



Volume ray casting — basics & principles

Projektarbeit 1

MTE7101

Studiengang: Informatik
Autor: Sven Osterwalder¹
Betreuer: Prof. Claude Fuhrer²
Datum: 4. Oktober 2015



Licensed under the Creative Commons Attribution-ShareAlike 3.0 License

¹sven.osterwalder@students.bfh.ch

²claudio.fuhrer@bfh.ch

Versionen

Version	Datum	Status	Bemerkungen
0.1	25.09.2015	Entwurf	Initiale Erstellung des Dokumentes

Todo list

Describe scope	3
Describe motivation	3
Loose some words about demoscene!	3
Describe initial situation	3
Describe objectives	3
Describe preliminaries	3
Describe new learning contents	3
Describe theoretical background	5
view plane	7
Since illumination returned to the viewer is deter- mined by a tree of “rays,” a ray tracing algorithm is ideally suited to this model. In an obvious approach to ray tracing, light rays emanating from a source are traced through their paths until they strike the viewer. Since only a few will reach the viewer, this approach is waste- ful. In a second approach suggested by Appel [1] and used successfully by MAGI [14], rays are traced in the opposite direction—from the viewer to the objects in the scene, as illustrated in Figure 4. Unlike previous ray tracing algorithms, the visibility calculations do not end when the nearest intersection of a ray with objects in the scene is found. Instead, each visible intersection of a ray with a surface produces more rays in the \vec{d} direction, the $\vec{r}/5$ direction, and in the direction of each light source. The intersection process is repeated for each ray until none of the new rays intersects any object.	10
Describe implicit surfaces	10

Management Summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus scelerisque, leo sed iaculis ornare, mi leo semper urna, ac elementum libero est at risus. Donec eget aliquam urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc fermentum nunc sollicitudin leo porttitor volutpat. Duis ac enim lectus, quis malesuada lectus. Aenean vestibulum suscipit justo, in suscipit augue venenatis a. Donec interdum nibh ligula. Aliquam vitae dui a odio cursus interdum quis vitae mi. Phasellus ornare tortor fringilla velit accumsan quis tincidunt magna eleifend. Praesent nisl nibh, cursus in mattis ac, ultrices ac nulla. Nulla ante urna, aliquet eu tempus ut, feugiat id nisl. Nunc sit amet mauris vitae turpis scelerisque mattis et sed metus. Aliquam interdum congue odio, sed semper elit ullamcorper vitae. Morbi orci elit, feugiat vel hendrerit nec, sollicitudin non massa. Quisque lacus metus, vulputate id ullamcorper id, consequat eget orci.

Inhaltsverzeichnis

Management Summary	i
1. Einleitung	1
2. Administratives	2
2.1. Beteiligte Personen	2
2.2. Aufbau des Dokumentes	2
2.3. Ergebnisse (Deliverables)	2
3. Aufgabenstellung	3
3.1. Motivation	3
3.2. Ausgangslage	3
3.3. Ziele und Abgrenzung	3
4. Vorgehen	4
4.1. Arbeitsorganisation	4
4.2. Projektphasen	4
4.3. Technologien	5
5. Theoretischer Hintergrund	6
5.1. Beleuchtungsmodelle	6
5.2. Ray Casting	8
5.3. Ray Tracing	10
5.4. Darstellung impliziter Oberflächen	10
6. Diskussion und Fazit	11
6.1. Diskussion	11
6.2. Erweiterungsmöglichkeiten	11
6.3. Fazit	11
Glossar	12
Literaturverzeichnis	13
Abbildungsverzeichnis	14
Tabellenverzeichnis	15
Auflistungsverzeichnis	16
Anhang	18
A. Meeting minutes	19

1. Einleitung

Seit dem Bestehen moderner Computer existiert auch die Computergrafik. Ziel der Computergrafik ist es unter Anderem den dreidimensionalen Raum auf eine zweidimensionale Fläche abzubilden, da die Ausgabe meist auf den zweidimensionalen Raum limitiert ist.

Dabei wird zwischen statischen Bildern und dynamischen Bildern unterschieden. Statische Bilder werden bei Bedarf dargestellt und ändern sich in der Regel nicht. Dynamische Bilder können sich hingegen ständig ändern und müssen — bedingt durch das menschliche Auge — mit 25 Bildern pro Sekunde ausgegeben werden. Es bestand bereits früh das Bestreben eine möglichst realistische Darstellung zu erhalten. Eine Darstellung also, die möglichst nahe an der menschlichen Wahrnehmung liegt.

Im Laufe der Zeit entstanden verschiedene Ansätze um eine solche Darstellung zu bieten. Ein Teilgebiet davon sind Beleuchtungsmodelle, welche die Beleuchtung einer Darstellung bzw. einer Szene berechnen. Dabei wird zwischen lokalen und globalen Beleuchtungsmodellen unterschieden.

Ein globales Beleuchtungsmodell ist Ray Tracing (zu deutsch Strahlenverfolgung), welches 1980 von Turner Whitted vorgestellt wurde. Das Verfahren besticht durch seine Einfachheit und bietet dabei eine hohe Bildqualität mit perfekten Spiegelungen und Transparenzen. Mit entsprechenden Optimierungen ist das Verfahren auch relativ schnell.

Mit schnell ist dabei die Zeit gemeint, die benötigt wird um ein einzelnes Bild darzustellen. Möchte man jedoch eine Darstellung in Echtzeit erreichen, so war das Verfahren lange zu langsam.

Im Rahmen der Weiterentwicklung der Computer und vor allem durch die Weiterentwicklung der Grafikkarten (GPUs), ist Ray Tracing jedoch wieder in den Fokus der Darstellung von Szenen in Echtzeit gerückt.

Diese Projektarbeit stellt ein spezielles Ray Tracing Verfahren zur Darstellung von Bildern in Echtzeit vor: Volume Ray Casting bzw. Sphere Tracing.

2. Administratives

Einige administrative Aspekte der Projektarbeit werden angesprochen, obwohl sie für das Verständnis der Resultate nicht notwendig sind.

Im gesamten Dokument wird nur die männliche Form verwendet, womit aber beide Geschlechter gemeint sind.

2.1. Beteiligte Personen

Autor Sven Osterwalder¹
Betreuer Prof. Claude Fuhrer²

Begleitet den Studenten bei der Projektarbeit

2.2. Aufbau des Dokumentes

Der Aufbau der vorliegenden Arbeit ist wie folgt:

- Einleitung zur Projektarbeit
- Beschreibung der Aufgabenstellung
- Vorgehen des Autors im Hinblick auf die gestellten Aufgaben
- Lösung der gestellten Aufgaben
- Verwendete Technologien

Der Schwerpunkt dieser Arbeit liegt in der Beschreibung der theoretischen Grundlagen (unter praktischen Aspekten) des Volume Ray Casting Verfahrens.

2.3. Ergebnisse (Deliverables)

Nachfolgend sind die abzugebenden Objekte aufgeführt:

- **Abschlussdokument**
Das Abschlussdokument beinhaltet die theoretischen Grundlagen (unter praktischen Aspekten) des Volume Ray Casting Verfahrens

¹sven.osterwalder@students.bfh.ch

²claudio.fuhrer@bfh.ch

3. Aufgabenstellung

Describe scope

3.1. Motivation

Describe motivation

3.1.1. Demoszene

Loose some words about demoscene!

3.2. Ausgangslage

Describe initial situation

3.3. Ziele und Abgrenzung

Describe objectives

3.3.1. Vorgängige Arbeiten

Describe preliminaries

3.3.2. Neue Lerninhalte

Describe new learning contents

4. Vorgehen

4.1. Arbeitsorganisation

4.1.1. Regelmässige Treffen

Regelmässige Besprechungen mit dem Betreuer der Arbeit halfen die gesteckten Ziele zu erreichen und Fehlentwicklungen zu vermeiden. Der Betreuer unterstützte den Autor dabei mit Vorschlägen. Die Treffen fanden mindestens alle zwei Wochen statt, sie wurden in Form eines Protokolles festgehalten.

4.2. Projekphasen

4.2.1. Meilensteine

Um bei der Arbeit ein möglichst strukturiertes Vorgehen einzuhalten, wurden folgende Projektphasen gewählt:

- Projektstart
- Erarbeitung und Festhalten der Anforderungen
- Erarbeitung der theoretischen Grundlagen
- Erstellung der abschliessenden Dokumentation

Die Phasen der Erarbeitung der theoretischen Grundlagen sowie die Erstellung der abschliessenden Dokumentation liefen parallel ab.

4.2.2. Zeitplan / Projektphasen

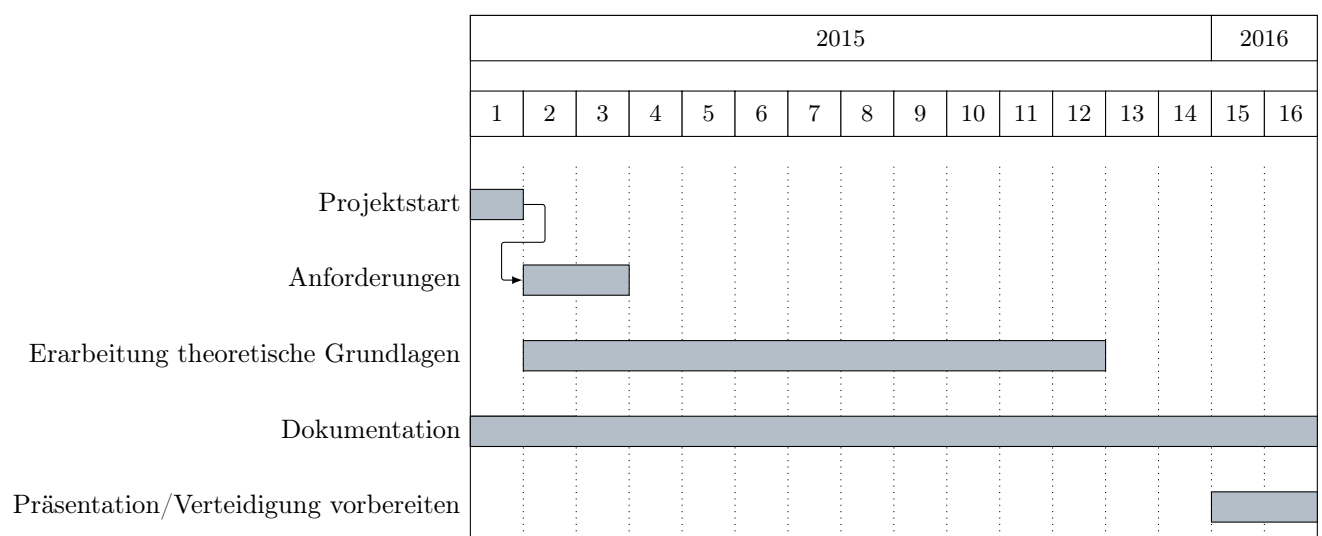


Abbildung 4.1.: Zeitplan; Der Titel stellt Jahreszahlen, der Untertitel Semesterwochen dar

Projektstart

In der ersten Phase wurden die Meilensteine der Arbeit identifiziert und skizziert. Um Details der Aufgabe zu verstehen, wurde das notwendige Vorwissen über globale Beleuchtungsalgorithmen erarbeitet. Weiter wurde das Grundgerüst dieser Dokumentation erstellt.

Anforderungen

In dieser Phase wurde das Ziel dieser Projektarbeit festgelegt. Vom Ziel ausgehend wurden die dazu erforderlichen Projektphasen festgelegt.

Erarbeitung theoretische Grundlagen

Describe theoretical background

Dokumentation

Die vorliegende Arbeit entspricht der Dokumentation. Sie wurde während der gesamten Projektarbeit stetig erweitert und diente zur Reflexion von fertiggestellten Teilen.

4.3. Technologien

4.3.1. Tools und Software

Dokumentation

L^AT_EX Eine Makro-Sammlung für das T_EX-System. Wurde zur Erstellung dieser Dokumentation eingesetzt. Diese Dokumentation wurde mittels L^AT_EX geschrieben.

Make Build-Automations-Werkzeug, wurde zur Erstellung dieses Dokumentes eingesetzt.

zotero Ein freies, quelloffenes Literaturverwaltungsprogramm zum Sammeln, Verwalten und Zitieren unterschiedlicher Online- und Offline-Quellen Wikipedia Foundation [2015].

VIM Vi IMproved. Ein freier, quelloffener Texteditor zur Textbearbeitung.

Arbeitsorganisation

Git Freie Software zur verteilten Versionsverwaltung, wurde für die Entwicklung dieser Dokumentation verwendet. Die Projektarbeit findet sich unter GitHub¹.

GitHub Eine freie Hosting-Plattform für Git mit Weboberfläche.

4.3.2. Standards und Richtlinien

Pseudocode

Da der Autor dieser Arbeit bedingt durch seine täglich Arbeit mit der Programmiersprache Python relativ bewandert ist, wird daher diese als Sprache zur Beschreibung von Pseudocode verwendet. Dabei wird aber kein Augenmerk auf die formale Korrektheit, geschweige denn der Lauffähigkeit des Pseudocodes gelegt.

¹<https://www.github.com/sosterwalder/mte7101-project1>

5. Theoretischer Hintergrund

5.1. Beleuchtungsmodelle

Sofern nicht anders vermerkt, basiert der folgende Abschnitt auf Whitted [1980][S. 343] sowie auf Hughes et al. [2013].

Beleuchtungsmodelle beschreiben, wieviel Licht von einem sichtbaren Punkt einer Oberfläche zum Betrachter emittiert wird. In der Regel wird das Licht als Funktion in Abhängigkeit folgender Faktoren beschrieben:

- Richtung der Lichtquelle
- Lichtstärke
- Position des Betrachters
- Orientierung der Oberfläche
- Oberflächenbeschaffenheit
- Globale Umgebung

Es wird dabei zwischen lokalen und globalen Beleuchtungsmodellen unterschieden.

5.1.1. Lokale Beleuchtungsmodelle

Lokale Beleuchtungsmodelle aggregieren Daten von benachbarten, eben lokalen, Oberflächen. Diese Modelle sind in deren Umfang allerdings limitiert, da sie normalerweise nur Lichtquellen sowie die Orientierung einer Oberfläche einbeziehen. Sie ignorieren dabei aber die globale Umgebung, in welcher sich eine Oberfläche befindet. Dies ist dadurch bedingt, dass die traditionell verwendeten Algorithmen zur Berechnung der Sichtbarkeit von Oberflächen, über keine globalen Daten verfügen.

Als Beispiel für ein lokales Beleuchtungsmodell dient das Phong-Beleuchtungsmodell, welches von Bui-Tong Phong entwickelt wurde. Es beschreibt die reflektierte (Licht-) Intensität als Zusammensetzung aus der ambienten, der diffusen und der ideal spiegelnden Reflexion einer Oberfläche:

$$I = I_{ambient} + I_{diffuse} + I_{specular} \quad (5.1)$$

oder mathematisch ausgedrückt:

$$I = I_a + k_d \sum_{j=1}^{ls} (\vec{N} \cdot \vec{L}_j) + k_s \sum_{j=1}^{ls} (\vec{N} \cdot \vec{L}_j) \quad (5.2)$$

wobei gilt:

- I : Die reflektierte (Licht-) Intensität
- I_a : Reflektion bedingt durch die Beleuchtung des Raumes
- k_d : Konstante für die diffuse Komponente des reflektierten Lichtes
- \vec{N} : Einheitsnormale der Oberfläche
- \vec{L}_j : Vektor in Richtung der j -ten Lichtquelle
- k_s : Koeffizient der spiegelenden Komponente

- \vec{L}_j : Vektor in der Hälfte zwischen dem Betrachter und der j -ten Lichtquelle
- n : Exponent, welcher von der Reflektion der Oberfläche abhängt
- ls : Anzahl Lichtquellen

5.1.2. Globale Beleuchtungsmodelle

Sofern nicht anders vermerkt, basiert der folgende Abschnitt auf Foley [1996][S. 775ff]

Globale Beleuchtungsmodelle beschreiben die reflektierte (Licht-) Intensität eines Punktes aufgrund direkter Lichteinstrahlung durch Lichtquellen sowie durch alles Licht, welches diesen Punkt nach Reflektion von bzw. Durchdringen der eigenen oder anderer Oberflächen erreicht.

Bei globalen Beleuchtungsmodellen unterscheidet man zwischen blickwinkelabhängigen Algorithmen, wie etwa Ray Tracing, und zwischen blickwinkelunabhängigen Algorithmen, wie etwa Photon Mapping.

Blickwinkelabhängige Algorithmen verwenden eine Diskretisierung der sichtbaren Fläche um zu entscheiden, an welchen Punkten, in Blickrichtung des Betrachters, die Beleuchtungsberechnung durchgeführt werden soll. Blickwinkelunabhängige Algorithmen hingegen diskretisieren und verarbeiten die Umgebung um genügend Informationen für die Beleuchtungsberechnung zu haben. Dies erlaubt ihnen die Beleuchtungsberechnung an einem beliebigen Punkt aus einer beliebigen Blickrichtung.

view plane

Beide Arten von Algorithmen haben jedoch Vor- und Nachteile. So sind blickwinkelabhängige Algorithmen gut geeignet um Spiegelungen, basierend auf der Blickrichtung des Betrachters, zu berechnen, eignen sich aber weniger um gleichbleibende diffuse Anteile über weiter Flächen eines Bildes zu berechnen. Bei blickwinkelunabhängigen Algorithmen verhält es sich genau umgekehrt.

Renderinggleichung

Die unter 5.1.2 genannten Verfahren versuchen auszudrücken, wie sich Licht von einem Punkt im Raum zu einem anderen bewegt. Dabei beschreiben sie die Intensität des Lichtes, ausgehend vom ersten Punkt zum zweiten Punkt. Zusätzlich wird die Intensität des Lichtes, ausgehend von allen anderen Punkten, welche den ersten Punkt erreichen, und zum zweiten Punkt emittiert werden, beschrieben.

James (Jim) Kajiya stellte 1986 die so genannte Renderinggleichung auf, welche genau dieses Verhalten beschreibt:

$$I(x, x') = g(x, x')[\epsilon(x, x') + \int_s \rho(x, x', x'') I(x', x'') dx''] \quad (5.3)$$

wobei gilt:

Tabelle 5.1.: Beschreibung der Komponenten der Renderinggleichung nach Kajiya [1986][S. 143]

x', x' und x'' :	Punkte in der Umgebung
$I(x, x')$:	Lichtintensität von Punkt x' nach Punkt x
$g(x, x')$:	Ein auf die Geometrie bezogener Term:
	0: x und x' verdecken sich
	$1/r^2$: x und x' sehen sich, wobei r die Distanz zwischen x und x' ist
$\epsilon(x, x')$:	Intensität des Lichtes, welches von x' nach x emittiert wird
$\rho(x, x', x'')$:	Intensität des Lichtes, welches von x'' durch die Oberfläche bei x' nach x gestreut wird
\int_s :	Integral über die Vereinigung aller Flächen, daher $S = \bigcup S_i$
	Dies bedeutet, dass die Punkte x , x' und x'' über alle Flächen aller Objekte der Szene "streifen".
	Wobei es sich bei S_0 um eine zusätzliche Fläche handelt, welche als Hintergrund verwendet wird.
	S_0 ist dabei eine Hemisphäre, welche die gesamte Szene umspannt.

5.2. Ray Casting

Sofern nicht anders vermerkt, basiert der folgende Abschnitt auf Hughes et al. [2013][Kapitel 15, S. 387ff].

Um ein Bild möglichst realistisch darzustellen muss berechnet werden, wieviel Licht zu jedem Pixel der sichtbaren Bildfläche (also dem Betrachter) transportiert wird. Da Photonen die Energie des Lichtes transportieren, muss man also das physikalische Verhalten dieser simulieren. Es ist allerdings nicht möglich *alle* Photonen zu simulieren, da der Aufwand schlicht zu gross wäre. Daher macht es Sinn nur einige Photonen (exemplarisch) zu betrachten und dann eine Abschätzung des gesamten Lichtes vorzunehmen.

Bei **Ray Casting** handelt es sich grundsätzlich um eine Strategie zur Simulation, wieviel Licht anhand eines (Licht-) Strahles zu der sichtbaren Bildfläche (also dem Betrachter) transportiert wird.

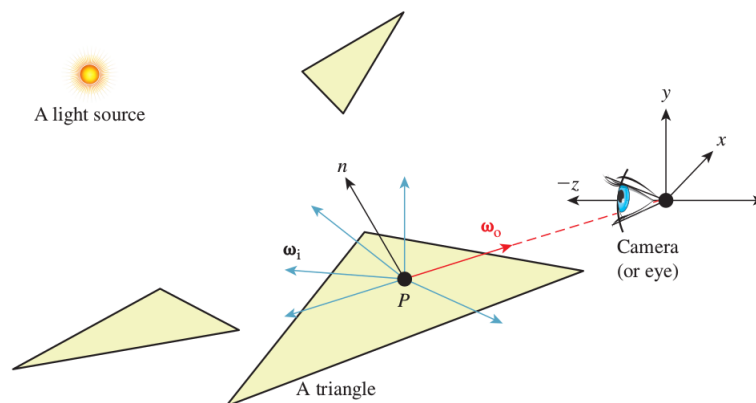


Abbildung 5.1.: Punkt P auf einer Oberfläche eines Dreiecks, welcher für die Kamera bzw. den Betrachter sichtbar ist. Der Betrachter nimmt dabei das Licht, welches aus verschiedenen Richtungen ω_i kommt, über den Punkt P in Richtung ω_o wahr.¹

Wie in Abbildung 5.1 ersichtlich, gelangt Licht aus vielen Richtungen durch den Punkt P zu dem Betrachter. Dies beinhaltet auch die Möglichkeit, dass Licht nicht nur von einer Lichtquelle aus, sondern von vielen Lichtquellen aus via P zum Betrachter gelangt. Weiter ist es möglich, dass Licht zuvor an anderen Punkten gestreut und/oder gespiegelt und erst dann via P zum Betrachter gelangte.

Dies führt zu den folgenden Schlussfolgerungen:

- Es müssen alle möglichen Richtungen, aus denen Licht kommen könnte, an Punkt P untersucht werden.
- Da, bedingt durch technische Limitierungen, nur diskretes Abtasten möglich ist, müssen die Richtungen auf eine endliche Anzahl beschränkt werden, was zu Abtastfehlern führen kann.

Um die Abtastfehler zu minimieren, können die Richtungen des Abtasten anhand der Lichtquellen priorisiert werden.

¹Darstellung von Hughes et al. [2013][Kapitel 15, Seite 389, Abbildung 15.1]

Ein möglicher Algorithmus, wie solch ein Verfahren umgesetzt werden kann, findet sich in 5.1.

Auflistung 5.1: *Eine abstrakte Umsetzung des Ray Castings.*²

```
def ray_cast():
    # "pixels" is a list of all pixels of the image plane
    for pixel in pixels:
        # Save all intersections for given pixel
        intersections = []

        # Returns the ray passing through the given
        # pixel from the eye
        ray = ray_at_pixel(pixel)

        # "scene_triangles" is a list of all triangles
        # coming from meshes contained in the scene to render
        for triangle in scene_triangles:
            p = intersect(ray, triangle)
            sum = 0

            for light in incoming_lights_at_p:
                sum = sum + l.value
            end

            if is_smallest_intersection(p, intersections):
                pixel = sum
            intersections.append(p)
```

²Algorithmus in Pseudocode gemäss Hughes et al. [2013][Kapitel 15, Seite 391, Auflistung 15.2]

5.3. Ray Tracing

Since illumination returned to the viewer is determined by a tree of “rays,” a ray tracing algorithm is ideally suited to this model. In an obvious approach to ray tracing, light rays emanating from a source are traced through their paths until they strike the viewer. Since only a few will reach the viewer, this approach is wasteful. In a second approach suggested by Appel [1] and used successfully by MAGI [14], rays are traced in the opposite direction—from the viewer to the objects in the scene, as illustrated in Figure 4. Unlike previous ray tracing algorithms, the visibility calculations do not end when the nearest intersection of a ray with objects in the scene is found. Instead, each visible intersection of a ray with a surface produces more rays in the \hat{d} direction, the $\hat{d}/5$ direction, and in the direction of each light source. The intersection process is repeated for each ray until none of the new rays intersects any object.

5.4. Darstellung impliziter Oberflächen

Describe implicit surfaces

6. Diskussion und Fazit

6.1. Diskussion

6.2. Erweiterungsmöglichkeiten

6.3. Fazit

Glossar

OWL Web Ontology Language; Ontologiesprache für das semantische Web. Mit dieser Sprache können Ontologien beschrieben werden..

Literaturverzeichnis

- Wikipedia Foundation. Zotero, August 2015. URL <https://de.wikipedia.org/wiki/Zotero>. Published: Website Abgerufen am 27. September 2015.
- Turner Whitted. An Improved Illumination Model for Shaded Display. *Commun. ACM*, 23(6):343–349, June 1980. ISSN 0001-0782. doi: 10.1145/358876.358882. URL <http://doi.acm.org/10.1145/358876.358882>.
- J.F. Hughes, A. Van Dam, J.D. Foley, and S.K. Feiner. *Computer Graphics: Principles and Practice*. The systems programming series. Addison-Wesley, 2013. ISBN 978-0-321-39952-6. URL <https://books.google.ch/books?id=OVpsAQAAQBAJ>.
- J.D. Foley. *Computer Graphics: Principles and Practice*. Addison-Wesley systems programming series. Addison-Wesley, 1996. ISBN 978-0-201-84840-3. URL <https://books.google.ch/books?id=-4ngT05gmAQC>.
- James T. Kajiya. The Rendering Equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, August 1986. ISSN 0097-8930. doi: 10.1145/15886.15902. URL <http://doi.acm.org/10.1145/15886.15902>.

Abbildungsverzeichnis

4.1. Zeitplan; Der Titel stellt Jahreszahlen, der Untertitel Semesterwochen dar	4
5.1. Punkt P auf einer Oberfläche eines Dreieckes, welcher für die Kamera bzw. den Betrachter sichtbar ist. Der Betrachter nimmt dabei das Licht, welches aus verschiedenen Richtungen ω_i kommt, über den Punkt P in Richtung ω_0 wahr. ¹	8

Tabellenverzeichnis

5.1. Beschreibung der Komponenten der Renderinggleichung nach Kajiya [1986][S. 143]	7
---	---

Auflistungsverzeichnis

5.1. Eine abstrakte Umsetzung des Ray Castings. ²	9
--	---

Anhang

A. Meeting minutes

20150921

No.: 01
Date: 21.09.2015 07:30 - 07:55
Place: Prof. Claude Fuhrer's office
Involved persons: Claude Fuhrer
Sven Osterwalder

- * Project issues
 - Requirement document needed?
 - * No, not directly
 - What are the requirements?
 - * Project schedule
 - * External inputs
 - * Conclusion
 - * Grading is analogous to bachelor thesis, so the requirements are the same
- * Goal
 - Read articles about the topic
 - Gain an understanding for the topic
 - Create a summary of read articles including small code segments in pseudo code, e.g. explaining an algorithm
- * Project 2 (MTE7102)
 - Building a software architecture regarding the master thesis
 - Proof of concept of the algorithms chosen in this project
- * Meetings
 - Will be held every 14 days
 - Time and location will be defined at the end of each meeting

TODO for next meeting:

- * Set up project repository
 - GitHub
 - Open source
- * Choose language for pseudo code

Next meeting:

Date: 04.10.2015 14:00
Place: Skype

No.: 02
Date: 04.10.2015 14:00 - 14:35
Place: Skype
Involved persons: Claude Fuhrer (CF)
Sven Osterwalder (SO)

* External documents

- Do external documents, e.g. papers, held in the project repository infringe copy rights? (CF)
 - * Both are not entirely sure about the copy rights, so it is decided to share the documents only between both persons via Dropbox (CF and SO)

* Theoretical background

- Phong equation
 - * Why is the half way vector used (as described in Whitted's paper)
instead of the more common usage of the angle (cosine) in direction of the light? (CF)
 - It is ok to use Whitted's originally proposed formula, as the results are the same for both formulas (CF)
 - Although the specular factor for the specular component is missing and has to be added (CF)
- Structuring / procedure
 - * Is the document structuring and are the plans for further development in good order? (SO)
 - The structuring of the document as well as the plans for further development are in a good order and the work may continue in the currently ongoing direction (CF)

* Literature

- Is it somehow possible to get the second edition of "Computer graphics: principles and practice"? (SO)
 - * Mr. Fuhrer possesses the mentioned book and proposes furthermore the lecture of a book dedicated to ray tracing which he also possesses. He will deposit both of the books on Monday, 5th of October 2015, in a room within the "Rolex" building of the Berne University of applied sciences in Biel (CF)

* Citations

- Is the current way of citing in good order or do citations need to be more precise? (SO)
 - * The way of citing is precise enough, the schemata is the following: (CF)
 - Source, [Chapter], Page(s)

* Document template

- Remove currently used font "cmbright" as invoked to the usage of the LaTeX template of the Berne University of applied sciences (CF)
- The lines shall be shortened so that they do not exceed 80 characters per line (CF)
- The margins, especially the left and the right margins, shall be enlarged as they are currently very narrow (CF)

TODO for next meeting:

- * Present the current state of the work
- * Discuss the current state of the work
- * Define further steps/proceeding

Next meeting:

Date: 11.10.2015 14:00
Place: Skype or in real life, if necessary

No.: 03
Date: 11.10.2015 14:00 - to be scheduled
Place: Skype/irl
Involved persons: Claude Fuhrer (CF)
Sven Osterwalder (SO)

- * Theoretical background
 - Ray casting
 - * Is the description of ray casting developed enough? (SO)
 - * Is the pseudo code of ray casting developed enough? (SO)

TODO for next meeting:

- * Present the current state of the work
- * Discuss the current state of the work
- * Define further steps/proceeding

Next meeting:

Date: To be scheduled
Place: Skype or in real life, if necessary
