Logarithmic Perspective Shadow Maps

Konni Hartmann

Universität Göttingen
Seminar Computergrafik, 2009

Inhaltsverzeichnis

Einleitung

Shadow Mapping ist ein von L. Williams [Williams1987] publiziertes Verfahren zur interaktiven Darstellung von Schatten in dreidimensionalen Szenen. Die 1978 vorgestellte Methode leidet unter blickwinkelabhängigen Aliasing-Artefakten und wurde bereits in vielen Varianten verbessert. Die in dieser Ausarbeitung vorgestellte Variante von Lloyd et al. [Lloyd2008] entwickelt hierzu eine Bildfehler-Metrik, die Vergleiche zwischen bestehenden Verfahren erlaubt. Aus dieser Metrik wird eine logarithmische Parametrisierung der Shadow Map bestimmt. Diese

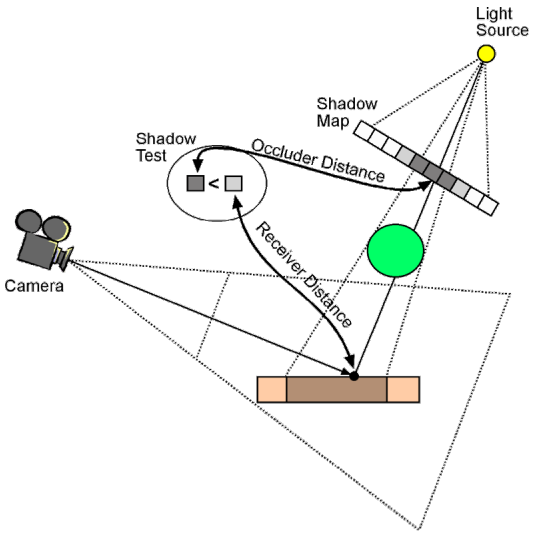


Abbildung 1: Shadow Mapping [Brabec2002, Fig.1, S.2].

Parametrisierung wird mit verschiedenen Techniken wie Warping (deutsch: perspektivische Verzerrung) und Partitionierung kombiniert und erzeugt so bei gleichen Vorraussetzungen deutlich geringere Bildfehler als andere auf Shadow Maps basierte Techniken.

1 Grundlagen

Zunächst wird in diesem Abschnitt das Grundprinzip der von Williams [Williams1987] entwickelten Uniform Shadow Maps dargestellt. Darauf aufbauend werden Probleme dieser erläutert und Lösungskonzepte wie Warping und Partitionierung vorgestellt. Hier wird die Grundlage für die vorgestellte Technik »logarithmic perspective shadow maps« beschrieben, welche im folgenden mit LogPSM abgekürzt wird.

1.1 Uniform Shadow Maps

Zu Beginn wird, wie in Abbildung 1 gezeigt, die Szene aus Sicht der Lichtquelle gezeichnet. Hierbei wird zu jedem Punkt des Rasterisierungsprozesses die Entfernung zur Lichtquelle in einer Textur gesichert. Diese Textur wird ihrer Funktion entsprechend Shadow Map genannt.

Im zweiten Schritt wird die Szene normal gezeichnet. Für jeden Punkt wird wieder die Entfernung zur Lichtquelle bestimmt. Diese wird mit der in der Shadow Map gesicherten Distanz verglichen. Ist die Entfernung größer als der Vergleichswert, so befindet sich ein Hindernis zwischen dem rasterisierten Punkt und der Lichtquelle. Somit liegt der Punkt im Schatten.

Die Shadow Map ist hierbei unabhängig von der Kamera und muss für statische Geometrie und Lichtquellen nur einmal bestimmt werden.

Probleme mit Uniform Shadow Maps Durch die gleichmäßige Verteilung der Auflösung einer Shadow Map über ihre gesamte Fläche können je nach Ausrichtung von Kamera und Licht-

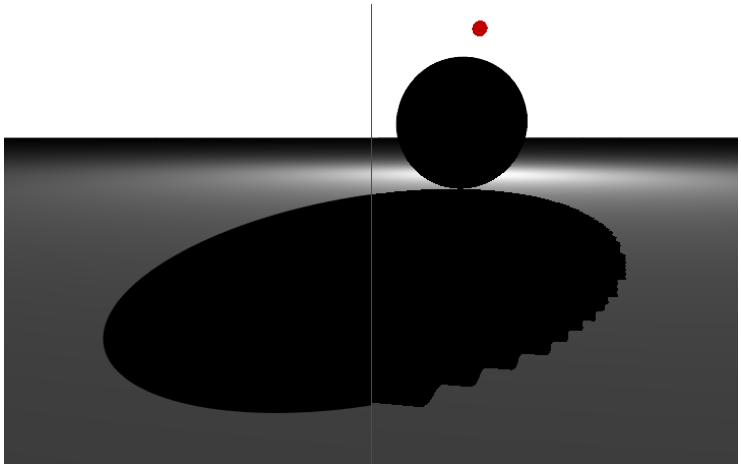


Abbildung 2: Beispiel für Aliasing Fehler. Der rote Punkt oben rechts stellt eine Punktlichtquelle dar. Der linke Abschnitt wurde mittels Raytracing bestimmt. Für den rechten Abschnitt wurde Uniform Shadow Mapping genutzt.

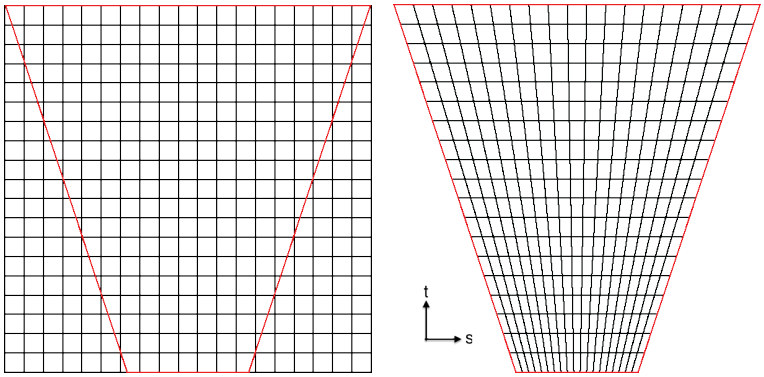
quelle starke Aliasing-Fehler enstehen (siehe Abbildung 2). Im Extremfall scheint die Lichtquelle frontal in die Kamera. Dies wird »duelling frustra« gennant, da Kamerakegel und Lichtkegel gegeneinander ausgerichtet sind. Ein weiterer Fall von Aliasing-Fehlern entsteht bei der Vergrößerung des Bildauschnittes. Da »Uniform Shadow Maps« nicht an den Sichtbereich angepasst werden, wird in diesem Fall ein Großteil der Auflösung für Bereiche der Shadow Map genutzt, die außerhalb des angezeigten Bildauschnittes liegen.

1.2 Warping

Um die zuvor beschriebenen Aliasing-Fehler zu reduzieren bieten sich verschiedene Möglichkeiten an. Warping beschreibt eine Reihe von Verfahren, die durch angepasste Transformationen der Shadow Map versuchen die Auflösung entsprechend des Sichtbereiches zu verteilen. Hierbei soll nach Möglichkeit der in der Shadow Map gezeichnete Bereich eng auf den angezeigten Bereich begrenzt werden. Desweiteren ist zu beachten, dass nahe der Kamera eine höhere Auflösung benötigt wird um Aliasing-Fehler zu vermeiden, als weiter entfernt.

Perspektivisches Warping Ein Beispiel für Warping-Verfahren bieten perspektivische Transformationen. Abbildung 3 zeigt ein solches Verfahren um die effektive Auflösung in s-Richtung der Textur zu erhöhen. Eine möglichst gute Anpassung der Auflösung in t-Richtung an den Sichtbereich ist eine der Hauptaufgaben dieser Arbeit.

Beispiele für perspektivisches Warping sind »perspective shadow maps« (PSM) [Stamm2002] und »light space perspective shadow maps« (LiPSM) [Wimmer2004]



(a) (b)

Abbildung 3: Warping der Texel in s-Richtung um effektive Auflösung zu erhöhen. In der Szene blickt die Kamera vom unteren zum oberen Bildrand. Die Lichtquelle beleuchtet die Szene orthogonal zur Sichtrichtung von oben. Bild 3a zeigt eine enge Begrenzung des Sichtbereiches mittels »Uniform Shadow Maps«. Bild 3b zeigt eine perspektivische Transformation in s-Richtung.

1.3 Partitionierung

Eine weitere Methode um die Auflösung der Shadow Map nahe des Betrachters zu erhöhen bietet Partitionierung. Der Sichtbereich wird hierbei in einzelne Abschnitte, genannt »Regions of Interest«, eingeteilt. Für jeden Abschnitt wird eine eigene Shadow Map genutzt.

Der Vorteil mehrerer Shadow Maps liegt darin, dass für jeden Bereich eine eigene Parametrisierung, also auch eine eigene »Warping«-Methode genutzt werden kann. Hierbei steigt der Verwaltungsaufwand, da die Szene entweder mehrfach gezeichnet oder in Shader-Programmen ermittelt werden muss, in welchem Bereich das aktuelle Bildfragment liegt.

z-Partitionierung Bei der z-Partitionierung wird das Sichtfeld, wie in Abbildung 4 gezeigt, entlang z-Achse der Kamera in einzelne Segmente unterteilt. Für jeden Abschnitt wird eine eigene Shadow Map erzeugt. Diese bieten eine wesentlich engere Abdeckung des Sichtbereiches. z-Partitionierung lässt sich in einem Shader-Programm auf der Grafikkarte realisieren, da der Abschnitt in der sich ein Fragment befindet direkt aus dessen z-Koordinate ermitteln lässt. Demnach ist dieses Partitionierungs-Verfahren einfach zu implementieren. Im Folgenden wird mit ZP# z-Partitionierung mit # Partitionen beschrieben.

Ein Beispiel für z-Partitionierung sind »cascaded shadow maps« [Engel2007]

Adaptive Partitionierung Die adaptive Partitionierung nach »adaptive shadow maps« [Fernando2001] führt z-Partitionierung im Zweidimensionalen fort. Hierzu wird der Sichtbereich durch einen Quad-Tree zerlegt. Es werden Bereiche mit vielen Hindernissen zusätzliche Partitionen zugewiesen. Das Verfahren muss hierbei in jedem Bild die Szene analysieren, eine Zerlegung bestimmen und bei Bedarf fehlende Partitionen zeichnen. Bei dieser Partitionierungsmethode können

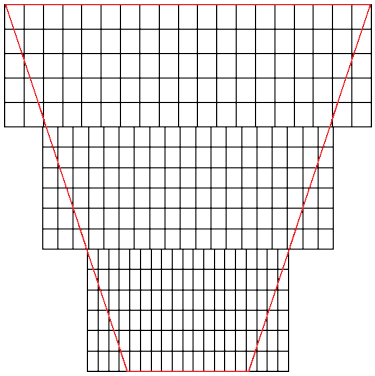


Abbildung 4: z-Partitionierung

Shadow Maps für sehr große Flächen erzeugt werden. Problematisch ist die Analyse der Szene, die auf der CPU stattfinden muss. Aufgrund des Aufwandes wird adaptive Partitionierung nicht weiter betrachtet.

Face-Partitionierung Eine weitere Partitionierungsmethode, die auch bei Punktlichtquellen genutzt werden kann ist Face-Partitionierung[1] (FP). Hierbei werden die Ebenen des Sichtbereiches in Shadow Maps abgebildet. Es werden entsprechend des zweidimensionalen Falls (siehe Abbildung 5) die Seiten des Sichtbereiches in Eingangs- und Ausgangsflächen unterteilt. Die von der Lichtquelle und den Ausgangsflächen aufgespannten Pyramiden zerlegen den Sichtbereich und den Bereich, in dem potentielle Hindernisse die Lichtquelle verdecken, in disjunkte Partitionen. Die Ausgangsflächen werden durch einzelne Shadow Maps abgedeckt. So wird üblicherweise eine geringe Zahl an Partitionen erzeugt (3-4 im Durchschnitt). Durch zusätzliche Nutzung von Warping-Methoden lassen sich Aliasing-Fehler stark reduzieren.

2 Fehler-Metrik

Die bisher subjektive Bestimmung von Aliasing-Fehlern soll in diesem Abschnitt mathematisiert werden. Es wird eine Metrik bestimmt, welche es erlaubt »Shadow Mapping« - Verfahren aufgrund ihrer Aliasing-Fehler zu vergleichen. Tabelle 1 führt einige wichtige Symbole ein, die im Folgenden genutzt werden.

2.1 Herleitung einer Aliasing-Fehler-Metrik

Zunächst wird nur der zweidimensionale Fall betrachtet. Um Aliasing-Fehler zu Quantifizieren werden die Sampling-Abstände auf der Bildebene, sowie auf der Lichtebene verglichen. Aliasing ensteht dann, wenn der Sample-Abstand auf der Lichtebene größer als der auf der Bildebene ist. Nach Abbildung 6 lässt sich für die normalisierten Bild- und Shadow Map Koordinaten eine Metrik

[1] Dieses Verfahren ergiebt sich aus den Analyzen von Lloyd et al. [Lloyd2008] und stellt die Basis von LogPSM für Punktlichtquellen dar.

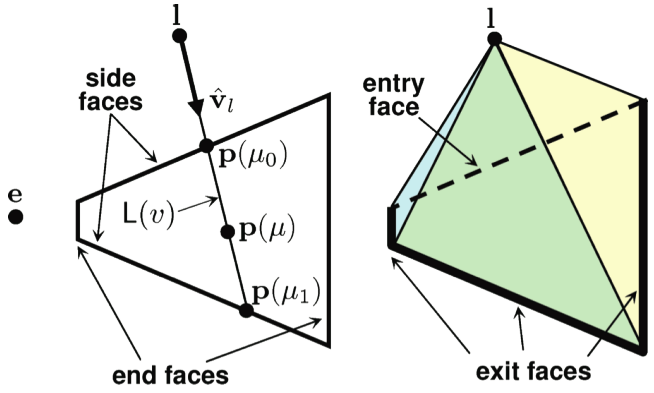


Abbildung 5: Face-Partitionierung [Lloyd2008, Fig.4, S.8]. Die Kamera befindet sich in Punkt e. Das Trapez stellt den zwei-dimensionalen Sichtbereich der Kamera dar. In Punkt l befindet sich eine Punktlichtquelle, die die Szene innerhalb des Sichtbereiches beleuchtet.

m	Aliasing-Fehler
$\tilde{m}$	perspektivischer Aliasing-Fehler
$\tilde{M}$	Maximum von $\tilde{m}$ über eine Linie $\mathbf{L}$ durch das Licht
F, G	Shadow Map Parametrisierung und die Inverse
(i, j)	normalisierte Koordinaten der Bildebene der Kamera
(u, v)	normalisierte Koordinaten der Lichtebene
(s, t)	normalisierte Shadow Map Koordinaten
$r_i \times r_j$	Auflösung der Bildebene
$r_s \times r_t$	Auflösung der Shadow Map

Tabelle 1: Wichtige Symbole der folgenden Rechnungen

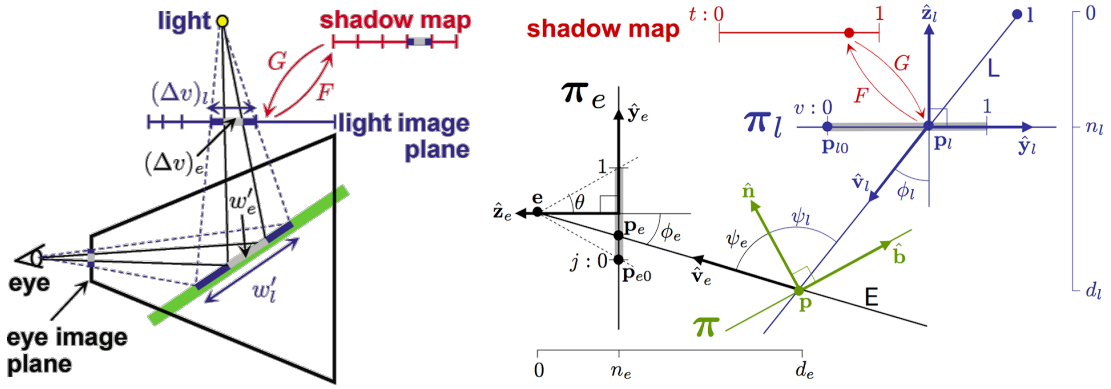


Abbildung 6: Bestimmung der Metrik für Shadow Maps nach[Lloyd2008, Fig.2, S.4]. (Links) Sample-Punkte für Bildpixel und Shadow Map Texel. (Rechts) Der Sample-Abstand hängt mit den Ableitungen der Funktion, welche einen Punkt $t \in [0,1]$ der Shadow Map auf $\mathbf{p}_l$ in der Lichtebene π_l. Dieser wird durch das Licht auf die planare Ebene π projeziert und schließlich von der Kamera auf den Punkt $\mathbf{p}_e$ auf der Bildebene π_e.

$$m = \frac{r_j}{r_t}\frac{\mathrm{d}j}{\mathrm{d}t}$$

bestimmen. Für $m > 1$ ensteht Aliasing. $m < 1$ bedeutet, dass unnötigt viele Informationen für einen Punkt in der Shadow Map gesichert sind.

Ableiten des Fehlers (geometrisch in 2D) Um das Verhältnis $\frac{\mathrm{d}j}{\mathrm{d}t}$ zu bestimmen lässt sich für den zweidimensionalen Fall ein vereinfachter, geometrischer Ansatz nutzen. Wie in Abbildung 6 im linken Bild zu erkennen lässt sich m als das Verhältnis zwischen w'_l und w'_e beschreiben:

$$m = \frac{w'_l}{w'_e} \tag{1}$$

Zunächst wird die Breite eines Pixels der Bildebene, welcher sich im Abstand n_e zur Kamera befindet, betrachtet. Über den Strahlensatz wird diese in einer Entfernugn von d_e, dem Punkt an dem der Strahl die Oberfläche π schneidet mit

$$w_e^{\perp \hat{\mathbf{z}}_e} = \frac{W_e}{r_j}\frac{d_e}{n_e}$$

wie in Abbildung 7 zu sehen, beschrieben.

Unter der Annahme, dass der Sampling Strahl der Kamera schmal genug ist, lassen sich die Begrenzungen des Strahls als parallel annehmen. Die Breite des projezierten Strahls lässt sich dann folgendermaßen bestimmen:

$$w'_e = \frac{w_e}{\cos\psi_e} = \frac{W_e}{r_j}\frac{d_e}{n_e}\frac{\cos\phi_e}{\cos\psi_e} \tag{2}$$

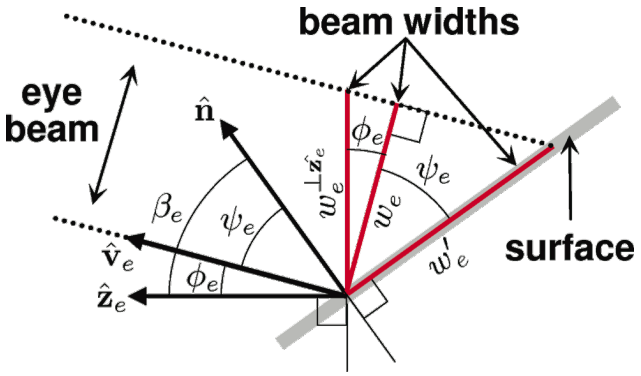


Abbildung 7: Geometrische Herleitung der Metrik m [Lloyd2008, Fig.3, S.6]

Für die Projektion eines Shadow Map Texels wird äquivalent vorgegangen. Somit ergibt sich die projezierte Breite eines Shadow Map Texels auf der Oberfläche π als

$$w'_l = \cdots = \frac{W_l}{r_t}\frac{\mathrm{d}G}{\mathrm{d}t}\frac{d_l}{n_l}\frac{\cos\phi_l}{\cos\psi_l} \tag{3}$$

Durch Einsetzen der Formeln 2 und 3 in die Formel 1 wird die folgende Beschreibung des Aliasing-Fehlers erhalten:

$$m = \frac{r_j}{r_t}\underbrace{\frac{\mathrm{d}G}{\mathrm{d}t}\frac{W_l}{W_e}\frac{n_l}{n_e}\frac{d_l}{d_e}\frac{\cos\phi_l}{\cos\phi_e}\frac{\cos\psi_l}{\cos\psi_e}}_{=\mathrm{d}j/\mathrm{d}t} \tag{4}$$

Diese Metrik ist sowohl für Punktlichtquellen als auch für direktionale Lichtquellen gültig. Wird eine Punktlichtquelle entlang einer Richtung von der Szene entfernt, so konvergiert sie im unendlichen Abstand gegen eine direktionale Lichtquelle.

2.2 Parametrisierung von m

Szenen-unabhängige Approximierung Nach der im vorherigen Abschnitt definierten Darstellung von m aus Gleichung 4 lässt sich m in die folgenden Faktoren zerlegen:

$$m = \frac{r_j}{r_t}\underbrace{\left(\frac{\mathrm{d}G}{\mathrm{d}t}\right)}_{=:\delta_l}\underbrace{\left(\overbrace{\left(\frac{W_l}{W_e}\frac{n_l}{n_e}\frac{d_l}{d_e}\frac{\cos\phi_l}{\cos\phi_e}\right)}^{=:1/\tilde{\delta}_e}\frac{\cos\psi_l}{\cos\psi_e}\right)}_{=:1/\delta_e}$$

Der Ausdruck $\frac{\cos\psi_l}{\cos\psi_e}$ basiert auf der Ausrichtung der Oberflächen in der Szene und kann allgemein nicht parametrisiert werden. In Ausnahmefällen ist dies möglich und erlaubt dann ein 1 zu 1 Matching von Schatten Texeln zu Bildpixeln, wie in den Verfahren von Chong und Gortler

[Chong2004] beschrieben. Durch Ignorieren des geometrie-abhängigen Termes $\frac{\cos\psi_l}{\cos\psi_e}$ in m wird der perspektivische Aliasing-Fehler

$$\tilde{m} := \frac{r_j}{r_t}\frac{\delta_l}{\tilde{\delta}_e} = \frac{w_e}{w_l}$$

eingeführt. Hierbei gilt $\tilde{m} = m$ wenn die Oberfläche π parallel zum Halbvektor zwischen Sicht- und Schattenstrahlen liegt.

Ideal wäre nun $\tilde{m} = 1$ über den gesamten Bildbereich. Da diese Beschränkung eine Parametrisierung erschwert, wird nur $\tilde{m} \leq 1$ gefordert. Somit kann für den Fall $\tilde{m} = m$ garantiert werden, dass jedem Bildpunkt mindestens ein Schatten-Texel zugeordnet wird.

Für den dreidimensionalen Fall kann ähnlich vorgegangen werden, allerdings werden die Berechnungen deutlich komplexer, weshalb hier auf die zugrundeliegende Publikation von Lloyd [Lloyd2006] verwiesen wird.

Strenge untere Grenze für $\tilde{\delta}_e$ Der Ausdruck $\frac{r_j}{r_t}\delta_l$ wird durch die Wahl der Auflösung sowie der perspektivischen Projektion der Szene festgelegt. Demnach soll nun eine strenge untere Grenze δ_b der perspektivischen Faktoren von δ_e bestimmt werden:

$$\delta_b(v) \in \left[\frac{1}{C}\min\left(\tilde{\delta}_e(v)\right), \min\left(\tilde{\delta}_e(v)\right)\right], C > 1$$

Hierzu wird der Sichtbereich nach der Idee der »Face«-Partitionierung in einzelne Bereiche unterteilt (siehe Abbildung 8) und δ_b jeweils für diese ausgewertet.

Hier werden entsprechend nur die Ergebnisse präsentiert:

$$\begin{aligned}
\delta_{b,t}^{end}(v) &= \frac{\cos\theta}{\cos\phi_{ly}(v)} \\
\delta_{b,s}^{end}(u) &= \frac{\cos\theta}{\cos\phi_{lx}(u)} \\
\delta_{b,t}^{side}(v) &= \frac{W_e}{(y_1 - y_0)}\frac{y(v)}{y_0}\frac{\cos\theta}{\cos\phi_{ly}(v)} \\
\delta_{b,s}^{side}(u,v) &= \frac{y(v)}{y_1}\frac{\cos\theta}{\cos\phi_{lx}(u)}
\end{aligned}$$

3 Vereinfachte Parametriesung der Verteilungsfunktion δ_b

Da δ_b selbst zu komplex ist um als Parametrisierung für Echtzeitanwendungen eingesetzt zu werden, soll im Folgenden eine Parametrisierung gefunden werden, deren Aliasing-Fehler vergleichbar mit denen von δ_b ist. Diese wird genutzt um das LogPSM-Verfahren zu motivieren.

3.1 Einfache Parametrisierungen

Zunächst sollen einige einfache Parametrisierungen der Shadow Map vorgestellt werden. Hierbei sind s,t die Koordinaten auf der Shadow Map und u,v die Koordinaten auf der Lichtebene

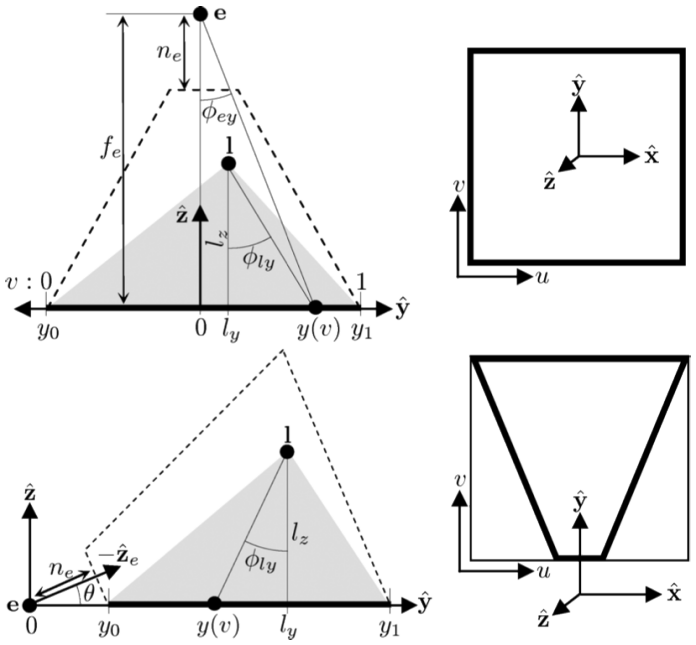


Abbildung 8: End- und Side-Face aus [Lloyd2008, Fig.5, S.9]. Das obere Bild zeigt links eine Draufsicht der Szene und am unteren Rand die End-Fläche des Sichtbereiches. Rechts ist die Lichtebene und deren Koordinatensystem aufgezeigt. Die beiden unteren Bilder zeigen eine Side-Fläche des Sichtbereiches und deren Lichtebene. Bei den Side-Flächen ist zu beachten, dass diese Trapezförmig sind.

(siehe Abbildung 6). Eine gleichverteile Parametrisierung $\mathbf{F}_{un}(u,v)$ führt zu den bereits bekannten Uniform Shadow Maps:

$$\begin{bmatrix} s \\ t \end{bmatrix} = \mathbf{F}_{un}(u,v) = \begin{bmatrix} u \\ v \end{bmatrix}$$

Perspektivische Parametrisierung wie in PSM und LiPSM liefern die folgende allgemeine Darstellung

$$\begin{bmatrix} s \\ t \end{bmatrix} = \mathbf{F}_{p}(u,v) = \begin{bmatrix} \frac{p_0 x(u)+p_1(y(v)+a)}{y(v)+a} \\ \frac{p_2(y(v)+a)+p_3}{y(v)+a} \end{bmatrix} \tag{5}$$

mit den frei wählbaren perspektivischen Parametern p_0, p_1, p_2, p_3.

Eine weitere Idee besteht in der logarithmische Parametrisierung in t-Richtung. Das eine logarithmische Parametrisierung eine gute Lösung für perspektivische Aliasing-Fehler darstellt wurde bereits von Wimmer et al., Zhang et al. sowie Lloyd et al. vorgeschlagen [Zhang2006, Wimmer2004, Lloyd2006]. Hierbei erhält man die folgende Darstellung

$$t = \mathbf{F}_{l}(v) = \frac{\log\left(\frac{y(v)+a}{y_0+a}\right)}{\log\left(\frac{y1+a}{y_0+a}\right)} \tag{6}$$

mit dem frei wählbaren Parameter a.

3.2 Logaritmisch-Perspektivische Parametrisierung

Die Idee der LogP-Parametrisierung besteht darin logarithmische- und perspektivische Parametrisierung (5 und 6) zu kombinieren. Somit wird eine Parametrisierung erhalten, die sowohl in s-, als auch in t-Richtung δ_b gut approximiert:

$$\begin{bmatrix} s \\ t \end{bmatrix} = \mathbf{F}_{lp}(u,v) = \begin{bmatrix} F_{p,s}(u,v) \\ c_0 \log\left(c_1 F_{p,t}(u,v)+1\right) \end{bmatrix}$$

$$c_0 = \frac{-1}{\log\left(\frac{y_1+a}{y_0+a}\right)}$$

$$c_1 = -\frac{y_1-y_0}{y_1+a}$$

Abbildung 9 zeigt F_{un} und F_{lp} im Vergleich.

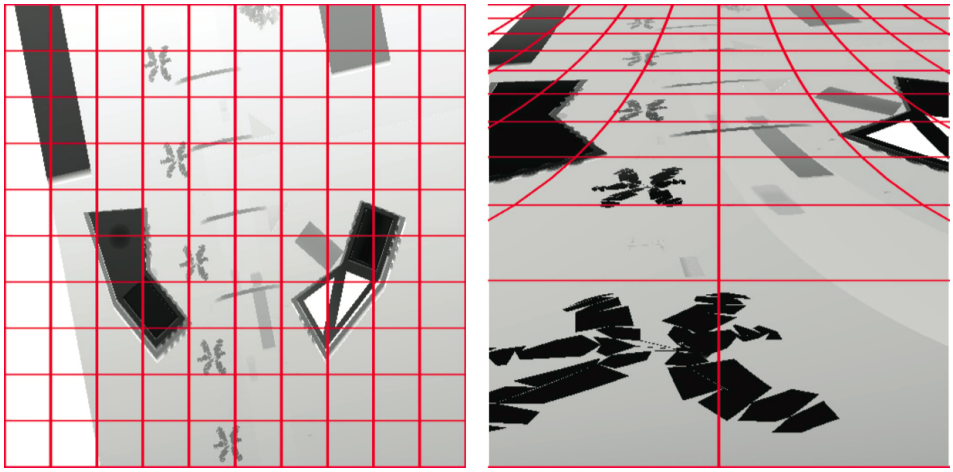


Abbildung 9: Vergleich F_{un} (links) mit F_{lp} (rechts) [Lloyd2008, Fig.8, S.15]. Die Bilder zeigen die Shadow Map einer direktionalen Lichtquelle, die von oben auf eine Szene scheint. Die Kamera befindet sich am unteren Rand des Bildes und blickt parallel zum Boden zum oberen Rand des Bildes.

Fehlerschranken für F_{un}, F_p, F_l und F_{lp} Um die gefundenen Parametrisierungen zu vergleichen wird die kritische Auflösung genutzt, ab der kein Aliasing im Sichtbereich auftritt. Diese ist für Side- und End-Faces in t-Richtung $r_t = R_t r_j$ mit dem kritischen Auflösungsfaktor $R_t = \max_v \left(\frac{\delta_l(v)}{\delta_b(v)} \right)$. Für die Side-Faces in s-Richtung ist diese $r_S^{side} = R_s^{side} r_i$ mit $R_s^{side} = \max_{(u,v)\in\mathcal{F}} \left(\frac{\delta_l(u,v)}{\delta_b^{side}(u,v)} \right)$. Die Kombination von R_t und R_s ergibt den Storagefaktor $S = R_s \times R_t$, der die minimale Auflösung in Texeln der Shadow Map pro Pixel der Bildebene angiebt. Da End-Faces bereits mit gleichverteilter Parametrisierung (F_{un}) bestmöglich bestimmt werden, genügt die Betrachtung der Side-Faces mit entsprechend $S^{side} = R_s^{side} \times R_t$. Für die Bestimmung von R und S müssen desweiteren n_e und f_e, die Nah- und Fernebene des Sichtbereiches, bekannt sein. Tabelle 2 zeigt R und S für die vorgestellten Parametrisierungen in O-Notation. Die

Tabelle 2: Übersicht der maximalen Fehler verschiedener Parametrisierungen [Lloyd2008, Table III, S.11]. F_b entspricht hierbei einer aus δ_b abgeleiteten Parametrisierung.

Parametrisierung	R_s^{end}	R_s^{side}	R_t^{side}	S^{side}
Untere Grenze (F_b)	$O(1)$	$O(1)$	$O\left(\log\left(\frac{f_e}{n_e}\right)\right)$	$O\left(\log\left(\frac{f_e}{n_e}\right)\right)$
Uniform (F_{un})	$O(1)$	$O\left(\frac{f_e}{n_e}\right)$	$O\left(\frac{f_e}{n_e}\right)$	$O\left(\left(\frac{f_e}{n_e}\right)^2\right)$
perspektivisch (F_p)	-	$O(1)$	$O\left(\sqrt{\frac{f_e}{n_e}}\right)$	$O\left(\frac{f_e}{n_e}\right)$
logarithmisch (F_l)	-	-	$O\left(\log\left(\frac{f_e}{n_e}\right)\right)$	-
LogP (F_{lp})	-	$O(1)$	$O\left(\log\left(\frac{f_e}{n_e}\right)\right)$	$O\left(\log\left(\frac{f_e}{n_e}\right)\right)$

logarithmisch perspektivische Parametrisierung (F_{lp}) liefert hier die selben oberen Schranken wie eine direkt aus δ_b bestimmte Parametrisierung (F_b) und ist somit die Basis des LogPSM Algorithmus.

3.3 LogPSM Algorithmus

Für alle Partitionierungsverfahren sind die groben Schritte des Algorithmus gleich:

1. Berechne Parametrisierung der Shadow Maps für jede Partition. Für Side-Faces LogPSM, für End-Faces UniformSM
2. Verteile Auflösung basierend auf maximalem Fehler der Partitionen.
   a) Bestimme je Partition δ_l/δ_b und so R_s, R_t und S.

      Bei direktionalen Lichtquellen ist Fehler über gesammte Partition monoton. Es genügt Auswertung an den Vertices.

      Bei Punktlichtquellen Fehler approximieren durch Abstandsvektor von Face und Lichtquelle sowie Auswertung an den Vertices der Partition.

   b) Verteile innerhalb Partition Auflösung nach R_s und R_t
3. Rendere Shadow Maps
4. Rendere das Bild mit Shadow Maps

Optimierung der Darstellung Wird LogP zusammen mit Face-Partitionierung genutzt, können einzelne Texel stark verzerrt werden. Insbesondere bei Shadow Maps niedriger Auflösung enstehen ungewollte Scherungsartefakte, wie in Abbildung 10 links zu sehen. Um Bildfehler zu reduzieren bieten sich zwei Korrekturverfahren an.

Beim **FPc** (Face-Partitioning with coordinate frame adjustment) werden die Texel einer Face-Partition am Halbvektor der Seitenkanten der Partition und nicht anhand der Partition selbst ausgerichtet. Dieser Effekt ist auf Abbildung 10 oben rechts zu erkennen.

In einigen Fällen genügt die Anpassung des Koordinatensystems nicht um Scherungsartefakte zu beseitigen. Hier hilft die Teilung der Partition entlang des Halbvektors beider Seitenkanten. Für die beiden Unterpartitionen wird jeweils die FPc-Methode eingesetzt um Scherungsartefakte zu beseitigen. Dieses Verfahren wird auch **FPcs** (Face-Partitioning with coordinate frame adjustment and face splitting) genannt.

4 Bewertung

4.1 Implementierung auf aktueller Hardware

Das beschriebene Verfahren LogPSM ist auf aktueller Hardware nicht ohne hohen Aufwand realisierbar. Ein großes Problem stellt die geforderte logarithmische Rasterisierung dar. Bisher unterstützen Grafikkarten nur lineare Rasterisierung. Mögliche Umsetungen auf bestehender Hardware sind:

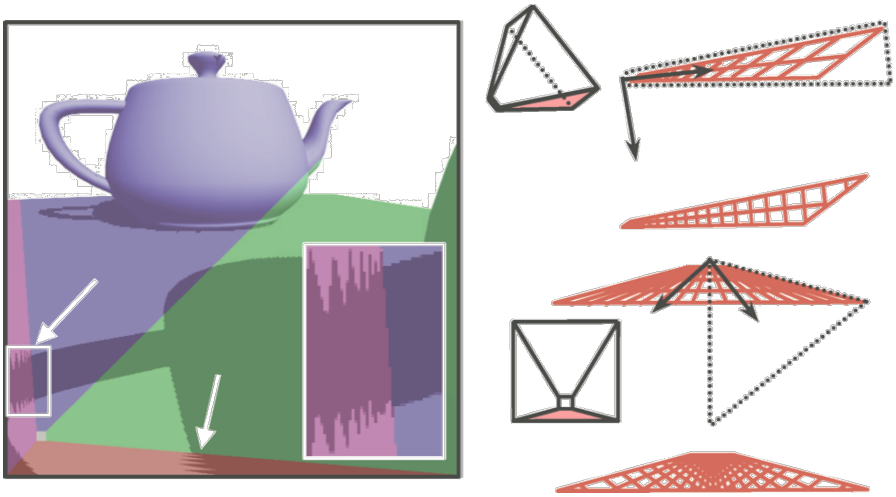


Abbildung 10: Scherungsartefakte beim Einsatz von Face-Partitionierung [Lloyd2008, Fig.10, S.16]. (Rechts) FPc und FPcs auf eine Partition angewandt.

Partitioning Durch eine große Anzahl an Partitionen wird die logarithmische Rasterisierung angenähert. Hierbei steigt allerdings mit der Zahl der Partitionen auch der Verwaltungsaufwand. Dies entspricht »Cascaded Shadow Maps«.

- große Anzahl an Partitionen simuliert Log-Transformation
- entspricht Cascaded Shadow Maps
- hoher Verwaltungsaufwand

Vertex-Warping Über die programmierbare Grafikpipeline der Grafikkarte können die Koordinaten einzelner Vertizes logarithmisch transformiert werden. Dies ist ohne Änderungen an bestehender Hardware möglich. Um Bildartefakte möglichst gering zu halten sollte bei dieser Methode die Szene etwa gleichmäßig durch eine hohe Zahl an Vertizes beschrieben werden. Ist dies nicht der Fall, so können über einen Geometrie-Shader weitere Vertizes generiert werden. Bisher wurde diese Möglichkeit noch nicht ausgiebig getestet. Problematisch ist jedoch, dass viele Vertextransformationen benötigt werden.

- große Anzahl an Vertizes notwendig
- adaptive Anpassung durch Geometrie-Shader möglich
- viele Vertextransformationen nötig

Fragment-Discarding Lloyd et al. [Lloyd2008] nutzen für ihre Simulation des LogPSM-Algorithmus Fragment-Shader um logarithmisch zu Rastern. Hierzu werden zunächst Vertizes transformiert und mittels Feedback-Methoden zurück in das Programm gelesen. Nun wird für jedes logarithmisch transformierte Dreieck ein dieses umschließendes Viereck (engl.: Bounding

Tabelle 3: Storage Faktor für direktionale Lichtquelle über alle Eintrittswinkel [Lloyd2008, Fig.12, S.18]

Algorithmus	$\min S$	$\max S$	$\min S / \max S$	$\varnothing S$	Ø Partitionen
Standard	8.65×10^5	2.00×10^6	2.31	1.32×10^6	1
FPcs+P	865	3170	3.66	1880	3.8
ZP5+P	12.9	159	12.3	53.7	5
FPcs+LogP	5.98	27.7	4.62	15.6	3.8

Box) berechnet und gezeichnet. Im Fragment-Shader wird dann für jedes Fragment bestimmt, ob es auf dem transformierten Dreieck liegt. Ist dies nicht der Fall, so wird das Fragment verworfen, ansonsten wird die Entfernunng bestimmt und die Tiefeninformation ausgeschrieben. Ein Großteil der Zeit wird für die Bestimmung der Bounding-Box benötigt. Dieser Schritt kann möglicherweise durch einen Geometrie-Shader beschleunigt werden, dennoch bleibt logarithmische Rasterisierung deutlich langsamer also rein Lineare.

- optimales Bildergebnis
- Verwerfen von Fragmenten verringert Leistung
- etwa um Faktor 10 langsamer als normales Sampling

4.2 Vergleiche

Aliasing-Fehler Im folgenden sollen verschiedene Shadow Mapping Verfahren in den vorgestellten Metriken verglichen werden. Die Abbildung 11

zeigt die Ergebnisse farbkodiert. Zu erkennen ist, dass LogP Fehler einheitlicher über das Bild verteilt. Desweiteren sind im Gegensatz zu ZP5+P (z-Partitionierung mit 5 Bereichen und perspektivischen Warping) die Übergänge zwischen einzelnen Partitionen nicht erkennbar.

Benötigte Auflösung Abschließend sollen verschiedene Shadow Mapping Verfahren in Hinsicht auf den benötigten Speicherplatz bewertet werden. Hierzu wird in Tabelle 3 der Storage Faktor S aufgezeigt. Aus der Tabelle ist zu sehen, dass LogP in Kombinationen mit Face-Partitionierung für das gleiche Bildergebnis wie andere Verfahren deutlich geringere Auflösungen in den Shadow Maps benötigt. Sowohl Durchschnitt, Minimum und Maximum des Storagefaktors liegen deutlich unter den Werten anderer Methoden.

5 Zusammenfassung und Ausblick

Dieser Abschnitt fasst die Ergebnisse der vorgestellten Arbeit »Logarithmic Perspective Shadow Maps« von Lloyd et al. [Lloyd2008] zusammen.

Unter den Shadow Mapping Verfahren stellt LogPSM die Methode mit dem bisher niedrigsten perspektivischen Aliasing-Fehler dar. Der Storagefaktor ist für beliebige Szenen mit $O\left(\log\left(\frac{f_e}{n_e}\right)\right)$ begrenzt.

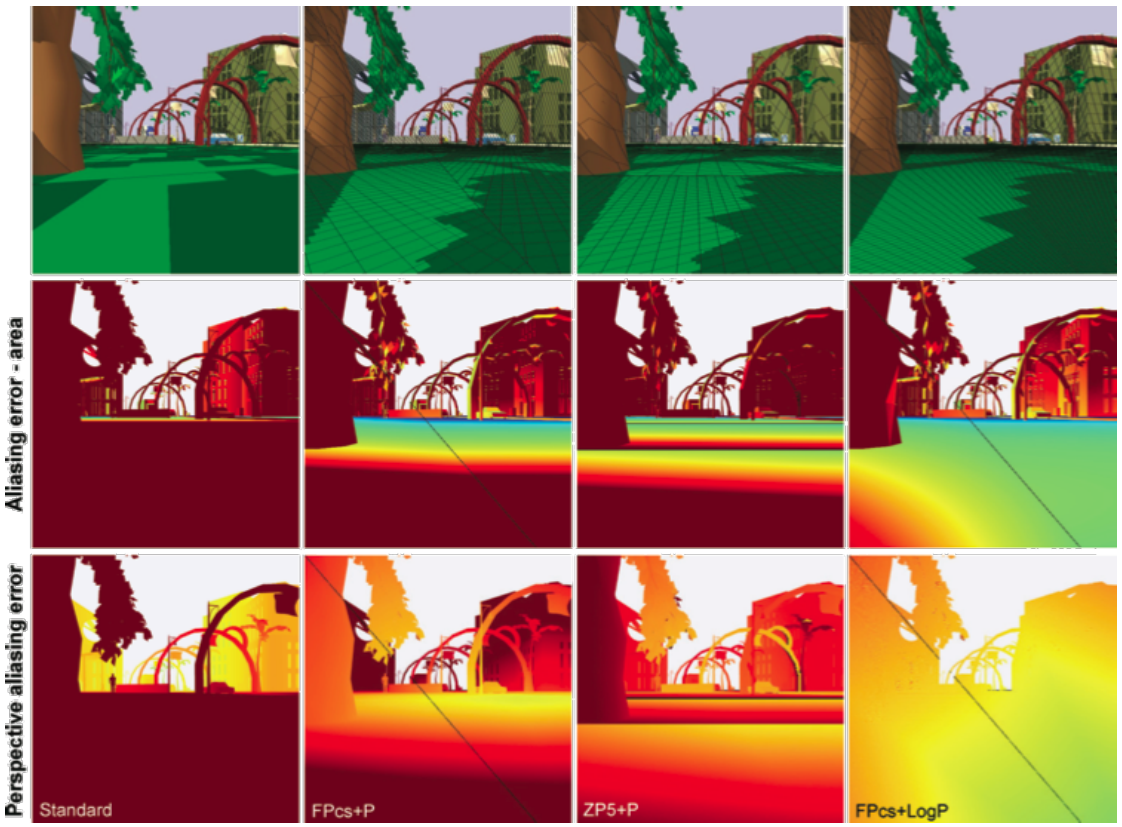


Abbildung 11: Farbkodierte Darstellung verschiedener Fehler-Metriken [Lloyd2008, Fig.12, S.18]. Rot bedeutet einen höheren Fehler. In der obersten Zeile ist die gerenderte Szene gezeigt. Hierbei wurde alle 5 Texel eine Linie über das Bild gezeichnet, die die Shadow Map visualisiert. In der Szene strahlt die Lichtquelle in Richtung der Kamera. In der zweiten Zeile ist die Fläche eines Shadow Map-Texels mit der Fläche eines Bildpixels verglichen. Somit wird hier m visualisiert. In der dritten Zeile ist die Fehlermetrik $\tilde{m}$ dargestellt.

Die vorgestellte logarithmische Parametrisierung lässt sich problemlos mit verschiedenen Partitionierungs-Verfahren verbinden. Sie ist für Punktlichtquellen und direktionale Lichtquellen gleichermaßen geeignet.

Der vorgestellte LogPSM-Algorithmus ist auf bisheriger Hardware nicht realisierbar, da fehlende logarithmische Rasterisierung auf der Hardware simuliert werden muss. Eine nicht optimierte Fassung des LogPSM-Algorithmus erzielte Bildraten von 2-3 BPS[2]. Somit sind spezielle Hardwareerweiterungen notwendig, die aufgrund ihres sehr spezifischen Characters mit Konsumergrafikkarten der nächsten Generationen nicht erwartet werden können. Intels Larabee-Plattform könnte hier deutlich bessere Bildraten erzielen, allerdings ist noch unklar, welchen Geschwindigkeitsvorteil diese bringt. Lloyd et al. zeigen weiterhin, dass Erweiterungen der Hardware inkrementell vorgenommen werden können und präsentieren die Umsetzung bestehender Verfahren auf logarithmische Rasterisierung wie Shadow Map Komprimierung.

Bisher ist die FPcs-Methode zur Reduzierung von Scherungsartefakten nur für direktionale Lichtquellen umgesetzt. Durch weitere Forschungsarbeit könnte FPcs auch für Punktlichtquellen realisiert werden.

Demnach bieten LogPSM ein gutes theoretisches Fundament für partitionierungs- und warpingbasierte Shadow Mapping Verfahren, allerdings fehlt bisher die Hardware für eine interaktive Umsetzung.

Literatur

[Brabec2002] St. Brabec, Th. Annen, H. Seidel. *Practical Shadow Mapping*. J. Graph. Tools, Vol. 7. 9–18 (2002)

[Chong2004] H. Chong und S. Gortler. *A lixel for every pixel*. In Proceedings of the Eurographics Symposium on Rendering, Eurographics Association, 167–172 (2004)

[Engel2007] W. Engel. *Cascaded shadow maps*. In ShaderX^5, W. Engel, Ed. Charles River Media, 197–206 (2007)

[Fernando2001] R. Fernando, S. Fernandez, K. Bala, und D. Greenberg. *Adaptive shadow maps*. In Proceedings of ACM SIGGRAPH 2001, 387–390 (2001)

[Lloyd2006] B. Lloyd, D. Tuft, S. Yoon und D. Manocha. *Warping and partitioning for low error shadow maps*. In Proceedings of the Eurographics Symposium on Rendering 2006, Eurographics Association, 215–226 (2006)

[Lloyd2008] B. Lloyd, N. Govindaraju, C. Quammen, St. Molnar, D. Manocha. *Logarithmic Perspective Shadow Maps*. ACM Trans. Graph. 27, 4, Article 106 (2008)

[Stamm2002] M. Stamminger und G. Drettakis. *Perspective Shadow Maps*. In Proceedings of ACM SIGGRAPH 2002, 557–562. (2002)

[2] Bilder pro Sekunde. Das Testsystem war mit einer GeForce 7800GT sowie einem 2 GHz Prozessor ausgestattet [Lloyd2008, S.15]

[Williams1987] L. Williams. *Casting curved Shadows on curved surfaces*. In Computer Graphics (SIGGRAPH '78 Proceedings), Vol. 12, 270–274 (1978)

[Wimmer2004] M. Wimmer, D. Scherzer und W. Purgathofer. *Light space perspective shadow maps*. In Proceedings of the Eurographics Symposium on Rendering, Eurographics Association, 143–152 (2004)

[Zhang2006] F. Zhang, L. Xu, C. Tao, und H. Sun. *Generalized linear perspective shadow map reparameterization*. In VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications, ACM Press, New York, NY, USA, 339–342 (2006)