

LEVELS OF DETAIL

OLIVER GASSNER

INSTITUTE OF COMPUTER GRAPHICS AND ALGORITHMS
VIENNA UNIVERSITY OF TECHNOLOGY

e9726248@stud3.tuwien.ac.at

1. ABSTRAKT

Das Ziel dieses Dokumentes soll sein dem Leser einen Überblick zum Thema Levels of Details (LOD) zu geben. Es ist als Begleitmaterial und Ergänzung zum gleichnamigen Vortrag von Christian Troger und mir gedacht. Anhand einiger Anwendungsgebiete aus dem Bereich der dreidimensionalen Grafik wird erläutert was Levels of Details sind, und wozu die Techniken benutzt werden. Behandelt werden Geometrische Levels of Detail bei der Darstellung von Polygonmodellen, Animation Levels of Detail. Eine Anwendung von Levels of Detail ausserhalb des 3D-Bereichs sind Levels of Detail bei Rastergrafiken. Dieser Anwendungsbereich wird im Paper von Troger Christian beschrieben .

1.1. Keywords:

Levels of Detail, Geometrische LOD, Simulation LOD, Animation LOD

2. EINFÜHRUNG

2.1. Was sind Levels of Detail

Der Begriff Levels of Detail allein bedeutet nicht mehr als die direkte Übersetzung des Selben ins Deutsche, also Detailstufen. Als Verfahren im Bereich der Computergrafik, und hier vor allem im 3D-Bereich versteht man darunter die Anpassung der Detailstufe der Darstellung/Animation/Textur in Abhängigkeit von mehreren Faktoren. Es wird dabei versucht so vorzugehen, daß es dem Benutzer so wenig wie moeglich auffällt, das nicht alles in der selben Detailstufe dargestellt wird.

2.2. Warum werden Levels of Detail verwendet

Levels of Detail werden verwendet um die Performance einer Anwendung zu steigern, da auch heute noch

die Hardware eine zu begrenzte Leistungsfähigkeit hat um auf Detailanpassung/-reduktion zu verzichten. Das Problem ist zwar in den letzten Jahren kleiner geworden aber nicht verschwunden. Auch die steigende Beliebtheit von netzwerkfaehigen Computerspielen (Quake, Unreal Tournament, ...) tragen ihren Teil dazu bei, daß LOD noch immer ein Thema ist. Hier stellt die verfügbare Netzbandbreite fast immer den Flaschenhals dar. Man ist also gezwungen die Datenmenge zu reduzieren. Reduziert wird vor allem dort wo der Benutzer es am schwersten wahrnimmt (z.B. Objekte mit großem Abstand zum Benutzer, sehr kleine Objekte). Der Grundsatz kann man „So detailliert wie notwendig, so grob wie moeglich“ nennen.

2.3. Welche Arten von Levels of Detail gibt es

2.3.1. Geometric Levels of Detail

Darunter versteht man die Verwendung unterschiedlicher Polygonmodelle für die Darstellung des selben Objektes. Je nach benötigtem Detaillevel wird zwischen den Varianten umgeschaltet (je weiter weg sich ein Objekt befindet desto weniger Polygone werden verwendet). Diese unterschiedlichen Polygonmodelle koennen entweder vorberechnet in die Applikation integriert werden oder sie werden in der Anwendung dynamisch aus einem hochdetaillierten Modell heruntergerechnet. Dieser Ansatz ermöglicht zwar eine bessere Darstellungsqualität, da es im Prinzip moeglich ist eine stufenlose Reduzierung des Detailgrades zu erreichen, ist aber rechenintensiver. Im Computer- und Videospielen, die oft mit sehr vielen Objekte klarkommen müssen und auch dann noch eine "spielbar" und keine Diashow sein sollen wird meist der erste Ansatz gewählt .

2.4. Simulation Levels of Detail

Nachdem man die mit Geometrischen LOD's die darzustellende Polygonanzahl optimiert hat wird bei Simulation Levels of Detail das Simulationsverhalten der dargestellten Objekte angepaßt. Man nennt das auch *Behavior Levels of Detail* (siehe [ES1] und [ES2]). Dabei wird das Verhalten der Objekte pro Detailstufe definiert und in der Anwendung nach vorgegebenen Bedingungen gewechselt. Für Simulation Levels of Detail bleibt im allgemeinen nur der Weg die einzelnen Varianten vorge-

fertigt in der Anwendung zu halten, da es nicht mit vertretbarem Aufwand möglich ist sie dynamisch während des Ablaufes zu berechnen.

2.4.1. Animation Levels of Detail

Genauso wie es in der Praxis möglich ist für weit entfernte Objekte die Anzahl der Polygone zu reduzieren ohne daß es dem Benutzer allzu sehr auffällt, ist der selbe Ansatz auch für die Animation der Objekte möglich. Bei Objekten die ihrerseits wieder aus mehreren Teilen bestehen und gegeneinander bewegt werden (z.B. Darstellung eines Humanoiden Körpers mit beweglichen Armen/Beinen/Kopf/...) ist es ab einer bestimmten Entfernung möglich die Animation auf weniger Teile zu beschränken (z.B. anstatt beim gehen das Knie abzubiegen wird nur mehr ein gestrecktes Bein bewegt, bei noch größerer Entfernung reicht es den Körper als Ganzes zu bewegen). Auch ist es möglich die Anzahl der Positions-Updates pro Zeiteinheit zu reduzieren (z.B. ein sich drehendes Objekt wird nur alle 10 Durchläufe neu gerendert, hat sich dabei dafür um den 10fachen Winkel weiterbewegt als wenn es bei jedem Durchlauf aktualisiert wird (wenn man sehr nah am Objekt ist)).

2.4.2. Levels of Detail in 2D

LOD Methoden auf 2D-Grafiken werden in 3D-Anwendungen für die Auswahl der Texturen verwendet. Je nach Entfernung werden unterschiedlich aufgelöste Texturen eingesetzt. Beim Wechsel der Texturen wird versucht harte, sichtbare Sprünge zu vermeiden. Diese unterschiedlichen Texturen nennt man Mipmaps. Weitere LOD-Anwendungen für 2D sind in [TRO1] zu finden.

3. GEOMETRISCHE LEVELS OF DETAIL

3.1. Verwendung von mehreren vorgefertigten Objektrepräsentationen

Mit diesem Verfahren erreicht man im allgemeinen die größte Performance in der Applikation da keine zusätzliche Rechenzeit benötigt wird um die Detailstufe zu wechseln. Ein Argument das dagegen spricht ist der Aufwand der zum Erstellen der Modelle anfällt, vor allem wenn sehr viele unterschiedliche Objekte benötigt werden. Bei Computerspielen ist diese Anzahl aber im allg. nicht sehr hoch, sodaß meist dieser Weg gewählt wird.

3.2. Dynamische Generierung der Detailstufen zu Laufzeit der Applikation

Als ein möglicher Lösungsweg zur dynamischen Erzeugung von LOD-Repräsentationen möchte ich den in [ES3] beschriebenen Hierarchic Clusteringalgorithmus kurz beschreiben.

3.2.1. Hierarchic Clustering

Der Grundgedanke des Verfahrens ist ein gegebenes Polygonmodell in eine Baumstruktur zu überführen. Die einzelnen Levels of Detail werden danach aus dieser Datenstruktur generiert, und nicht mehr aus dem Original. Je nach gewünschtem Detaillevel werden mehr oder weniger Elemente des Baumes zu Clustern zusammengefaßt die jeweils durch einen Punkt repräsentiert werden.

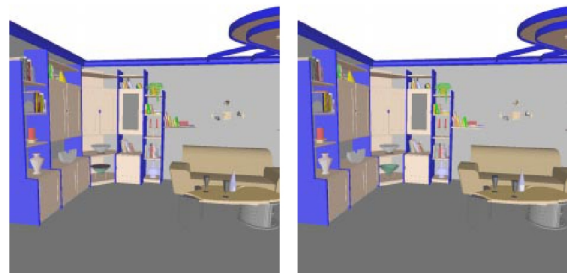


Abbildung 1 Raum mit/ohne LOD (links 6810, rechts 13847 Polygone)

3.2.1.A ERZEUGEN DER BAUMDARSTELLUNG

Der erste Schritt des Verfahrens ist das Erzeugen des Baumes der die einzelnen Vertices des Objektes enthält.

- Die Vertices des Originalmodells stellen die Blätter(leaves) des Baumes dar.

- Es werden nun im Modell jene zwei Punkte gesucht die den geringsten Abstand zueinander aufweisen. Sind diese Punkte gefunden werden sie zu einem Cluster zusammengefaßt der im weiteren diese Punkte repräsentiert.
- Dieser Schritt wird solange wiederholt bis nur mehr ein Cluster übrig bleibt, der dann das gesamte Objekt repräsentiert.

Als Ergebnis erhält man einen Baum dessen Blätter (Leaves) die Originalpunkte des Modell sind, und dessen Wurzelknoten (Root) das gesamte Objekt repräsentiert.

Die Position des bei der Verbindung von 2 Punkten/Clustern entstehenden Clusters wird wie folgt berechnet:

$$g = \frac{|i|g_i + |j|g_j}{|i| + |j|} \quad |i| \dots \text{Anzahl der Punkte innerhalb}$$

von Cluster i (falls i ein Punkt war 1).

Als Maß für den Abstand gilt

$$d = d(i, j) = \|g_i - g_j\|$$

3.2.1.B AUTOMATISCHE GENERIERUNG VON LOD REPRÄSENTATIONEN

- Der im vorigen Schritt generierte Baum dient als Ausgangsbasis zur Generierung eines angenäherten Objektes. Dabei wird im Prinzip der umgekehrte Algorithmus durchlaufen. Es wird ein Abstand definiert bis zu welchem Details berücksichtigt werden sollen.
- Es wird nun von oben in den Baum gegangen bis ein Cluster gefunden wird, bei dem der Abstand der Unterpunkte diesen Abstand unterschreitet. Ist ein solcher Punkt gefunden repräsentiert dieser Punkt alle unter ihm hängenden und wird in das Modell aufgenommen.
- Aus allen Modellen wird eine neue Polygondarstellung erzeugt die im allgemeinen eine stark reduzierte Vertexanzahl aufweist, aber die Gestalt des Ursprungsobjektes sehr gut wiedergibt.

3.2.1.C BEREINIGEN DES ERZEUGTEN MODELLS

Das im vorangegangenen Schritt erzeugte Modell wird in einem weiteren Durchlauf von unnötigen Punkte befreit. Darunter fallen Polygone die durch die Reduktion zu einer Linie, im Extremfall zu einem Punkt degeneriert sind. Fallen diese Linien/Punkte mit Kanten/Eckpunkten anderer Polygone zusammen können sie aus dem Modell entfernt werden ohne die Darstellungsqualität zu beeinflussen.

3.2.1.D BEISPIEL

Anhand eines virtuellen Blumentopfes werden die Auswirkungen auf die Darstellung gezeigt. Man sieht deutlich die geringe Änderung der Gesamtgestalt bei doch massiven Polygoneinsparungen.



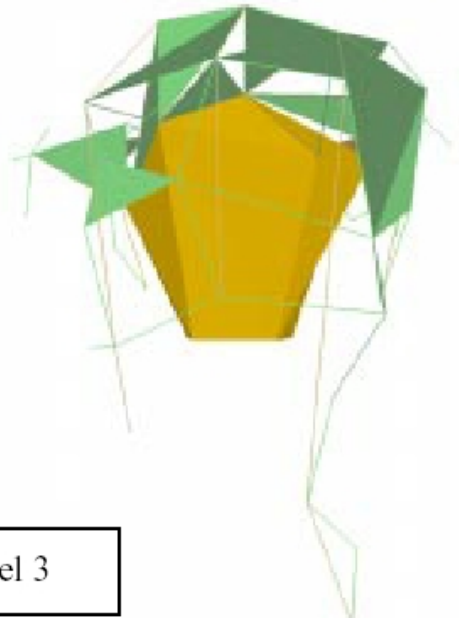
Abbildung 2 Blumentopf, Originalmodell 6064 Polygone

Das Originalmodell besteht aus 6064 Polygonen, die auf 3674, 1225 und schließlich in Level 3 auf 339 reduziert werden. In der Darstellung mit größerer Entfernung erkennt man daß sich die Reduktion auf die Darstellungsqualität nur in geringem Maß auswirkt, obwohl sehr viele Polygone eingespart wurden.



Level 1

Abbildung 3 Blumentopf, 3675 Polygone



Level 3

Abbildung 5 Blumentopf, 339 Polygone



Level 2

Abbildung 4 Blumentopf, 1225 Polygone



Abbildung 6 Level 2,3 im Vergleich zum Original in der Entfernung ab der sie verwendet werden

4. SIMULATION LEVELS OF DETAIL

In einer Simulation soll eine möglichst genaue Reproduktion der Realität erreicht werden. Das ist meist mit sehr hohem Rechenaufwand verbunden. Und das nicht nur bei grafischen Simulationen.

4.1. Simulation von Bewegung

In [ES1] wird die Anwendung von Levels of Detail für Echtzeit-Animationen beschrieben. Für die im Beispiel verwendeten Akteure werden 3 verschiedene Simulationslevel verwendet, zwischen denen umgeschaltet wird.

Simuliert wird ein rechteckiger Bereich, der durch Mauern abgegrenzt ist. In diesem befinden sich mehrere einbeinige Figuren die versuchen einem Puck auszuweichen.

An der doch einfachen Konstruktion der Figuren wird klar, welcher Berechnungsaufwand hinter einer realitätsgetreuen Simulation von Bewegung liegt.

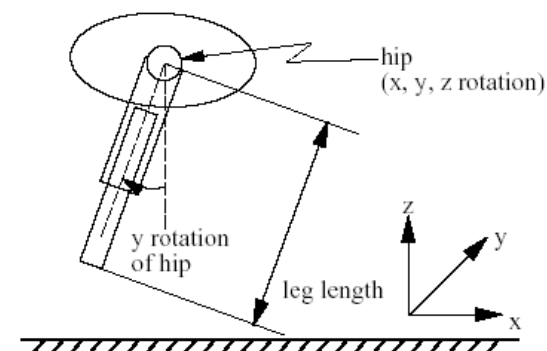


Abbildung 7 Aufbau der simulierten Figur

Wie man in Abb.7 sieht bestehen die Bewegungsmöglichkeiten der Figur in der Rotation im „Hüftgelenk“ in drei Freiheitsgraden und in der variablen Beinlänge. Die Fortbewegung erfolgt durch hüpfen auf dem einen Bein.

4.1.1. Das volldynamische Modell

Im höchsten Detaillevel wird das komplette Verhalten, soweit möglich, dynamisch berechnet. In die Simulation werden Ausfahrgeschwindigkeit/-Beschleunigung des Beines beim Absprung, die Anstellwinkel des Beines zum Körper und die Drehmomente in der „Hüfte“ berücksichtigt.

4.1.2. Kinematisch/dynamisches Modell

In diesem Level unterteilt man die Simulation in eine dynamische Simulation des „Körpers“ (der Kugel die auf dem Bein sitzt) und einer kinematischen Simulation des Beines. Die Bewegung des Körpers erfolgt anhand eines Punkt-Masse-Modells mit einem Beschleunigungslimit. Das entspricht fast dem dynamischen Modell. Die Ausrichtung der Figur erfolgt in Bewegungsrichtung. Die Länge und die Anstellwinkel des Beines sowie die Höhe des Körpers über dem Boden werden mit Hilfe von vorberechneten Lookup-Tables und linearer Interpolation berechnet. Die Lookup-Tables wurden mit dem Dynamischen Modells erstellt.

4.1.3. Einfache Punkt-Masse Repräsentation

Das stellt den detailärmsten Simulationslevel dar, bei dem nur mehr das Verhalten der auf einen Punkt reduzierten Masse der Figur simuliert wird. Durch den hier auftretenden starken optischen Unterschied zu den beiden anderen Levels, ist ein Einsatz nur in den Fällen sinnvoll, wo sich die Figur ausserhalb des Sichtbereiches des Betrachters befindet.

4.1.4. Umschalten zwischen den Modellen

Das Umschalten zwischen den drei genannten Repräsentationen erfolgt unter Miteinbeziehung mehrerer Kriterien.

4.1.4.A WIE WICHTIG IST DYNAMISCHES VERHALTEN:

Jedes Objekt hat einen definierten Einflusbereich in dem es zu Interaktionen mit anderen Objekten kommen kann. Falls sich zwei oder mehrere Objekte so nahe kommen, daß sich ihre Einflusbereiche überschneiden ist es sinnvoll das dynamische Modell anzuwenden um die Interaktion realitätsgetreu zu simulieren.

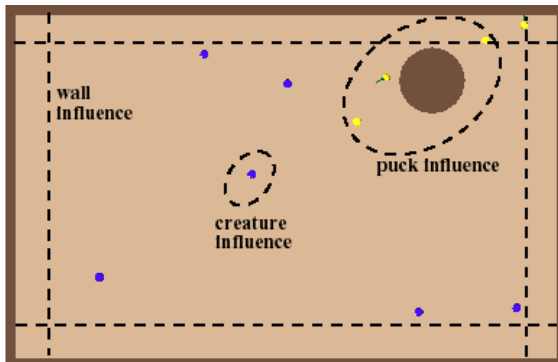


Abbildung 8 Objektinflussbereiche

4.1.4.B WIE IST DIE POSITION DER FIGUR ZUM BETRACHTER:

Einfluß hat ob die Figur im Sichtbereich des Betrachters ist und wenn sie sichtbar ist ist weiters die Entfernung ausschlaggebend. Ist die Figur sichtbar aber weit entfernt kann das kinematisch/dynamische Modell verwendet werden (falls sich nicht zwei Figuren zu nahe kommen). Bei Nichtsichtbarkeit wird bevorzugt die Punkt-Masse Repräsentation verwendet. Das dynamische Modell wird nur eingeschaltet wenn sich die Figur nah zum Betrachter befindet.

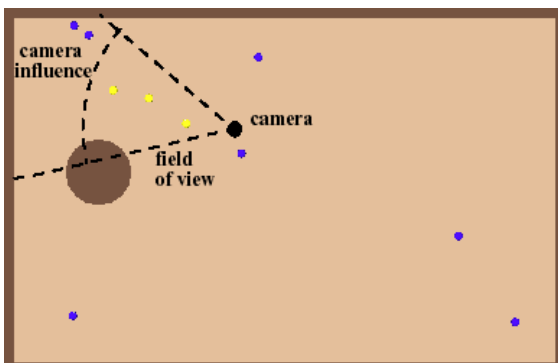


Abbildung 9 Sichtbereich des Betrachters

4.2. Simulation Levels of Detail bei Partikelsystemen

In [ES4] wird die Anwendung von Simulation Levels of Detail bei der Implementierung von Partikelsystemen beschrieben. Das Ziel ist eine vom Benutzer vorgegebene Framerate und Genauigkeit der Simulation in allen Fällen zu erreichen.

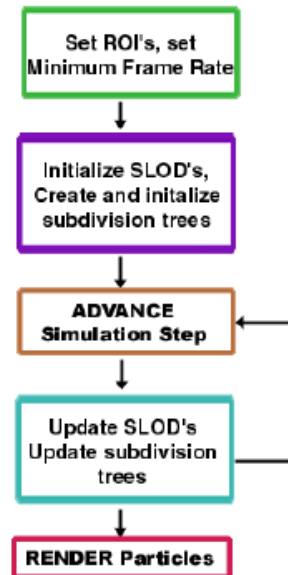


Abbildung 10 Ablaufschema

Dazu wird das gesamte Partikelsystem in mehrere kleinere Bereiche unterteilt die als eigene kleine Partikelsysteme behandelt werden. Diese werden dann in unterschiedlichen Levels of Detail simuliert und angezeigt. Die Unterteilung wird in jedem Simulationsschritt aktualisiert. In die Unterteilung gehen neben der Position der Partikel auch physikalische Parameter wie Masse, Geschwindigkeit und Beschleunigung ein.

Die Genauigkeit der Simulation der Teilbereiche wird von sog. *Regions of Interest* bestimmt. Diese werden z.B. durch den Sichtbereich des Benutzers definiert (im Zentrum des Sichtbereiches, wo der Benutzer das Geschehen am deutlichsten wahrnimmt erfolgt nur eine Unterteilung in kleine Gruppen, oder es werden sogar die Einzelpartikel behandelt, in Randbereichen können größere Gruppen verwendet werden, die als ein Partikel mit der gewichteten Masse ihrer Teile simuliert werden) oder durch Bereiche in denen es zu Kollisionen mit Hindernissen kommen kann. Diese Bereiche ändern sich im Laufe der Simulation immer wieder, was ein ständiges Update LOD Bereiche erfordert. (siehe Abb. 11).

Ein weiterer Grund für ein nötiges Update der Unterteilung ist falls sich zwei oder mehrere Teilsysteme begegnen und zusammenstoßen. Die Kollision wird auf

Ebene der Einzelpartikel behandelt und danach wird eine neue Einteilung getroffen.

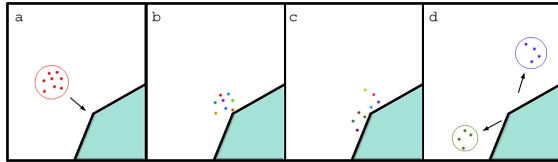


Abbildung 11 Annäherung eines zusammengefaßten Partikelreiches an ein feststehendes Hindernis.

Kurz vor dem Aufprall erfolgt der Wechsel auf die einzelnen Partikel um die Simulation realitätsgetreuer zu machen. Bei der anschließenden Fortbewegung werden die Partikel wieder zu Gruppen zusammengefaßt.



Abbildung 12 Darstellung mit (rechts) und ohne (links) Levels of Detail

Abbildung 12 zeigt sehr gut welche Ergebnisse sich mit diesem Ansatz erzeugen lassen. Die beiden Darstellungen unterscheiden sich nur minimal obwohl der Brunnen ohne Levels of Detail circa den sechsfachen Rechenaufwand verursacht (also einen sechsmal schnelleren Rechner verlangt um die gleiche Framerate zu erreichen)

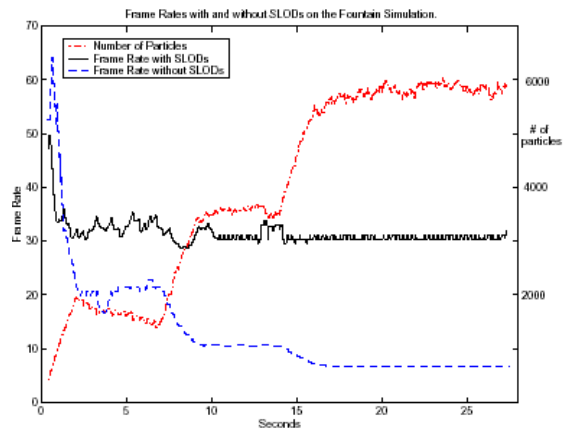


Abbildung 13 Vergleich der Framerate mit/ohne Levels of Detail

5. ERGEBNISSE UND AUSBLICKE

Als Abschluß läßt sich sagen, daß die Anwendungsbereiche von Levels of Detail nicht nur auf die am weitesten verbreiteten Geometrischen Levels of Detail beschränkt sind. Das Prinzip läßt sich eigentlich auf faßt alle Bereiche der zwei und dreidimensionalen Grafik anwenden in denen uns die verfügbare Hardware Grenzen setzt. Diese Beschränkungen werden zwar immer geringer, aber sie werden wohl auch noch die nächsten Jahre ein Thema bleiben da auch mit der verfügbaren Leistung auch die Ansprüche der Anwender steigen. Levels of Detail werden also in einem gewissen Maß immer ein Thema bleiben.

1 LITERATURVERZEICHNIS

ES1: Deborah A. Carlson, Jessica K. Hodgins, Simulationlevels of detail for real-time animation, 1997
ES2: Jinseok Seo, Gerard Jounghyun Kim, Kyo Chul-Kang, Levels of Detail Engineering of VR Objects, 1999
TRO1: Troger Christian, Levels of Detail, 2001
ES3: G. Schaffler, W. Stürzlinger, Generating multiple Levels of Details from Polygonal Modells,
ES4: David O'Brien, Susan Fisher, Ming C. Lin, Automatic Simplification of Particle System Dynamics,

2 ABBILDUNGSVERZEICHNIS

Abbildung 1Raum mit/ohne LOD (links 6810,rechts 13847 Polygone) III
Abbildung 2Blumentopf, Originalmodell 6064 Polygone IV
Abbildung 3Blumentopf, 3675 Polygone V
Abbildung 4Blumentopf, 1225 Polygone V
Abbildung 5Blumentopf, 339 Polygone V
Abbildung 6Level 2,3 im vergleich zum Original in der Entfernung ab der sie verwendet werden V
Abbildung 7Aufbau der simulierten Figur VI
Abbildung 8Objekteinflußbereiche VI
Abbildung 9Sichtbereich des Betrachters VII
Abbildung 10Ablaufschema VII
Abbildung 11Annäherung eines zusammengefaßten Partikelbereiches an ein feststehendes Hindernis. VII
Abbildung 12Darstellung mit(rechts) und ohne (links) Levels of Detail VII
Abbildung 13 Vergleich der Framerate mit/ohne Levels of Detail VIII