

Nutzung von Echtzeit-Tiefenschärfe-Effekten zur Beeinflussung von wahrgenommenen Größen- und Abstandsverhältnissen

Benjamin Glatzel*
Hochschule Darmstadt

Dominik Wilhelm†
Hochschule Darmstadt



(a)



(b)

Abbildung 1: Die linke Abbildung zeigt eine Aufnahme der Beispielszene, bei der die Distanz zur Schärfeebele auf 75.0 m eingestellt und sowohl die Brennweite als auch die Größe der Linse an die Eigenschaften des menschlichen Auges angepasst wurden. Innerhalb der rechten Abbildung wurde die Größe der Linse so angepasst, dass ein Abstand von 6.0 cm zur Schärfeebele simuliert wird. Nach Anwendung des in diesem Paper vorgestellten Algorithmus, wirken die Bäume und das Terrain innerhalb der Szene merklich miniaturisiert.

Abstract

Innerhalb des vorliegenden Papers wird ein Algorithmus vorgestellt, der die Manipulation von wahrgenommenen Größen- und Abstandsverhältnissen in Echtzeit ermöglicht. Hierzu wird ein Modell zur Berechnung von Tiefenschärfe eingeführt, das mithilfe von Shadern und einem Grafikchip effizient zur Berechnung verwendet werden kann. Die Anwendung des Algorithmus für die Erzeugung des sogenannten Tilt-Shift-Effektes unter Berücksichtigung der menschlichen Wahrnehmung lieferte gute Ergebnisse und wurde anhand einer speziellen, fraktal generierten Beispielszene erprobt und validiert. Auch der Vergleich mit Offline-Verfahren war zufriedenstellend und zeigte lediglich die erwarteten, durch die Unterschiedlichkeiten der Berechnung bedingten, Differenzen.

Keywords: Tiefenschärfe, Echtzeit, Shader, Tilt-Shift-Effekt

1 Einleitung

Unschärfe in Fotografien und Grafiken kann wahrgenommene Größen- und Abstandsverhältnisse merklich beeinflussen [Held et al. 2010]. Dieses Phänomen kann einerseits dazu verwendet werden, Szenen viel größer wirken zu lassen, als sie tatsächlich sind, oder aber für die Miniaturisierung von Motiven eingesetzt werden. Der

erste Fall wird häufig für Spezialeffekte mithilfe von sehr kleinen Kameraobjektiven, die die Variation und die Intensität von Unschärfe merklich reduzieren, eingesetzt [Held et al. 2010]. Die Miniaturisierung kann mit einer fotografischen Manipulation, dem sogenannten Tilt-Shift-Effekt, erzeugt werden.

Für die Realisierung des Tilt-Shift-Effektes existieren drei bewährte Umsetzungsmöglichkeiten: die Verwendung von Tilt-Shift-Objektiven, die Anwendung eines vertikalen Unschärfefilters durch Postprocessing und die realitätsnahe, physikalische Berechnung von Tiefenschärfe. Während die ersten beiden Möglichkeiten bei Fotografien leicht angewendet werden können, sind für die realitätsnahe Berechnung Tiefeninformationen eine Grundvoraussetzung. Da bei der Entwicklung von Echtzeit-3D-Applikationen Tiefeninformationen ein grundlegender Bestandteil sind und die Ausrichtung eines vertikalen Unschärfefilters geschickt gewählt werden muss, eignet sich in diesem Szenario eine realitätsnahe Berechnung.

Abbildung 1 (a) und Abbildung 1 (b) zeigen die Ergebnisse nach einer Echtzeit-Tiefenschärfe-Berechnung angewandt auf eine fraktal generierte Beispielszene. Die Parameter wurden für die Generierung der Abbildungen so gewählt, dass, unter Berücksichtigung der Eigenschaften des menschlichen Auges, eine fokussierte Distanz von 75.0 m auf 6.0 cm simuliert wird.

Wie zu erkennen ist, eignen sich Echtzeitverfahren für die Erzeugung des Tilt-Shift-Effektes. Innerhalb des vorliegenden Papers wird ein einfacher Algorithmus beschrieben, der die Berechnung von Tiefenschärfe unter Berücksichtigung der menschlichen Wahrnehmung in Echtzeit ermöglicht. Durch dessen Anwendung können Größenverhältnisse betrachteter Szenen durch Defokussierung und Fokussierung manipuliert und an gewünschte Größenverhältnisse angepasst werden. Abschließend werden die Ergebnisse validiert und mit einer offline Reproduktion verglichen.

*E-Mail: benjamin.glatzel@me.com

†E-Mail: dominik_wilhelm@arcor.de

2 Hintergrund

Die Berechnung von Tiefenschärfe ist in vielerlei Hinsicht bedeutsam. Unschärfe kann dazu eingesetzt werden, den Blick des Betrachters zu steuern und auf bestimmte Bildregionen auszurichten [Held et al. 2010]. Dies kann durch die Scharfzeichnung des zu betrachtenden bzw. zu fokussierenden Objektes und die Weichzeichnung der verbleibenden Bildelemente erreicht werden. Tiefenschärfe eignet sich zudem, wie bereits erwähnt wurde, zur Manipulation von wahrgenommenen Größen- und Abstandsverhältnissen. Einerseits können durch die Nutzung von sehr kleinen Kameraobjektiven Miniaturen so aufgenommen werden, dass diese im Vergleich zum Original deutlich größer wirken. Dieser Effekt wird primär beim Filmdreh verwendet [Held et al. 2010].

Andererseits hat sich der Tilt-Shift-Effekt zur Umkehrung des Effekts etabliert. Die ursprüngliche Namensgebung stammt von sogenannten Tilt-Shift-Objektiven, die die Ausrichtung der Linse relativ zur Bildebene ermöglichen. Beim Einsatz dieser können unter Berücksichtigung des Scheimpflugprinzips Ebenen in den Fokus gerückt werden, die anderenfalls außerhalb des Fokusbereiches liegen würden [Held et al. 2010]. Umgekehrt können Bereiche aus der Fokusebene bewegt werden, die unter Verwendung eines handelsüblichen Kameraobjektives fokussiert wären.



Abbildung 2: Durch Postprocessing erzeugter Tilt-Shift-Effekt. Hierbei wurde ein linearer, vertikaler Grauverlauf sowie der von Photoshop CS5 bereitgestellte Linsenfilter mit einem hexagonalen Filterkern verwendet.

Die für die Nachbearbeitung verwendete Aufnahme der Shibuya Kreuzung in Tokio wurde vom Flickr-Benutzer Maynard erstellt.

Statt auf Tilt-Shift-Objektive zurückzugreifen, kann der beschriebene Effekt gleichwertig mithilfe von linearen Weichzeichnungsfilters reproduziert werden [McCloskey and Langer 2009]. Für die Umsetzung eignet sich ein linearer, vertikaler Grauverlauf zur Beschreibung der Unschärfeintensität für den jeweiligen Bildpunkt. Abbildung 2 zeigt eine Aufnahme der Shibuya Kreuzung in Tokio nach Anwendung des Tilt-Shift-Effektes. Dieser wurde auf die beschriebene Art und Weise, nach der Aufnahme des Bildes mit einer handelsüblichen Kamera, durch Postprocessing in Adobe Photoshop CS5 hinzugefügt. Durch die Leichtigkeit und die Wirkung des Effektes, erfreut sich die Erzeugung von Miniaturbildern reger Beliebtheit [Flickr 2011].

Der Einfluss von Unschärfe innerhalb einer Szene auf wahrgenommene Größen- und Abstandsverhältnisse ist unverkennbar. Dennoch kann die Distanz von Objekten zu dem Punkt der Aufnahme einer Fotografie oder Grafik nicht allein anhand der Intensität der Unschärfe rekonstruiert werden. Held, et. al. haben hierzu ein Modell vorgestellt, dass für die Rekonstruktion des Abstandes zur Fokusebene

zusätzlich zur Unschärfe relative Distanzen innerhalb der Szene berücksichtigt. Dieses Modell wurde anhand eines semi-automatischen Algorithmus, der Fotografien durch Anwendung von Tiefenschärfe akkurat in gewünschte Größenverhältnisse versetzt, und einem psychophysiologischen Experiment validiert [Held et al. 2010].

2.1 Tiefenschärfe in Echtzeit-3D-Applikationen

Die Berechnung von Tiefenschärfe ist trotz Hardwarebeschleunigung ein rechenaufwändiger Prozess [Held et al. 2010]. Dennoch hat sich der Einsatz von Tiefenschärfe als stilistisches Mittel innerhalb von Computerspielen etabliert und findet zudem bei der Entwicklung von Echtzeit-3D-Applikationen in verschiedenen Formen und Ausführungen Anwendung.

Die erste Variante nutzt die Berechnung des Circle of Confusion (kurz CoC) anhand der Linsengleichung als Grundlage. Dieser gibt für einen betrachteten Punkt unter Berücksichtigung der Brennweite, der Distanz zur Schärfeebeine, der relativen Distanz zur Schärfeebeine für den betrachteten Punkt und der Größe der Linse, den Durchmesser der Unschärfe an. Nach der Ermittlung des CoC-Wertes wird ein Unschärfefilter mit dem entsprechenden Durchmesser auf jeden Bildpunkt angewandt [Dudash 2004; Scheuermann 2004]. In Computerspielen wird dieser Effekt meist als stilistisches Mittel verwendet, sodass statt der Linsengleichung lineare Grauverläufe zwischen bestimmten Distanzen interpoliert werden [Earll Hammon 2007; Filion and McNaughton 2008]. Zur Verbesserung des Effekts existieren verschiedene Ansätze, um durch das Verfahren bedingte Artefakte zu minimieren.

Unter Verwendung des Circle of Confusion kommt in der Praxis zudem eine rechenaufwändigere Technik zum Einsatz. Bei diesem wird die sogenannte Wärmeleitungsgleichung (engl. heat equation), eine partielle Differentialgleichung, zur Approximation von Unschärfe in einem Punkt verwendet [Bertalmio et al. 2004; Kass et al. 2006]. Ausgehend von einem Lochkameramodell wird eine Wärmeverteilung entsprechend den CoC-Werten berechnet [Kass et al. 2006]. Für einen großen CoC-Durchmesser wird die thermische Leitfähigkeit des Materials als hoch modelliert und für einen entsprechend kleinen CoC-Durchmesser gering. Dieses Verfahren wird beispielsweise von den Pixar Animation Studios für die Generierung von Echtzeit-Previews genutzt.

Die Publikationen von Sungkil Lee et. al. präsentieren weitere Verfahren zur Echtzeitberechnung von Tiefenschärfe und Unschärfe. Bei einem dieser Verfahren wird der Effekt durch den Einsatz von anisotroper Filterung und MIP-Maps erzeugt, wobei mögliche Artefakte durch den Einsatz eines Gauß-Filters unterdrückt werden [Lee et al. 2008]. In einem weiteren Verfahren wird die Szene innerhalb eines einzigen Renderingdurchlaufs in mehrere Schichten aufgeteilt. Die Berechnung wird abschließend ähnlich zu Multi-View-Synthesis-Verfahren durchgeführt [Lee et al. 2009]. Die aktuellste Publikation beschreibt ein Renderingsystem, das speziell für die Generierung von Defokussierungs- sowie Unschärfeeffekten entwickelt wurde und zudem die Simulation von verschiedenen Linsenmodellen ermöglicht [Lee et al. 2010].

2.2 Modell zur Berechnung von Tiefenschärfe

Die Berechnung des Circle of Confusion bildet die Grundlage für den in diesem Paper vorgestellten Tiefenschärfe-Algorithmus. Die Berechnung dessen baut auf den Ergebnissen von Held et. al. auf [Held et al. 2010]. Eine ideale Linse fokussiert parallel einfallende Lichtstrahlen in einem Punkt f , dem sogenannten Brennpunkt. Lichtstrahlen, ausgesendet von einem beliebigen Punkt in der Entfernung z_1 , werden in einem Punkt s_1 auf der entgegengesetzten

Seite der Linse fokussiert. Der Zusammenhang zwischen diesen Abständen ist durch die Linsengleichung definiert (Formel 1).

$$\frac{1}{s_1} + \frac{1}{z_1} = \frac{1}{f} \quad (1)$$

Bei gewöhnlichen Bildaufnahmegeräten liegt die Bildebene parallel zur Linse, sodass Punkte in einer Distanz von $z_0 = (f^{-1} - s_0^{-1})^{-1}$ im Fokus liegen und somit vollständig scharf dargestellt werden. Die Ebene in der Distanz z_0 zur Linse wird als Schärfeebebe bezeichnet. Punkte, die nicht auf der Schärfeebebe liegen, werden entsprechend unscharf dargestellt. Die Intensität der Unschärfe kann durch den Durchmesser c des Unschärfekreises auf der Bildebene beschrieben werden. Das Verhältnis $\frac{z_1}{z_0}$ (Formel 2), sprich das Verhältnis von einem Punkt auf der Schärfeebebe und einem Punkt, der nicht auf dieser liegt, wird als relative Distanz d bezeichnet. Bei Berücksichtigung des Durchmessers A der Linse ergibt sich für ein Objekt in der Distanz z_1 die in Formel 3 dargestellte Gleichung.

$$d = \frac{z_1}{z_0} \quad (2)$$

$$c_1 = \left| A \frac{s_0}{z_0} \left(1 - \frac{1}{d} \right) \right| \quad (3)$$

Als Tiefenschärfe (engl. depth of field) wird der Bereich im Umkreis zur Schärfeebebe bezeichnet, in dem der betrachtete Punkt vollständig scharf dargestellt wird. Zur Veranschaulichung sind sämtliche, soeben beschriebenen Verhältnisse innerhalb von Abbildung 3 verbildlicht.

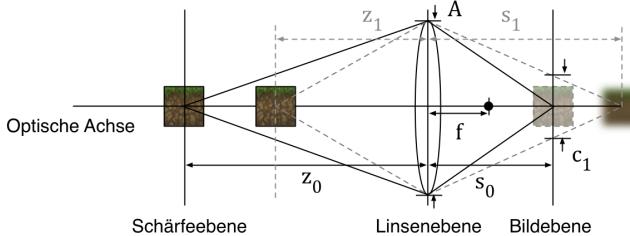


Abbildung 3: Der Circle of Confusion bildet die Grundlage für die Berechnung von Tiefenschärfe. z_0 kennzeichnet den Abstand zur Schärfeebebe, A die Größe der Linse, f die Brennweite der Linse und s_0 den Abstand zwischen Linse und Bildebene. Bei Betrachtung eines Punktes mit der Distanz z_1 entsteht ein Streupunkt mit dem Durchmesser c_1 .

Der zuvor beschriebene Tilt-Shift- bzw. Miniaturisierungseffekt lässt sich mit dem Verhältnis zwischen z_0 und c_1 erklären. Durch Verringerung von z_0 , sprich einer Reduzierung des Abstandes zwischen Linse und Schärfeebebe, erhöht sich der Durchmesser des Circle of Confusion c_1 bei Betrachtung eines Bildpunktes in der relativen Distanz d . Entsprechend erzeugt eine aus der Nähe aufgenommene Szene mehr Unschärfe als eine hochskalierte Version der Szene, aufgenommen aus größerer Distanz.

3 Konzept

Der vorgestellte Algorithmus basiert, mit kleinen Variationen, auf dem von T. Scheuermann beschriebenen Advanced-Depth-of-Field-Effekt [Scheuermann 2004]. Hierbei werden die eigentlichen Berechnungen in die Grafikpipeline verlagert und in Form von Shadern

realisiert. Da der Kernpunkt des Algorithmus die Berechnung des Tilt-Shift-Effektes ist, muss beim Entwurf der Szene darauf geachtet werden, dass die Größenverhältnisse von Entitäten (zumindest näherungsweise) auf die reale Welt übertragbar sind. Um spätere Berechnungen zu erleichtern, sollte eine Einheit der 3D-Umgebung so gewählt werden, dass sie einem Meter in der realen Welt entspricht.

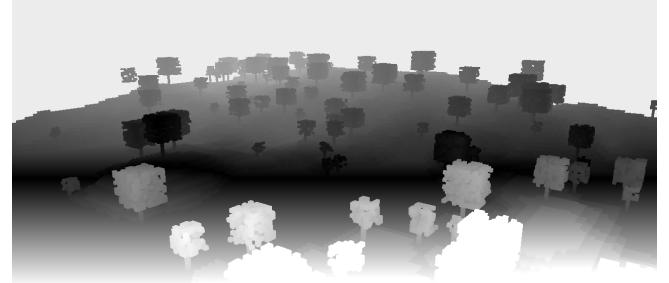


Abbildung 4: Visualisierung des Circle of Confusion als Grauwertbild. Die Intensität der Unschärfe pro Bildpunkt $c_{x,y}$ wurde skaliert, so dass $c_{x,y} \in [0,0, 1,0]$ gilt.

Für die eigentliche Berechnung des Echtzeit-Tiefenschärfe-Effektes sind folgende Schritte notwendig: Zunächst müssen die jeweiligen Tiefenwerte für das aktuelle Blickfeld der Kamera ermittelt und für spätere Berechnungen bereitgestellt werden. Im zweiten Schritt muss, ausgehend von den ermittelten Tiefenwerten, der entsprechende Diameter für den Unschärfefilter unter Verwendung der zuvor genannten Formel für c_1 (Formel 3) für jeden Bildpunkt ermittelt werden. Sowohl die Tiefenwerte als auch die Durchmesserwerte des CoCs werden in der Praxis normalisiert und als Lookup-Table, in Form einer Textur, codiert. Abbildung 4 zeigt die Visualisierung einer solchen Tabelle für den Circle of Confusion als Grauwertbild.

Anhand der ermittelten Werte für den Circle of Confusion wird im letzten Schritt der eigentliche Tiefenschärfe-Effekt berechnet. Hierzu muss für jeden Bildpunkt entsprechend dem zuvor ermittelten Diameter $c_{x,y}$ ein Unschärfefilter angewendet werden. Die Schwierigkeit liegt an dieser Stelle darin, dass der Durchmesser und die Intensität des Filters variabel wählbar sein muss.

3.1 Nutzung einer Poisson-Disk-Verteilung als Weichzeichnungsfilter

Um den Vorgang möglichst effizient zu gestalten, ist von der Verwendung eines Gauß-Filters abzusehen. Für jeden Diameter $c_{x,y}$ müsste an dieser Stelle eine entsprechend große Filter-Matrix bereitgestellt oder dynamisch berechnet werden. Eine Mehrfachausführung des Gauß-Filters ist ebenfalls keine geeignete Lösung.

Als geeignete Alternative erweist sich die Poisson-Disk-Verteilung [Engel 2004]. Zur Erzeugung dieser wird zunächst eine Menge von Zufallswerten entsprechend der Poisson-Verteilung generiert. In einem zweiten Schritt werden Punkte eliminiert, die einen bestimmten Mindestabstand zu Nachbarpunkten unterschreiten. Die Poisson-Verteilung eignet sich, bei entsprechender Wahl der Parameter, als Näherung der Gauß-Verteilung, was die Nutzung als Unschärfefilter rechtfertigt. Für die Anwendung als Filterkern werden lediglich Punkte in einem bestimmten Radius betrachtet. Um den Filterkern zu verallgemeinern, ist der Radius des Einheitskreises zu wählen. Die Anzahl der ermittelten Samples ist für die Qualität des Effektes zuständig. Sollten für ein entsprechend großen Radius zu wenige Samples vorhanden sein, können sich im

Endergebnis Artefakte in Form von Geisterbildern äußern.

Ausgehend von der Betrachtung eines bestimmten Punktes $\vec{p} = (x, y)^T$ kann der pro Sample \vec{s}_i zu berücksichtigende Punkt \vec{p}_i entsprechend Formel 4 ermittelt werden.

$$\vec{p}_i = \vec{p} \cdot \vec{s}_i \cdot c_{x,y} \quad (4)$$

Die Farbwerte der entsprechenden Punkte können an dieser Stelle auf den betrachteten Punkt im Ursprung aufaddiert werden. Im Anschluss muss der Farbwert lediglich der Anzahl der verwendeten Samples entsprechend normalisiert werden. Abbildungen 5 (a) und (b) zeigen die Anwendung eines Filters mit dem Radius $r = 0.0$ bzw. $r = 1.0$ und 6 Samples.

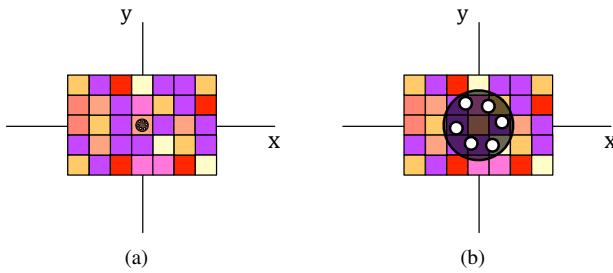


Abbildung 5: Anwendung eines Poisson-Disk-Filters mit 6 Samples für einen Radius $r = 0.0$ (a) bzw. $r = 1.0$ (b). Der aktuell betrachtete Bildpunkt liegt im Ursprung des Koordinatensystems.

3.2 Simulation des Tilt-Shift-Effektes

Um möglichst wirklichkeitstreue Ergebnisse zu erzielen, müssen die Eigenschaften des menschlichen Auges berücksichtigt werden, sodass sich bei Betrachtung von Formel 3 für $A = 4.6$ mm und $s_0 = 17.0$ mm ergeben [Held et al. 2010].

Für die Simulation des Tilt-Shift-Effektes muss nun die Größe der Linse, ausgehend von den ursprünglichen Dimensionen, angepasst werden. Bei Simulation einer Distanz, die um Faktor m verschoben wird, ergibt sich das in Formel 5 und 6 beschriebene Verhältnis [Held et al. 2010].

$$\hat{z}_0 = m \cdot z_0 \quad (5)$$

$$\hat{A} = \frac{A}{m} \quad (6)$$

Geht man beispielsweise von $s_0 = 0.017$ m sowie $A = 0.0046$ m und einer Szene, die aus einer Distanz $z_0 = 75.0$ m aufgenommen wurde, aus, muss für die Simulation einer Distanz von $\hat{z}_0 = 0.06$ m eine Linse mit $\hat{A} = 5.75$ m verwendet werden.

3.3 Abbildung der Größenverhältnisse

Unter Berücksichtigung der zuvor beschriebenen Parameter für die menschliche Wahrnehmung und Formel 3, kann der Durchmesser des Circle of Confusion bei einer Projektion auf die menschliche Netzhaut berechnet werden. Für eine Simulation muss die Verbindung zwischen dem beschriebenen Durchmesser des Poisson-Disk-Filters und dem Durchmesser des CoCs hergestellt werden. Hierzu wurden die Ergebnisbilder von Held et. al. betrachtet und

der jeweilige Durchmesser des angewendeten Filters anhand eines Gauß-Weichzeichnungs-Filters approximiert [Held et al. 2010]. Mithilfe von Photoshop CS5 wurde für eine der beschriebenen Szenen aus einer simulierten Distanz $\hat{z}_0 = 0.06$ m mit $\hat{A} = 60.0$ m, $z_0 = 785.0$ m, $s_0 = 0.017$ m und $A = 0.046$ m ein maximaler Unschärfe-Radius von $r = 4.6$ px ermittelt. Bei Anwendung des Poisson-Disk-Filters muss für den Durchmesser des maximalen CoCs $c_{\max} = 4.6$ px $\cdot 2 = 9.2$ px gewählt werden.

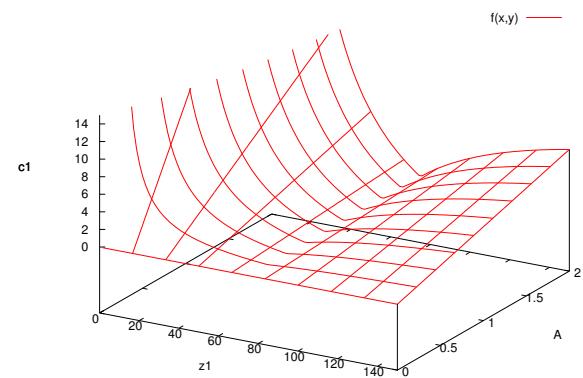


Abbildung 6: Durchmesser des CoCs ermittelt anhand von Formel 3 für eine Linse der Größe $A \in [0.0 \text{ m}, 2.0 \text{ m}]$, $z_1 \in [0.0 \text{ m}, 150 \text{ m}]$, $z_0 = 75.0 \text{ m}$, $s_0 = 0.017 \text{ m}$ bei Verwendung des Skalierungsfaktors $b = 28000$.

Anhand der Linsen-Parameter der betrachteten Beispieldaten kann der Skalierungsfaktor $b = 28000$ abgeschätzt werden. Unter Verwendung dessen entspricht die Intensität der dargestellten Unschärfe in etwa der, die bei Betrachtung der Szene aus der simulierten Distanz tatsächlich entstehen würde [Held et al. 2010]. Abbildung 6 zeigt den Durchmesser des CoC für ein Beispieldaten nach Anwendung der beschriebenen Skalierung.

4 Berechnung der Tiefenschärfe mithilfe von Shadern

Für die Implementierung des Tiefenschärfe-Effektes kommen zwei spezifische Fragment-Shader und ein Vertex-Shader zum Einsatz. Der Vertex-Shader ist für die Erzeugung der Tiefenwerte zuständig, während die zwei Fragment-Shader für die Berechnung der CoC-Werte und die Anwendung des eigentlichen Tiefenschärfe-Effektes zuständig sind. Anhand der ermittelten Werte wird eine Depth-Map bzw. CoC-Map erzeugt, wobei die CoC-Map an den finalen Shader für die Berechnung des Tiefenschärfe-Effektes übergeben wird. In diesem kommen die erzeugten Maps zusammen mit einer Textur, die ein Rendering der betrachteten Szene beinhaltet, zur Erzeugung des finalen Tiefenschärfe-Effektes zur Verwendung.

Bei der Realisierung des Effektes wird davon ausgegangen, dass die Werte des Z-Puffers nicht unmittelbar zur Verfügung stehen und diese zunächst mithilfe eines entsprechenden Shaders ermittelt werden müssen.

4.1 Erzeugung der Depth-Map

Der Vertex-Shader für die Depth-Map muss auf sämtliche Vertices der Szene angewendet werden. Ausgehend von einem Punkt in Weltkoordinaten, wird eine Transformation in das Kamerako-

ordinatensystem und abschließend in das Projektionskoordinaten- system durchgeführt. Hierzu müssen die entsprechenden Matrix- Transformationen der Reihe nach oder in Form einer einzigen Matrix, oft als World-View-Projection-Matrix oder Model-View-Projection- Matrix bezeichnet werden.

Nach der Ermittlung der Koordinaten des Vertex muss, um die Darstellung als Graubild zu ermöglichen, der entsprechende Wert auf der Z-Achse des Koordinatensystems auf einen Bereich zwischen 0.0 und 1.0 abgebildet werden. Ein Tiefenwert von 1.0 steht hierbei für sehr nahe und 0.0 für Punkte in der maximal möglichen Entfernung. Dies kann durch eine Beschränkung der Auflösung erreicht werden, bei der die maximal darstellbare Tiefe der Szene entsprechend festgelegt wird. Der Dividend muss für spätere Berechnungen und die Rückrechnung in die Ausgangseinheit bereitgestellt werden.

Listing 1 zeigt eine mögliche Implementierung eines entsprechenden Shaders in Pseudocode.

```

1 vertex program DepthVertexShader with parameters
2   worldPosition, worldViewProjectionMatrix,
3   sceneDepth
4
5   set position to worldViewProjectionMatrix *
6     worldPosition
7   set colour to 1.0 - position.z / sceneDepth
8
9   return saturate (colour) and position
10 end
```

Listing 1: Pseudocode für den Vertex-Shader zur Erzeugung der Depth-Map.

Der ermittelte Farbwert muss mithilfe eines Fragment-Shaders ausgegeben werden. Da dieser sehr einfach umzusetzen ist, wird an dieser Stelle auf eine detaillierte Beschreibung verzichtet.

4.2 Erzeugung der CoC-Map

Unter Verwendung der zuvor beschriebenen Depth-Map (siehe Abschnitt 4.1) wird nun innerhalb eines Fragment-Shaders die Gleichung für c_1 (siehe Formel 3) zur Berechnung der CoC-Map eingesetzt. Hierzu wird für jeden Pixel der Ausgabetextur der entsprechende Wert für c_1 als Grauwert kodiert. Da die Unschärfeintensitäten bei Verwendung der Parameter des menschlichen Auges je nach Abstand verschwindend gering ausfallen, muss der Wert vor der eigentlichen Ausgabe skaliert werden (mit Hinblick auf die Genauigkeit von einfachen Gleitkommazahlen). Diese Skalierung muss bei einer erneuten Verwendung berücksichtigt werden.

Listing 2 zeigt eine exemplarische Implementierung eines entsprechenden Fragment-Shaders als Pseudocode.

```

1 fragment program DepthFragmentShader with
2   parameters currentPoint, depthSample, z0, a,
3   sceneDepth
4
5   // z0 entspricht z.B. 75.0
6   // a entspricht z.B. 0.0046
7   set s0 to 0.017
8   set depthValue to depthSample at currentPoint
9   set d to ( sceneDepth * ( 1.0 - depthValue ) ) /
10    z0
11  set cocDiameter to absolute ( a * ( s0 / z0 ) *
12    ( 1.0 - ( 1.0 / d ) ) ) * 1000.0;
13
14  return saturate ( cocDiameter );
15 end
```

Listing 2: Pseudocode für den Fragment-Shader zur Berechnung der CoC-Map.

4.3 Abschließende Berechnung des Tiefenschärfe-Effektes

Abschließend wird nun unter Verwendung der CoC-Map, einem Abbild der aktuell betrachteten Szene als Textur und unter Verwendung einer im Voraus definierten Menge von Poisson-Disk-Samples der Tiefenschärfe-Effekt für jeden Bildpunkt berechnet. Hierzu wird vom aktuell betrachteten Bildpunkt für jedes Sample der zu betrachtende Punkt gemäß Formel 4 berechnet, die entsprechenden Farbwerte addiert, normalisiert und der finale Farbwert zurückgegeben. Unter Berücksichtigung von Abschnitt 3.3 muss an dieser Stelle der Skalierungsfaktor $b = 28000$ bei Berechnung des CoC-Diameters berücksichtigt werden.

An dieser Stelle wird davon ausgegangen, dass die Poisson-Samples an die Darstellungsauflösung der Szene angepasst sind und der Multiplikator in Pixeln beschrieben werden kann.

Listing 3 zeigt eine exemplarische Implementierung eines entsprechenden Fragment-Shaders als Pseudocode.

```

1 fragment program DepthOfFieldFragmentShader with
2   parameters currentPoint, depthSample,
3   cocSample, sceneSample
4   set cocValue to cocSample at currentPoint
5   set cocDiameter to (cocValue / 1000.0) * 28000
6   set result to 0.0 0.0 0.0
7
8   for each point p in poissonSamples do
9     set newPoint to currentPoint + p * cocDiameter
10    set result to result + sceneSample at newPoint
11  end
12
13  return normalize ( result )
14 end
```

Listing 3: Pseudocode für den Fragment-Shader zur Berechnung der Tiefenschärfe.

5 Ergebnisse

Die Referenzimplementierung wurde mithilfe von Ogre (Kurzform für *Object-Oriented Graphics Rendering Engine*) [Torus Knot Software 2011] unter Verwendung von Mac OS 10.6.6 und OpenGL sowie der Shader-Hochsprache Cg (Kurzform für *C for Graphics*) durchgeführt. Als Grafikkarte wurde eine ATI Radeon 5870 mit 1 GB Grafikspeicher eingesetzt. Der Testrechner besitzt einen Intel Xeon 6-Kern-Prozessor mit 3.33 GHz pro Kern, 12 Threads und insgesamt 8 GB Arbeitsspeicher. Für die Darstellung wurde ein Weitbildmonitor mit einer Bilddiagonale von 24" und einer Auflösung von 1920×1200 px verwendet.

Die beschriebenen Shader wurden mithilfe des von Ogre bereitgestellten Material- sowie Kompositor-Systems zugeordnet. Da die Tiefenwerte getrennt von der eigentlichen Berechnung ermittelt werden müssen, wird eine sogenannte Render-Queue-Invocation verwendet, bei der der Shader für die Erzeugung der Depth-Map auf die Szenen-Objekte angewendet und das Gesamtergebnis in Form einer Textur für die weitere Verwendung vorbereitet wird. Für den Unschärfefilter wurden 32 Poisson-Disk-Samples pro Frame verwendet, wobei der Tiefenschärfe-Shader zur Verstärkung des Effektes zweifach ausgeführt wird.

Für die Demonstration wurde eine spezielle Szene entwickelt, die eine Art *Lego-Effekt* erzeugen soll. Die Szene besteht aus vielen einzelnen, quadratischen Blöcken, die durch einen fraktalen Algorithmus (dem Square-Diamond-Algorithmus) zu einem Terrain ausgelegt werden. Zudem werden für bestimmte, zufällig ermittelte

Regionen, Seen und Schneeabschnitte erzeugt sowie Bäume in der Landschaft platziert. Die Szene besteht aus Blätter-, Sand-, Dreck-, Schnee-, Gras-, Holz- sowie Wasserblöcken, die mit einfachen Texturen à 16×16 px versehen wurden. Die Dimensionen eines Blockes wurden so gewählt, dass die Proportionen geschätzt einem halben Meter in der realen Welt entsprechen. Bäume besitzen eine maximale Höhe von ca. 6.00 m. Der Stil der Beispielszene orientiert sich am Computerspiel Minecraft [Mojang Specifications 2011]. Zur Verdeutlichung des Effektes kann sich innerhalb der Szene frei bewegen und bei Bedarf auf eine Vogelperspektive umgeschaltet werden. Abbildung 7 zeigt eine Aufnahme der Szene aus der Sicht des Betrachters.



Abbildung 7: Blick in die Beispielszene bei normalen Größenverhältnissen.

Abbildungen 8 (a) und (b) zeigen die Anwendung des Tiefenschärfe-Effektes aus einer Distanz von ca. 50.0 m zur Oberfläche in zwei unterschiedlichen Regionen. Abbildung 8 (c) und (d) zeigen die gleichen Szenen bei Betrachtung aus einer simulierten Distanz von 6.0 cm zur Schärfeebe. Abbildungen 8 (e) und (f) zeigen zum Vergleich die Anwendung eines linearen Unschärfeffilters, der nach dem in Abbildung 2 gezeigten Verfahren mithilfe eines linearen, vertikal verlaufenden Weichzeichnungsfilters erzeugt wurde.

Die Beispielanwendung wird ohne Anwendung des Tiefenschärfe-Effektes bei einer Darstellung eines Gebietes von 512×512 Blöcken mit 52 Frames pro Sekunde (kurz FPS) ausgeführt. Nach Anwendung des Effektes sinkt der Wert auf 51 FPS.

6 Diskussion

Die Ergebnisse, nach Anwendung des beschriebenen Echtzeit-Tiefenschärfe-Effektes, entsprechen den Erwartungen. Die durch die Anwendung des Tilt-Shift-Effektes entstehenden Größenverhältnisse sind, aus subjektiver Sicht, korrekt und lassen sich mit einem Vergleich zu den von Held et. al. bereitgestellten Referenzbildern bestätigen. Der entstehende Effekt ist somit nicht nur für photorealistische Szenen interessant, sondern kann auch, wie durch die von uns vorgestellte Beispielszene bewiesen wird, auf synthetische Szenarien übertragen werden.

Im Vergleich zu einem nachträglich auf das Bild angewendeten Linsenfilter (siehe Abbildung 8), ergeben sich nur wenig Differenzen. Da das vorgestellte Verfahren physikalisch korrekt unter der Verwendung von Tiefenwerten arbeitet, kann ein genaueres Ergebnis erzielt werden. Dies äußert sich darin, dass, bei Nutzung des linear verlaufenden Unschärfeffektes, Objekte, die eigentlich auf der Schärfeebe liegen müssten, unscharf dargestellt werden und vice versa. Betrachtet man beispielsweise in Abbildung 8 (e) einen der Bäume im Bereich der Fokusebene, kann es zur Weichzeichnung von Baumspitzen kommen, obwohl diese bei Betrachtung der Tiefenwerte auf

der Schärfeebe liegen müssten.

Die beschriebenen Effekte können den in Abbildung 9 gezeigten Farbdifferenzbildern entnommen werden. Deutliche Unterschiede sind innerhalb der Bilder schwarz hervorgehoben, während schwache oder nicht vorhandene Unterschiede als grau bzw. weiß zu erkennen sind. Das vorgestellte Verfahren zeigt, zusammen mit vielen weiteren Verfahren, Artefakte an den Übergängen zwischen Vordergrundebene, Schärfeebe und Hintergrundebene. Objekte, die im Vordergrund im äußeren Bereich der Schärfeebe liegen und die Schärfeebe überlagern, erhalten eine harte Kante, da der Unschärfefilter lediglich einzelne Punkte innerhalb der CoC-Map betrachtet. Zudem lässt sich ein *Ineinanderbluten* beobachten, da für die Erzeugung des Unschärfeeffektes die Farbwerte der umliegenden Pixel benötigt werden. Dieser Effekt tritt ebenfalls am Übergang zwischen Schärfe- und Hintergrundebene auf, was zu einer Art *Glühen* an scharf dargestellten Entitäten führen kann.

7 Zusammenfassung und Ausblick

Innerhalb des vorliegenden Papers wurde ein Verfahren vorgestellt, das die Berechnung eines Tiefenschärfe-Effektes unter Berücksichtigung der Eigenschaften des menschlichen Auges in Echtzeit ermöglicht. Unter Verwendung dessen kann der sogenannte Tilt-Shift-Effekt physikalisch korrekt reproduziert und auf synthetische Szenen angewendet werden.

Das vorgestellte Verfahren basiert auf der Verwendung von Tiefenwerten zur Ermittlung des Circle of Confusion anhand der Linsengleichung. Um eine korrekte Abbildung erzeugen zu können, werden die Eigenschaften des menschlichen Auges berücksichtigt. Die Simulation verschiedener Effekte wird durch die Anpassung der Linsengröße durchgeführt.

Ausgehend von den Ergebnissen dieses Papers, kann das vorgestellte Verfahren auf erweiterte Algorithmen, wie zum Beispiel den Einsatz der Wärmeleitungsgleichung [Bertalmio et al. 2004; Kass et al. 2006] zur Minimierung von Artefakten durch das Einbeziehen von benachbarten Punkten bei der Anwendung der Unschärfe, übertragen werden. Hierbei muss jedoch beachtet werden, dass erweiterte Verfahren deutlich höhere Anforderungen an die Leistung des Grafikprozessors stellen und für komplexe Szenarien ungeeignet sind.

Literatur

- BERTALMIO, M., FORT, P., AND SANCHEZ-CRESPO, D. 2004. Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. *Proceeding 3DPVT '04 Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*.
- DUDASH, B. 2004. Depth of field - controlling camera parameters. Tech. rep., NVIDIA Corporation, 7.
- EARLL HAMMON, J. 2007. Chapter 28. practical post-process depth of field. In *GPU Gems 3*. Addison-Wesley Professional.
- ENGEL, W. 2004. *ShaderX3: advanced rendering with DirectX and OpenGL*, 1 ed. Charles River Media.
- FILION, D., AND MCNAUGHTON, R. 2008. Starcraft 2 – effects techniques. *Advances in RealTime Rendering in 3D Graphics and Games Course – SIGGRAPH 2008*.
- FLICKR, 2011. Flickr group: Tilt-shift miniature fakes. <http://www.flickr.com/groups/tilt-shift-fakes/>, January.



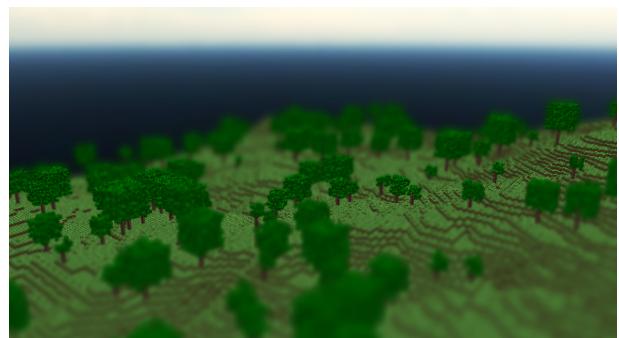
(a) See mit $z_0 = 75.0$ m und $A = 4.6$ mm



(b) Wald mit $z_0 = 75.0$ m und $A = 4.6$ mm



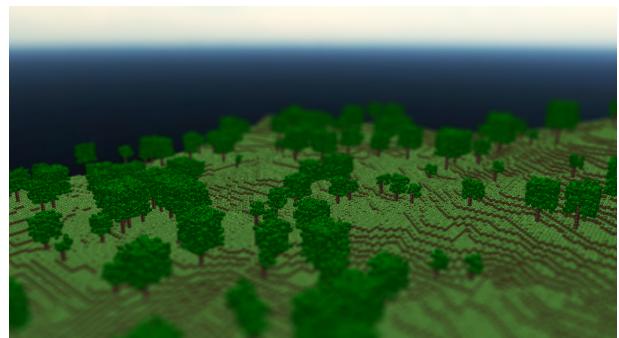
(c) See mit $\hat{z}_0 = 6.0$ cm und $\hat{A} = 5.75$ m



(d) Wald mit $\hat{z}_0 = 6.0$ cm und $\hat{A} = 5.75$ m

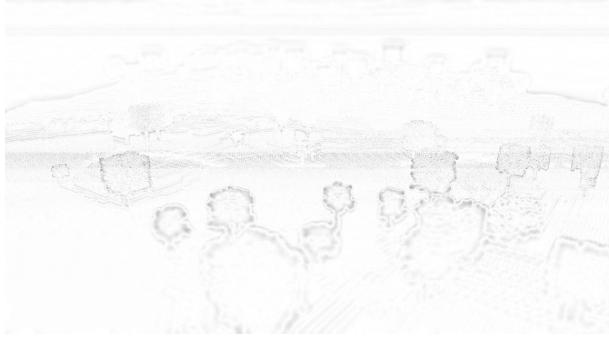


(e) See mit linearem, vertikalem Linsenfilter

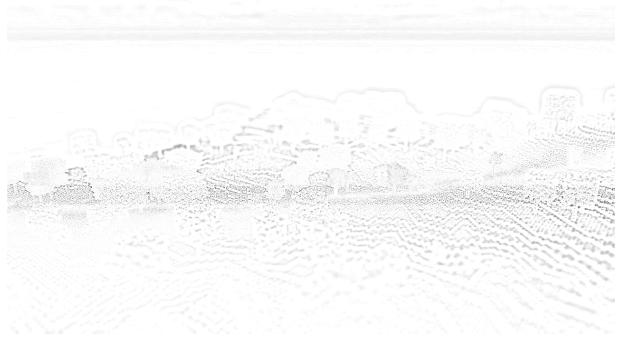


(f) Wald mit linearem, vertikalem Linsenfilter

Abbildung 8: Zu Demonstration des Algorithmus wurden zunächst zwei Bilder aus einer großen Distanz von 75.0 m aufgenommen (a und b). Im Anschluss daran wurde mithilfe des vorgestellten Algorithmus eine Simulation einer Distanz von ca. 6.0 cm durch Vergrößerung der Linse durchgeführt (c und d). Zum Vergleich wurde ausgehend von der CoC-Karte der Ergebnisse ein linearer Grauverlauf abgeschätzt und zur Weichzeichnung mithilfe des Linsenfilters von Adobe Photoshop CS5 verwendet (e und f).



(a)



(b)

Abbildung 9: Farbdifferenzbilder der Beispielszenen zur Verdeutlichung der Unterschiede zwischen dem vorgestellten Echtzeit-Tiefenschärfe-Effekt und der Offline-Berechnung mittels Photoshop.

HELD, R. T., COOPER, E. A., O'BRIEN, J. F., AND BANKS, M. S. 2010. Using blur to affect perceived distance and size. *ACM Transactions on Graphics* 29, 2 (3).

KASS, M., LEFOHN, A., AND OWENS, J. 2006. Interactive depth of field using simulated diffusion on a gpu. Tech. rep., Pixar.

KILGARD, M. J. 2003. Cg in two pages. Tech. rep., NVIDIA Corporation, 1.

LEE, S., KIM, G. J., AND CHOI, S. 2008. Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation. *IEEE Transactions on Visualization and Computer Graphics*.

LEE, S., EISEMANN, E., AND SEIDEL, H.-P. 2009. Depth-of-field rendering with multiview synthesis. *ACM Transactions on Graphics*.

LEE, S., EISEMANN, E., AND SEIDEL, H.-P. 2010. Real-time lens blur effects and focus control. *ACM Transactions on Graphics*.

MCCLOSKEY, S., AND LANGER, M. 2009. Planar orientation from blur gradients in a single image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2318–2325.

MOJANG SPECIFICATIONS, 2011. Minecraft. <http://www.minecraft.net/>.

RANDIMA FERNANDO, M. J. K. 2003. *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*, 1 ed. Addison-Wesley Professional, 3.

RIGUER, G., TATARCHUK, N., AND ISIDORO, J. 2003. Real-time depth of field simulation. *ShaderX² – Shader Programming Tips and Tricks with DirectX 9*.

SCHEUERMANN, T. 2004. Advanced depth of field. Tech. rep., ATI Research, Inc.

TORUS KNOT SOFTWARE, 2011. Ogre. <http://www.ogre3d.org/>.