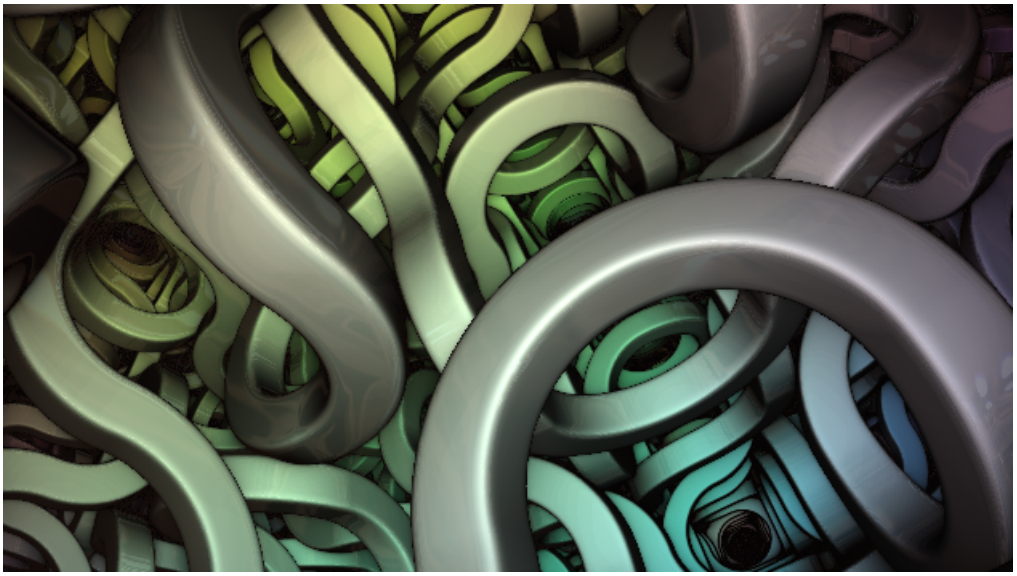

Volume ray casting — basics & principles

Projektarbeit 1

MTE7101

Studiengang:	Informatik
Autor:	Sven Osterwalder[1]
Betreuer:	Prof. Claude Fuhrer[2]
Datum:	11. Oktober 2015

[1] sven.osterwalder@students.bfh.ch
[2] claude.fuhrer@bfh.ch

Versionen

Version	Datum	Status	Bemerkungen
0.1	25.09.2015	Entwurf	Initiale Erstellung des Dokumentes

Todo list

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

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 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 Osterwalder[1]	
Betreuer	Prof. Claude Fuhrer[2]	*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

[1] sven.osterwalder@students.bfh.ch
[2] claude.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

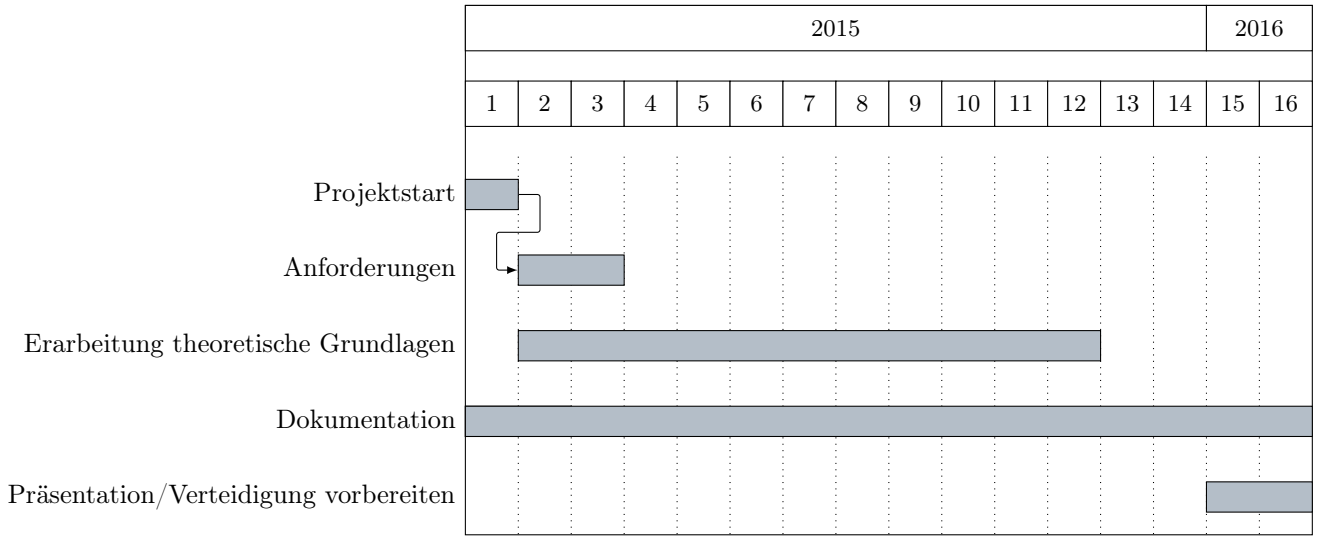

Abbildung 4.1.: *Zeitplan; Der Titel stellt Jahreszahlen, der Untertitel Semesterwochen dar*

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

LaTeX Eine Makro-Sammlung für das TeX-System. Wurde zur Erstellung dieser Dokumentation eingesetzt. Diese Dokumentation wurde mittels LaTeX 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 GitHub[1].

GitHub Eine freie Hosting-Platform für Git mit Weboberfläche.

4.3.2. Standards und Richtlinien

Pseudecode

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 Pseudcode 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 emitiert wird. In der Regel wird das Licht als Funktion in Abhängigkeit folgender Faktoren beschrieben:

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

Es wird dabei zwischen lokalen und globalen Belechtungsmodellen 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:

$$I = I_{ambient} + I_{diffuse} + I_{specular} \tag{5.1}$$

oder mathematisch ausgedrückt:

$$I = I_a + k_d \sum_{j=1}^{ls} (\overrightarrow{N} \cdot \overrightarrow{L_j}) + k_s \sum_{j=1}^{ls} (\overrightarrow{N} \cdot \overrightarrow{L_j}) \tag{5.2}$$

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
- $\overrightarrow{N}$: Einheitsnormale der Oberfläche
- $\overrightarrow{L_j}$: Vektor in Richtung der j-ten Lichtquelle
- k_s: Koeffizient der spiegelenden Komponente

- $\overrightarrow{L_j}$: 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 Betrachtes, zu berechnen, eignen sich aber weniger um gleichbleibende diffuse Anteile über weiter Flächen eines Bildes zu berechnen. Bei blickwinkelabhängigen 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 emitiert werden, beschrieben.

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

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

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

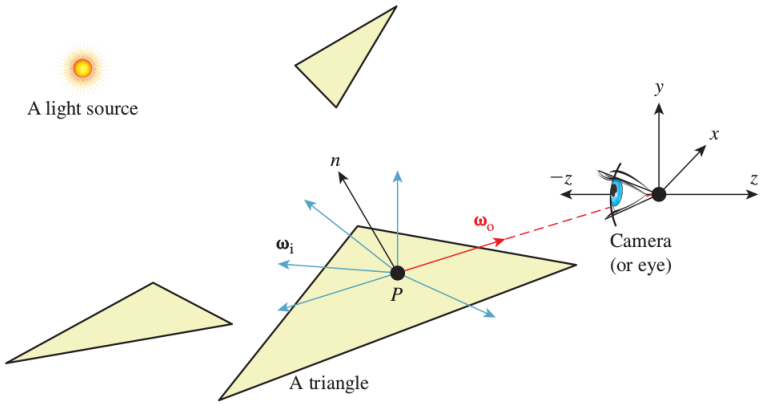

Abbildung 5.1.: *Punkt P auf einer Oberfläche eines Dreieckes, welcher für die Kamera bzw. den Betrachter sichtbar ist. Der Betrachter nimmt dabei das Licht, welches aus verschiedenen Richtungen ω_i kommt, über den Punkt P in Richtung ω_0 wahr.*[1]

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 minieren, können die Richtungen des Abtasten anhand der Lichtquellen priorisiert werden.

[1] Darstellung 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 **??**.
[caption=Eine abstrakte Umsetzung des Ray Castings[2].,label=fig:ray$_c asting$: $high_l evel, captionpos = b]def ray_c ast() : "pixelsïsalistofallpixelsoftheimageplaneforpixelinpixels : Saveallintersectionsforgivenpixelintersections = []$
Returns the ray passing through the given pixel from the eye ray = ray$_a t_p ixel(pixel)$
ßcene$_t rianglesïsalistofalltrianglescomingfrommeshescontainedinthescenetorenderfortriangleinscene_t riangles : p = intersect(ray, triangle)sum = 0$
for light in incoming$_l ights_a t_p : sum = sum + l.valueend$
if is$_s mallest_i ntersection(p, intersections) : pixel = sumintersections.append(p)$
Das Verfahren wurde erstmals 1968 in der Publikation "Some techniques for shading machine renderings of solids" von A. Appel vorgeschlagen und auch 1968 von der Matthematical Applications Group Inc. in "3-D Simulated Graphics Offered by Service Bureau" erfolgreich umgesetzt.

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

5.3. Ray Tracing

Bei dem heute als Ray Tracing bekannten Verfahren, handelt es sich um eine verbesserte Version des unter 5.2 genannten Ray Casting Verfahrens. Dieses wurde im Juni 1980 durch Turner Whitted in der Publikation "An Improved Illumination Model for Shaded Display" verbessert.

So schlägt Turner vor, dass die Berechnung der Sichtbarkeit (von Objekten) nicht bei dem nähesten gefundenen Schnittpunkt abgebrochen wird, sondern dass jedes Auftreffen eines (Licht-) Strahles mehr (Licht-) Strahlen durch Transmission bzw. Reflektion sowie in Richtung jeder Lichtquelle gesendet werden. Dieser Prozess wird so lange wiederholt, bis keiner der neu generierten (Licht-) Strahlen mehr auf ein Objekt trifft Whitted [1980][S. 345].
Es handelt sich dabei also um ein rekursives Verfahren und wird daher teilweise auch rekurisves Ray Tracing genannt.

5.4. Oberflächen

Sofern nicht anders vermerkt, basiert der folgende Abschnitt auf Division and Menon [1996][S. 1 ff].

Um in Computergrafiken überhaupt etwas darstellen zu können, muss erst einmal definiert werden, was dargestellt werden soll. Häufig orientiert sich die Computergrafik dabei an der realen Welt. In der realen Welt haben Oberlächen von Objekten häufig keine starken Übergänge (Kanten) sondern sind eher von glatter Natur Foley [1996][S. 471].

Die Darstellung von Kurven und Oberflächen führt zu zwei Fällen: Modellierung von bestehenden Objekten und Modellierung von Grund auf.

Zur Modellierung von Oberflächen werden hauptsächlich zwei Techniken verwendet: Parametrische Modellierung und implizite Modellierung.

Bei der parametrsichen Darstellung wird eine Oberfläche überlicherweise als eine Menge von Punkten definiert, so zum Beispiel:

$$\boldsymbol{p}(s,t) = (x(s,t), y(s,t), z(s,t)) \tag{5.4}$$

Bei der impliziten Darstellung wird eine Oberfläche überlicherweise als Kontur einer Funktion mit Wert 0 definiert, so zum Beispiel:

$$f(\boldsymbol{p}) = f(x,y,z) = 0 \tag{5.5}$$

Die parametrische Darstellung bringt Vorteile wie die Unabhängigkeit von einem Koordinatensystem oder eine effiziente Berechnung von Punkten auf einer Oberfläche. Die implizite Darstellung erlaubt hingegen eine grössere Einflussnahme aus mathematischer Sicht und ist daher sehr nützlich für Operationen wie Biegung, Vermischung, Schnitte (Intersektion) oder Bool'sche Operationen.
Meh. Not sure if intro is good enough.

5.4.1. Implizite Oberflächen

Wie in Gleichung 5.4 beschrieben, ist eine implizite Oberfläche gemäss Hart [1993][S. 1] als Funktion $f(\boldsymbol{x}) = \mathbb{R}^3 \rightarrow \mathbb{R}$ definiert. Es wird also jedem Punkt einer Menge $\boldsymbol{p} \in \mathbb{R}^3$ ein skalarer Wert $s \in \mathbb{R}$ zugewiesen. Dabei besteht die Oberfläche aus der Punktemenge $\boldsymbol{x} \equiv (x,y,z) \in \mathbb{R}^3$.

Angenommen $\boldsymbol{A}$ ist ein geschlossener Festkörper, welcher durch die Funktion f beschrieben wird, dann kann gemäss Hart [1993] Folgendes angenommen werden:

$$x \in \overset{\circ}{\boldsymbol{A}} \Leftrightarrow f(\boldsymbol{x}) < 0 \tag{5.6}$$
$$x \in \partial \boldsymbol{A} \Leftrightarrow f(\boldsymbol{x}) = 0 \tag{5.7}$$
$$x \in \mathbb{R}^3 - \boldsymbol{A} \Leftrightarrow f(\boldsymbol{x}) > 0 \tag{5.8}$$

Dies bedeutet, dass die implizite Funktion $f(\boldsymbol{x})$

- negativ ist, wenn sich ein Punkt $\boldsymbol{x}$ innerhalb von $\boldsymbol{A}$ befindet
- 0 ist, wenn sich ein Punkt $\boldsymbol{x}$ auf der Oberfläche von $\boldsymbol{A}$ befindet
- Positiv ist, wenn sich ein Punkt $\boldsymbol{x}$ ausserhalb von $\boldsymbol{A}$ befindet

Dies gilt, da es sich bei $\boldsymbol{x}$ um eine Punktemenge mit topologischer Struktur handelt.

Gemäss Division and Menon [1996] finden hauptsächlich drei Methoden Anwendung zur Beschreibung impliziter Oberflächen: algebraische Oberflächen, Blobby-Objekte sowie die funktionale Repräsentation. Hart [1994] gibt jedoch an, dass die gebräuchlichste Form von impliziten Oberflächen die algebraischen Oberflächen sind. Diese werden implizit durch polynomiale Funktionen definiert.

Algebraische und geometrische implizite Oberflächen

Ein Beispiel für eine algebraische Oberfläche ist die Beschreibung der Einheitskugel anhand einer impliziten algebraischen Gleichung zweiten Grades:

$$x^2 + y^2 + z^2 - 1 = 0 \tag{5.9}$$

Wobei es sich bei dem letzten Parameter um den Radius r handelt, welcher — bedingt durch die Einheitskugel — den Wert 1 hat.
Wie Division and Menon [1996] schreibt, handelt es sich bei impliziten Oberflächen, welche durch eine polynomiale Funktion zweiten Grades beschrieben werden, um quadrische implizite Oberflächen.
Hart [1994] gibt weiter an, dass — unter Nuztung einer Metrik — die Einheitskugel durch die implizite Gleichung

$$\|\boldsymbol{x}\| - 1 = 0 \tag{5.10}$$

beschrieben werden kann, was unter Anwendung der allgemeinen Form 5.5 einer impliziten Gleichung 5.5 zu folgender Gleichung führt:

$$f(\boldsymbol{x}) = \|\boldsymbol{x}\| - 1 \tag{5.11}$$

Dabei ist $\|\boldsymbol{x}\|$ als euklidische Metrik definiert und entspricht $\sqrt{x^2 + y^2 + z^2}$.

Die Gleichung 5.9 gibt die algebraische Distanz zurück, Gleichung 5.10 gibt die geometrische Distanz zurück.

Distanzfunktionen

Gemäss Hart [1994] wird die geometrische Darstellung von quadrischen Oberflächen bevorzugt, da deren Parameter unabhängig von Koordinaten sind, sie robuster und intuitiver sind. Es handelt sich dabei um eine **Distanzfunktion**.

Wie anfangs erwähnt, definiert Hart [1994] die allgemeine Form zur Beschreibung bzw. Darstellung von impliziter Oberflächen als Zuweisung von Punkten zu einem skalaren Wert: $f : \mathbb{R}^n \to \mathbb{R}$.

Unter Anwendung der unter 5.6 definierten Bedingungen kann geschlossen werden, dass eine Menge von Punkten A existiert, welche Teil von $\mathbb{R}^n$, also $A \subset \mathbb{R}^n$ ist. Dies heisst, dass alle Punkte in A die folgende Bedingung erfüllen:

$$A = \{x : f(x) \leq 0\} \tag{5.12}$$

Hart [1994] liefert zwei Definition, welche der Beschreibung von Distanzfunktionen dienen:

Definition 5.4.1. *Point-to-set distance*
Die Distanz eines Punktes zu einer Menge von Punkten definiert die Distanz eines Punktes $\boldsymbol{x} \in \mathbb{R}^3$ zu einer Menge von Punkten $A \subset \mathbb{R}^3$ als Distanz von $\boldsymbol{x}$ zum nähesten Punkt in A:

$$d(\boldsymbol{x}, \boldsymbol{A}) = \min_{\boldsymbol{y} \in \boldsymbol{A}}(\|\boldsymbol{x} - \boldsymbol{y}\|) \tag{5.13}$$

Definition 5.4.2. *Signed distance bound*
Eine Funktion $f : \mathbb{R}^3 \to \mathbb{R}$ ist eine Obergrenze ihrer impliziten Oberfläche $f^{-1}(0)$, wenn gilt:

$$|f(\boldsymbol{x})| \leq d(\boldsymbol{x}, f^{-1}(0)) \tag{5.14}$$

Wenn die Gleichung 5.14 für eine Funktion f gilt, dann ist f eine *vorzeichenabhängige Distanzfunktion (signed distance function).*

Distanzfelder (distance fields)

Introduce distance fields.

5.5. Darstellung von impliziten Oberflächen

Wie Hart [1994][S. 1] angibt, existieren verschiedene Möglichkeiten zur Darstellung (zum Rendering) von impliziten Oberflächen. So wandeln indirekte Methoden implizite Oberflächen in Polygonmodelle um, was die Nuztung bestehender Techniken und Hardware zur Darstellung von polygonalen Modellen erlaubt. Obwohl die Umwandlung der impliziten Oberflächen mit gängigen Systemen zur Darstellung problemlos dargestellt werden kann, ist die Umwandlung jedoch nicht in jedem Fall gegeben und kann zu nicht zusammenhängenden Flächen oder einer Verminderung des Detailgrades führen.

Eine andere Methode zur Darstellung von impliziten Oberflächen ist das unter 5.3 vorgestellte Ray Tracing Verfahren.
Ein (Licht-) Strahl wird dabei parametrisch als

$$r(t) = r_0 + t * r_d \tag{5.15}$$

beschrieben. Der Strahl startet dabei bei Punkt r_0 in Richtung des Einheitsvektors r_d, wobei t die zurückgelegte Distanz des Strahles ist. $r(t)$ räpresentiert also den Punkt in Raum, welchen der Strahl nach dem Zurücklegen der Distanz t — ausgehend von seinem Ursprung r_0 — erreicht.

Um nun die Schnittpunkte eines Strahles mit einer impliziten Oberfläche zu finden, wird die Gleichung des Lichtstrahles r (5.15) in die Funktion einer impliziten Oberfläche f (5.5) eingesetzt. Wobei $r : \mathbb{R} \to \mathbb{R}^3$ und $f : \mathbb{R}^3 \to \mathbb{R}$. Dies ergibt die zusammengesetzte Funktion $F = f \circ r$ wobei $F : \mathbb{R} \to \mathbb{R}$.

Die Lösungen dieser Gleichung sind alle Distanzen t, welche ein gegebener Strahl zurücklegt und welche die folgende Bedingung erfüllen:

$$F(t) = f \circ r = f(r(t)) = 0 \tag{5.16}$$

Um die Gleichung 5.16 zu lösen, können numerische Verfahren zur Nullstellensuche angewendet werden, wobei die Verfahren vom Typ der Funktion $F(t)$ abhängig sind. Bei polynomialen Funktionen bis zum vierten Grad existieren analytische Lösungen, für eine beliebige Funktion muss jedoch ein generisches, robustes Verfahren zur Nullstellensuche verwendet werden. Dies bedingt jedoch meist, dass mehr Informationen über die Funktion zur Verfügung stehen, was beispielsweise durch Ableiten dieser gelöst werden kann.

Die erwähnten Verfahren zur Nullstellensuche haben jedoch häufig den Nachteil, dass sie mehrere Schnittpunkte eines Strahles mit einer impliziten Oberfläche liefern. Um diese Problematik zu umgehen, wird nur der kleinste Wert von t berücksichtigt. Die Ray Marching und Sphere Tracing Algorithmen gehen hier sogar noch einen Schritt weiter, in dem sie nur die erste Nullstelle der Gleichung 5.16 betrachten.

5.5.1. Ray Marching

Perlin and Hoffert [1989] schlagen eine Abtastung des Strahles mit fixen Abständen $\Delta\mu$ vor:

$$x = x_{\mu_0} + k * \Delta x_\mu \tag{5.17}$$

wobei $k = 0, 1, 2, \ldots$ und $\mu_0 + k\Delta\mu \leq \mu_1$.

Auf die parametrische Darstellung eines (Licht-) Strahles, Gleichung 5.16, angwendet:

$$r(k) = r_0 + \Delta t * k * r_d \tag{5.18}$$

wobei Δt die Grösse der Abstände und $k = 0, 1, 2, \ldots$ die Nummer der Schritte darstellt. Wie Hart et al. [1989] schreiben, bildet das Abtasten des (Licht-) Strahles mit fixen Abständen die Basis für gewisse Verfahren des volumetrischen Renderings.

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

```
def ray_march():
    step         = 0
    intersection = 0
    max_steps    = 10

    while step < max_steps:
        intersection = test_intersection(k)

        if intersection <= 0:
            # An intersection has happened
            #   intersection <  0: ray is inside surface
            #   intersection == 0: ray is excatly on surface
            return ray_travel_distance(step)

        step = step + 1

    # When we reach this step, after max_steps, no intersection
    # has happened, so distance is 0
    return 0
```

Auflistung 5.1: *Eine abstrakte Umsetzung des Ray Marchings*[3].

[3] Algorithmus in Pseudocode gemäss Perlin and Hoffert [1989][S. 259, Abschnitt 3.1]

Dabei ist jedoch zu beachten, dass der Abstand zur Abtastung eines Strahles Δt so gering als möglich sein sollte um eine Punktemeng bzw. ein Objekt — definiert durch implizite Oberflächen — A möglichst gut abschätzen zu können. Ist der gewählte Abstand zu gross gewählt, so findet ggf. eine Abtastung weit im Inneren des Objektes statt und somit geht Präzision verloren. Es ist auch möglich dass der erste eigentliche Punkt gar nicht abgetastet wird und erst der zweite abgetastete Punkt das Objekt "erkennt". Die Grafik veranschaulicht diese Problematiken. insert referen

Provide illustration for ray marching

Hart [1994] weist darauf hin, dass Ray Marching durch den möglichst geringen Abstand zwischen den Abtastungen entsprechend langsam und paralleles Abtasten praktisch unumgänglich ist. In der von Hart [1994] vorgestellten Technik des Sphere Tracings ist der Abstand zwischen den Abtastungen nicht konstant sondern variiert in Abhängigkeit der Geometrie.

5.5.2. Sphere Tracing

Das von Hart [1994] vorgestellte Sphere Tracing Verfahren ist ein Ray Tracing (5.3) Verfahren für implizite Oberflächen. Es handelt sich nach wie vor um Ray Marching (5.5.1), die Distanz der Schritte zum Abtastens eines (Licht-) Strahles wird jedoch aufgrund einer Distanzfunktion (5.4.1) bestimmt.

Hart [1994] verweist auf den Term *"unbounding volumes"*, welcher in Hart et al. [1989] eingeführt wurde. "Unbounding volumes" (zu Deutsch etwa "negativer Hüllkörper") wird genutzt um Sphere Tracing zu beschreiben und darzustellen. Der Term steht im Gegensatz zu dem gängigen Konzept des Hüllkörpers ("bounding volumes") — ein Volumen, welches einen Körper umschliesst. Ein "negativer Hüllkörper" ("unbounding volume") umschliesst also eine Fläche im Raum, welche den Körper *nicht* beinhaltet.

Man möchte für einen abzutastenden Punkt im Raum ein Volumen finden, wessen Zentrum im abzutastenden Punkt liegt. Ist der Abstand des Punktes zum nähesten Punkt der Oberfläche eines Objektes bekannt, kann dieser Abstand als Radius einer Kugel angenommen werden. Diese Kugel dient als negativer Hüllkörper ("unbounding Volume") und ist *garantiert nicht* Teil des Objektes und schneidet dieses auch nicht ($\overline{\boldsymbol{A}}^{\circ}$) — nur der äusserste Punkt des Abstandes (also des Radius der Kugel) liegt genau auf der Oberfläche des Objektes ($\partial \boldsymbol{A}$). Der Radius solch einer Kugel wird durch Evaluation der Distanzfunktion eines abzutastenden Punktes im Raum bestimmt.

Gemäss Hart [1994] kommt daher auch der Name Sphere Tracing: Die Schnittpunkte eines (Licht-) Strahles werden durch eine Folge von negativen Hüllkörper ("unbounding volumes") — bzw. in diesem Fall Kugeln ("unbounding spheres") — beschrieben.

Da Hart et al. [1989] die Darstellung von Fraktale im dreidimensionalen Raum beschreibt, wird dort von einer Abschätzung der Distanz gesprochen. Dies ist dadurch bedingt, dass die Distanz für Fraktale nicht effizient berechnet werden kann. Betrachtet man jedoch die Darstellung von "regulären" Objekten, wie zum Beispiel eine Kugel, kann der zur Oberfläche am nähesten gelegene Punkt von einem beliebigen Punkt derselben Domäne exakt berechnet werden. Dies ist durch die implizite Gleichung 5.11 gegeben.

Gemäss Hart et al. [1989] verläuft die Strahlenverfolgung bei dem Sphere Tracing Verfahren wie folgt. Ein Strahl wird vom Betrachter (Auge bzw. Lochkamera) durch die Bildebene zu einem Objekt geschossen. Dabei wird beim initialen Augangspunkt der Radius eines negativen Hüllkörpers in Form einer Kugel — so wie oben beschrieben — berechnet. Dies ist die Distanz, welcher der Strahl in einem ersten Schritt effektiv zurücklegen wird. Bei jedem Schnittpunkt der Kugel mit dem Strahl wird dasselbe Verfahren wiederholt. Dies geschieht so lange, bis schliesslich der Strahl durch einen Schnittpunkt mit einem Radius auf die Oberfläche des Objektes trifft. Ein weiteres Abbruchkriterium ist eine definierte maximale Distanz eines Strahles. Ist diese erreicht und der Strahl erreicht die Oberfläche des Objektes nicht — weil der Strahl das Objekt nicht schneidet oder das Objekt zu weit weg ist —, wird abgebrochen. Somit ist auch ersichtlich, dass Sphere Tracing nicht die unter 5.5.1 genannten Problematiken aufweist.

Provide illustration for sphere tracing

Ausgehend von der parametrischen Beschreibung eines (Licht-) Strahles (Gleichung 5.15), beschreibt Hart [1994] die Richtung r_d eines Strahles als Einheitsvektor:

$$r_d = \frac{p_{x,y} - r_0}{|p_{x,y} - r_0|} \tag{5.19}$$

wobei r_0 der Ursprung eines Strahles und $p_{x,y}$ ein Punkt der Bildebene ist.

Um nun den Schnittpunkt eines Strahles r_d mit der Oberfläche eines Objektes zu finden, muss die Gleichung $F(t)$ (5.16) gelöst werden. Dabei ist — wie oben definiert — die Funktion $f(x)$ nun eine Distanzfunktion wie die geometrische Distanzfunktion zur Beschreibung einer Kugel (Gleichung 5.10).

Evaulaiert man nun die Gleichung $F(t)$ unter Anwendung der eben beschriebenen Strahlenverfolgung, findet man so die erste positive Nullstelle der Gleichung $F(t)$. Diese Nullstelle ist die Grenze der Folge von negativen Hüllkörpern ("unbounding spheres"), welche durch die rekrusive Gleichung

$$t_{i+1} = t_i + F(t_i) \tag{5.20}$$

definiert ist. Der Ursprungspunkt ist dabei als t_0 definiert. Diese Folge konvergiert genau dann — und nur dann –, wenn der Strahl auf die implizite Oberfläche eines Objkektes trifft. Diese Folge bildet den Kern des Algorithmus zur Darstellung von geometrisch definierten, impliziten Oberflächen.

```
def sphere_trace():
    ray_distance          = 0
    estimated_distance    = 0
    max_distance          = 9001
    convergence_precision = 0.000001

    while ray_distance < max_distance:
        # sd_sphere is a signed distance function defining the implicit surface
        # cast_ray defines the ray equation given the current travelled /
        # marched distance of the ray
        estimated_distance = sd_sphere(cast_ray(ray_distance))

        if estimated_distance < convergence_precision:
            # the estimated distance is already smaller than the desired
            # precision of the convergence, so return the distance the ray has
            # travelled as we have an intersection
            return ray_distance

        ray_distance = ray_distance + estimated_distance

    # When we reach this point, there was no intersection between the ray and a
    # implicit surface, so simply return 0
    return 0
```

Auflistung 5.2: *Eine abstrakte Umsetzung des Sphere Tracings*[4].

[4] Algorithmus in Pseudocode gemäss Hart [1994][S. 531, Fig. 1]

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.

International Business Machines Corporation Research Division and J. Menon. *An Introduction to Implicit Techniques*. Research report. IBM T.J. Watson Research Center, 1996. URL https://books.google.ch/books?id=Ew4OGwAACAAJ.

John C Hart. Ray tracing implicit surfaces. *Siggraph 93 Course Notes: Design, Visualization and Animation of Implicit Surfaces*, pages 1–16, 1993.

John C. Hart. Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces. *The Visual Computer*, 12:527–545, 1994.

K. Perlin and E. M. Hoffert. Hypertexture. *SIGGRAPH Comput. Graph.*, 23(3):253–262, July 1989. ISSN 0097-8930. doi: 10.1145/74334.74359. URL http://doi.acm.org/10.1145/74334.74359.

J. C. Hart, D. J. Sandin, and L. H. Kauffman. Ray Tracing Deterministic 3-D Fractals. *SIGGRAPH Comput. Graph.*, 23(3):289–296, July 1989. ISSN 0097-8930. doi: 10.1145/74334.74363. URL http://doi.acm.org/10.1145/74334.74363.

Abbildungsverzeichnis

Tabellenverzeichnis

Auflistungsverzeichnis

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
```

20151004

```
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 / procdeure
    * 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 posseses the mentioned book and proposes furthermore the lecture of a book dedicated to
      ray tracing which he also posesses. 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
    Univeristy of applied sciences (CF)
  - The lines shall be shortened so that they do not exceed 80 caracters 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
```

20151011

```
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)
  * Ray tracing
    * Is an explanation of transmission and refraction needed? (SO)
    * Is the material sufficient? (SO)
  * Implicit surfaces
    * Does one need to explain euclidean distance? (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
```