|  |
| --- |
| Rendering Denoising  Path Tracing Denoising mit neuronalen Netzwerken  **Bachelorthesis**  [Kurztext einfüngen, falls gewünscht] |
| Studiengang: CPCVR  Autor: Pascal Cornu  Betreuer: Marcus Hudritsch  Experten: Harald Studer  Datum: 07.04.2025 |

Management Summary

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit.

Inhaltsverzeichnis

[Path Tracing Denoising mit neuronalen Netzwerken 1](#_Toc194867298)

[Management Summary 2](#_Toc194867299)

[Inhaltsverzeichnis 3](#_Toc194867300)

[1 Einleitung 4](#_Toc194867301)

[2 Projektplanung 5](#_Toc194867302)

[2.1 Projektziele 5](#_Toc194867303)

[2.2 Meilensteine 5](#_Toc194867304)

[2.2.1 Planung 5](#_Toc194867305)

[2.2.2 Forschung 5](#_Toc194867306)

[2.2.3 Dataset 5](#_Toc194867307)

[2.2.4 Modell 5](#_Toc194867308)

[2.2.5 Präsentation 5](#_Toc194867309)

[2.3 Projektablauf 6](#_Toc194867310)

[3 Path Tracing 7](#_Toc194867311)

[3.1 Die Rendering-Gleichung 7](#_Toc194867312)

[*3.2* Monte-Carlo-Integration beim Path Tracing 7](#_Toc194867313)

[3.3 Der Path Tracing Algorithmus 8](#_Toc194867314)

[*3.4* Das Rauschen 9](#_Toc194867315)

[4 Abbildungsverzeichnis 11](#_Toc194867316)

[5 Tabellenverzeichnis 11](#_Toc194867317)

[6 Glossar 11](#_Toc194867318)

[7 Literaturverzeichnis 11](#_Toc194867319)

[8 Anhang 12](#_Toc194867320)

[9 Selbständigkeitserklärung 13](#_Toc194867321)

# Einleitung

Path Tracing ist eine physikalisch genaue Rendering-Technik, die in der Computergrafik weit verbreitet ist, um das Verhalten von Licht zu simulieren. Sie ist in der Lage, äußerst realistische Bilder zu erzeugen, indem sie die Pfade einzelner Lichtstrahlen bei der Interaktion mit Oberflächen in einer Szene nachzeichnet. Diese Methode ist jedoch rechenintensiv und anfällig für Rauschen, insbesondere in Szenarien mit komplexen Beleuchtungseffekten wie globaler Beleuchtung, Kaustik und indirekter Beleuchtung.

Die Hauptquelle des Rauschens beim Path Tracing liegt in der stochastischen Natur der Monte-Carlo-Integration, die zur Annäherung an die Rendering-Gleichung verwendet wird. Da die Farbe jedes Pixels durch Mittelwertbildung aus einer endlichen Anzahl zufälliger Lichtpfade geschätzt wird, führt eine geringe Anzahl von Stichproben zu einer hohen Varianz, die sich im endgültigen Bild als körniges oder fleckiges Rauschen äussert. Eine Erhöhung der Anzahl der Abtastwerte pro Pixel kann das Rauschen zwar verringern, doch ist dieser Ansatz aufgrund der exponentiell ansteigenden Rendering-Zeit oft nicht praktikabel.

Um dieses Problem zu lösen, wurden Entrauschungstechniken entwickelt, die qualitativ hochwertige Bilder mit deutlich weniger Stichproben erzeugen. Diese Entrauschungsverfahren nutzen statistische, analytische und auf maschinellem Lernen basierende Methoden, um ein sauberes Bild aus einem verrauschten Eingangssignal zu rekonstruieren. Herkömmliche Entrauschungsmethoden beruhen auf Filtertechniken, die verrauschte Regionen glätten und dabei wichtige Details erhalten, während moderne, auf Deep Learning basierende Entrauschungsmethoden neuronale Netzwerke verwenden, die auf grossen Datensätzen trainiert wurden, um Rauschen effektiv vorherzusagen und zu entfernen. Durch die intelligente Unterscheidung von Rauschen und echten Details ermöglichen diese Verfahren schnellere Rendering-Workflows und machen Path Tracing für Echtzeitanwendungen wie interaktives Rendering und Spielgrafik möglich.

Diese Arbeit erforscht die Rauschunterdrückung beim Path Tracing mit Verwendung von neuronalen Netzwerken.

# Projektplanung

## Projektziele

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabellenkopf** |  |  |  |
| Path Tracing und Denoising erforschen | Erarbeitung eines tiefen Verständnisses von Path Tracing.  Untersuchung von Rauschquellen, bestehenden Rauschunterdrückungstechniken und neuronalen Netzen zur Rauschunterdrückung. | | |
| Implementierung eines einfachen Path Tracers | Mit dem erlernten Wissen aus der Forschung soll entweder ein eigener Path Tracer erstellt werden, oder den Path Tracer aus dem SLProject4 oder SmallPT übernehmen und auf die Bedurfnisse anpassen. | | |
| Implementierung eines Denoisers | Für den Path Tracer soll ein Denoiser für die Rauschunterdrückung erstellt werden. Dazu gehört die Auswahl oder Erstellung eines Datasets für das Trainieren vom Dataset. | | |
| Denoiser Vergleich | Den Denoiser mit Nvidias Optix Denoiser oder den verschiedenen Methoden in Blender vergleichen. | | |

Tabelle : Projektziele

## Meilensteine

### Planung

* Meetings mit dem Betreuer sind festgelegt.
* Projektziele, so wie Meilensteine sind erstellt.
* Projektablauf ist festgelegt.

### Forschung

* Wissen über Path Tracing und Denoising ist erlangt und dokumentiert.
* Der Path Tracer im SLProject4 wurde analysiert und die Fehler so weit möglich behoben.

### Dataset

* Öffentliche Datasets für Denoising wurden erkundet.
* Ein passendes Dataset wurde ausgewählt, falls möglich.
* Existiert kein passendes öffentliches Dataset, wurde ein eigenes Dataset erstellt.

### Modell

* Ein Modell wurde erstellt. Es übt anhand des erstellten Datasets. Das Modell kann neue Bilder akzeptieren und gibt als Resultat eine entrauschte Version desselben Bildes zurück.
* Falls mehrere Arten von Modellen sich als sinnvoll erweisen, wurden zwei ausgewählt und ein Modell pro Ansatz erstellt und trainiert.

### Präsentation

* Dokumentation, Video, Plakat und Präsentation wurden erstellt und finalisiert.
* Die Präsentation ist geübt und die Ausstellung ist vorbereitet.

## Projektablauf

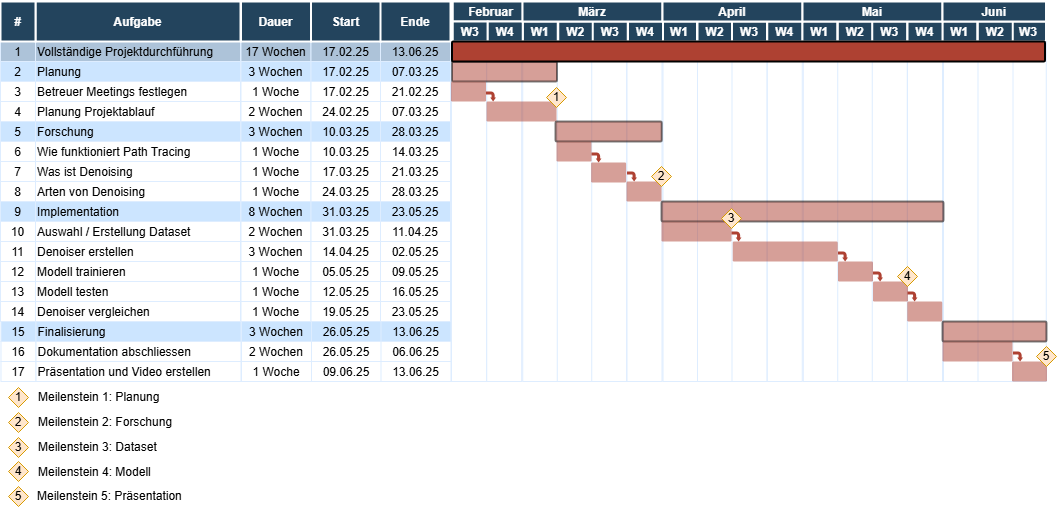


Abbildung : Projektablauf Gantt Diagramm

# Path Tracing

Path Tracing ist ein globaler Beleuchtungs-Rendering-Algorithmus, der die Physik des Lichttransports genau simuliert. Im Gegensatz zu rasterbasierten Methoden, die sich auf Heuristiken stützen, um Beleuchtungseffekte anzunähern, löst Path Tracing die Rendering-Gleichung direkt durch die Verfolgung von Lichtpfaden auf physikalische Weise. Aufgrund der Komplexität der Lichtinteraktionen in realen Szenen ist es nicht möglich, die Rendering-Gleichung analytisch zu lösen. Stattdessen kommt beim Path Tracing die Monte-Carlo-Integration zum Einsatz, ein statistisches Verfahren, das die Gleichung mithilfe von Zufallsstichproben approximiert.

## Die Rendering-Gleichung

Die Rendering-Gleichung, die von Kajiya (1986) eingeführt wurde, beschreibt den Transport von Licht in einer Szene:

* ist die ausgehende Strahldichte (radiance) am Punkt x in Richtung .
* ist die emittierte Strahldichte (radiance) von x, zum Beispiel von einer Lichtquelle.
* ist die bidirektionale Reflexionsverteilungsfunktion (BRDF), die beschreibt, wie das Licht gestreut wird.
* ist die einfallende Strahldichte (radiance) aus der Richtung .
* berücksichtigt den Kosinus des Einfallswinkels und sorgt für eine korrekte Energieerhaltung.
* Das Integral summiert die Beiträge aus allen einfallenden Lichtrichtungen über die Hemisphäre .

Diese Gleichung modelliert, wie Licht mit Oberflächen interagiert. Aber sie direkt für jedes Pixel zu lösen, ist aufgrund ihrer rekursiven Natur unpraktisch. Licht, das an einem Punkt ankommt, kann mehrfach zurückgeworfen worden sein, bevor es die Kamera erreicht.

## Monte-Carlo-Integration beim Path Tracing

Die Monte-Carlo-Integration bietet eine praktische Möglichkeit zur Annäherung an die Rendering-Gleichung, indem eine endliche Anzahl von Stichproben ausgewertet wird, anstatt das vollständige Integral analytisch zu berechnen. Die Monte-Carlo-Approximation für ein Integral der Form

wird gegeben durch:

wobei:

* ist die Anzahl der Proben.
* sind zufällig ausgewählte Punkte in der Domäne .
* ist die Wahrscheinlichkeitsdichtefunktion (PDF), für die Stichprobe .

Durch Anwendung auf die Rendering-Gleichung schätzt der Path-Tracing-Algorithmus die ausgehende Strahldichte, indem er Lichtpfade abtastet und deren Beiträge mittelt:

wobei die BRDF als Wichtigkeitssamplingfunktion verwendet wird, um die Varianz zu verringern und die Konvergenz zu verbessern.

## Der Path Tracing Algorithmus

Path Tracing folgt einem stochastischen Ansatz zur Annäherung an die globale Beleuchtung:

1. Strahlenerzeugung: Für jedes Pixel wird ein Primärstrahl von der Kamera in die Szene geworfen.
2. Strahlenschnittpunkt: Der erste Schnittpunkt mit einer Oberfläche wird gefunden.
3. Schattierung und BRDF-Sampling: Die Oberflächen-BRDF bestimmt, wie das Licht reflektiert oder durchgelassen wird.
4. Rekursive Pfadverfolgung: Eine neue Richtung wird auf der Grundlage der BRDF abgetastet, und der Prozess wird fortgesetzt, bis eine Abbruchbedingung erfüllt ist. Zum Beispiel das Erreichen einer Lichtquelle oder maximaler Tiefe.
5. Monte-Carlo-Schätzung: Die akkumulierte Strahldichte entlang des Pfades wird gemittelt, um die Pixelfarbe zu schätzen.

Jeder Pfad ist eine Monte-Carlo-Stichprobe der Rendering-Gleichung. Bei einer ausreichenden Anzahl von Pfaden konvergiert die Annäherung zur richtigen Lösung.

Die Monte-Carlo-Integration ist zwar unvoreingenommen, konvergiert aber langsam und erfordert Tausende von Stichproben pro Pixel, um rauschfreie Ergebnisse zu erzielen. Niedrige Stichprobenzahlen führen zu Monte-Carlo-Rauschen, das als körnige Artefakte sichtbar wird. Dies ist die Hauptmotivation für Rauschunterdrückungstechniken, die darauf abzielen, qualitativ hochwertige Bilder aus verrauschten Renderings mit geringer Abtastung zu rekonstruieren.

## Path Tracing Rauschen

Das durch Path Tracing erzeugte Rauschen hat deutliche Merkmale, die es vom natürlichen Bildrauschen unterscheiden. Es ist das Ergebnis des Monte-Carlo-Samplings, bei dem die Farbe jedes Pixels anhand einer begrenzten Anzahl von zufälligen Lichtpfaden geschätzt wird. Dies führt zu einer hochfrequenten Varianz, die als gesprenkelte oder körnige Muster erscheint, insbesondere in Bereichen mit komplexer Beleuchtung wie indirekter Beleuchtung, Kaustik und weichen Schatten. Im Gegensatz zu natürlichem Rauschen, wie etwa dem Sensorrauschen von Digitalkameras, das in der Regel zufällig, additiv und statistisch gleichmäßig über das Bild verteilt ist, ist das Rauschen durch Path Tracing stark strukturiert und szenenabhängig. Seine Intensität und Verteilung variiert mit der Anzahl der Proben pro Pixel, den Oberflächenmaterialien, den Lichtverhältnissen und der geometrischen Komplexität. Natürliches Rauschen ist in der Regel unabhängig vom Bildinhalt, während das Rauschen durch Path Tracing stark mit den visuellen und physikalischen Eigenschaften der Szene korreliert.



Abbildung : Verrauschtes Bücherregal mit Blender (Cycles)

Schlussfolgerungen/Fazit

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit.

# Abbildungsverzeichnis

[Abbildung 1: Projektablauf Gantt Diagramm 6](#_Toc194865737)

[Abbildung 2: Verrauschtes Bücherregal mit Blender (Cycles) 9](#_Toc194865738)

# Tabellenverzeichnis

[Tabelle 1: Projektziele 5](#_Toc194865739)

# Glossar

**Auinweon**

Et ut aut isti repuditis qui ium 7

**Batnwpe**

Et ut aut isti repuditis qui ium 9

**Cowoll**

Et ut aut isti repuditis qui ium 11

# Literaturverzeichnis

**Literatureintrag**

*Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr* 7

**Literatureintrag**

*Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr* 9

**Literatureintrag**

*Autorname, Autorvorname, Buchtitel, Verlag, Ort, Ausgabe, Jahr* 11

# Anhang

Et ut aut isti repuditis qui ium nonsecturia quis incientiae laborem elliquis et quatur, sitiur aut od moluptatur aut ea conseque peri sim erro essequisit remporia dem et landi dest, cone poris quunt volecab ipidero quatur ad quibusamus.

# Selbständigkeitserklärung

Ich bestätige, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Sämtliche Textstellen, die nicht von mir stammen, sind als Zitate gekennzeichnet und mit dem genauen Hinweis auf ihre Herkunft versehen.

Ort, Datum:

Unterschrift: