TU Clausthal

Institut für Informatik

Bachelorarbeit

Denoising-Algorithmen für Monte-Carlo-Rendering

vorgelegt von

Marvin Stümmel

Matrikelnummer 426136

Studiengang Informatik

6. Fachsemester

Betreuer: M.Sc. Johannes Jendersie

Erstgutachter: Prof. Dr. Thorsten Grosch

Zweitgutachter: Prof. Dr. Sven Hartmann

8. Oktober 2018

Schriftliche Erklärung

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass alle Stellen dieser Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht wurden und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsstelle vorgelegt wurde.

Des Weiteren erkläre ich, dass ich mit der öffentlichen Bereitstellung meiner Abschlussarbeit in der Instituts- und/oder Universitätsbibliothek nicht einverstanden bin.

Eine öffentliche Bereitstellung der PDF Datei auf Internetseiten der Universität ist gestattet.

Clauthal-Zellerfeld, den 8. Oktober 2018

Marvin Stümmel

Zusammenfassung

Fotorealistisches Rendering hat ein breites Anwendungsgebiet in der Computergrafik, jedoch erfordert es viel Zeit, um gute Ergebnisse zu erzielen. Um Zeit einzusparen, können Ergebnisse verwendet werden, welche verrauscht sind. Um die Qualität der Ergebnisse zu verbessern, können Algorithmen zur Entfernung des Rauschens auf diese angewandt werden. Dafür wurden hier verschiedene Filtermethoden getestet, die zu einer Verbesserung Führen können. Die Ergebnisse zeigen, dass ein cross-bilateraler Filter mit neu entwickelten Guides ~~es~~ dafür gut geeignet ist das Rauschen zu entfernen. Dadurch kann beim fotorealistischen Rendering mit Monte-Carlo-Verfahren Zeit eingespart werden, ohne dass die Bildqualität darunter leidet.

Inhaltsverzeichnis

1 Einleitung **1**

2 Grundlagen **3**

2.1 Monte-Carlo-Rendering 3

2.2 Filter 4

2.3 SSIM 5

3 Bilateraler Filter **9**

3.1 Vergleich RGB mit CIE-Luv 1976 9

3.2 Vergleich RGB mit L 10

4 Guided Filter **15**

4.1 Artefakte bei der Rauschreduktion 16

5 Cross-Bilateraler Filter **19**

5.1 Einfache und Kumulative Guides 19

5.2 Verwendete Guides 21

5.2.1 Tiefen-Guide 21

5.2.2 Positionen-Guide 22

5.2.3 Normalen-Guide 24

5.2.4 Texturen-Guide 26

5.2.5 Ambiente Verdeckung Guide 28

5.2.6 Licht-Guide 29

5.3 Varianzen zur Rauschverringerrung 30

5.4 Vorfiltern von Verrauschten Guides 32

5.5 Fazit zum Cross-Bilateralen Filtern 34

5.5.1 Fazit zum Positionen-Guide und Normalen-Guide 35

5.5.2 Fazit zum Texturen-Guide 35

5.5.3 Fazit zum Ambiente Verdeckung Guide 36

5.5.4 Fazit zum Licht-Guide 36

5.5.5 Fazit zu den Einfachen und Kumulativen Guides 37

5.5.6 Verwendete Einstellungen 38

6 SURE Filter **41**

6.1 SURE 41

6.2 SURE basiertes Filtern 41

6.2.1 Verschiedene Filtergrößen 42

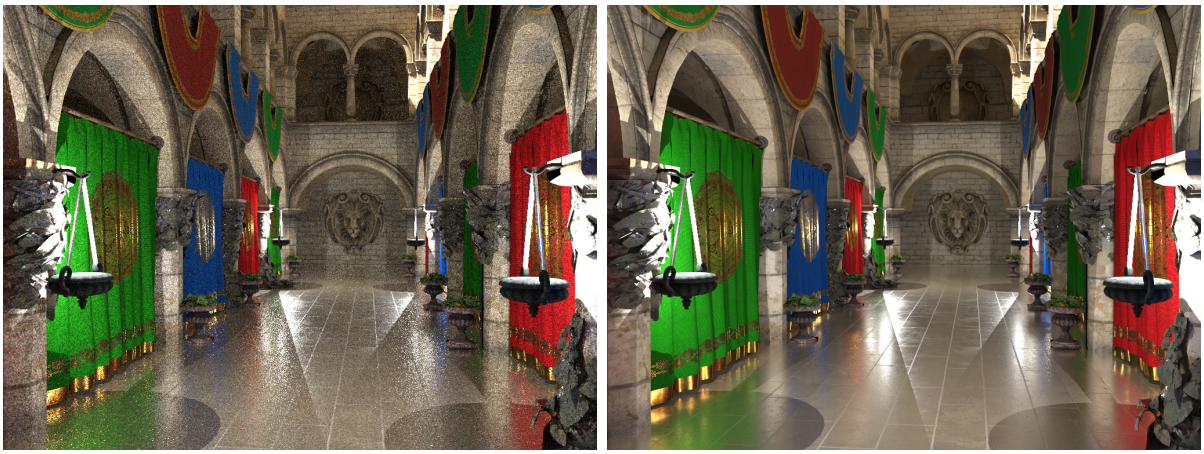
6.2.2 Cross-Bilateral mit und ohne Varianz 42

6.3 Fazit zum SURE basierten Filtern 44

7 Fazit **47**

7.1 Zukünftige Arbeiten 47

7.2 Ähnliche Arbeiten 48



(a) Verrauscht, benötigte Zeit: 158s **(b)** Referenzbild, benötigte Zeit: 10h

Abbildung 1: Sponza Szene. Das linke Bild wird soll so verbessert werden, dass es möglichst nahe an das rechte herankommt.

1 Einleitung

Beim fotorealistischen Rendern wird versucht, Szenen so zu rendern, dass das Ergebnis möglichst nahe an die Realität herankommt. Es wird beispielsweise bei der Animation von Filmen, wie von der Firma Pixar verwendet [Ren]. Aber auch bei der Produktentwicklung ist es hilfreich, um das Aussehen von zukünftigen Produkten zu visualisieren, ohne dass dafür ein physischer Prototyp hergestellt werden muss.

Für komplexe Szenen wird jedoch viel Zeit benötigt, damit ein realistisches Ergebnis erhalten werden kann. Mit wenig Zeitaufwand bekommt man hingegen ein Ergebnis, welches zwar akzeptabel aussieht, jedoch viel Rauschen enthält, siehe Abbildung 1. Wenn es geschafft werden kann, dieses Rauschen zu entfernen, dann kann Zeit beim Rendern eingespart werden. Es wird also ein Algorithmus gesucht, welcher ein Bild mit relativ wenig Zeitaufwand entrauschen kann. Eine weitere Anforderung an den Algorithmus ist, dass er möglichst stabil für eine breite Auswahl an Szenen sein soll, ohne dass dafür die Parameter angepasst werden müssen.

Dafür werden in dieser Arbeit verschiedene existierende Algorithmen getestet, darunter der Guided Filter, der cross-bilaterale Filter und der SURE Filter und teilweise angepasst und verbessert. Für den cross-bilateralen Filter werden neue Guides verwendet und verschiedene Kombinationen von diesen evaluiert, außerdem werden die Varianzen von Guides verwendet, um diese zu verbessern.

2 Grundlagen

In diesem Kapitel werden Grundlagen zum besseren Verständnis der Arbeit behandelt. Dafür wird das Monte-Carlo-Rendering und Path Tracing, als Grundlage zum Rendern, der Gauß-Filter, zur Erklärung von Filtern, und die „structural similarity“ (SSIM), zum besseren Verständnis für das verwendete Fehlermaß, erläutert.

2.1 Monte-Carlo-Rendering

Monte-Carlo-Integration wird oft zum Rendern von fotorealistischen Bildern benutzt. Dabei wird versucht, die Rendergleichung [Kaj86] mithilfe von Zufallszahlen zu lösen. Das Integral aus der Rendergleichung wird dabei als Erwartungswert einer Zufallsvariable dargestellt und dieser durch Stichproben abgeschätzt. Je mehr Stichproben dabei genommen werden, desto besser wird das Ergebnis. Standardgemäß wird in einer Iteration für jeden Pixel im zu rendernden Bild eine Stichprobe genommen, so hat ein Bild mit 50 Iterationen genau 50 Stichproben pro Pixel. Es gibt aber auch Ansätze, die Pixel zu bewerten und die Stichproben auf die schlecht bewerteten Pixel zu konzentrieren [LWC12]. Das resultierende Bild ist der Mittelwert von allen Stichproben pro Pixel. Das Verfahren konvergiert nur langsam und erfordert je nach Szene mehrere tausend Iterationen, um zu einem guten Ergebnis zu kommen. Bilder mit nur wenigen Iterationen hingegen sind stark verrauscht.

Bidirektionales Path Tracing

Beim Bidirektionalen Path Tracing nach E. Veach [Vea98] wird bei jeder Stichprobe jeweils ein Pfad von der Kamera durch den betrachteten Pixel und ein zufälliger Pfad von einer Lichtquelle ausgehend berechnet. Wenn ein Objekt getroffen wird, wird der Pfad in eine zufällige Richtung, die Streuung hängt dabei von Material des getroffenen Objektes ab, reflektiert. Der Pfad wird bis zu einer vorher festgelegten Anzahl an Reflexionen weiterverfolgt und die Punkte, an denen Reflexion stattgefunden hat, der beiden Pfaden miteinander verbunden. Die so verbundenen Pfade werden anhand der von ihnen getroffenen Materialien gewichtet und daraus der Farbwert des betrachteten Pixels berechnet.

2.2 Filter

Bilder können mit Filteroperationen bearbeitet werden, dabei werden die Pixel des Bildes anhand anderer Pixel im Bild verändert. Das hat mehrere Anwendungsgebiete, wie z. B. Kantenerkennung oder Rauschunterdrückung. Um das Rauschen aus einem gerenderten Bild mit wenigen Iterationen, in dieser Arbeit meist 50, zu entfernen, können auf das Bild verschiedene Filter angewendet werden. Die Filter betrachten die jeweiligen Pixel p, ihre umgebenden Pixel q, sowie gegebenenfalls andere Zusatzinformationen, um daraus ein neues Bild zu rekonstruieren.
Ein einfaches Beispiel für einen solchen Filter ist der Gauß-Filter. Beim Gauß-Filter werden alle Pixel in einem bestimmten Radius um den Ausgangspixel betrachtet, anhand ihres Abstandes zum Ausgangspixel unterschiedlich gewichtet und dann daraus der gefilterte Pixel berechnet. Die Funktion $\text{dist}(p, q)$ berechnet den euklidischen Abstand zwischen den zwei Pixeln p und q, wobei x und y die Koordinaten der Pixel sind.

$$\text{dist}(p, q) = \sqrt{\Delta x^2 + \Delta y^2}$$

Das Gewicht wird wie in (1) berechnet, dabei wird mit den Abstand der Wert an einer Stelle einer Gaußschen Normalverteilung bestimmt, wobei σ_s der Parameter der Standartabweichung dieser ist und der Filterradius r abhängig von ihm gewählt wird.

$$w_{\text{Gauss}}(p, q) = \exp\left(-\frac{\text{dist}(p, q)^2}{2\sigma_s^2}\right) \tag{1}$$

Das führt dazu, das Pixel die eine größere Distanz haben, ein kleineres Gewicht bekommen. Eine Einschränkung des Filterradius ist notwendig, da zwar die Gaußsche Normalverteilung eine unendliche Ausdehnung hat, es aber viel Rechenleistung erfordert, alle Pixel mit allen anderen zu filtern. In dieser Arbeit wird für den Gauß-Filter die Belegung $r = 6 \cdot \sigma_s$ relativ groß, damit keine sichtbaren Kanten entstehen, wenn sehr helle Pixel gefiltert werden. Der neue Pixel ergibt sich dadurch, dass die Pixel q aus der Umgebung U im Filterradius von p mit ihren Gewichten multipliziert, zusammenaddiert und anschließend durch die Summe der Gewichte geteilt werden.

$$\text{Gauss}(p) = \frac{1}{\sum_{q \in U_p} w_{\text{Gauss}}(p, q)} \sum_{q \in U} (w_{\text{Gauss}}(p, q) \cdot q) \tag{2}$$

Wird (2) mit großem σ_s auf alle Pixel angewandt, werden viele Pixel in die Berechnung des neuen Pixels mit einbezogen und haben ein, im Vergleich zu den jeweiligen p hohen Einfluss auf das Ergebnis. Dadurch wird zwar das Rauschen stark verringert, jedoch werden auch die Kanten sehr verschwommen und das Bild sehr weichgezeichnet. Mit kleinen σ_s kommt es nur zu wenig Rauschverringerung, da nur wenige umgebende Pixel betrachtet werden und diese einen niedrigen Einfluss auf das Resultat haben. Die negativen Effekte mit großem σ_s kommen daher, dass der Filter nur anhand der Entfernung im Bild entscheidet, welche Pixel wie viel Einfluss auf das Ergebnis haben und somit die Kanten im Bild nur schlecht erhalten bleiben. Für Beispielbilder mit verschiedenen σ_s siehe Abbildung 2. Die in Abschnitt 3 bis Abschnitt 6 vorgestellten Filter lösen das Problem durch eine explizite Detektion von Kanten.

2.3 SSIM

SSIM steht für structural similarity oder zu Deutsch strukturelle Ähnlichkeit und wurde von Zhou Wang et al. in [WBSS04] vorgestellt. Die strukturelle Ähnlichkeit ist ein Fehlermaß, das verwendet wird, um die wahrgenommene Qualität von digitalen Bildern und Videos zu schätzen. Dafür wird ein störungsfreies Bezugsbild benötigt, in dieser Arbeit ein Bild mit vielen Iterationen, je nach Szene 3000 oder 16100, ist. Der Vorteil gegenüber anderen Fehlermaßen wie den MSE ist, dass die SSIM mehr der Wahrnehmung eines Menschen entspricht. So wird nicht nur der absolute Fehler betrachtet, sondern auch Phänomene der Wahrnehmungspsychologie mit einbezogen. So fallen z. B. Fehler in sehr hellen Bereichen des Bildes tendenziell weniger auf und haben dementsprechend auch weniger Einfluss auf den SSIM-Wert.
Die Berechnung erfolgt über mehrere gleich große Bildausschnitte α und β von den Originalbildern:

$$\mathrm{SSIM}(\alpha, \beta) = \frac{(2\mu_\alpha\mu_\beta + c_1)(2\sigma_{\alpha\beta} + c_2)}{(\mu_\alpha^2 + \mu_\beta^2 + c_1)(\sigma_\alpha^2 + \sigma_\beta^2 + c_2)} \tag{3}$$

Wobei μ_α und μ_β die Mittelwerte und σ_α^2 und σ_β^2 die Varianzen von α und β sind. $\sigma_{\alpha\beta}$ ist die Kovarianz von α und β und c_1 und c_2 sind Variablen, um die Division bei kleinen Nennern zu stabilisieren. Hier verwendet wird $c_1 = 0.0004$ und $c_2 = 0.000324$. Die Mittelwerte und Varianzen beziehen sich auf die RGB Werte der Pixel in 8x8 großen Bildausschnitten. Der errechnete Wert liegt im Bereich von -1 bis 1, wobei 1 nur bei identischen Bildausschnitten

(a) Input 50 Iterationen

(b) Referenzbild

(c) $\sigma_s = 1$

(d) $\sigma_s = 2$

(e) $\sigma_s = 4$

(f) $\sigma_s = 8$

Abbildung 2: Gauss gefilterte Bad Szene mit verschiedenen σ_s, wobei $r = 6 \cdot \sigma_s$.

erreicht werden kann. Je höher der Wert ist, desto ähnlicher sind sich die Bildausschnitte. Der SSIM(α, β)-Wert wird für mehrere Bildausschnitte berechnet und der Durchschnitt aller Werte als Endergebnis gewählt.

3 Bilateraler Filter

Eine Verbesserung des Gauß-Filters, im Hinblick auf die Kantenerhaltung, ist der bilaterale Filter. Dieser verwendet nicht nur den Abstand der Pixel zueinander, sondern zusätzlich auch noch die Farben der Pixel, um Bildkanten zu erhalten. D. h. es fließt auch die Distanz der Pixel im Farbraum zueinander in die Gewichtung der einzelnen Pixel mit ein. Dadurch fließen Pixel, die unterschiedliche Farben haben nicht mehr so stark in die Gewichtung mit ein und das Verschwimmen wird vermindert. Die Funktion $\mathrm{cdist}(p, q)$ berechnet diese Distanz im Farbraum, wobei für die Abstände verschiedene Berechnungsmöglichkeiten existieren, denn die Distanz von Farben ist nicht eindeutig definiert. Denkbar sind z. B. der euklidische Abstand im linearen RGB Raum, der euklidische Abstand im Luv Raum oder die Differenz der Luminanz. Auf die Auswahl der $\mathrm{cdist}(p, q)$ Funktion wird in Unterabschnitt 3.1 und Unterabschnitt 3.2 noch eingegangen.

$$\begin{aligned}
\mathrm{cdist}_{\mathrm{RGB}}(p, q) &= \sqrt{\Delta\mathrm{rot}^2 + \Delta\mathrm{grün}^2 + \Delta\mathrm{blau}^2} \\
\mathrm{cdist}_{\mathrm{Luv}}(p, q) &= \sqrt{\Delta L^2 + \Delta u^2 + \Delta v^2} \\
\mathrm{cdist}_{\mathrm{L}}(p, q) &= \sqrt{\Delta L^2}
\end{aligned}$$

Hierbei ist Δ der Differenzoperator des jeweiligen Parameters von p und q. Das Gewicht zwischen p und q berechnet sich dann folgendermaßen:

$$w_{\mathrm{Bilateral}}(p, q) = \exp\left(-\frac{\mathrm{dist}(p, q)^2}{2\sigma_s^2}\right) \cdot \exp\left(-\frac{\mathrm{cdist}(p, q)^2}{2\sigma_r^2}\right) \tag{4}$$

Das Filterergebnis (5) wird analog zu der für (2) beschriebenen Vorgehensweise mit einer angepassten Gewichtungsfunktion berechnet.

$$\mathrm{Bilateral}(p) = \frac{1}{\sum_{q \in U} w_{\mathrm{Bilateral}}(p, q)} \sum_{q \in U} \left(w_{\mathrm{Bilateral}}(p, q) \cdot q\right) \tag{5}$$

3.1 Vergleich RGB mit CIE-Luv 1976

Der CIE-Luv 1976 Farbraum wurde so definiert, dass die Abstände in diesen Farbraum ungefähr den empfundenen Abständen der Farben entsprechen. Ein möglicher Nutzen den Luv Farbraum zu verwenden besteht darin, dass wenn die gemessenen Farbabstände den sichtbaren Farbabständen entsprechen, das Ergebnis visuell ansprechender sein könnte.

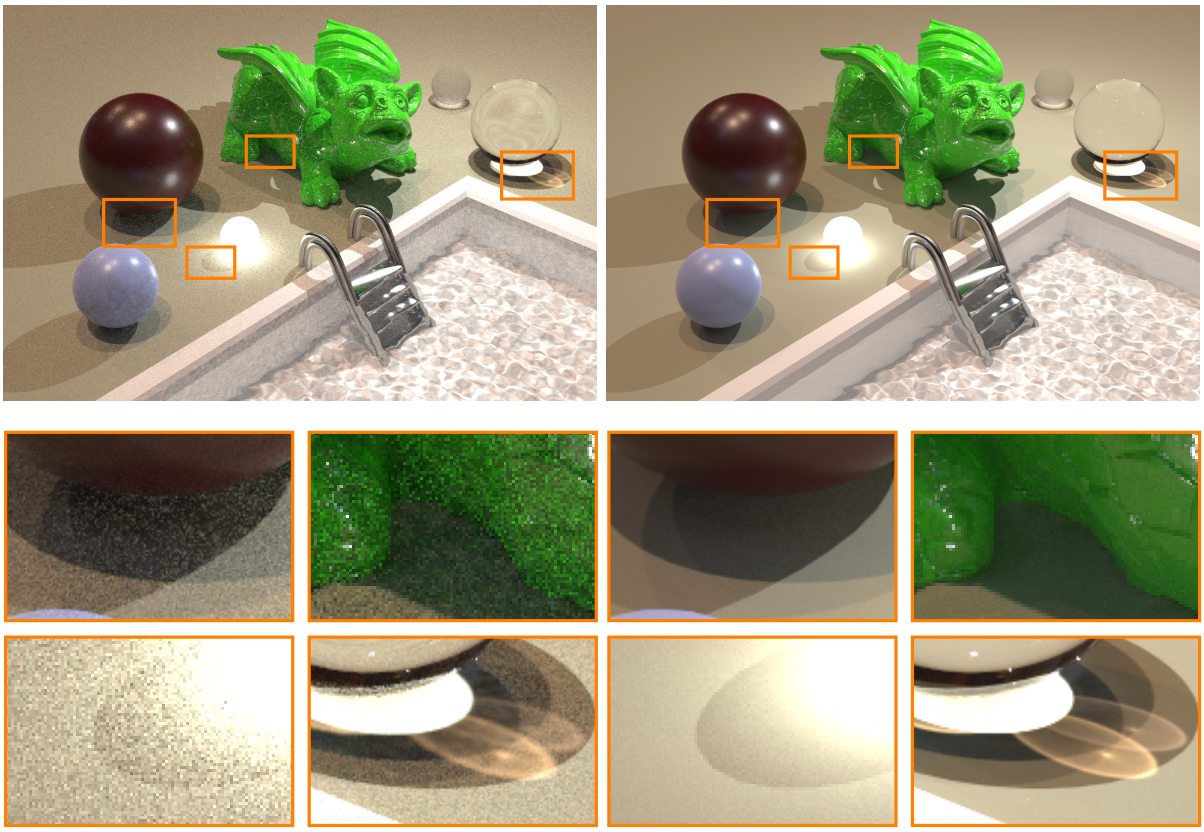
Tabelle 1: Beste SSIM von $\text{cdist}_{\text{RGB}}$, $\text{cdist}_{\text{Luv}}$ und cdist_{L}.

Szene	Ungefiltert	Gauß	$\text{cdist}_{\text{RGB}}$	$\text{cdist}_{\text{Luv}}$	cdist_{L}
Drachenpool	0.906791	0.940007	0.967166	0.966911	0.967735
Bad	0.664969	0.643708	0.857605	0.853476	0.818366
Sphären	0.927265	0.950812	0.984500	0.983690	0.984862

In Abbildung 3 sind dafür drei Szenen mit $\text{cdist}_{\text{RGB}}$ und $\text{cdist}_{\text{Luv}}$ gefiltert worden. Die σ_r wurden dabei so gewählt, dass ein möglichst hoher SSIM-Wert erzeugt wird. Da optisch nur wenige Unterschiede erkannt werden können, werden die SSIM-Werte aus Tabelle 1 als Grundlage der Entscheidung genommen, $\text{cdist}_{\text{RGB}}$ anstelle von $\text{cdist}_{\text{Luv}}$ zu verwenden.

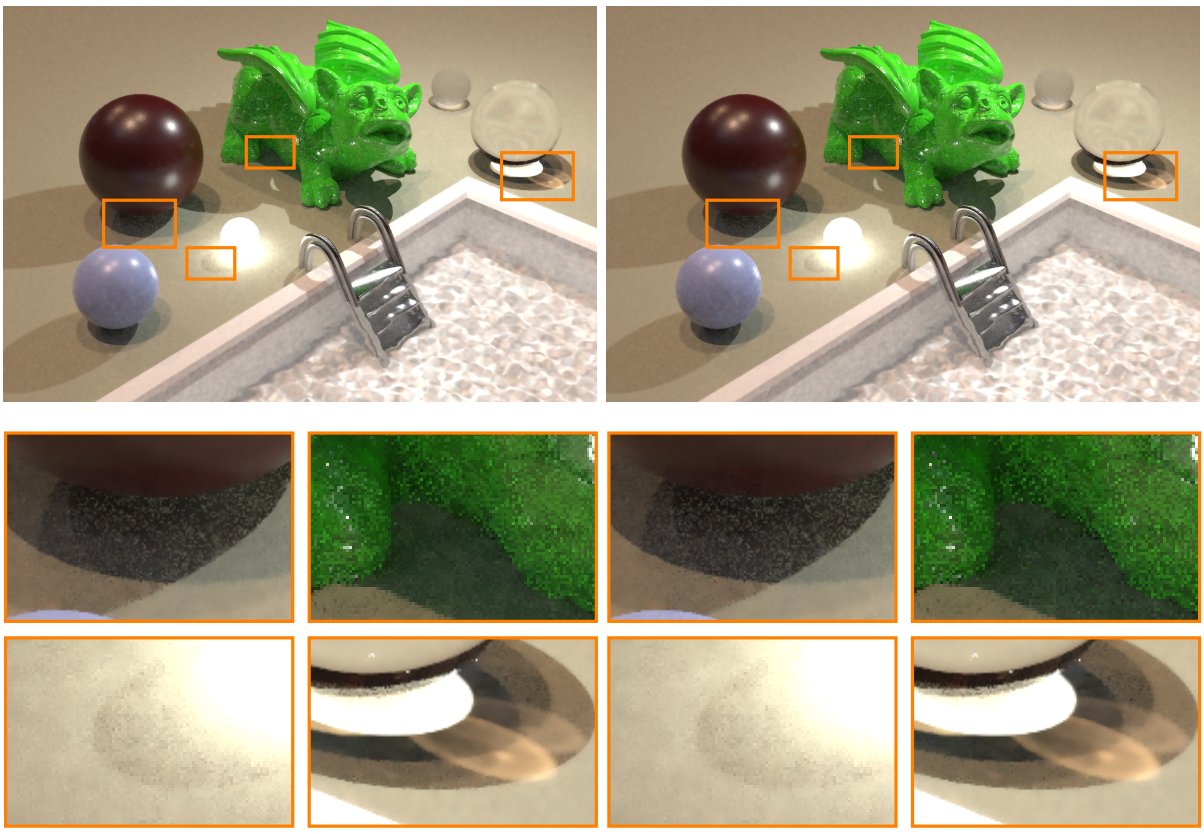
3.2 Vergleich RGB mit L

Eine andere Möglichkeit ist nur die Luminanz zu verwenden. In Abbildung 4 sind wieder drei Szenen mit $\text{cdist}_{\text{RGB}}$ und cdist_{L} gefiltertt worden. Die SSIM-Werte aus Tabelle 1 bewerten cdist_{L} teilweise besser als $\text{cdist}_{\text{RGB}}$, bei der Bad Szene jedoch deutlich schlechter und die Ergebnisse sehen optisch eher schlechter aus. Aus diesen Gründen wird im weiteren Verlauf $\text{cdist}_{\text{RGB}}$ anstelle von cdist_{L} verwendet.



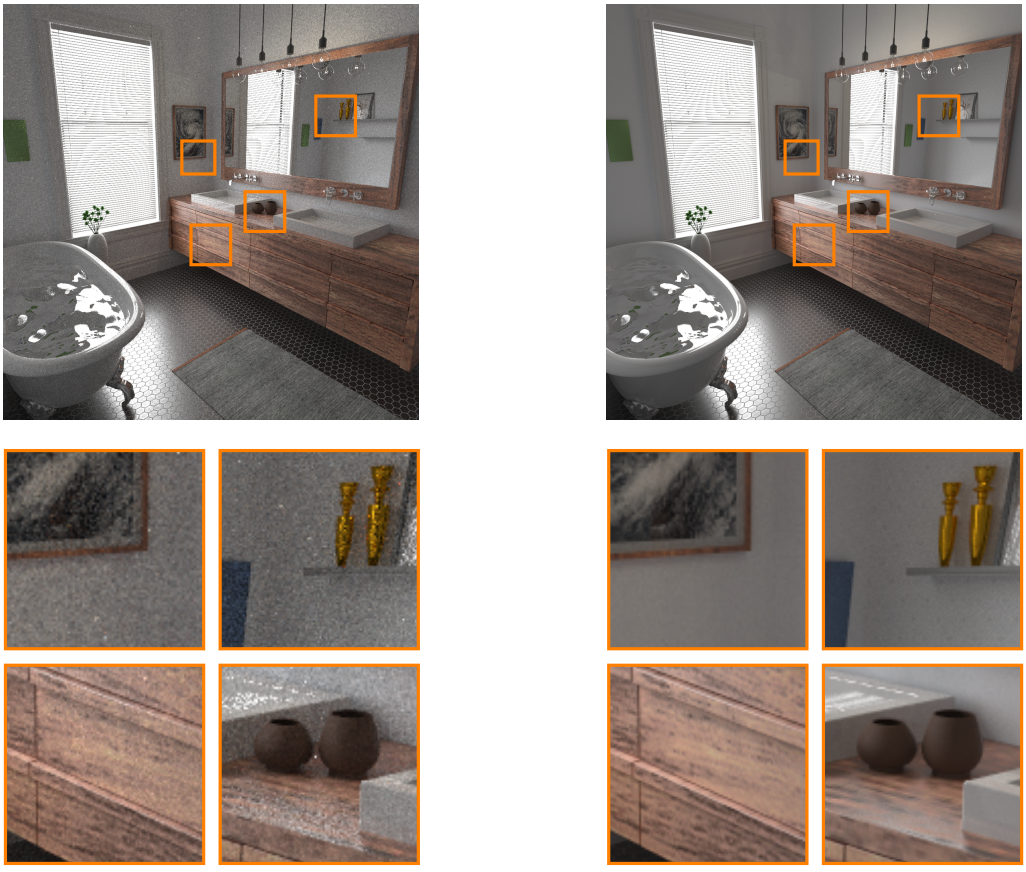
(a) Input 50 Iterationen

(b) Referenzbild



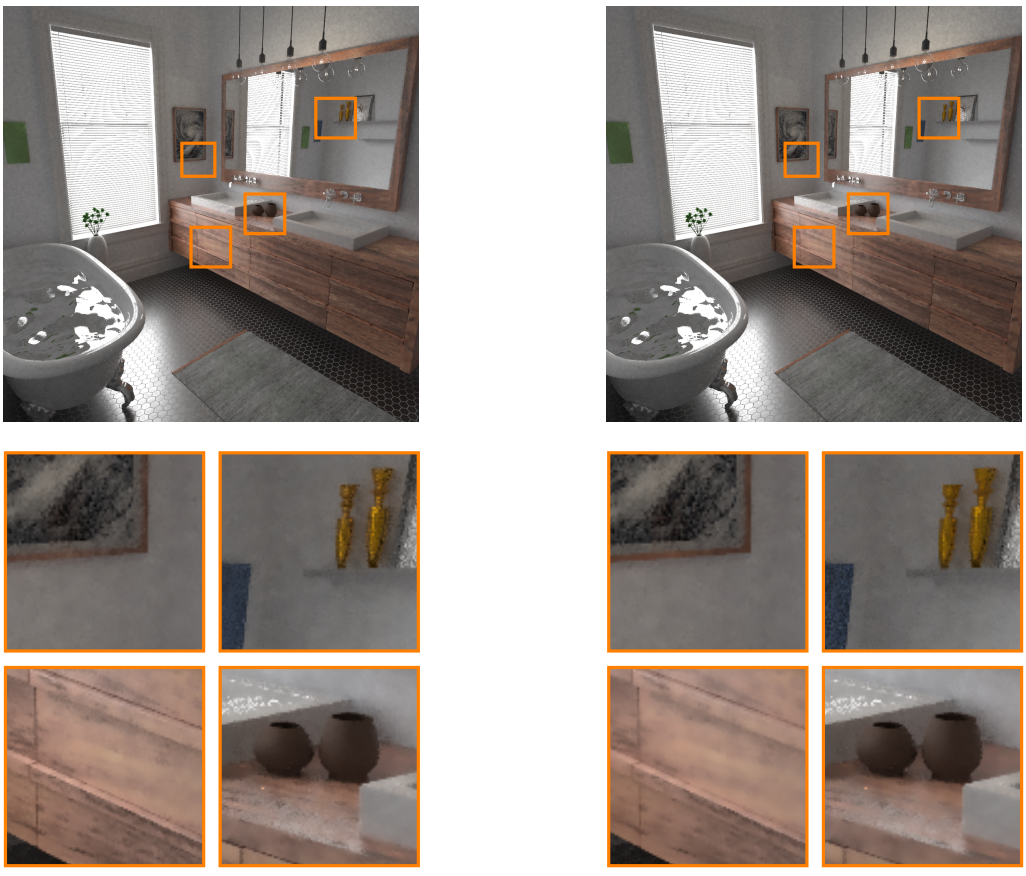
(c) RGB $\sigma_r = 0.7$

(d) Luv $\sigma_r = 0.4$



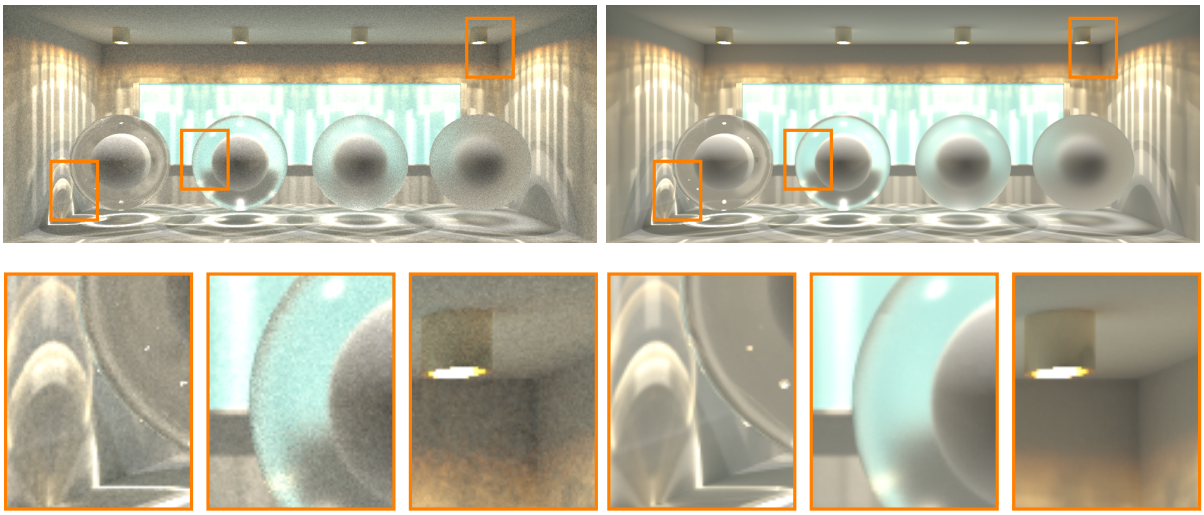
(e) Input 50 Iterationen

(f) Referenzbild

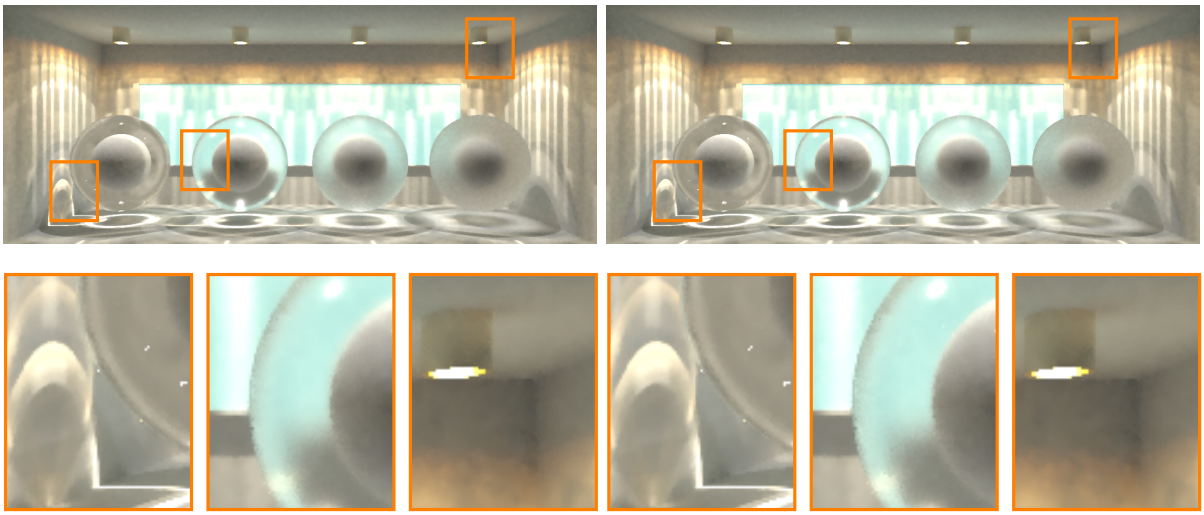


(g) RGB $\sigma_r = 0.7$

(h) Luv $\sigma_r = 0.5$

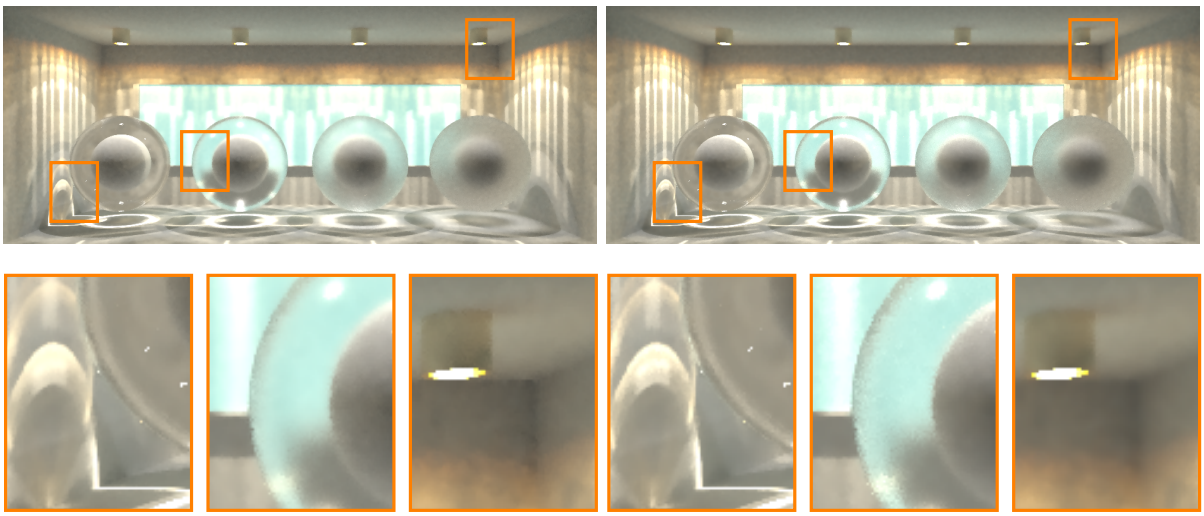


(i) Input 50 Iterationen (j) Referenzbild

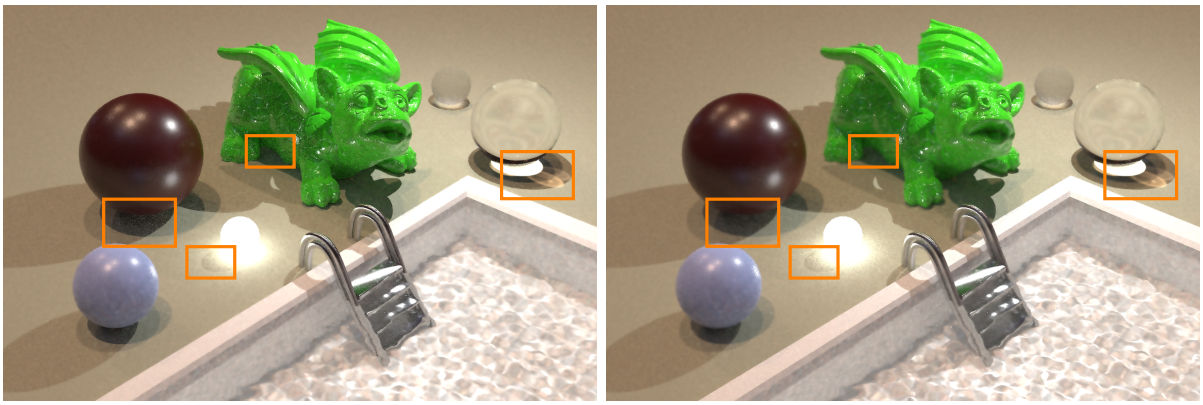


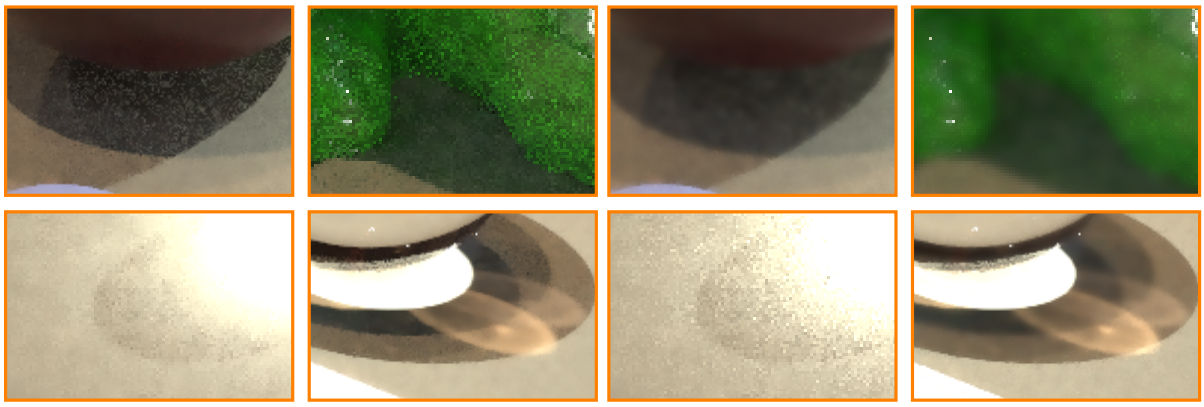
(k) RGB $\sigma_r = 0.5$ (l) Luv $\sigma_r = 0.3$

Abbildung 3: Vergleich von $\text{cdist}_{\text{RGB}}$ mit $\text{cdist}_{\text{Luv}}$, wobei $\sigma_s = 2$ und der Filterradius $= 5$.



(a) RGB $\sigma_r = 0.5$ (b) L $\sigma_r = 0.1$





(c) RGB $\sigma_r = 0.7$ (d) L $\sigma_r = 0.15$

(e) RGB $\sigma_r = 0.7$ (f) L $\sigma_r = 0.065$

Abbildung 4: Vergleich von $\text{cdist}_{\text{RGB}}$ mit cdist_{L}, wobei $\sigma_s = 2$ und der Filterradius $= 5$.

4 Guided Filter

Guides enthalten Informationen, welche beim Filtern des Bildes mit einbezogen werden können. Das Ausgangsbild enthält z. B die Farbinformationen zu jedem Pixel. Bei dem von Kaming He et al. vorgestellten Guided Filter [HST13] wird angenommen, dass es zwischen dem Guide g und dem resultierenden Bild R einen linearen Zusammenhang gibt. Genauer wird angenommen, dass R eine lineare Transformation von g in einem Fenster U_k mit Filterradius r ist, welches um den Pixel k liegt und i ist der Index für die Pixel in U_k.

$$R_i = a_k g_i + b_k \tag{6}$$

Dabei sind a_k und b_k linear Koeffizienten welche als konstant in U_k angenommen werden. Der lineare Zusammenhang versichert, dass R nur dann eine Kante hat, wenn g eine Kante hat, da $\Delta R = a\Delta g$. Um a_k und b_k zu berechnen, werden Bedingungen von dem zu filternden Bild I benötigt. Es wird R so modelliert, dass es I ohne die unerwünschten Komponenten (Rauschen) n ist.

$$R_i = I_i - n_i \tag{7}$$

Es wird eine Lösung gesucht, welche die Differenz zwischen R und I minimiert, während das lineare Modell (6) eingehalten wird.

$$E(a_k, b_k) = \sum_{i \in U_k} \left((a_k g_i + b_k - I_i)^2 + \epsilon {a_k}^2 \right) \tag{8}$$

Dabei ist ϵ ein Regularisierungsparameter, welche große a_k bestraft. Die Lösung von (8) ist gegeben von (9) und (10).

$$a_k = \frac{\frac{1}{|U|} \sum_{i \in U_k} (g_i I_i) - \mu_k \bar{I}_k}{\sigma_k^2 + \epsilon} \tag{9}$$

$$b_k = \bar{I}_k - a_k \mu_k \tag{10}$$

μ_k ist der Mittelwert und σ_k^2 die Varianz von g in U_k, $|U|$ ist die Anzahl der Pixel in U_k und $\bar{I}_k = \frac{1}{|U|} \sum_{i \in U_k} I_i$ ist der Mittelwert von I in U_k. Durch Einsetzen in (6) kann jetzt R_i berechnet werden, jedoch ist ein Pixel an Position i in mehreren überlappenden U_k enthalten und R_i wird in diesen unterschiedlich berechnet. Aus diesem Grund wird der

Durchschnitt von allen möglichen R_i berechnet.

$$R_i = \frac{1}{|U|} \sum_{k:i \in U_k} (a_k g_i + b_k) \tag{11}$$

Da $\sum_{k:i \in U_k} a_k = \sum_{k \in U_i} a_k$ aufgrund der Symmetrie des Fensters ist, kann (11) in (12) umgewandelt werden.

$$R_i = \bar{a}_i g_i + \bar{b}_i \tag{12}$$

Dabei sind $\bar{a}_i = \frac{1}{|U|} \sum_{k \in U_i} a_k$ und $\bar{b}_i = \frac{1}{|U|} \sum_{k \in U_i} b_k$ die durchschnittlichen Koeffizienten von allen Fenstern in denen der Pixel an Position i enthalten ist.

4.1 Artefakte bei der Rauschreduktion

Bei dem Guided Filter können verschiedene Guides benutzt werden, aber zum Entrauschen eignet sich besonders das Ausgangsbild. In unseren Fall ist also der Guide gleich dem Ausgangsbild und somit kann (9) zu (13) reduziert werden.

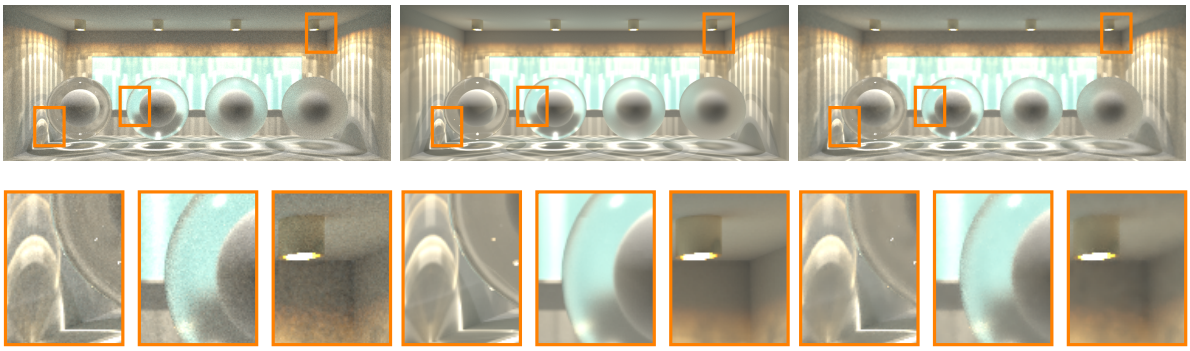
$$a_k = \frac{\sigma_k^2}{\sigma_k^2 + \epsilon} \tag{13}$$

D. h. der Guided Filter erkennt, wenn das Fenster über eine Kante geht und somit die Varianz im Vergleich zum ϵ hoch ist. An diesen Stellen wird $a_k \approx 1$, $b_k \approx 0$ und somit $R_i \approx g_i = I_i$, also wird der ursprüngliche Pixel beibehalten. Wenn die Varianz im Vergleich zum ϵ kleiner wird, wird auch a_k kleiner und somit der Anteil von dem Mittelwert $\bar{I}_k$ aus dem Fenster im resultierenden Pixel größer.
Da aber dadurch die Pixel in der Nähe von Kanten nicht gefiltert werden, wird auch das Rauschen dort nicht vermindert, dies ist besonders auffällig, wenn größere Filterradien verwendet werden. In den Ergebnisbildern Abbildung 5 ist dieser Effekt noch leicht erkennbar bei der Bad Szene, in der Vergrößerung oben links nahe den Bilderrahmen, und in der Sphären Szene nahe der Lampe, in der rechten Vergrößerung (besser erkennbar in der digitalen Version der Bachelorarbeit). Eine Lösung dafür stellt [ZCW14] mit *Nonlocal Image Guided Averaging* (NLGA) vor.
In Tabelle 2 sind die bestmöglichen SSIM-Werte der Szenen, welche bei einem sehr kleinen Filterradius = 2 erreicht worden sind, dargestellt. Da der Guided Filter Rauschen in der Nähe von Kanten bei größeren Filterradien nicht zuverlässig entfernt und die SSIM-Werte

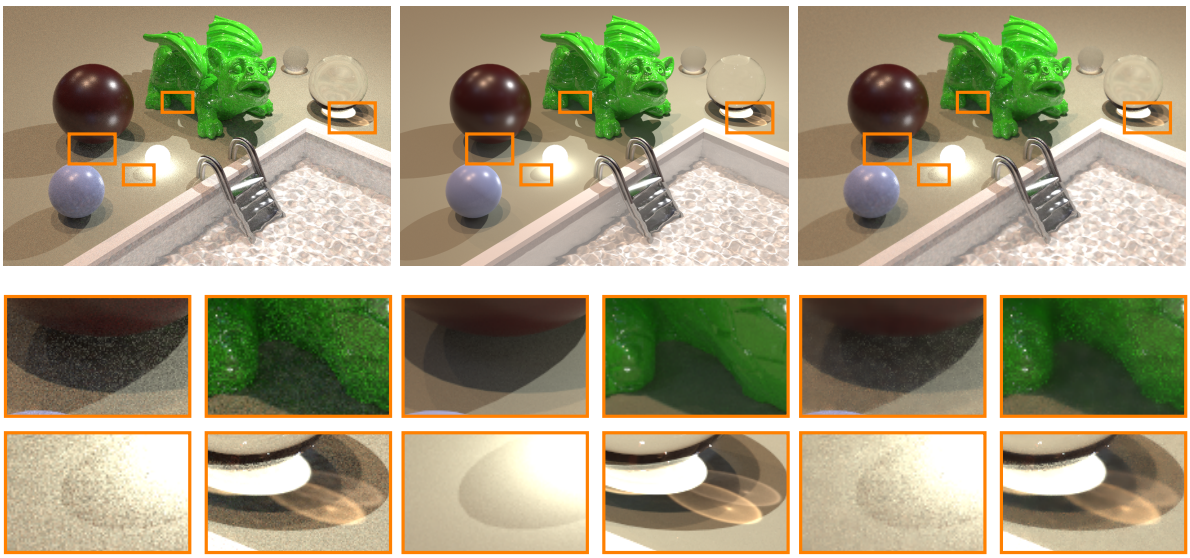
Tabelle 2: Beste SSIM von Guided Filter.

Szene	Ungefiltert	Gauß	Bilateral	Guided
Drachenpool	0.906791	0.940007	0.967166	0.964417
Bad	0.664969	0.643708	0.857605	0.84073
Sphären	0.927265	0.950812	0.984500	0.983992

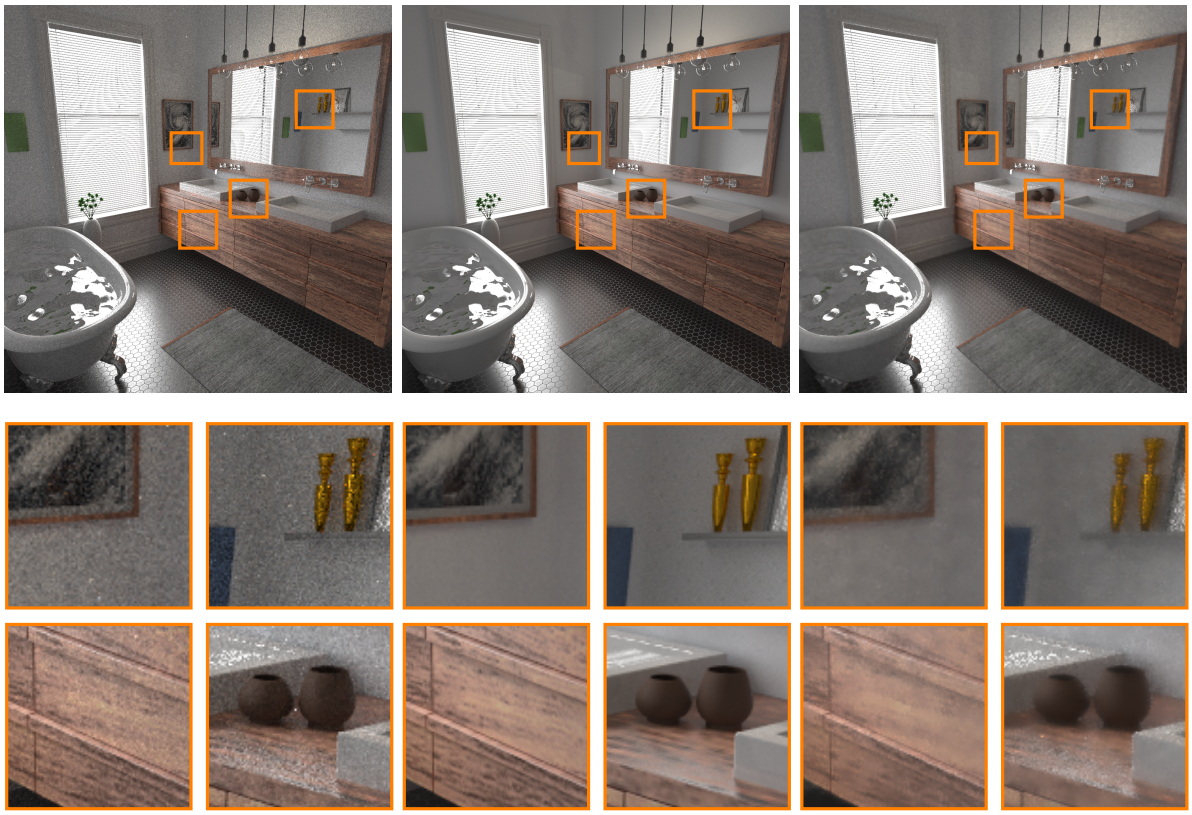
schlechter als die des bilateralen Filters sind, wird der Guided Filter aus der weiteren Untersuchung ausgeschlossen.



(a) Input 50 Iterationen **(b)** Referenzbild **(c)** Guided



(d) Input 50 Iterationen **(e)** Referenzbild **(f)** Guided



(g) Input 50 Iterationen **(h)** Referenzbild **(i)** Guided

Abbildung 5: Guided Filter Ergebnisse Filterradius = 2.

5 Cross-Bilateraler Filter

Bei den bisher verwendeten Filtermethoden, wurden jeweils höchstens die Ausgangsbilder als Guides verwendet, um Kanten beim Entfernen des Rauschens zu erhalten. Es ist aber auch möglich, mehrere Guides zu verwenden, wodurch bessere Ergebnisse bei der Kantenerhaltung erzielt werden können. Die Guides enthalten dabei unterschiedliche Informationen zu dem Bild, wie z. B. die Normalen oder Materialwerte. Wenn in einem der Guides eine Kante enthalten ist, dann sollte diese auch im gefilterten Bild erhalten bleiben. Dies ist von Nachteil, wenn die Guides selber Rauschen enthalten, deshalb sollten sie möglichst frei von Rauschen sein. Ein gutes Beispiel dafür, dass sich das aber nicht immer vermeiden lässt, ist das Ausgangsbild, welches selber verrauscht ist. Wenn es als Guide verwendet wird, werden Kanten an Stellen erkannt, an denen das Rauschen zu stark ist. Es als Guide wegzulassen, ist aber auch nicht möglich, da in dem Ausgangsbild wichtige Informationen, vor allem von der Beleuchtungssituation, enthalten sind.
Der cross-bilateraler Filter erweitert den bilateralen Filter, in welchem nur das Ausgangsbild als Guide verwendet wird, indem er mehrere Guides einsetzt. Dafür muss die neue Gewichtsfunktion (14) verwendet werden. g ist dabei ein Guide aus der Menge aller verwendeten Guides G, σ_g^2 ist die Varianz und $\mathrm{dist_g}$ die verwendete Distanzfunktion des Guides.

$$w_{\mathrm{Cross}}(p,q) = \exp\left(-\frac{\mathrm{dist}(p,q)^2}{2\sigma_s^2}\right) \cdot \prod_{g \in G} \exp\left(-\frac{\mathrm{dist_g}(p,q)^2}{2\sigma_g^2}\right) \tag{14}$$

Die Distanzfunktionen sind von der Art der Informationen in den jeweiligen Guides abhängig. So ist es z. B. nicht sinnvoll die Distanzfunktion, die für Farben genutzt wird, auch auf Guides, die Normalen enthalten, anzuwenden, näheres zu den verwendeten Guides in Unterabschnitt 5.2. Die Gleichung (15) beschreibt das Filterergebnis für den cross-bilateralen Filter.

$$\mathrm{Cross}(p) = \frac{1}{\sum_{q \in U} w_{\mathrm{Cross}}(p,q)} \sum_{q \in U} (w_{\mathrm{Cross}}(p,q) \cdot q) \tag{15}$$

5.1 Einfache und Kumulative Guides

Für den cross-bilateralen Filter werden in dieser Arbeit zwei verschiedene Guide-Typen verwendet, welche im Folgenden einfach und kumulativ genannt werden. Beide Typen können direkt beim Rendern erhalten werden, was keinen zusätzlichen Zeitaufwand benötigt. Bei den kumulativen Guides werden die Ergebnisse der Pfade bis zur maximalen Pfadlän-

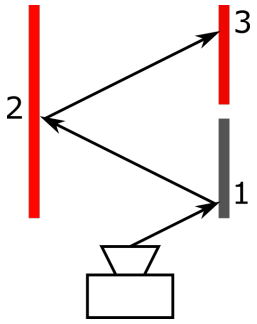


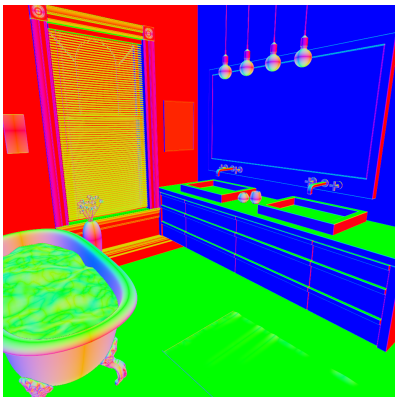
Abbildung 6: Grafik zu den Pfadgewichtungen.

ge berechnet und entsprechend ihrer Sichtbarkeit gewichtet. In (16) ist die Formel für die Gewichtung beschrieben und wird hier erstmalig vorgestellt und ausgewertet. Sie wurde experimentell aus einer Auswahl von verschiedenen möglichen Formeln für die Gewichtungen bestimmt und angepasst. Es bekommt das Ergebnis jedes Pfads P ein Gewicht, welches von den Pfadwahrscheinlichkeiten der vorherigen Pfade abhängt. Die pdf ist eine Wahrscheinlichkeitsdichtefunktion, welche von dem Material des getroffenen Objekts abhängt. Genaueres zur pdf und Importance Sampling kann in [Vea98] nachgeschlagen werden.

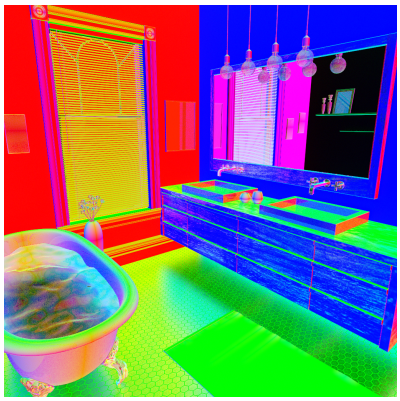
$$w(P) = \prod_{i=1}^{len(P)-1} \left(1 - \exp\left(-\frac{\mathrm{pdf}_i^2}{5}\right)\right) \tag{16}$$

Als kleines Beispiel für die Gewichtung dient Abbildung 6, mit einer maximalen Pfadlänge von drei. Das graue Material ist ein Spiegel und das rote ein diffuses Material. Am Punkt 1 ist $w(P_1) = 1$, weil das leere Produkt immer eins ist. Am Punkt 2 ist $w(P_2) = 1$, da der Wert der pdf bei Spiegeln unendlich ist. Und am Punkt 3, unter Annahme einer uniformen pdf, wodurch der Wert für alle Richtungen bei einem diffusen Material $1/2\pi$ beträgt, ist das Gewicht $w(P_3) = 1 \cdot (1 - \exp(-1/20\pi^2)) \approx 0.00505$. Für jeden Punkt wird ein Wert für den Guide ausgerechnet, z. B. die Normale, dieser Wert wird dann mit den Gewichten multipliziert und anschließend werden die Werte für alle Punkte aufaddiert. Dies wird in jeder Iteration wiederholt und dabei werden, je nach Wahrscheinlichkeit der einzelnen Pfade, unterschiedliche Punkte getroffen.

Bei den einfachen Guides werden nur die Ergebnisse mit einer Pfadlänge von eins berechnet und nicht wie bei den kumulativen die Ergebnisse bis zur maximalen Pfadlänge benutzt. Das führt dazu, dass bei den einfachen Guides wenig Rauschen vorhanden ist, aber auch keine Informationen zu Reflexionen oder transparenten Objekten erkannt werden können. Bei den kumulativen Guides sind diese zu erkennen, jedoch enthalten die



(a) Einfacher Normalen-Guide



(b) Kummulativer Normalen-Guide

Abbildung 7: Normalen-Guides für die Bad Szene.

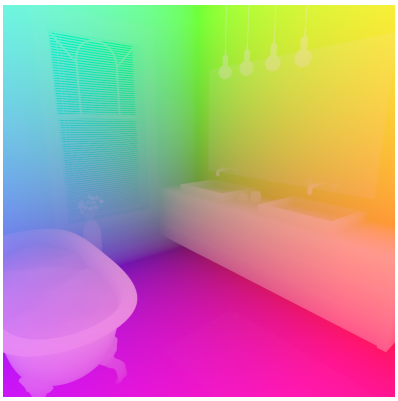
Guides selber mehr Rauschen. Erkennbar ist das beispielsweise in Abbildung 7, welches den einfachen und kumulativen Normalen-Guide für die Bad Szene zeigt.

5.2 Verwendete Guides

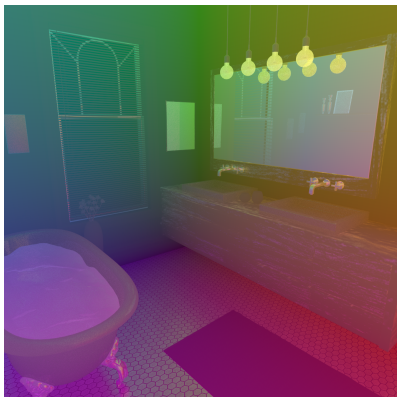
In diesem Abschnitt werden die verwendeten Guides erklärt und die Wirkungen der einzelnen Guides genauer betrachtet. Dafür werden die jeweiligen Szenen cross-bilateraler gefiltert, wobei das Bild selbst und der zu prüfenden Guide als Guides verwendet werden. Das gefilterte Ergebnis wird mit dem rein bilateral gefilterten Bild verglichen.

5.2.1 Tiefen-Guide

Ein erster Guide-Typ ist der Tiefen-Guide. Bei diesen wird geschaut, in welchen Abstand zur Kamera die einzelnen Pixel liegen, denn wenn eine große Differenz in der Tiefe besteht, gehören die Pixel wahrscheinlich zu unterschiedlichen Objekten und sollen nicht zusammen gefiltert werden. Problematisch dabei ist zu entscheiden, ab wann ein Abstand als groß bezeichnet werden soll. Wenn dafür ein fester Wert genommen wird, bedeutet eine Skalierung der Szene ein anderes Filterergebnis, was kein gewünschtes Verhalten ist. Deshalb wird die Tiefe mit der maximalen Tiefe normiert, um ein von der Skalierung unabhängiges Filterergebnis zu erzeugen. Das Normieren ist aber auch nicht unbedenklich, da es vorkommen kann, dass Objekte, wie beispielsweise Wolken oder der Himmel, sehr weit entfernt liegen. Wenn dann normiert wird, ist der Unterschied zwischen den anderen Objekten sehr klein und dadurch ist die Unterscheidbarkeit der Objekte nicht mehr gegeben. Auf dieses Problem wird aber hier nicht weiter eingegangen, da in diesen Fall der



(a) Einfach



(b) Kumulativ

Abbildung 8: Positionen-Guides der Bad Szene.

Guide zwar keinen Beitrag zur Kantenerhaltung liefert, aber auch keine falschen Kanten erhält. Die Distanzfunktion für die Tiefe (17) normiert mit der maximalen und minimalen Distanz des Guides, damit ist $\text{dist}_{\text{depth}}(p, q) \in [0, 1]$. Δdepth ist die Differenz zwischen den Tiefenwerten der beiden Pixel.

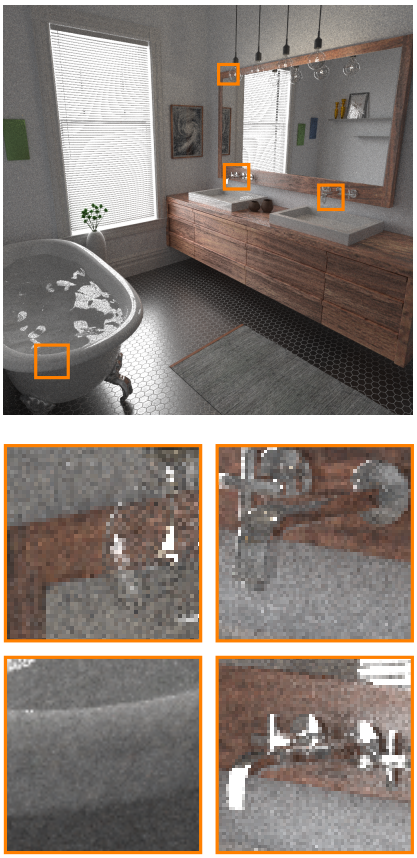
$$\text{dist}_{\text{depth}}(p, q) = \sqrt{\left(\frac{\Delta\text{depth}}{\text{max}_{\text{depth}} - \text{min}_{\text{depth}}}\right)^2} \tag{17}$$

Da der Tiefen-Guide aber im-nachfolgenden Positionen Guide (siehe Unterunterabschnitt 5.2.2) enthalten ist, wird er nicht einzeln betrachtet.

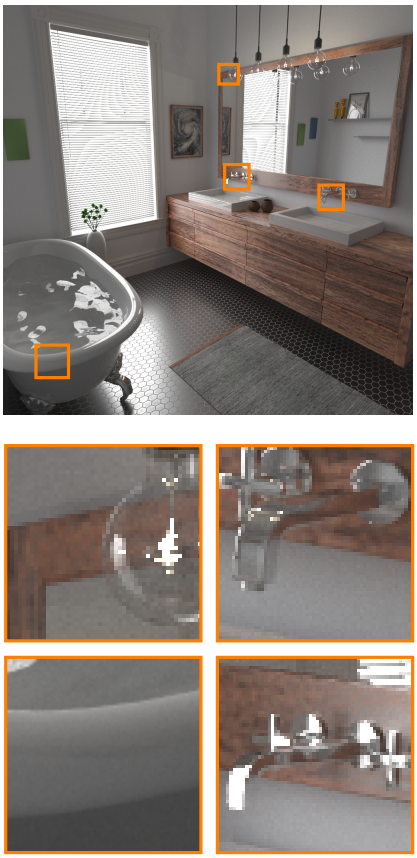
5.2.2 Positionen-Guide

Der Positionen-Guide benutzt nicht nur die Tiefen, sondern gleich die ganzen Positionen der einzelnen Pixel in der Szene. Die Positionen wurden dabei jeweils um das Minimum der Szene verschoben, damit sie sich nur im positiven Bereich befinden, was praktische Gründe hat. Wenn zwei Pixel einen großen Abstand in ihrer dreidimensionalen Position haben, sollen sie nicht zusammen gefiltert werden. Die Normierung der einzelnen Komponenten ist auch bei den Positionen vorteilhaft und erfolgt in (18) mit den jeweiligen Minima und Maxima. Der kumulative und einfache Positionen-Guide der Bad Szene ist in Abbildung 8 zu sehen und die Bad Szene cross-bilateral mit den kumulativen und einfachen Guide zusammen gefiltert in Abbildung 9.

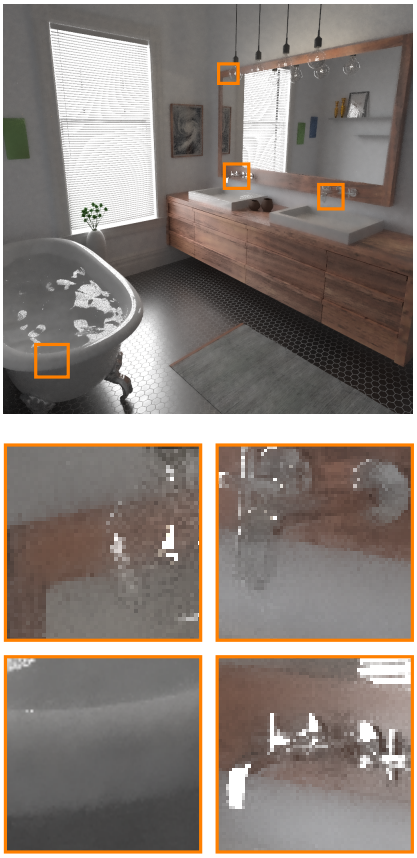
$$\text{dist}_{\text{pos}}(p, q) = \sqrt{\left(\frac{\Delta\text{pos}_\text{x}}{\text{max}_\text{x} - \text{min}_\text{x}}\right)^2 + \left(\frac{\Delta\text{pos}_\text{y}}{\text{max}_\text{y} - \text{min}_\text{y}}\right)^2 + \left(\frac{\Delta\text{pos}_\text{z}}{\text{max}_\text{z} - \text{min}_\text{z}}\right)^2} \tag{18}$$



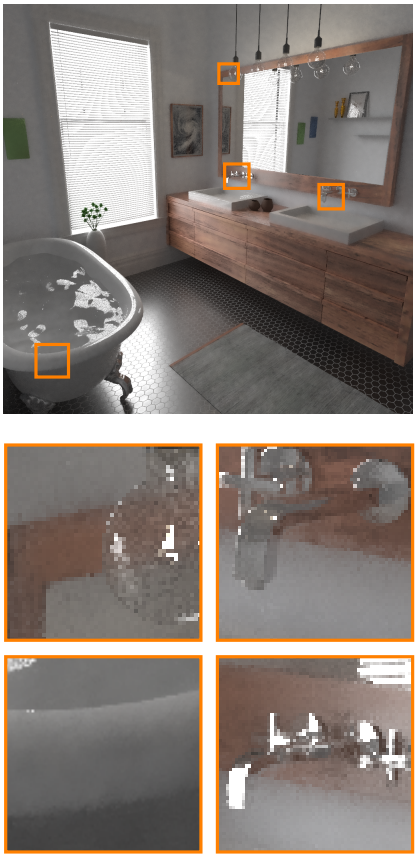
(a) Input 50 It.



(b) Referenzbild

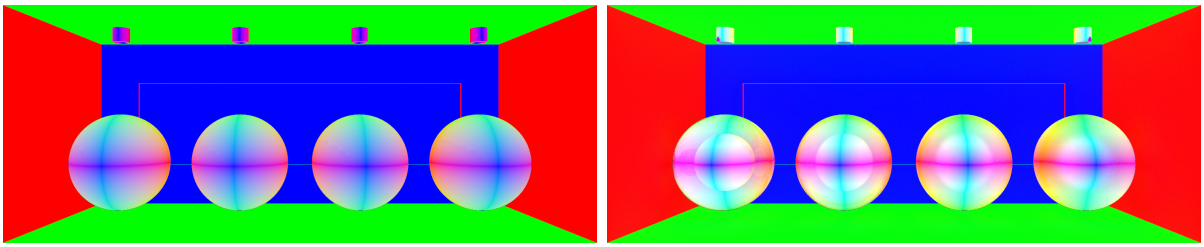


(c) Bilateral



(d) Kum. + ein.

Abbildung 9: Cross-bilateral mit den Positionen-Guides für die Bad Szene: Filterradius = 12, $\sigma_s = 4$, $\sigma_r = 0.7$, $\sigma_{\text{kumPos}} = 0.1$, $\sigma_{\text{einPos}} = 0.15$.



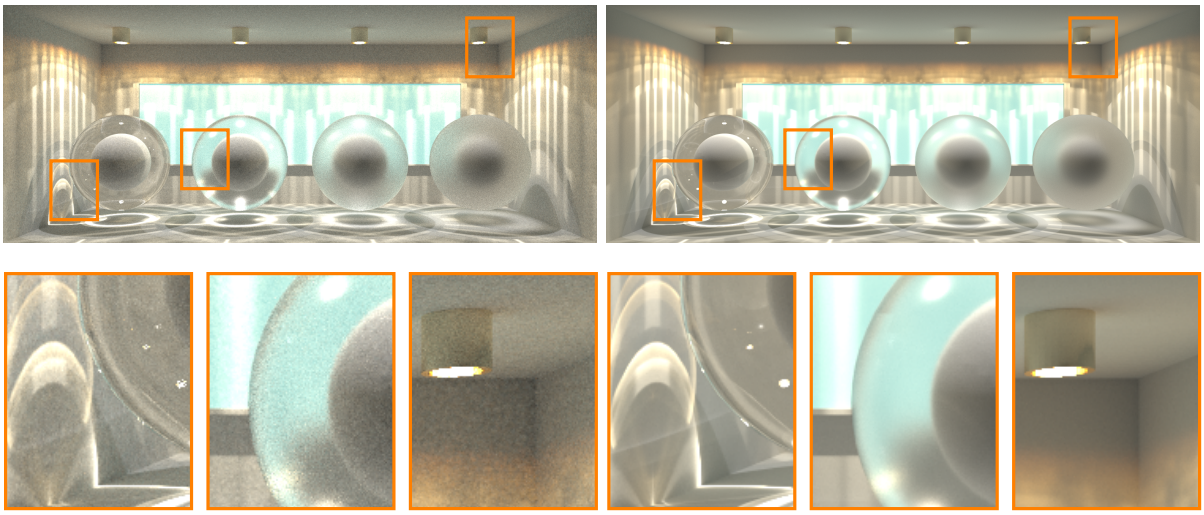
(a) Einfach **(b)** Kumulativ

Abbildung 10: Normalen-Guides der Sphären Szene.

5.2.3 Normalen-Guide

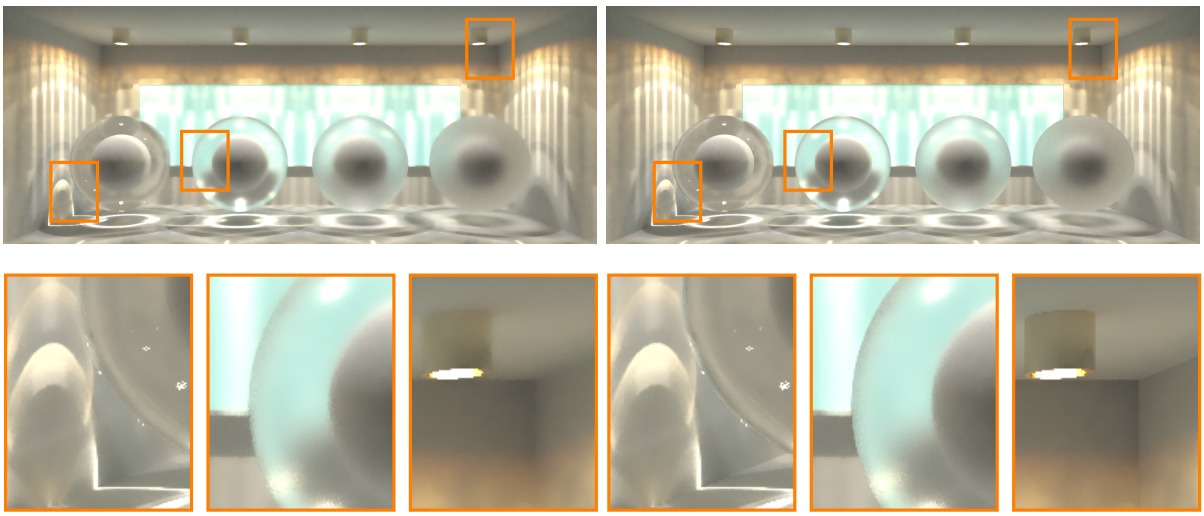
Ein weiterer Guide-Typ ist der Normalen-Guide, welcher die Normalen enthält. Damit lassen sich z. B. Kanten von Objekten besser erkennen, da sich dort die Normalen ändern. Der Positionen-Guide bringt an solchen Stellen wenig, da sich dort nur wenig ändert und wenn das Objekt eher gleichfarbig ist, wie beispielsweise eine Wand, ist das Ursprungsbild als Guide auch nicht hilfreich. Für die Abstandsfunktion (19) wird der euklidische Abstand der Normalenvektoren genommen. Der kumulative und einfache Normalen-Guide der Sphären Szene ist in Abbildung 10 zu sehen und das gefilterte Ergebnis in Abbildung 11.

$$\text{dist}_{\text{normal}}(p, q) = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \tag{19}$$



(a) Input 50 It.

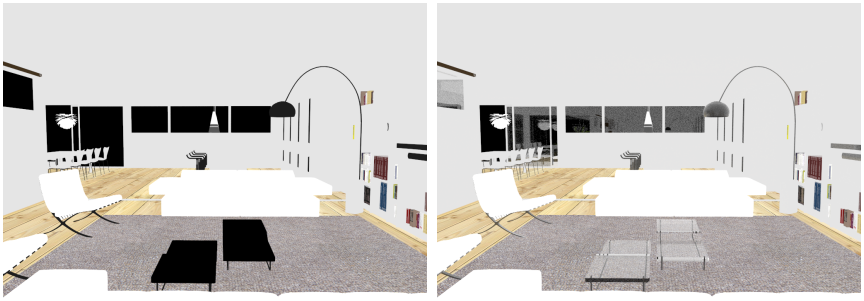
(b) Referenzbild



(c) Bilateral

(d) Kum. + ein.

Abbildung 11: Cross-bilateral mit den Normalen-Guides für die Sphären Szene: Filterradius = 12, $\sigma_s = 4$, $\sigma_r = 0.7$, $\sigma_{\text{kumNormal}} = 0.45$, $\sigma_{\text{einNormal}} = 0.45$.



(a) Einfach **(b)** Kumulativ

Abbildung 12: Texturen-Guides der Villa Szene.

5.2.4 Texturen-Guide

Beim Texturen-Guide wird mithilfe einer bidirektionale Reflexionsverteilungsfunktion (engl. Bidirectional Reflectance Distribution Function, BRDF) [NRH+92] ein Wert für das Material der Objekte bestimmt. Die hier verwendete Formel zur Berechnung des Texturwertes ist (20), welche die Normale und eine Tangente von p verwendet und einen RGB Wert zurückgibt. Die Formel (20) und seine Parameter wurden experimentell bestimmt, und wird hier erstmalig als Guide für cross-bilaterales Filtern verwendet. ρ ist dabei die BRDF Funktion, die zwei Vektoren als Parameter erhält, die Normale sowie eine Interpolation zwischen Normalen und Tangente der getroffenen Oberfläche, ähnlich Abbildung 13.

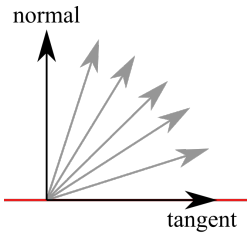
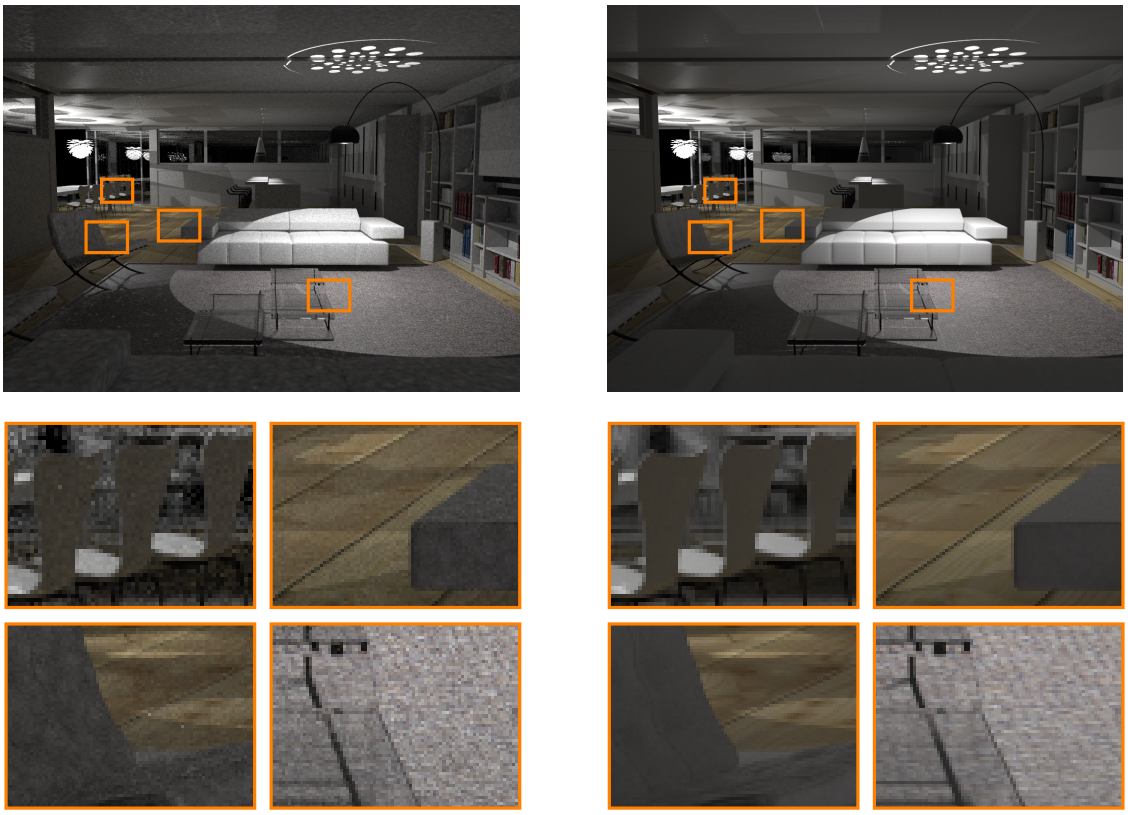


Abbildung 13: Interpolation zwischen Normalen und Tangente einer Oberfläche.

$$\mathrm{Tex}(p) = \sum_{i=1}^{7} \rho \left(\mathbf{n}_p, \frac{\mathbf{n}_p + \mathbf{t}_p \cdot \frac{i^2}{18}}{\left\| \mathbf{n}_p + \mathbf{t}_p \cdot \frac{i^2}{18} \right\|} \right) \tag{20}$$

Dieser Guide ermöglicht das Erhalten von Strukturen in Materialien, wie beispielsweise Holz und Stoffen, und von Kanten zwischen Objekten mit unterschiedlichen Materialien. Materialien haben häufig viele Parameter z. B. ein rau glänzendes Objekt ist anders als ein gleichfarbiges diffuses. Das Auswerten von (20) ermöglicht eine Unterscheidung ohne genaue Kenntnis der BRDF Parameter. Für die Distanzfunktion (21) wird der euklidische Abstand der Materialwerte genommen. Die Texturen-Guides der Villa Szene sind in Abbildung 12 zu sehen und das gefilterte Ergebnis in Abbildung 14.

$$\mathrm{dist}_{\mathrm{tex}}(p, q) = \sqrt{\Delta\mathrm{rot}^2 + \Delta\mathrm{grün}^2 + \Delta\mathrm{blau}^2} \tag{21}$$



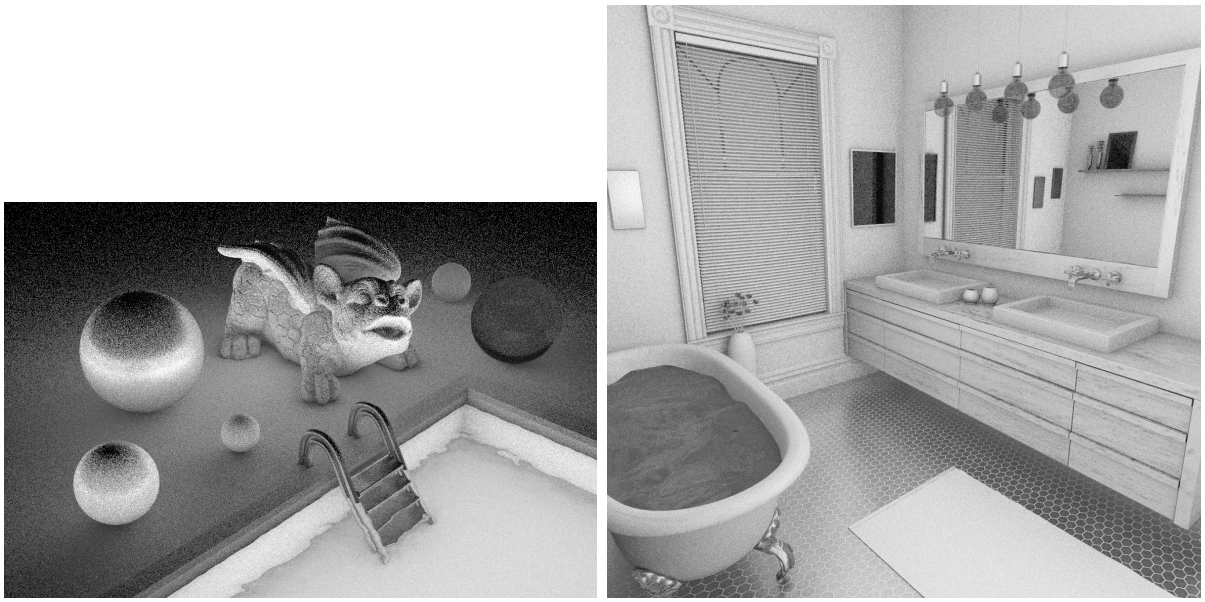
(a) Input 50 It.

(b) Referenzbild

(c) Bilateral

(d) Kum. + ein.

Abbildung 14: Cross-bilateral mit den Textur-Guides für die Villa Szene: Filterradius = 12, $\sigma_s = 4$, $\sigma_r = 0.7$, $\sigma_{\text{kumTexl}} = 0.4$, $\sigma_{\text{einTex}} = 0.4$.



(a) **(b)**

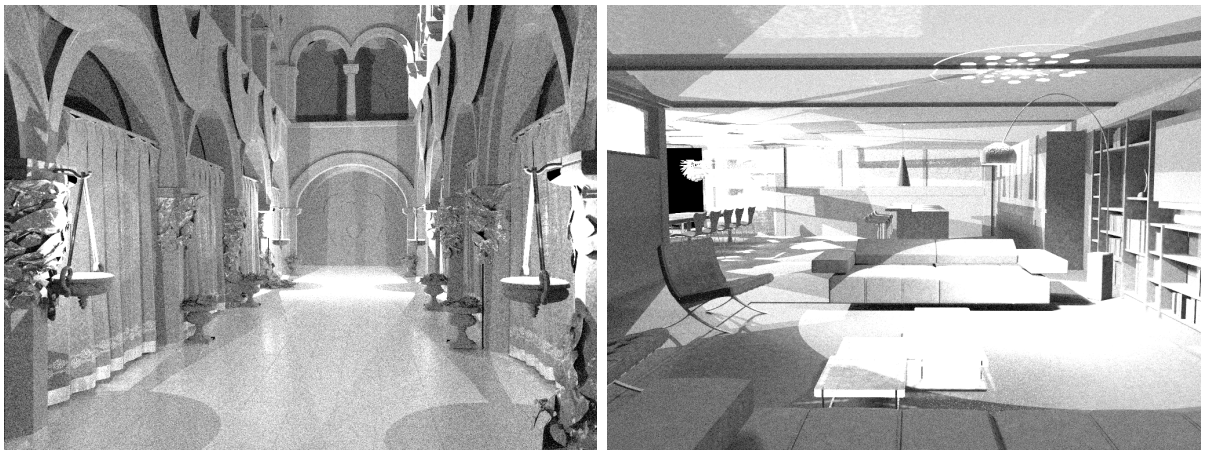
Abbildung 15: Kumulative ambiente Verdeckung Guides der Drachenpool und Bad Szene.

5.2.5 Ambiente Verdeckung Guide

Mit Hilfe eines weiteren neu entwickelten Guides, den ambiente Verdeckung Guide wird versucht Schattenkanten, welche durch Verdeckung von anderen Objekten entstehen, zu erhalten. Dafür wird das Integral über alle Richtungen für die Pfadlängen bestimmt und abhängig vom Material gewichtet (14). Wenn ein Pixel nahe an Objekten ist, sind viele Pfade kurz, weil diese mit den Objekten kollidieren, wodurch das Integral kleiner wird und wenn er entfernt von Objekten ist, sind die Pfade länger. Für die Distanzfunktion (22) wird der Wert wie in (17) normiert und die Differenz gebildet.

$$\mathrm{dist}_{\mathrm{amb}}(p,q) = \sqrt{\left(\frac{\Delta \mathrm{amb}}{\mathrm{max}_{\mathrm{amb}} - \mathrm{min}_{\mathrm{amb}}}\right)^2} \tag{22}$$

Der ambiente Verdeckung Guide enthält ein starkes Rauschen und ist somit in diesen Zustand nicht nützlich. Auch die einfache Variante des Guides enthält starkes Rauschen und wurde für diesen Guide weggelassen, da kein Vorteil gegenüber den kumulativen Guide besteht, welcher für die Drachenpool und Bad Szene in Abbildung 15 zu sehen ist. Da die höchste SSIM-Werte nur bei großen σ_{amb} erreicht wurden, was praktisch den Guide aus der Gewichtung herausnimmt, gibt es keinen sichtbareren Unterschied bei der Verwendung des Guides. Aus diesem Grund sind hier für diesen Guide keine Vergleichsbilder.



(a) **(b)**

Abbildung 16: Kumlative Licht-Guides der Sponza und Villa Szene.

5.2.6 Licht-Guide

Es wird auch noch ein anderer neu entwickelter Guide, der Licht-Guide, verwendet. Darin wird die Beleuchtungssituation dargestellt, was unter anderen hilft Kaustiken besser zu erhalten. Dafür wird jedes Mal, wenn ein View-Pfad mit einem Licht-Pfad verbunden wird, die Gewichte beider Pfade nach (14) miteinander multipliziert und am Ende pro Pixel aufaddiert. Die kumulativen Licht-Guides für die Sponza und Villa Szene sind in Abbildung 16 zu sehen.

Der Licht-Guide enthält wie der ambiente Verdeckung Guide starkes Rauschen und kann somit in diesen Zustand nicht verwendet werden und auch hier wird keine einfache Variante verwendet.

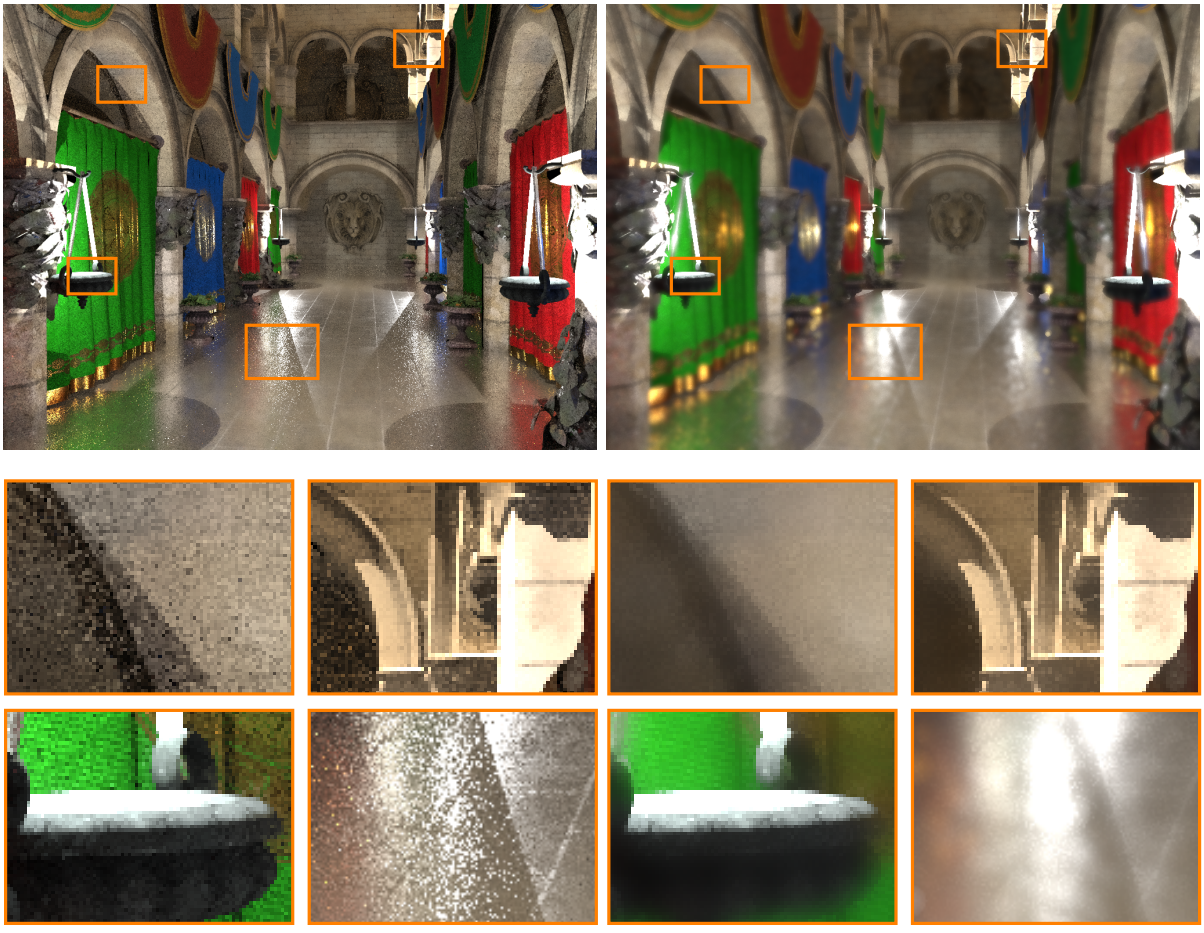
5.3 Varianzen zur Rauschverringerrung

Die Licht- und ambiente Verdeckung Guides sind sehr verrauscht und es können daher fehlerhaft Kanten erkannt werden. Eine Möglichkeit, um den Einfluss, den das Rauschen auf das Ergebnis nimmt, zu verringern, ist die Guides jeweils mit ihren Varianzen zu normalisieren, welches in [LWC12] vorgeschlagen wurde. Dafür wird durch die Summe der Varianzen der betrachteten Pixel des Guides geteilt und damit die Normalisierung, falls vorhanden, ersetzt. Für den ambiente Verdeckung Guide beispielsweise in (23), wobei $\sigma^2_{\text{amb},p}$ die Varianz von p im Guide ist.

$$\text{dist}_{\text{ambVar}}(p,q) = \sqrt{\left(\frac{\Delta\text{amb}}{\sigma^2_{\text{amb},p} + \sigma^2_{\text{amb},q}}\right)^2} \tag{23}$$

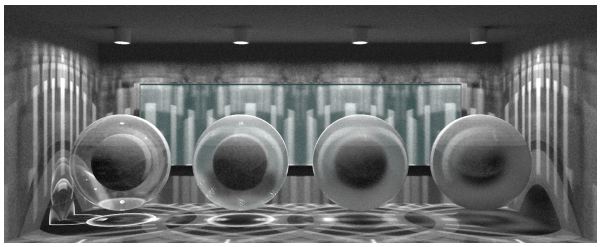
Die Stichproben-Varianzen der einzelnen Pixel können einfach beim Rendern der Guides mit berechnet werden und sie werden für den Licht- und ambiente Verdeckung Guide sowie für das Ausgangsbild verwendet, da diese die stärker verrauschten Guides sind.

(a) Input 50 It. **(b)** Varianzbild **(c)** Referenzbild

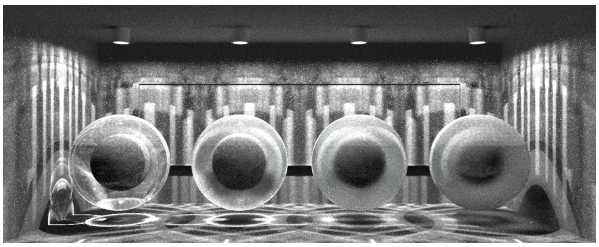


(d) Bilateral **(e)** Bilateral Var.

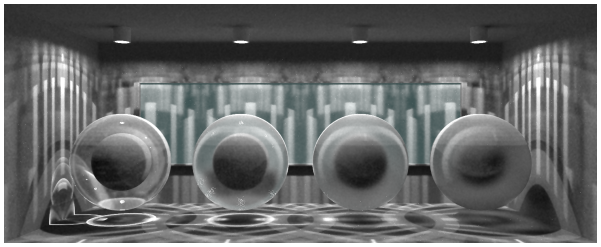
Abbildung 17: Bilateral ohne und mit Varianz für die Sponza Szene: Filterradius = 12, $\sigma_s = 4$, Bilateral: $\sigma_r = 0.7$, Bilateral Var.: $\sigma_r = 0.5$.



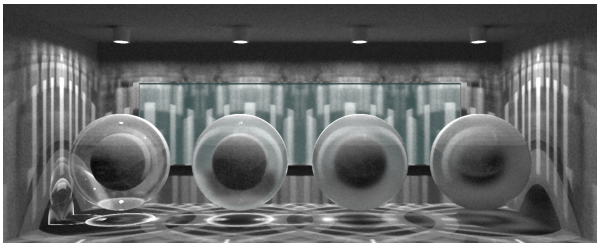
(a) Licht-Guide



(b) Varianz des Licht-Guides



(c) Cross-bilateral gefiltert

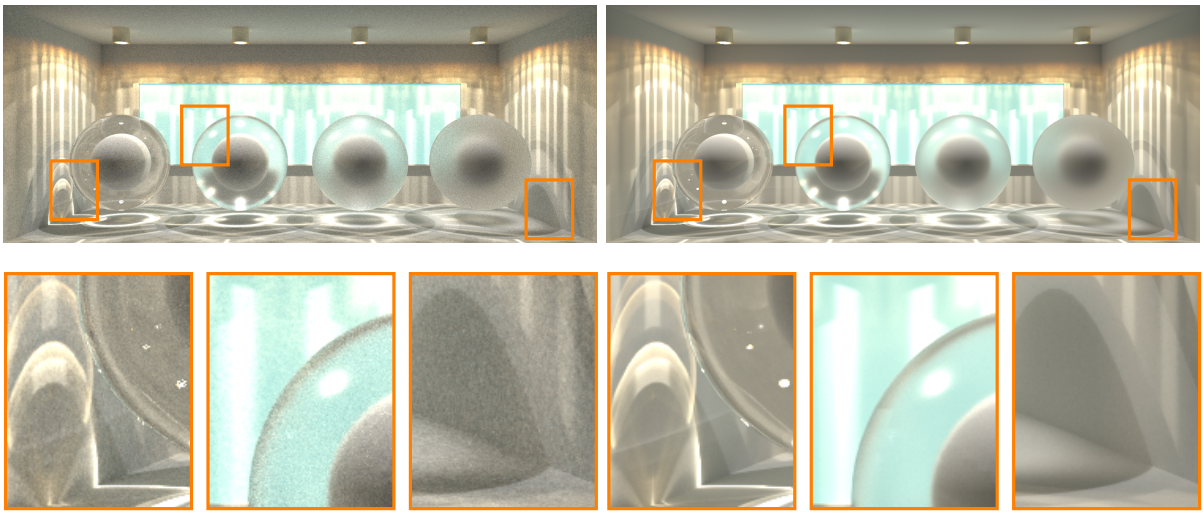


(d) Median gefiltert

Abbildung 18: Licht-Guides der Sphären Szene. Cross-bilateral: Filterradius = 2, $\sigma_s = 3$, $\sigma_{\text{light}} = 1.0$, $\sigma_{\text{normal}} = 0.45$, $\sigma_{\text{position}} = 0.1$. Med: Filterradius = 1.

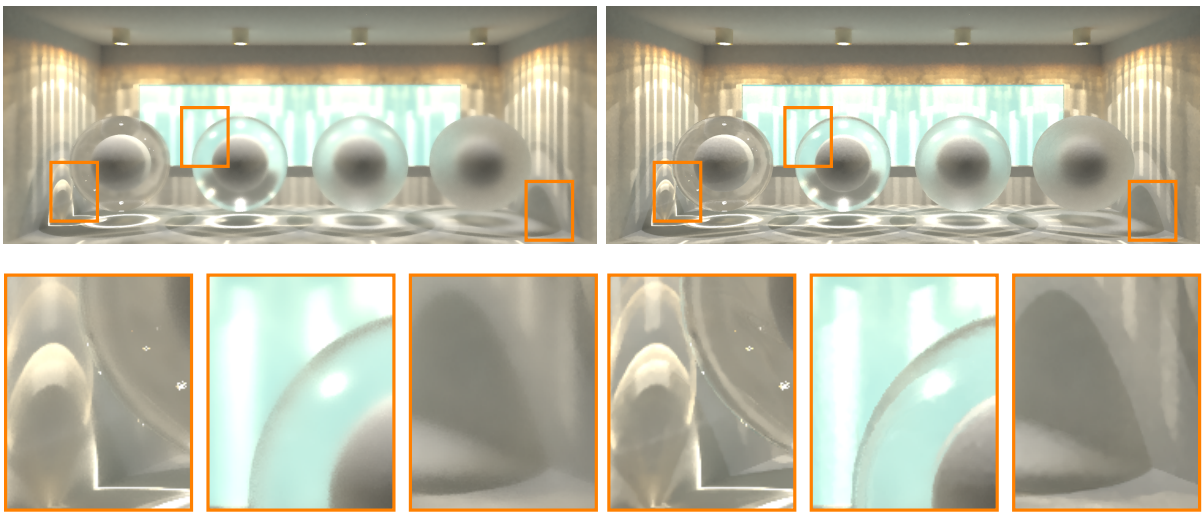
5.4 Vorfiltern von Verrauschten Guides

Die Guides selber können natürlich auch selber gefiltert werden, um ihr Rauschen zu verringern. Dafür werden diese mit einem cross-bilateralen Filter, der den Guide selbst, sowie den Normalen- und Positionen-Guide verwendet, gefiltert. Beim ambienten Überdeckungen Guide wird noch zusätzlich das Varianzbild zur weiteren Verringerung des Rauschens benutzt. Die Varianz wird nicht beim Filtern des Licht-Guides verwendet, da dies zu Artefakten in der Nähe von Lichtquellen führt. Den Texturen-Guide zu verwenden ist dabei nicht sinnvoll, da die zu filternden-Guides nicht davon profitieren, weil sie keine Textur Informationen enthalten. Zusätzlich werden auch die Varianzbilder parallel mit denselben Guides gefiltert, um ein gefiltertes Varianzbild zur weiteren Verwendung zu erhalten. Da bei den cross-bilateralen Vorfiltern des Licht-Guides manche Kanten nicht gut erhalten werden, wird es auch nur mithilfe eines Median-Filters, mit Radius von eins, gefiltert und das resultierende Bild zusätzlich als weiterer Guide benutzt.



(a) Input 50 It.

(b) Referenzbild



(c) Bilateral

(d) c.b. Var. + Med.

Abbildung 19: Cross-bilateral mit den gefilterten Licht-Guides für die Sphären Szene: Filterradius = 12, $\sigma_s = 4$, $\sigma_r = 0.7$, $\sigma_{\text{cbVar}} = 0.3$, $\sigma_{\text{Med}} = 0.075$.

5.5 Fazit zum Cross-Bilateralen Filtern

Mit dem cross-bilateralen Filter ohne Varianz kann das Rauschen verringert werden, ohne dabei viele Kanten zu zerstören. Jedoch bleibt starkes Rauschen erhalten und einzelne fälschlicherweise sehr helle Pixel, welche auch Fireflys genannt werden, werden nicht zuverlässig entfernt. Bei dem cross-bilateralen Filter mit Varianz, ist das Rauschen fast vollständig verschwunden und auch Fireflys werden gut entfernt, jedoch werden Glanzpunkte nicht gut erhalten und verschwimmen stark. Die Ergebnisbilder sind am Ende des Abschnitts in Abbildung 22 dargestellt.

In Tabelle 3 sind die SSIM-Werte von jeden einzelnen Guide, sowie die SSIM-Werte von den gesamten Filtern zu sehen. Im folgenden Abschnitt werden die einzelnen Guides ausgewertet.

Tabelle 3: SSIM-Werte durch cross-bilateralesFiltern mit dem jeweiligen Guide, sowie unter Verwendung aller Guides. Die Werte für das Bilaterale und Ungefilterte Bild unterscheiden sich von den Vorherigen, da hier andere Ausgangsbilder genommen worden sind.

Szene	Bad	Drachenpool	Sphären	Sponza	Villa
Ungefiltert	0.659069	0.89769	0.928013	0.806774	0.97439
Bilateral	0.846762	0.956919	0.980799	0.874094	0.986999
Bilateral Var.	0.829476	0.940578	0.986222	0.914532	0.984408
Normalen kum.	0.858708	0.958878	0.983149	0.8764	0.98818
Normalen ein.	0.854299	0.958754	0.982677	0.876168	0.987964
Normalen kum. + ein.	0.860094	0.959345	0.98325	0.876653	0.988312
Positionen kum.	0.851579	0.957731	0.981998	0.874705	0.987211
Positionen ein.	0.847253	0.957312	0.982133	0.874293	0.987086
Positionen kum. + ein.	0.851946	0.957871	0.982372	0.874873	0.987281
Texturen kum.	0.883531	0.957789	0.981795	0.880807	0.987814
Texturen ein.	0.880506	0.95766	0.98195	0.881397	0.988177
Texturen kum. + ein.	0.888496	0.957969	0.981993	0.881278	0.987975
Ambiente kum.	0.847049	0.956924	0.980853	0.87388	0.987042
Ambiente Var.	0.855047	0.957254	0.981768	0.874891	0.987611
Ambiente c. b. Var.	0.858587	0.957697	0.983356	0.876102	0.988059
Licht kum.	0.849099	0.958015	0.986913	0.874152	0.987827
Licht Var.	0.849033	0.959348	0.99018	0.873661	0.988678
Licht c. b. Var.	0.861032	0.963735	0.992636	0.876517	0.989918
Licht Med.	0.853104	0.962871	0.992304	0.874732	0.989251
Licht c. b. Var.+ Med.	0.861296	0.964102	0.992687	0.876518	0.99009
Cross-bilateral	0.89881	0.965186	0.993892	0.881371	0.990478
Cross-bilateral Var.	0.911225	0.957531	0.99368	0.971107	0.991598

(a) Normalen Differenz **(b)** Positionen Differenz

Abbildung 20: Differenzbilder der cross-bilateral gefilterten Bilder mit den rein bilateral gefilterten Bildern.

5.5.1 Fazit zum Positionen-Guide und Normalen-Guide

Als einzige Guides sind diese beiden direkt miteinander vergleichbar, da sie ähnliche Kanten enthalten. So stellt sich die Frage, ob einer dieser beiden Guides dem anderen direkt überlegen ist. Laut den Werten aus Tabelle 3 ist der Normalen-Guide besser, aber es werden auch Kanten vom Positionen-Guide erkannt, welche nicht von Normalen-Guide erkannt werden. Dies ist der Fall, wenn Objekte voreinander sind, die ähnliche Normalen haben, siehe dafür Abbildung 20, wo dies an den oberen und unteren Kanten des Bildes vorkommt. Im Allgemeinen ist es so, dass der Normalen-Guide viele Kanten erkennt und diese auch sehr gut erhält. Der Positionen-Guide hingegen erkennt weniger Kanten und erhält diese auch nicht so gut wie der Normalen-Guide, aber er ergänzt diesen um einige Kanten und ist somit nicht überflüssig. Zusammen erhalten sie fast alle Kanten, die durch die Form und räumliche Anordnung von Objekten entstehen.

5.5.2 Fazit zum Texturen-Guide

Der erste der neuen Guides ist der Texturen-Guide, welcher sehr gut darin ist Strukturen zu erhalten. Beispiele dafür sind die Teppiche, wie sie in der Bad, Sponza und Villa Szene zu finden sind, Holzmaserungen, welche in der Bad Szene vorhanden sind, oder auch bei anderweitigen Strukturen, wie bei dem Bild in der Bad Szene. Diese Strukturen können

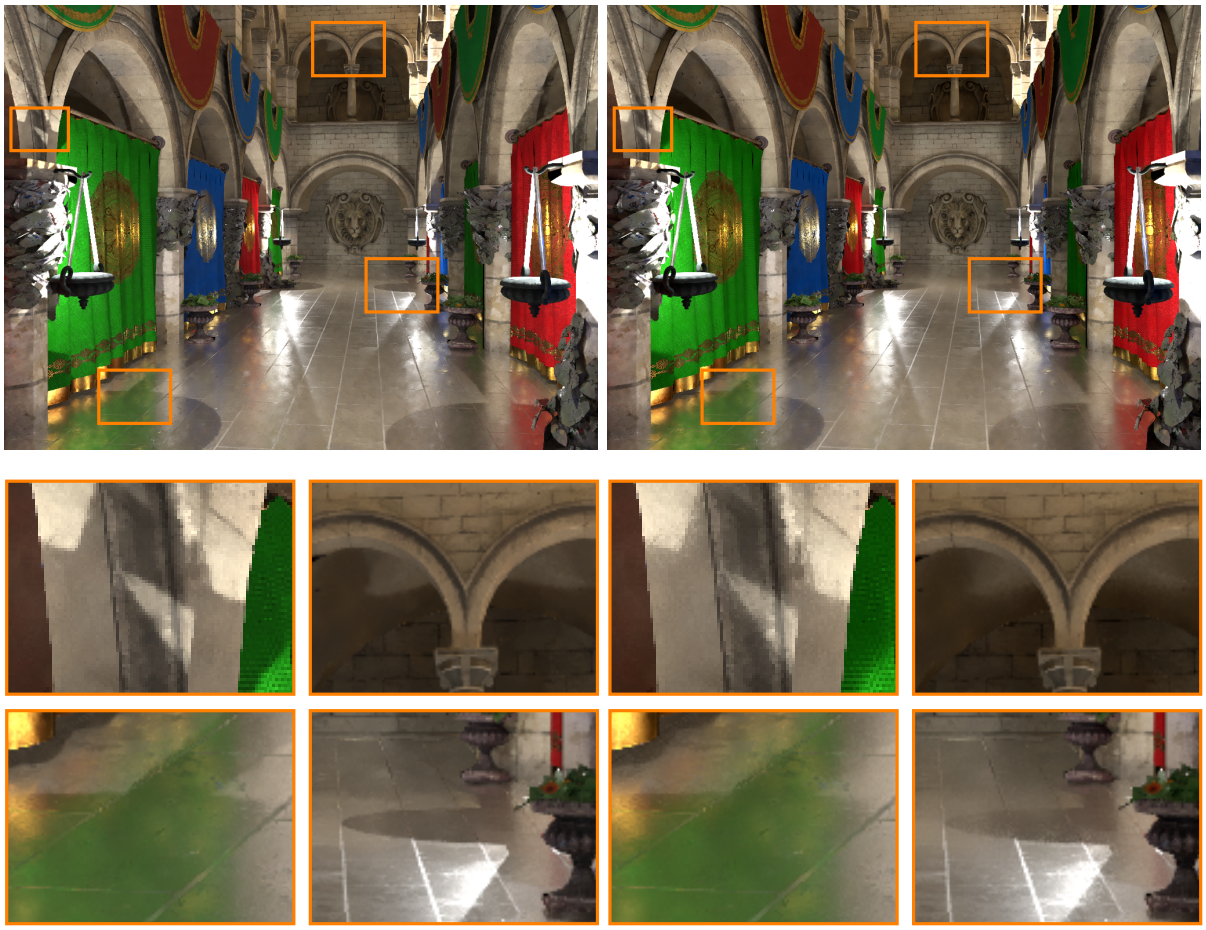
in keinen anderen Guide erhalten werden, da sie dort nicht vorhanden sind, bzw. bei dem Ausgangsbild als Rauschen erkannt werden. Er wird zwar bei einigen Szenen in Tabelle 3 nicht so hoch bewertet, wie andere Guides, was aber daran liegt, das in diesen Szenen nicht viele solcher Strukturen vorhanden sind. Der Texturen-Guide kann in diesen Szenen nicht viel helfen, aber er verursacht dort auch keine Fehlentscheidungen. Insgesamt ist es ein sehr guter Guide und es lohnt sich diesen zu verwenden.

5.5.3 Fazit zum Ambiente Verdeckung Guide

Der Ambiente Verdeckung Guide ist in der Version ohne Vorfiltern und ohne Varianz zu verrauscht, um hilfreich zu sein. In der gefilterten Variante mit Varianz hingegen, kann der Guide verwendet werden, um Schattierungen, welche durch Verdeckung von Objekten entstehen, besser zu erhalten. Ein einfaches Beispiel für eine solche Schattierung ist, wenn man die Ecke eines Raumes betrachtet, sofern diese nicht direkt beleuchtet wird. Nahe der Kanten ist es dunkler als weiter von dieser entfernt. Die Effekte des Filters sind dabei eher gering, da diese Schattierungen eher schwach und nicht so häufig sind. Auf die Verwendung kann deshalb ohne großen negativen Effekt verzichtet werden, aber es schadet auch nicht ihn zu verwenden, da ein paar Stellen des Bildes damit verbessert werden können. Die Stellen sind eher schlecht zu erkennen, weshalb an dieser Stelle kein Beispielbild hierzu gezeigt wird.

5.5.4 Fazit zum Licht-Guide

Der Licht-Guide ist auch erst in der gefilterten Variante mit Varianz hilfreich und mit ihm können Schatten und Kaustiken besser erhalten werden. Durch das Vorfiltern werden aber teilweise die Kaustiken verschlechtert und nicht so gut erhalten wie erhofft. Dennoch hat er einen größeren Effekt und ist in mehreren Szenen, laut Tabelle 3, der beste Guide. Ohne ihn sind Schattenkanten sichtbar schlechter erhalten, siehe Abbildung 21, und Kaustiken sind eher verschwommen. Deswegen lohnt es sich auch diesen neuen Guide zu verwenden. Wenn mehr Zeit zum Rendern investiert werden kann, lohnt es sich auch die ungefilterte Variante des Guides zu verwenden, da der Guide mit mehr Iterationen auch weniger verrauscht ist. Eine Variante mit 500 anstelle der bisherigen 50 Iterationen hat gut funktioniert und damit können vor allem die Kaustiken besser erhalten werden. Das gilt natürlich



(a) Cross-bilateral Var. **(b)** Cross-bilateral Var. ohne Licht-Guide

Abbildung 21: Cross-bilateral gefilterte Sponza Szene mit und ohne Licht-Guide.

auch für den Ambiente Verdeckung Guide.

5.5.5 Fazit zu den Einfachen und Kumulativen Guides

Die neu entwickelten kumulativen Guides und die einfachen Guides sind ähnlich gut bei der Kantenerhaltung, siehe Tabelle 3. Der einfache Guide enthält weniger Rauschen und ist so in manchen Szenen besser, jedoch enthält er keine Informationen zu beispielsweise Reflexionen in Spiegeln, weswegen in Szenen mit solchen, z. B. in der Bad Szene, der kumulative Guide besser ist. Außerdem hat sich herausgestellt, dass eine gemeinsame Verwendung von beiden oft zu Verbesserungen, über die SSIM-Wert des besseren von Beiden hinaus, geführt hat. In keiner der verwendeten Szenen, hat eine Verwendung von beiden gleichzeitig zu einer Verschlechterung, unter den SSIM-Wert des schlechteren von beiden, geführt. Jedoch wurde manchmal ein Mittelwert zwischen den beiden SSIM-Werten erreicht. Dennoch empfiehlt sich eine Verwendung von beiden gleichzeitig.

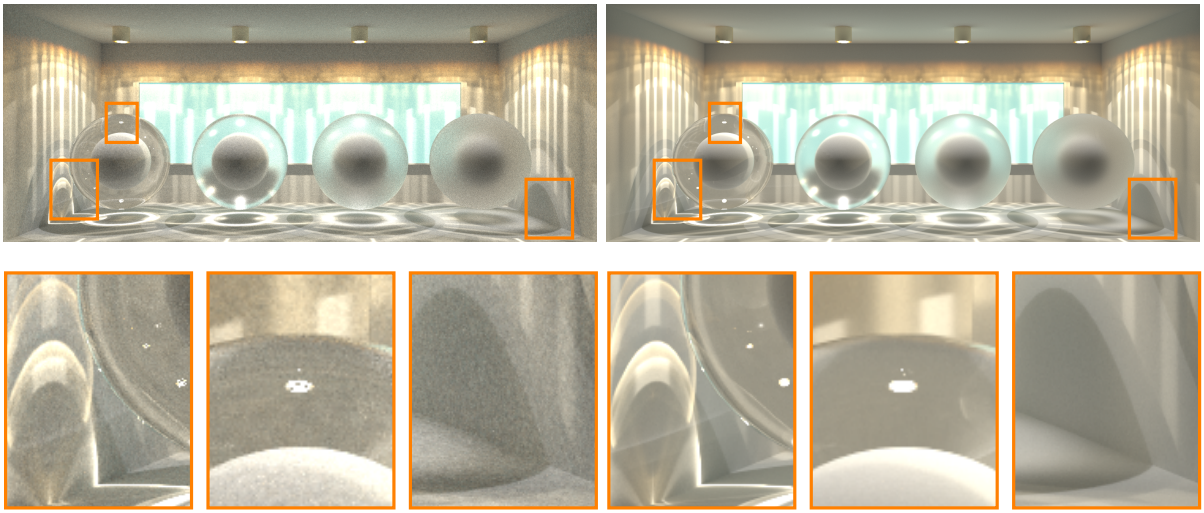
5.5.6 Verwendete Einstellungen

Die Sigmas wurden so gewählt und angepasst, dass ein möglichst guter SSIM-Wert bei den gefilterten Szenen herauskommt. Dabei sind die optimalen Sigma-Werte für jede Szene unterschiedlich und nicht so einfach ohne Kenntnis des Referenzbildes zu Bestimmen. Die in Tabelle 4 bestimmten Werte haben aber für alle ausprobierten Szenen gute Ergebnisse geliefert, mit Ausnahme bei den ungefilterten Licht- und ambiente Verdeckung Guide, da diese zu verrauscht sind. Der Filterradius wurde auf 12 und $\sigma_s = 4$ festgelegt. Bei den Filtern für die Guides selbst, wurden die Sigmas mit den gleichen Werten belegt, aber Filterradius = 2 und $\sigma_s = 3$ wurden kleiner gewählt, damit Kanten besser erhalten bleiben. Bei den Medianfilter wurde auch ein sehr kleiner Filterradius = 1 aus demselben Grund gewählt.

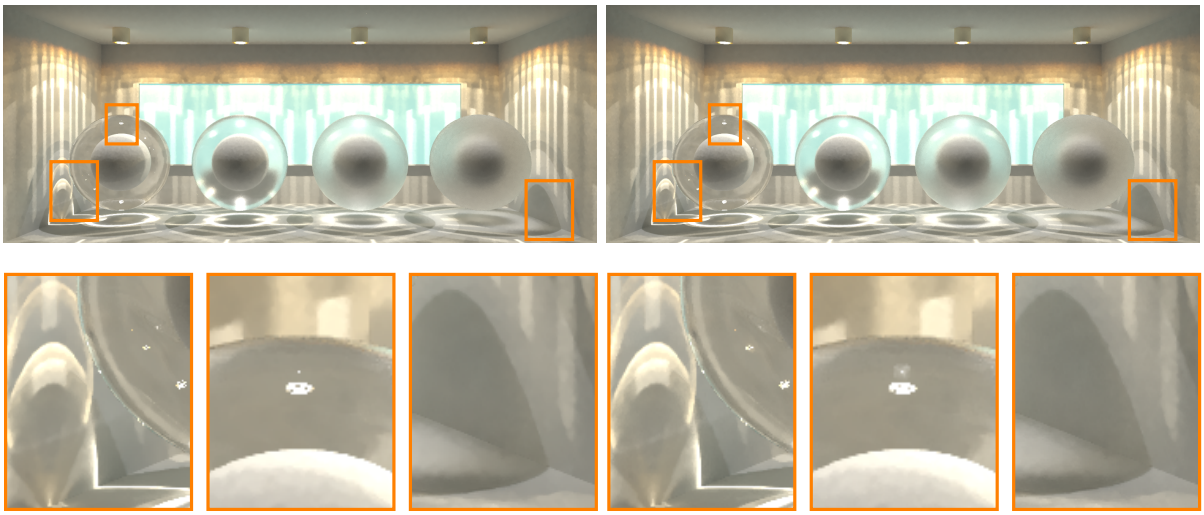
Tabelle 4: Verwendete Sigmas der einzelnen Guides.

Guide	Sigma
Bilateral	0.7
Bilateral Var.	0.5
Normalen kum.	0.45
Normalen sim.	0.45
Positionen kum.	0.1
Positionen sim.	0.15
Texturen kum.	0.4
Texturen sim.	0.4
Ambiente kum.	2
Ambiente Var.	0.3
Ambiente c. b. Var.	0.15
Licht kum.	1
Licht Var.	0.425
Licht c. b. Var.	0.075
Licht Med.	0.3

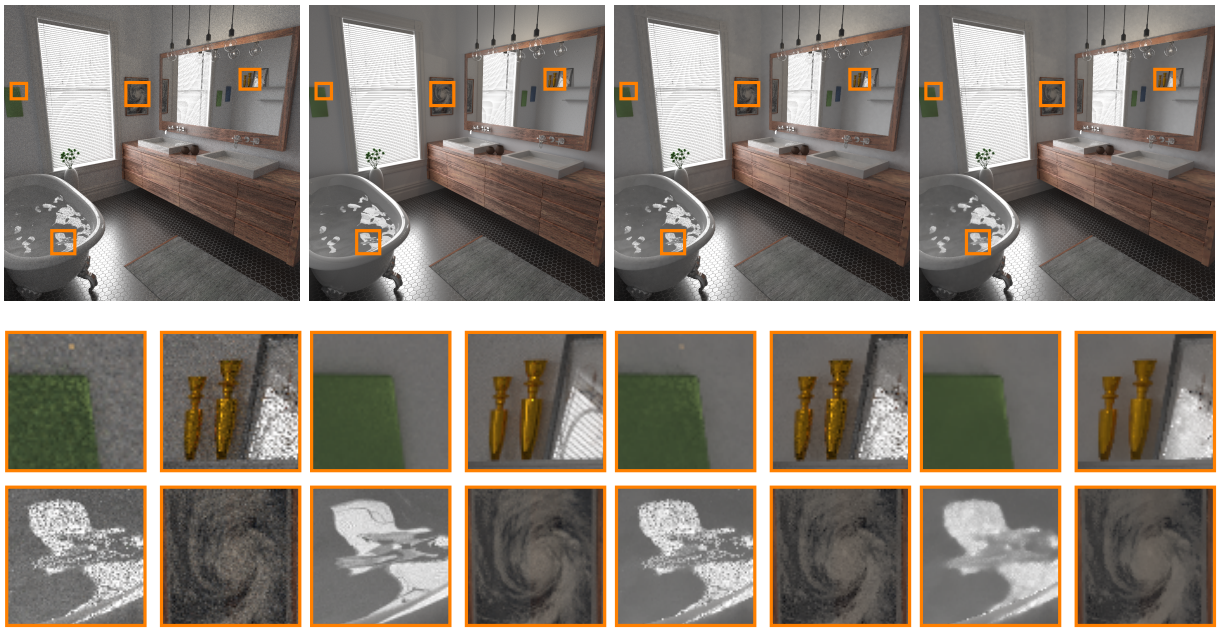
Bei dem cross-bilateralen Filter wurden die Guides mit ihren kumulativen und einfachen Variante gewählt, für den ambienten Guide, die cross-bilateral gefilterte Variante mit der gefilterten Varianz und für den Licht-Guide auch diese Variante zusammen mit der Median gefilterten verwendet. Es wurde jeweils einmal mit Varianz von dem Ausgangsbild und einmal ohne gefiltert. Die Ergebnisse sind in Abbildung 22 für die fünf verwendeten Szenen zu sehen.



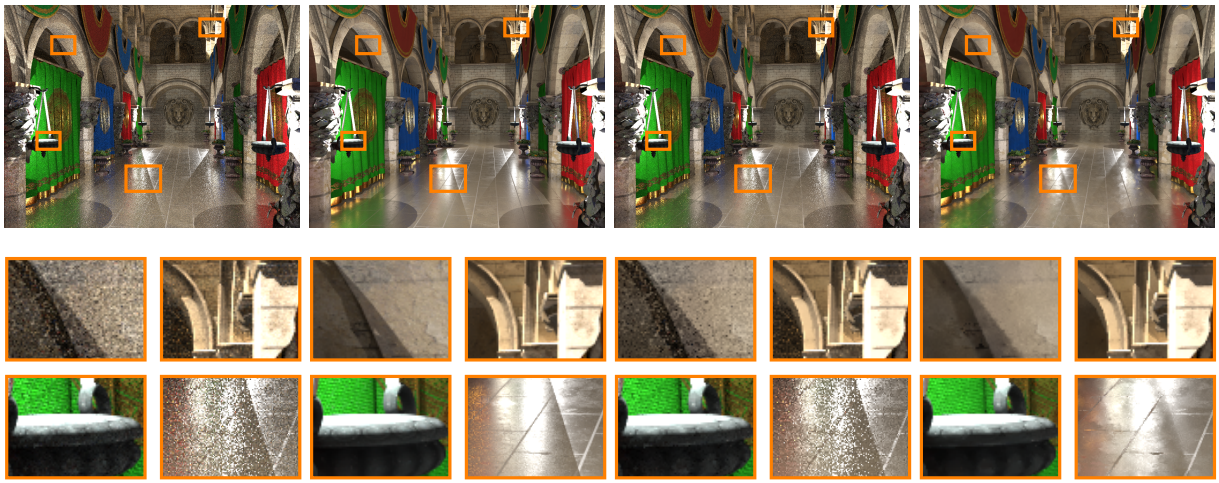
(a) Input 50 It. (b) Referenzbild



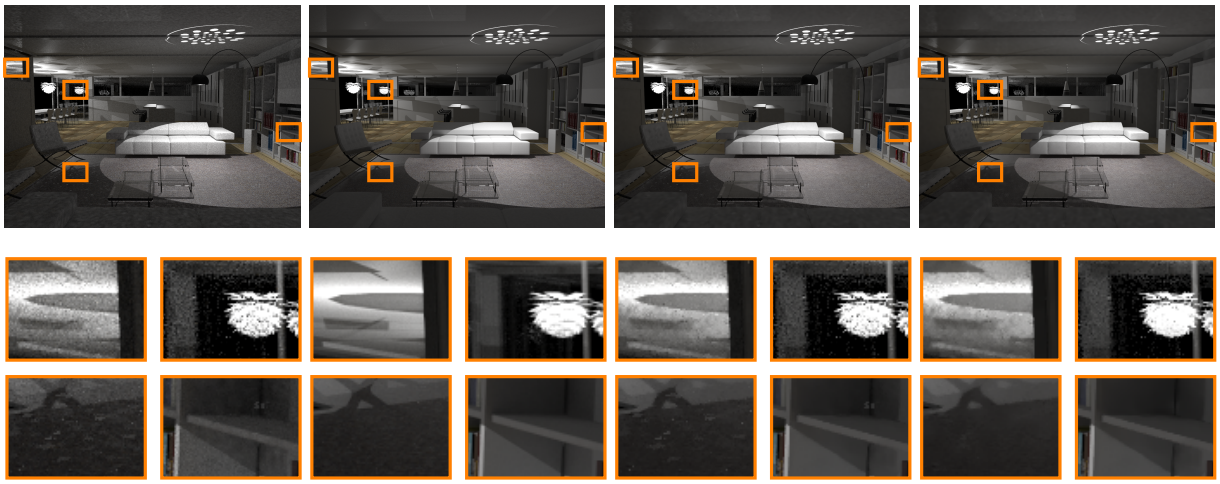
(c) Cross-bilateral (d) Cross-bilateral Var.



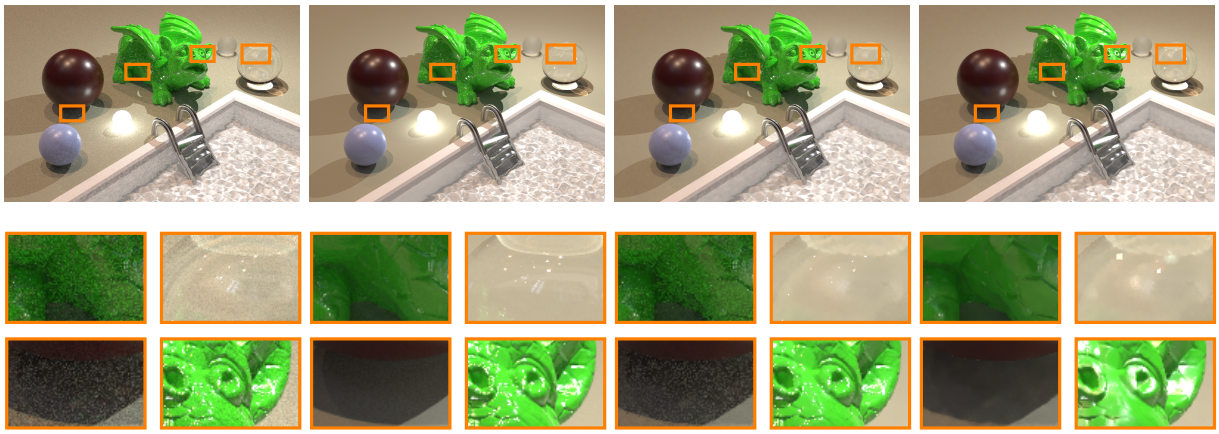
(e) Input 50 It. (f) Referenzbild (g) Cross-bilateral (h) Cross-bilateral Var.



(i) Input 50 It. (j) Referenzbild (k) Cross-bilateral (l) Cross-bilateral Var.



(m) Input 50 It. (n) Referenzbild (o) Cross-bilateral (p) Cross-bilateral Var.



(q) Input 50 It. (r) Referenzbild (s) Cross-bilateral (t) Cross-bilateral Var.

Abbildung 22: Cross-bilateral gefilterte Szenen.

6 SURE Filter

Der SURE Filter [LWC12] benutzt „Stein's unbiased risk estimate“ (SURE) [Ste81], um damit den Fehler von verschiedenen Filtern zu schätzen und den optimalen Filter auszuwählen. Dafür werden mehrere Filter, hier cross-bilaterale Filter mit verschiedenen Radien, auf das Bild angewendet und für jeden Pixel der SURE bestimmt. Anschließend wird der Filter ausgewählt, welcher den besten SURE geliefert hat.

6.1 SURE

SURE ist ein Schätzer für den Fehler eines anderen gegebenen Schätzers. Wenn w eine Messung von v mit Normalverteilung $N(v, \sigma_w^2)$ und F eine schwach differenzierbare Funktion ist, dann schätzt (24) den Fehler, welcher durch F erzeugt wird, ab

$$\mathrm{SURE}(F(w)) = \|F(w) - w\|^2 + 2\sigma_w^2 \frac{\Delta F(w)}{\Delta w} - \sigma_w^2 \tag{24}$$

und ist ein erwartungstreuer Schätzer des „mean squared error“(MSE).

$$E[\mathrm{SURE}(F(w))] = \|F(w) - v\|^2 \tag{25}$$

Wenn jetzt σ_w und $\frac{\Delta F(w)}{\Delta w}$ ausgerechnet werden können, kann der Fehler von F geschätzt werden ohne den Wert von v zu kennen.

6.2 SURE basiertes Filtern

Das Ziel von SURE basierten Filtern ist es, die Formeln aus Unterabschnitt 6.1 zu benutzen, um die Fehler verschiedener Filter zu schätzen und so den besten Filter zu finden. Da die Stichproben beim Monte-Carlo-Rendering normalverteilt sind, kann SURE als Schätzer für den MSE der cross-bilateral gefilterten Pixel $Cross(p)$ benutzt werden. Der cross-bilaterale Filter wird also als Schätzer für den Pixel p betrachtet. Um den SURE zu bestimmen, muss nur $\frac{\Delta \mathrm{Cross}(p)}{\Delta p}$ berechnet werden, denn σ_p, also die Varianz der Pixel, kann direkt beim Rendern von den Monte-Carlo-Stichproben erhalten werden. Für die Berechnng von $\frac{\Delta \mathrm{Cross}(p)}{\Delta p}$ leitet [LWC12] Formel (26) her.

$$\frac{\Delta \mathrm{Cross}(p)}{\Delta p} = \frac{1}{\sum_{q \in U} w_{\mathrm{Gauss}}(p, q)} + \frac{1}{\sigma_r^2} \left(\mathrm{Cross}^2(p) - \mathrm{Cross}(p)^2 \right) \tag{26}$$

wobei

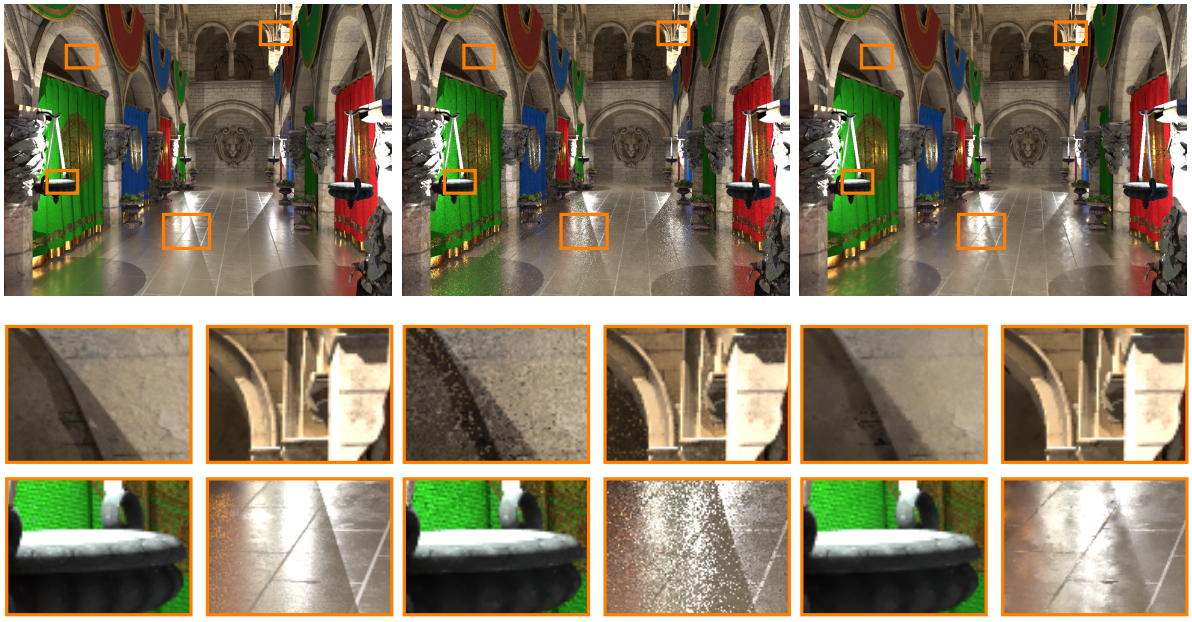
$$\text{Cross}^2(p) = \frac{\sum_{q \in U} w_{\text{Gauss}}(p,q) p^2}{\sum_{q \in U} w_{\text{Gauss}}(p,q)} \tag{27}$$

6.2.1 Verschiedene Filtergrößen

Damit die Pixel nun optimal ausgewählt werden können, müssen verschiedene Filter auf das Bild angewendet werden. Eine Möglichkeit dafür ist es, die Filtergrößen zu variieren und dadurch verschiedene Ergebnisse zu erhalten. Hier wird einfach jede mögliche kleinere Filtergröße genommen, da mit Speicherung von Zwischenergebnissen nur wenig Mehraufwand entsteht. Da die Selektierung der Filtergrößen sehr verrauscht ist und dadurch ein verrauschtes Endergebnis entsteht, wird die Filtergrößenauswahl mit einem Gauß-Filter geglättet. Dieses Verfahren wurde jeweils für den cross-bilateralen Filter mit und ohne Varianz ausprobiert und die Ergebnisse sind in Abbildung 23 zu sehen.

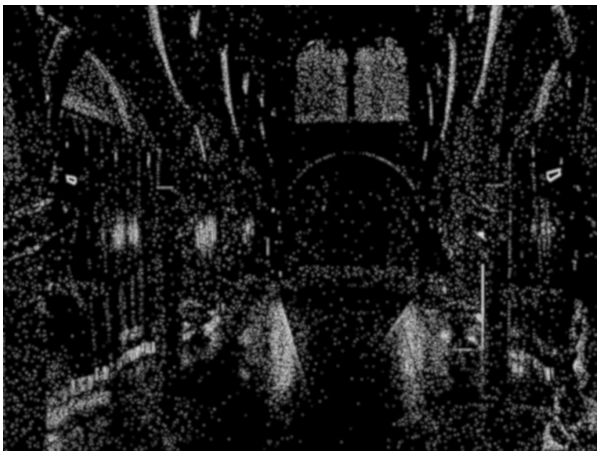
6.2.2 Cross-Bilateral mit und ohne Varianz

Da die cross-bilateralen Filter mit und ohne Varianz verschiedene Stärken und Schwächen haben, kann auch versucht werden den jeweils besseren Pixel aus beiden Bildern mit SURE auszuwählen. Es gibt aber keine Zwischenstufen zwischen den beiden Filtern, deshalb sind an manchen Stellen Ränder erkennbar. Auch wird nicht zuverlässig der Pixel ausgewählt, der auch näher an die Referenz herankommt, was besonders bei verrauschten Schatten auffällt, welche zwar beim cross-bilateralen mit Varianz geglättet werden aber nicht bei

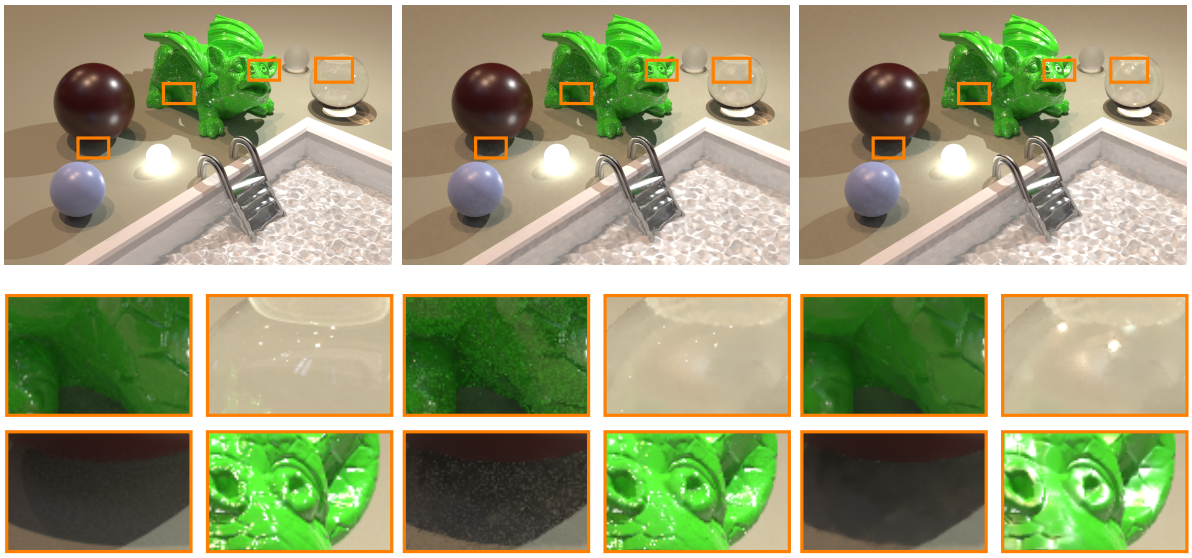


(a) Referenzbild **(b)** Sure c. b. **(c)** Sure c. b. Var.

(d) Filtergrößen Sure c. b.

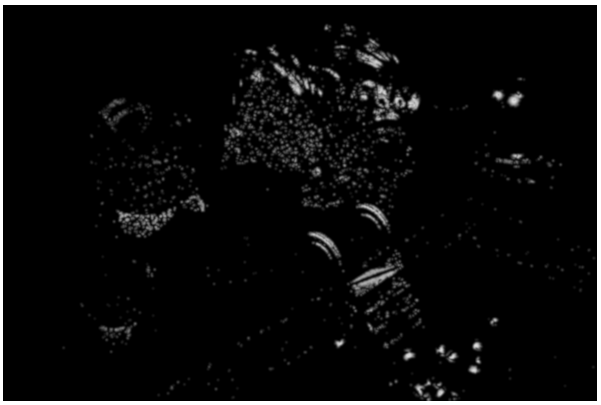


(e) Filtergrößen Sure c. b. Var.



(f) Referenzbild **(g)** Sure c. b. **(h)** Sure c. b. Var.

(i) Filtergrößen Sure c. b.



(j) Filtergrößen Sure c. b. Var.

Abbildung 23: SURE gefilterte Szenen und Filtergrößen, je heller der Pixel desto kleiner der Filter.

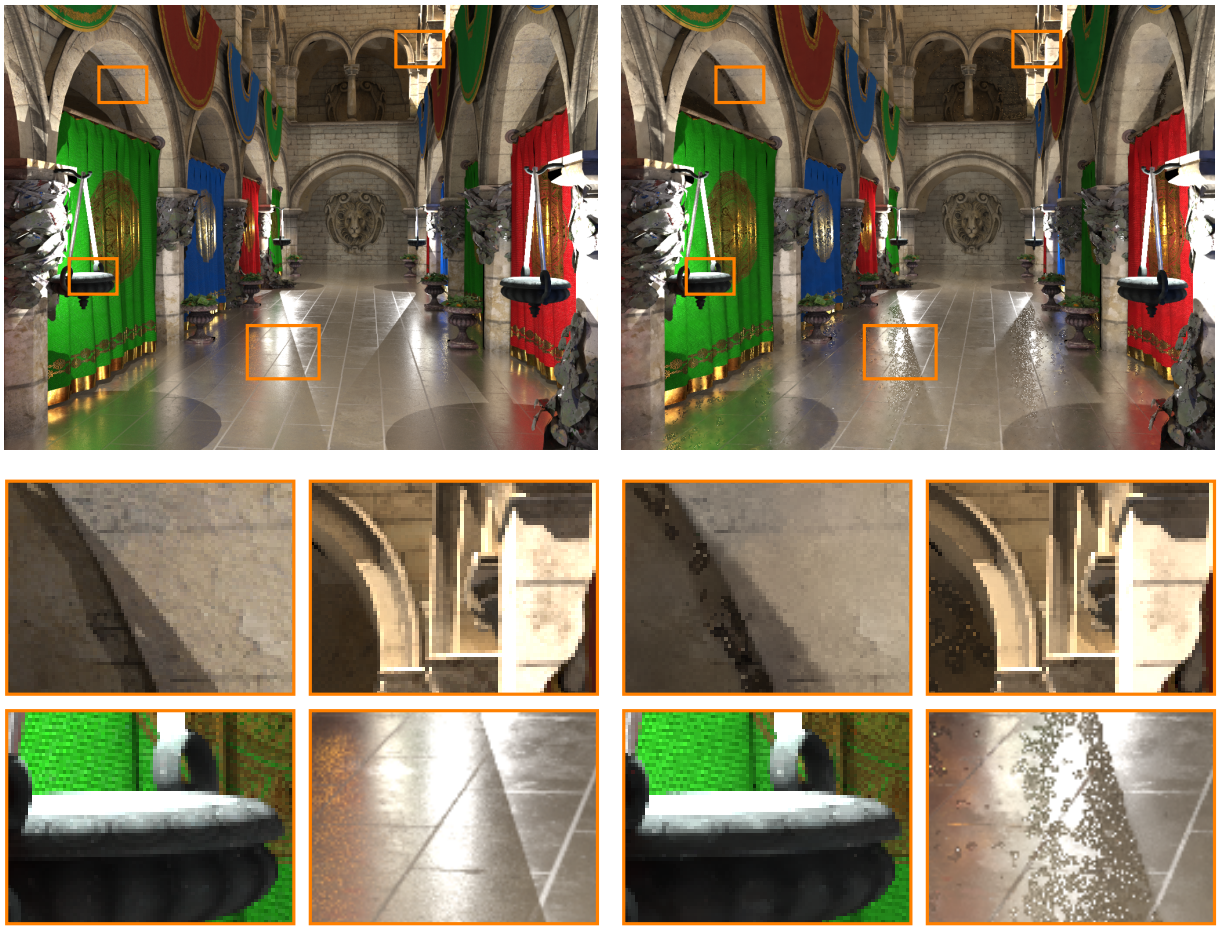
Tabelle 5: SSIM-Werte der SURE Filter.

Szene	Bad	Drachenpool	Sphären	Sponza	Villa
Cross-Bilateral	0.89881	0.965186	0.993892	0.881371	0.990478
Cross-Bilateral Var.	0.911225	0.957531	0.99368	0.971107	0.991598
SURE c. b.	0.900201	0.965282	0.993911	0.884486	0.990848
SURE c. b. Var	0.911365	0.958402	0.993672	0.970872	0.991696
SURE c. b. mit und ohne Var.	0.910632	0.96483	0.99388	0.943536	0.991341

dem ohne Varianz. An diesen Stellen ist der SURE beim letzteren geringer zu sein und es werden eher die Pixel von diesen genommen, wie in Abbildung 24 (e) auf der Vergrößerung unten links zu erkennen ist. Die gefilterten Ergebnisse sind in Abbildung 24 abgebildet.

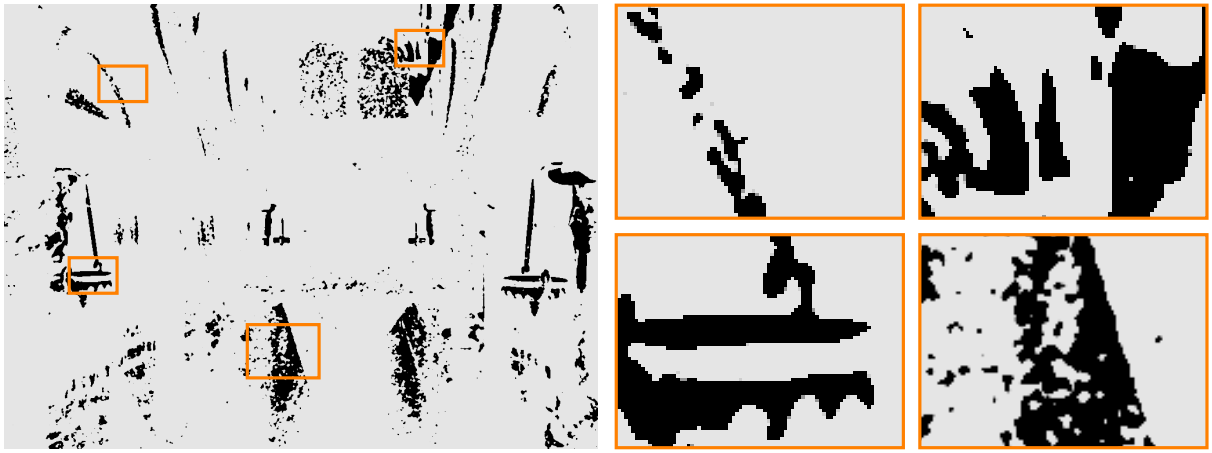
6.3 Fazit zum SURE basierten Filtern

Der SURE basierte Filter führt laut Tabelle 5 zwar in vielen Fällen zu einer Verbesserung, jedoch ist diese nur sehr klein. Aus diesem Grund lohnt sich der zusätzliche Aufwand, der gemacht werden muss, oft nicht und es kann darauf verzichtet werden. Die kombinierte Variante liefert zwar SSIM-Werte zwischen den beiden Varianten, jedoch kann sie optisch nicht überzeugen, siehe Abbildung 24, und es ist deshalb nicht sinnvoll diese zu verwenden.

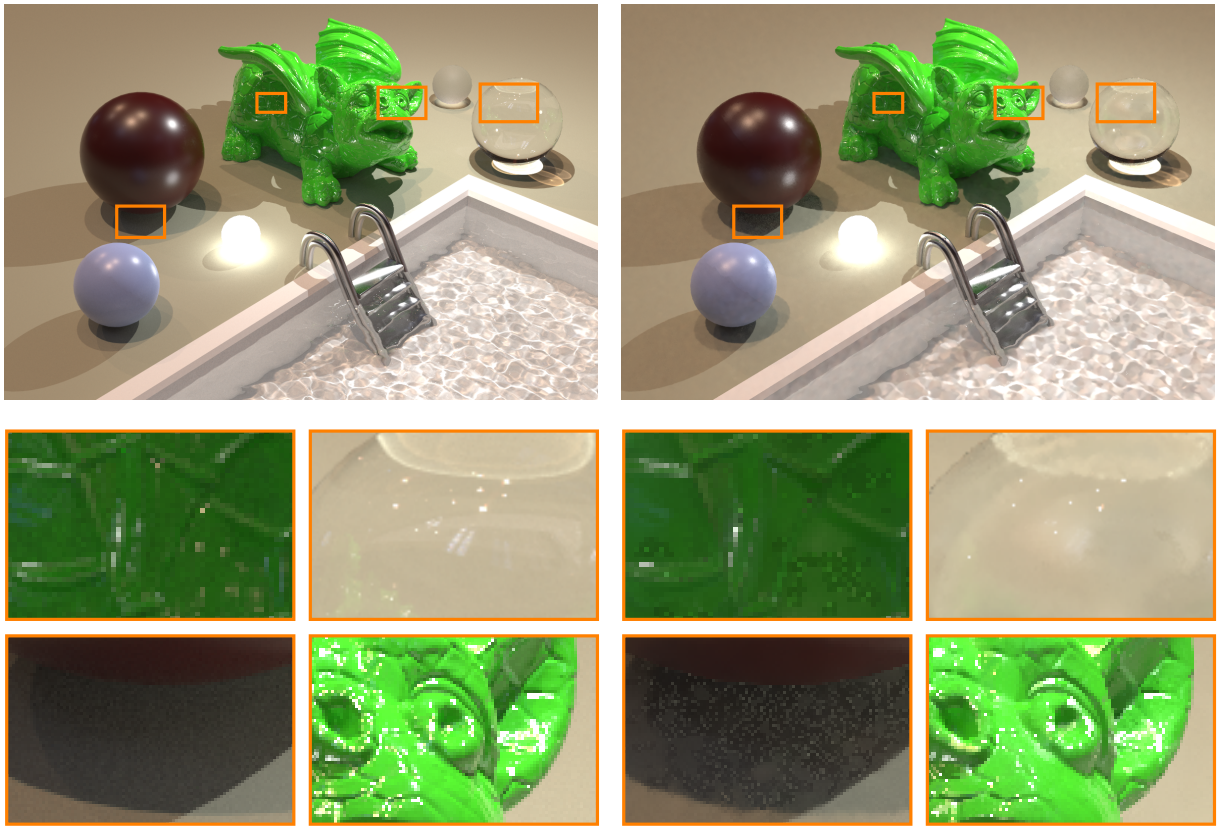


(a) Referenzbild

(b) Sure c. b. + c. b. Var.

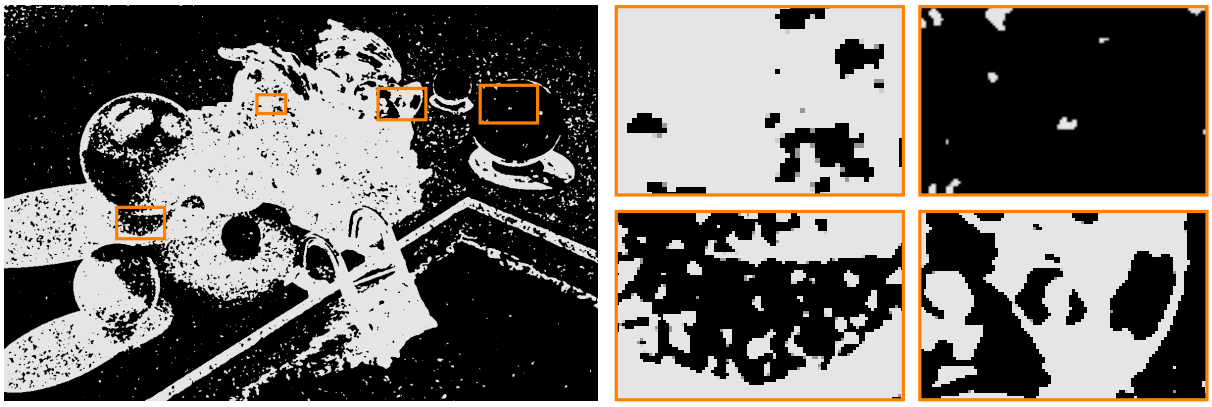


(c) Filterauswahl



(d) Referenzbild

(e) Sure c. b.



(f) Filterauswahl

Abbildung 24: SURE gefilterte Szenen und Filterauswahl, schwarz ist ohne und Grau mit Varianz.

7 Fazit

Es wurden der Guided, cross-bilaterale und SURE Filter zur Entfernung von Rauschen in Monte-Carlo gerenderten Bildern getestet und neue Guides für den cross-bilateralen Filter vorgestellt. Der Guided Filter, welcher mit einem lokalen linearen Modell arbeitet, hat Probleme nahe an Kanten und arbeitet deswegen am besten mit kleinen Filterradien, aus diesem Grund wurde er schnell verworfen.
Der cross-bilaterale Filter und die neuen Guides wurden ausgiebig getestet, was zu dem Schluss führt, dass die Verwendung von allen vorgestellten Guides und der Varianz oft zu guten Ergebnissen führt. Fireflys und viel Rauschen werden zuverlässig entfernt. Es gibt dabei aber Probleme, wenn Glanzpunkte vorhanden sind. Auch Kaustiken werden nicht allzu gut erhalten. Wenn die Varianz nicht verwendet wird, wird weniger Rauschen entfernt aber dafür Glanzpunkte besser erhalten und Kaustiken sind weniger verschwommen. Die neuen Guides, vor allem der Texturen- und Licht-Guide haben zu wesentlichen Verbesserungen bei der Erhaltung von Kanten geführt. Dabei ist anzumerken, dass die Kantenerhaltung durch die neuen Guides besonders bei der Verwendung von Varianz stark dazu beigetragen hat, dass diese sehr gut funktionieren. Wenn einzelne Guides weggelassen werden, führt das zu einer stärkeren Verschlechterung, als bei der Variante ohne Varianz. Das liegt unter anderen daran, dass durch die Verwendung von Varianz das Ausgangsbild weniger stark gewichtet wird und deshalb Texturen und Schatten weniger erhalten werden. Die in Tabelle 4 gezeigten Parameter für die Guides haben dabei zuverlässig für alle verwendeten Szenen gute Ergebnisse geliefert.
Der SURE Filter hat beide Varianten nur geringfügig verbessert und eine Kombination von beiden funktioniert auch nicht optimal. Deswegen wurde entschieden keine weiteren Experimente mit diesen anzustellen.

7.1 Zukünftige Arbeiten

Auch wenn die hier entwickelten Filtermethoden schon gut sind, gibt es noch weitere Verbesserungsmöglichkeiten, die in Zukunft angegangen werden können. So kann versucht werden eine Heuristik zu finden, um Glanzlichter zu erkennen und damit die Filtervariante mit Varianz zu verbessern. Erste Versuche dazu haben schon stattgefunden und es

wurde geschafft, diese in einigen Szenen besser zu erhalten, jedoch führt das in anderen Szenen zu Artefakten, deswegen ist dieser Guide hier nicht aufgeführt. Aber, wenn eine zuverlässigere Heuristik dafür gefunden werden kann, ist es möglich, dass die Variante mit Varianz der Variante ohne Varianz nur noch in sehr wenigen Fällen unterlegen ist.
Auch ist es denkbar eine bessere Heuristik zu finden, die besten Pixel aus beiden Varianten auszuwählen und auch Zwischenstufen zwischen den beiden zu schaffen, damit die kombinierte Variante weniger starke Kanten enthält, wenn zwischen ihnen gewechselt wird.
Es kann auch versucht werden, die Heuristik für den Licht-Guide so zu verbessern, dass dieser weniger Rauschen enthält und nicht mehr gefiltert werden muss. Dadurch würde die Erhaltung von Kaustiken und Schattenkanten besser funktionieren.
Eine weitere Möglichkeit wäre es, anstatt von einfachen und kumulativen Guides, Guides für jede Pfadlänge zu erstellen und die einzelnen Ergebnisse der Pfadlängen zu filtern und am Ende zu einem Bild zusammenzufügen. Das würde allerdings zu Speicherproblemen beim Rendern führen, da so sehr viele Guides parallel errechnet werden müssen.

7.2 Ähnliche Arbeiten

Es gibt mehrere Arbeiten, die sich mit der Entfernung von Rauschen in Bildern befassen. Darunter fällt auch das *Nonlocal Image Guided Averaging* von J. Zhang et al. [ZCW14]. Dort wird der Guided Filter so angepasst, dass die hier aufgefallenen negativen Effekte nicht mehr Auftreten. Dafür wird ein nichtlokales lineares Modell verwendet. Der Filter ist damit den hier verwendeten Guided Filter überlegen. In Abbildung 3 aus diesem Paper sieht man aber auch, dass Texturen Details verloren gehen. Die Filtermethode könnte damit potentiell von hier vorgestellten Texturen-Guide profitieren.
Eine weitere Methode ist *Spatiotemporal Variance-Guided Filtering* von C. Shied et al. [SKW+17], in welcher aus einem räumlich-zeitlichen Stream von Bildern mit jeweils einer Iteration in Echtzeit versucht wird, das Rauschen zu entfernen. Dafür werden die vorherigen Bilder mitbetrachtet, um die effektive Anzahl an Iteration zu erhöhen und die räumlich-zeitliche Varianz der Luminanz geschätzt, um einen hierarchischen Wavelet-Filter zu steuern.
In *An Efficient Denoising Algorithm for Global Illumination* von M. Mara et al. [MMBJ17] wird ein System zum Entrauschen vorgestellt, welches auf eine faktorisierte Approxima-

tion eines Material-basierten Monte-Carlo Integrators aufbaut. Dafür wird dieser unter anderem in einen Teil für glänzendes und einen Teil für mattes Material aufgeteilt. Dieses System arbeitet auch mit einem räumlich-zeitlichen Stream von Bildern mit jeweils einer Iteration und in Echtzeit. Der Material-basierte Teil ist dabei ähnlich zu den hier verwendeten Texturen-Guide, jedoch haben sie Probleme damit, dass Reflexionen zu stark geglättet werden. Die hier vorgestellte kumulative Guide Methode kann dabei helfen dieses Problem zu lösen.

Falls weiteres Interesse an dem Themengebiet besteht wird hier auf *Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering* von M. Zwicker et al. [ZJL+15] verwiesen. Darin sind verschiedenen Algorithmen zu dem Thema gesammelt.

Danksagung

Ich möchte mich bei meinem Betreuer Johannes Jendersie für das Bereitstellen der Monte-Carlo gerenderten Bilder und Guides bedanken. Auch für seine weitere Unterstützung und das Korrekturlesen bin ich ihm sehr dankbar. Für das Korrekturlesen möchte ich mich zusätzlich noch bei Dr. Tobias Paetz und Felix Brüll bedanken.

Literatur

[HST13] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, Juni 2013.

[Kaj86] James T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, pages 143–150. ACM, 1986.

[LWC12] Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. Sure-based optimization for adaptive sampling and reconstruction. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2012)*, 31(6):186:1–186:9, November 2012.

[MMBJ17] Michael Mara, Morgan McGuire, Benedikt Bitterli, and Wojciech Jarosz. An efficient denoising algorithm for global illumination. In *Proceedings of High Performance Graphics*. ACM, Juli 2017.

[NRH+92] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Radiometry. chapter Geometrical Considerations and Nomenclature for Reflectance, pages 94–145. Jones and Bartlett Publishers, Inc., 1992.

[Ren] What is renderman? `https://renderman.pixar.com/about`. Zugriff am: 20.09.2018.

[SKW+17] Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*, HPG '17, pages 2:1–2:12. ACM, 2017.

[Ste81] Charles M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, November 1981.

[Vea98] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, 1998. AAI9837162.

[WBSS04] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[ZCW14] J. Zhang, Y. Cao, and Z. Wang. A new image filtering method: Nonlocal image guided averaging. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2460–2464, Mai 2014.

[ZJL+15] Matthias Zwicker, Wojciech Jarosz, Jaakko Lehtinen, Bochang Moon, Ravi Ramamoorthi, Fabrice Rousselle, Pradeep Sen, Cyril Soler, and Sung-Eui Yoon. Recent advances in adaptive sampling and reconstruction for monte carlo rendering. *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports)*, 34(2):667–681, Mai 2015.