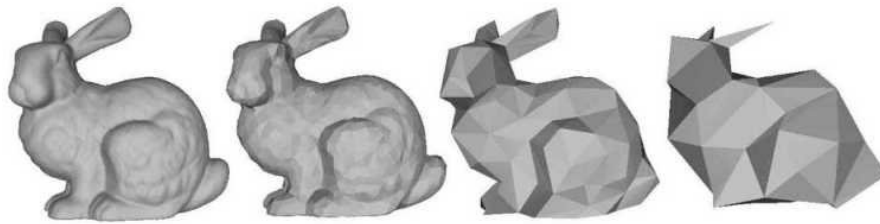


Abbildung 2.1.: Level Of Detail Visualisierung vier Hasen

Wie in der Abbildung 2.1 zu erkennen ist, wird von links nach rechts der Detailgrad und somit die Komplexität des Objektes reduziert. Sind es im Bild ganz links noch 69'451 Polygone, wird es bereits im ersten Schritt auf 2'502 Polygone reduziert. Dies ist eine enorme Reduktion von ca 96.5%. Im dritten Schritt wird die Anzahl Polygone wiederum um ca. 90% auf 251 reduziert. Schlussendlich hat das letzte Objekt noch 76 Polygone was knapp 0.1% der ursprünglichen Anzahl entspricht.

### ***Parallel rendering***

Lorem ipsum

### ***Frustum culling***

Polygone, welche nicht im Kamera Frustum enthalten sind, werden bei dieser Methode nicht weiter prozessiert. Dies reduziert die Anzahl Polygone drastisch.

### ***Occlusion culling***

Polygone bzw. Objekte, welche komplett von anderen Objekten überdeckt werden, werden bei dieser Variante nicht prozessiert.

### ***Backface culling***

Bei dieser Methode wird berechnet welche Polygone zur Kamera orientiert sind. Alle Polygone, welche in die andere Richtung zeigen werden nicht gezeichnet. Dies ist nicht immer gewünscht, für die meisten Anwendungen ist diese Optimierung jedoch aktiviert.

### ***Image-based rendering***

Lorem ipsum

## **2.1.3. Verschiedene Ansätze zur Generierung von LOD Modellen**

Es gibt verschiedene Ansätze, 3D-Modelle mittels LOD zu vereinfachen. In diesem Abschnitt werden einige davon detaillierter erläutert so wie ihre Vor- und Nachteile aufzeigt.

### ***Diskrete LOD (DLOD)***

Bei diskreten LOD werden für ein detailliertes Modell mehrere weniger detaillierte Modelle verwendet. Abhängig von der Distanz zum Betrachter wird das optimale Modell gewählt.

- + Simplizität: Keine Anpassungen am Scene Graph notwendig
- Harte Grenzen: Veränderung des Objektes kann merkbar sein
- Kein Clustering möglich: Probleme bei sehr grossen oder vielen kleinen Modellen

### ***Kontinuierliche LOD (CLOD)***

Im Gegensatz zu DLOD wird bei CLOD vereinfachende Veränderungen an einem Modell gespeichert.

- + Weiche Grenzen: Interpolation zwischen Auflösungen ist möglich
- Runtime Performance
- Kein Clustering möglich

### ***Hierarchische LOD (HLOD)***

Bei HLOD werden mehrere Objekte in einen Cluster gruppiert.

- + Clustering möglich

## 3. Vorgehen

### 3.1. Nutzen LOD

Um den Nutzen von LOD quantifizieren zu können, wird in einer ersten Phase ein Benchmark aufgestellt. Ziel ist es, das Laufzeitverhalten unter Einsatz eines optimierten Modells zu analysieren und somit den maximal möglichen Einfluss von LOD auf die Leistung klassifizieren zu können.

#### 3.1.1. Definition Benchmark

Benchmarking im Web Frontend wird in der Praxis eher selten eingesetzt. Im Gegensatz zum Backend wo zum Beispiel Lasttests zur Norm gehören. Es gibt jedoch einige Tools, die das Durchführen eines Benchmarks vereinfachen. In diesem Abschnitt werden die verschiedenen erwägten Optionen erläutert und die Abgrenzungen aufgezeigt.

Um den Umfang des Benchmarks überschaubar zu halten, wurde ausschliesslich ein Benchmark für Google Chrome entwickelt. Google Chrome basiert auf Chromium, dieselbe Engine, welche auch Microsoft Edge oder Opera verwenden. Einen Benchmark basierend auf Google Chrome liefert somit auch Indizien für diese beiden Browser, auch wenn gewisse Abweichungen möglich sind. Neben dem Marktführer Chrome sind Mozilla Firefox oder Safari von Apple ebenfalls Optionen. Jedoch wurde primär aufgrund des Marktanteils zugunsten von Google Chrome entschieden. Um die Tests durchzuführen, wird ein Testautomationstool benötigt; unter anderem der Einsatz von Selenium wurde in Erwägung gezogen. Der Vorteil von Selenium ist insbesondere, dass der Benchmark für weitere Browser ausgeweitet werden könnte. Da jedoch das Analysieren der GPU sowie CPU Daten stark vom System abhängig ist, wird in diesem Benchmark die im Google Chrome integrierten Chrome DevTools eingesetzt. Puppeteer ist eine Bibliothek, die eine vereinfachte Schnittstelle zu einer Headless Chrome Instanz bietet. Sie wird zudem direkt von Google entwickelt und bietet eine stabile Grundlage zur Kommunikation mit den DevTools.

#### 3.1.2. Profiling

Die Chrome DevTools erlauben es, ein detailliertes Profil einer Applikation anzulegen. Im Profil befinden sich Informationen zu CPU/GPU Auslastung aber auch generelle Informationen bzgl. der Rendering Engine werden gesammelt. Die Analyse dieser Daten ermöglicht es, eine Aussage zum Laufzeitverhalten einer Applikation zu tätigen.

#### **3.2. Vergleich LOD Systeme**

Die unterschiedlichen LOD Systeme bieten allesamt ihre Vor- und Nachteile. In diesem Schritt wird erläutert, welche Art LOD System in dieser Arbeit eingesetzt werden soll.

#### **3.3. Vergleich LOD Algorithmen**

Abhängig vom LOD System wird ein passender Algorithmus ausgesucht.

#### **3.4. Pipeline Integration**

Die Lösung soll in eine wiederverwendbare und konfigurierbare Pipeline integriert werden.

#### **3.5. Automatische Generierung von Detail Levels**

Damit für den Endbenutzer die Konfiguration übersichtlich bleibt, soll der Einsatz von zum Beispiel heuristischen Methoden hilfreiche Basiskonfigurationen liefern. Zum einen geht es hier um das Festlegen der Thresholds, aber auch das Definieren des Dezimierungs- / Vereinfachungsfaktors spielt eine wichtige Rolle.

#### **3.6. Levelwahl während Laufzeit**

Die optimale Levelwahl kann stark von der Laufzeitumgebung abhängig sein. So sollte zum Beispiel auf mobilen Geräten früher ein einfacheres Modell gezeigt werden als auf leistungsstarken Geräten. Deshalb eignet sich eine Wahl der Levels zur Laufzeit, um ein optimales Erlebnis auf allen Geräten zu ermöglichen.

## 4. Resultate

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 5. Diskussion und Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## 6. Verzeichnisse



# Abbildungsverzeichnis

1.1. Bildli . . . . .	7
2.1. Level Of Detail Visualisierung vier Hasen . . . . .	9

# Tabellenverzeichnis

1.1. Eine Tabelle . . . . .	7
-----------------------------	---



## **A. Anhang**

### **A.1. Aufgabenstellung**

Aufgabenstellung

## A.2. Anhang 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.