

Das ist eine Überarbeitete Version der Ausarbeitung von Maciej Makowski –
Überarbeitung Andreas Bauer.

Die Ausarbeitung stützt sich auf die Folien von der Vorlesung, das Buch “Computer Graphics with OpenGL” (3rd Edition), auf die Unterlagen, die auf <http://www-lehre.inf.uos.de/~cg/2006/> zu finden sind, auf <http://www.wikipedia.org> (siehe Links bei den jeweiligen Fragen) und auf die Ausarbeitung der mündlichen Fragen, die hier im Forum zu finden war.

Für die Richtigkeit wird keine Haftung übernommen. ;)

Kapitel 1 – A Survey of Computer Graphics (S. 2)

1) Was ist Computergraphik?

Computergraphik ist der Vorgang, der aus einer Beschreibung ein Bild generiert (description ⇒ image, siehe Graphik Folie 1, Kapitel 1).

- Umkehrung = Mustererkennung (pattern recognition): image ⇒ description
- Bildverarbeitung (image processing): „bad“ image ⇒ „better“ image

2) Welche Anwendungsbereiche von Computergraphik gibt es?

- Presentation Graphics: Repräsentation von business data (z.B. bar charts, line graphs)
- Computer Art: paintbrush programmes, commercial art (z.B. logos, commercials), animation, video clips usw.
- Entertainment: Filme, Musikvideos, TV-shows, usw.
- Education and Training: Computer-generierte Modelle von physikalischen, finanziellen und ökonomischen Systemen, Simulatoren (Militär, Fahrschule, usw.)
- Visualization: scientific & information visualization. Dabei sind die Algorithmen von Daten-Charakteristika und dem Untersuchungsziel abhängig. Techniken: Color coding, contour plots, volume visualization
- Image Processing: Modifikation oder Interpretation von Bildern. Ziele: Qualität verbessern bzw. maschinen-unterstützte Verarbeitung visueller Information
- Graphical User Interfaces (GUIs): WIMP (windows, icons, menus, pointer)

Kapitel 2 – Overview of Graphics Systems (S. 34)

3) Welche video display devices werden in der Computergraphik verwendet und welche Eigenschaften haben sie? (S. 35)

- CRT Monitore (Röhrenmonitore, S. 42):
http://en.wikipedia.org/wiki/Cathode_ray_tube
4 Magnete links und rechts sowie oben und unten, dazwischen liegt der Elektronenstrahl, dieser trifft dann gelenkt durch die Magneten auf der Oberfläche, welche mit einer phosphorizierenden Schicht beschichtet ist, auf und beginnt in diesem Punkt zu leuchten. Ein Bild wird von links nach rechts und von oben nach unten Punkt für Punkt aufgebaut. Bei Farbmonitoren hat man für jede

Farbe einen separaten Elektronenstrahl (RGB-fokussieren alle am selben Punkt am Bildschirm)

- Raster-Scan Monitore (S.39, S.52): zeichnen Punkt für Punkt und Zeile für Zeile
- Random-Scan Monitore (S. 41, S.56): zeichnen das Objekt direkt auf den Bildschirm

- Flachbildschirme

http://en.wikipedia.org/wiki/Flat_panel_display

Man kann Flachbildschirme in zwei Arten unterteilen:

- Emmisive displays = Emitter (S.44):
konvertieren elektrische Energie in Licht, können also Licht selbst erzeugen. Bsp: *Plasmabildschirme*: 2 Elektroden werden elektrisch geladen, dazwischen befindet sich leuchtendes Gas, am Schnittpunkt der Elektroden/Leiter beginnt das Gas zu leuchten. Die Trennung zw. den Bildpunkten wird durch das elektrische Feld der einzelnen Elektroden erreicht.
Thin-film-electroluminescent-displays: sind ähnlich aufgebaut wie die Plasmabildschirme, jedoch zw. den Glasplatten befindet sich Phosphor. Nachteil: braucht mehr Strom als Plasmabildschirmen und gute Farb/Graudarstellungen sind nur schwer zu erreichen.
 - Nonemissive displays = Nonemitter
erzeugen das Licht nicht selber sondern nutzen Licht aus anderen Quellen zur graphischen Darstellung (z.B.: LCD-Bildschirme)
- LED (light emitting diodes, Flachbildschirm):
<http://en.wikipedia.org/wiki/Led>
es gibt schon blaue Leuchtdioden ⇒ sehr teuer in der Produktion. In jedem Bildpunkt eine Leuchtdiode
 - LCD (liquid crystal displays, Flachbildschirm):
http://en.wikipedia.org/wiki/Liquid_crystal_display
Verwendung bei Notebook-Displays.
verwenden polarisierendes Licht entweder aus der Umgebung oder von einer anderen Quelle, welches durch die Flüssigkeitskristalle geschickt wird, um ein Bild zu erzeugen.
 - Passive-matrix-LCDs: Die Flüssigkeitskristalle polarisieren das Licht (S. 46). Man hat 2 Glasplatten jeweils mit einem Polarisierer, die im rechten Winkel zu einander liegen, durch die Flüssigkeitskristalle wird das hereinkommende Licht gedreht ⇒ nur wenn sie unter Strom stehen, ansonsten sind sie in Längsrichtung angeordnet und lassen das Licht nicht durch.
 - Active-matrix-LCDs: bei jedem Bildpunkt ein Transistor der die Spannung kontrolliert.

4) Was versteht man unter interlacing? (S. 41)

<http://en.wikipedia.org/wiki/Interlacing>, <http://de.wikipedia.org/wiki/Interlace>

Interlacing wird bei Raster-Scan Monitoren verwendet. Es ist eine Methode zur Aufnahme von Bewegtbildern, der Übertragung von Bewegtbildern und der Darstellung von Bildern auf sequenziellen Darstellungsgeräten (Bildröhren, Laser-Projektoren). Das Verfahren findet Verwendung bei Fernsehgeräten (dort noch üblich) und Computerbildschirmen (dort mittlerweile eher unüblich). Dabei bilden zwei Halbbilder (engl. Fields) erst ein vollständiges Bild mit voller vertikaler Auflösung. Beim Bildaufbau des 2:1-Interlacings wird jeweils eine Zeile übersprungen, daher der Name. Die übersprungenen Zeilen wurden beim vorhergehenden und werden beim nächsten Aufbau dargestellt.

Der Vorteil des Verfahrens besteht in der Verdopplung der Bildwiederholfrequenz bei gleicher vertikaler Auflösung. Der Nachteil ist ein erhöhtes Flimmern des Bildes an dünnen waagerechten Linien und anderen feinen Strukturen, Artefakte an langsam bewegten Objekten sowie Inhomogenitäten der Darstellung von strukturarmen Flächen.

5) Wie funktioniert eine (Farb)-Kathodenstrahlröhre (CRT)? (S. 42)

Siehe Frage 3)

6) Was ist ein Plasmabildschirm und wie funktioniert er? (S. 45)

Siehe Frage 3)

7) Was ist Virtual Reality? (s. 48)

= Stimulation der menschlichen Sinne, um ihnen eine fiktive Umgebung vorzutäuschen.

Die 3 Ebenen:

- Augen: privat eye, head mounted display (HMD)
- Ohren: headphones, loudspeakers in HMD
- Tastsinn: data glove, data suit, threadmill

8) Was ist ein "head-mounted display" (HMD)? (s. 51)

http://de.wikipedia.org/wiki/Head_Mounted_Display

Ist ein (VR-)Helm mit zwei LCD-Bildschirmen und Lautsprechern. Er wird verwendet und eine Virtual-Reality-Umgebung vorzutäuschen. Außerdem verfügen HMDs oft über einen Tracker, der die momentane Blickrichtung und Position des Benutzers erfassen kann.

9) Was ist ein "data-glove" und wozu wird er verwendet? (S. 60)

http://en.wikipedia.org/wiki/Data_glove

Ein „data-glove“ ist ein mit Sensoren besetzter Handschuh, der es dem Benutzer erlaubt Objekte in einer VR-Umgebung anzufassen und zu bewegen. Durch die Sensoren werden auch die Bewegungen der einzelnen Finger erfasst. In Kombination mit einem HMD ist es auch möglich, die eigene Hand in der VR-Umgebung zu sehen.

10) Welche Arten von Scannern mit CCD-Elementen gibt es?

- Flachbett-Scanner

<http://de.wikipedia.org/wiki/Flachbettscanner>

CCD-Elemente: Lichtsensoren, die bei einfallendem Licht einen elektrischen Strom erzeugen. Die Lichtsensoren sind farbenblind, das heißt, es wird nur die Intensität der Farbe aber nicht die Farbe selbst erfasst.

(http://de.wikipedia.org/wiki/Charge-coupled_Device)

Die zeilenweise Abtastung kann je einmal pro Farbe erfolgen, oder einmal mit 3-fach-Zeile und je einem Farbfilter.

Durch die CCD-Elemente kann es durch abweichende Lichtempfindlichkeiten der Fotozellen zu Bildfehlern kommen. Ein Trommelscanner ist deshalb beim momentanen Stand der Technik immer noch qualitativ besser, als ein Flachbettscanner.

- Dokumenten-Scanner

<http://de.wikipedia.org/wiki/Dokumentenscanner>

Die Scanner sind meist als Durchzugsscanner ausgeführt und können je nach Ausführung Dokumentenstapel von 50 bis 1000 Seiten selbstständig abarbeiten.

Die meisten Modelle sind in der Lage mittels zweier CCD-Sensoren-Leisten die Vorder- und die Rückseite gleichzeitig in einem Scan zu erfassen (Duplex). Einige Modelle werden als Kombination von Durchzugs- und Flachbettscanner ausgeführt, um auch gebundene Dokumente ohne Zerstörung scannen zu können.

Die meisten der am Markt befindlichen Dokumentenscanner sind reine Schwarz/weiß-Geräte (mit grün als Blindfarbe). Neuerdings kommen aber immer mehr Geräte auf den Markt, die auch farbig in guter Qualität scannen können.

11) Welche Drucker gibt es und wie funktionieren sie generell? (s. 72)

- Impact Drucker: Impact-Drucker sind von der Funktionsweise vergleichbar mit einer Schreibmaschine. Sie drucken vorgefertigte Zeichen über ein Farbband auf Papier.

- Nonimpact Drucker: verwenden keine fixvorgefertigten Zeichen. Bsp Tintenstrahl-, Laserdrucker.
 - Laserdrucker: Man hat eine Rolle, welche mit einem photoelektrischen Material ummantelt ist. Diese wird durch einen Laser elektrisch aufgeladen wird. Auf dieser Rolle befinden sich verschiedene Ladungen. Anschließend wird der Toner auf die Rolle aufgetragen und an den geladenen Stellen bleibt der Toner haften, daraufhin wird der Toner auf das Papier übertragen.
 - Tintenstrahldrucker: Dabei wird die Tinte durch elektronische Ladungen unterschiedlich abgelenkt um die Tinte genau zu positionieren.
 - Electric Static Devices: Kopierer: Papier wird negativ geladen, Toner wird positiv geladen, dort wo das Papier neg. geladen ist, bleibt der Toner haften.

12) Wie arbeiten Sublimationsdrucker?

<http://de.wikipedia.org/wiki/Thermosublimation>, <http://www.drucker-onkel.de/drucktechnik/sublimation.htm> Sublimationsdrucker arbeiten mit Farbbändern oder Farbfolie. Pro Grundfarbe (cyan, magenta, yellow) wird ein Druckdurchlauf benötigt. Die Farbe wird mit Heizelementen von den Farbbändern gelöst und auf das Papier aufgedampft. Die Temperatur bestimmt dabei die Farbstoffmenge, die auf das Papier übertragen wird und steuert damit die Helligkeit des Bildpunktes. Die meisten Sublimationsdrucker arbeiten mit Farbfolien in der Größe des zu bedruckenden Mediums. Aus diesem Grund sind alle Ausdrücke gleich teuer, egal ob nur ein Punkt gedruckt wird oder eine vollflächige Seite.

Kapitel 3 – Output Primitives (S. 84)

13) Welche line drawing Algorithmen kennen Sie und wie funktionieren Sie?

- DDA (digital differential analyzer, S. 94):
 Die Linie wird in Einerkoordinaten-Einheitsintervalle unterteilt $y_{k+1} = y_k + m$
 $k \dots$ geht von 0 (Startpunkt) bis Endpunkt \rightarrow nimmt nur Integerwerte an und wird immer um 1 erhöht
 $m \dots$ ist die Steigung, jeder Wert zw. 0 und 1
 $y \dots$ muss zur nächsten Pixelposition gerundet werden
 Ist die Steigung größer als 1, werden die x- und y-Rollen vertauscht.
 Nachteil: es wird mit float gerechnet und ist daher rechenaufwändiger – ist aber immer noch schneller, als wenn man die Pixelpositionen direkt implementieren würde. Er ist schneller als normale mathematische Berechnungen für Geraden $y=kx+d$, da er sich die Eigenschaft vom Rastersystem zu Nutze macht und dadurch die Multiplikation durch Addition ersetzen kann.

- Bresenham Line Algorithmus (S. 95):
<http://de.wikipedia.org/wiki/Bresenham-Algorithmus>,
http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm
 Verwendet Integerzahlen → weniger rechenaufwändig; weiters können auch Kreise und Kurven dargestellt werden.
 Die vertikale Achse definiert die Scanlinienposition und die horizontale Achse die Pixel-Spalten. Es wird überprüft welche Pixelposition der Linie näher ist (obere oder untere).
 Berechnung mit Hilfe des Entscheidungsparameters p_k . Ist $p_k < 0$, dann wird y_k geplottet, andernfalls y_{k+1} .
 Horizontale Linien (mit $\Delta y=0$), vertikale Linien (mit $\Delta x=0$) und diagonale Linien ($|\Delta x|=|\Delta y|$) können direkt gezeichnet werden.

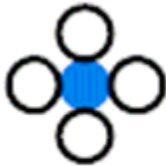
14) Welche zwei grundsätzlichen Arten von Polygonen kennen Sie? Wie kann man geht die Umwandlung vor sich?

Polygon = eine ebene Figur mit mind. 3 Koordinatenpunkten (vertices), welche durch Gerade (edges) miteinander verbunden sind. Edges eines Polygons haben ausser den Eckpunkten keine gemeinsamen Punkte ⇒ dürfen sich nicht schneiden

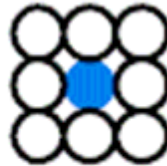
- Konvexe Polygone:
http://en.wikipedia.org/wiki/Convex_polygon
 Alle Winkel innerhalb des Polygons sind $< 180^\circ$
- Konkave Polygone:
http://en.wikipedia.org/wiki/Concave_polygon
 Mindestens ein Winkel $> 180^\circ$
 Verlängerungen mancher Kanten können andere Polygonkanten schneiden
- Erkennen konvexer Polygone:
 Vektormethode: Alle Vektor-Kreuzprodukte haben das gleiche Vorzeichen → konvex
 Rotationsmethode: Drehe die Polygon-Kanten auf die x-Achse. Immer gleiche Drehrichtung → konvex
- Aufspaltung konkaver Polygone:
 - Vektormethode: Man geht gegen den Uhrzeigersinn vor, hat man irgendein Kreuzprodukt mit neg. z-Komponente, kann man das Polygon aufteilen: man verlängert den Kantenvektor des ersten Punktes des negativen Kreuzprodukt-Paares
 - Rotationsmethode: Man geht wieder gegen den Uhrzeigersinn vor. Der Knoten V_k wird in den Ursprung gebracht und das Polygon wird rotiert bis der Knoten V_{k+1} auf der x-Achse ist. Ist der Knoten V_{k+2} unter der x-Achse, dann ist das Polygon konkav und wird entlang der x-Achse in zwei konvexe Polygone aufgeteilt.
 Man macht den jeweiligen Test dann für die beiden neuen Polygone, solange bis man alle Knoten getestet hat.

15) Welche Füll-Algorithmen kennen Sie und wie funktionieren Sie? (S. 123 ff.)

- Boundary-Fill Verfahren (S. 201):
Startet bei einem „inneren“ Punkt und füllt von innen nach außen, von Pixel zu Pixel, bis die Polygonränder erreicht sind. Diese Methode ist bei Polygonen mit komplexen Grenzen und interaktiven Zeichnungssystemen sinnvoll.



4-way-stepping



8-way-stepping

Über 4er-Nachbarschaft bzw. 8er-Nachbarschaft

Fülle rekursiv alle Nachbar-Pixel bis Grenze erreicht ist

Bei 4er-Nachbarschaft muss die Grenze die 8er-Nachbarschaft erfüllen und umgekehrt

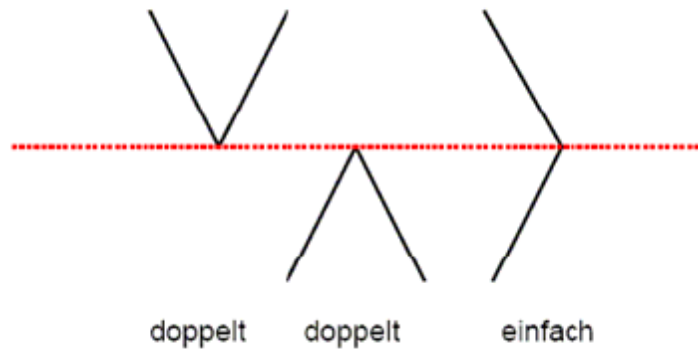
Rekursive Boundary-Fill Algorithmen füllen Flächen nicht immer korrekt, wenn ein Pixel schon in der Füllfarbe dargestellt wird. Stößt man auf ein Pixel, das schon die Füllfarbe hat, kann es passieren, dass das Programm beendet wird, ohne dass alle Pixel gefärbt sind. Dem kann man vorbeugen, indem man zuerst die Farbe aller inneren Pixel ändert, die schon zu Beginn die Füllfarbe haben, bevor man Boundary-Fill anwendet.

Anwendung siehe Buch S. 204

Da diese Methode bedenkliche Häufungen von benachbarten Punkten benötigt werden meistens effizientere Methoden verwendet.

- Scan-Line Fill Verfahren (S. 196)
Berechnet alle Schnittpunkte der Scanlinie mit den Polygonrändern. Danach werden alle Pixel, die in diesem Intervall liegen, auf die entsprechende Farbe gesetzt.
Das Verfahren funktioniert folgendermaßen:
Die Kanten werden nach dem größten y-Wert sortiert. Dann bewegt sich die Scanline von oben nach unten durch. Für jede Position der Scanline werden die aktiven Kanten ermittelt und die Schnittpunkte berechnet. Die Schnittpunkte werden nach den x-Werten sortiert und dann wird die Scanline eingefärbt. Nach jedem ungeraden Schnittpunkt wird gestartet, und nach jedem geraden gestoppt!
Probleme:

Geht eine Scan-Line genau durch einen Eckpunkt des Polygons, so muss je nachdem wie der Eckpunkt aussieht, eine Unterscheidung getroffen werden, wie oft dieser Eckpunkt gezählt wird. Die Unterscheidung funktioniert folgendermaßen. Wenn die beiden Kanten auf der selben Seite liegen (oben oder unten) dann wird zweimal gezählt, wenn sie auf unterschiedlichen Seiten liegen zweimal.



- Flood Fill Verfahren (S. 205)

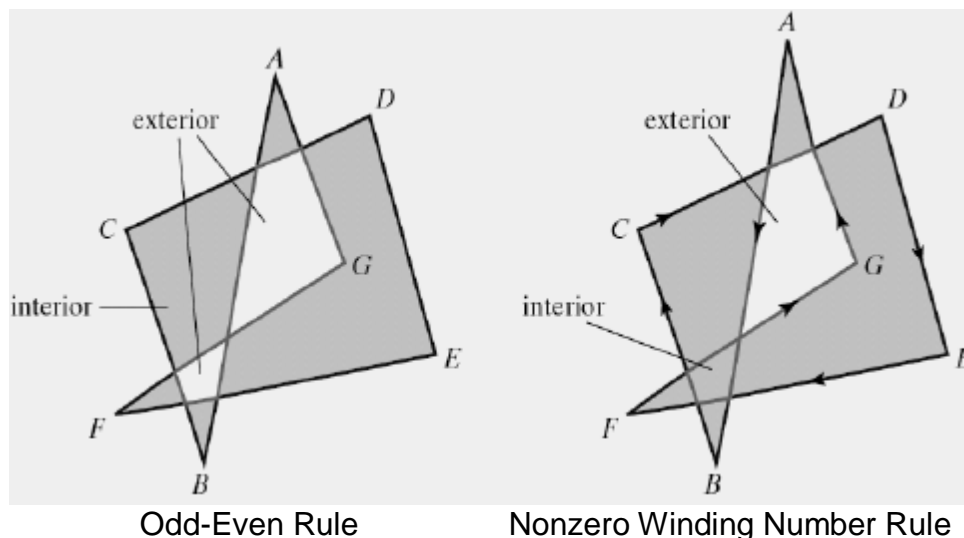
http://en.wikipedia.org/wiki/Flood_fill

Will man eine Fläche füllen (oder umfärben), deren Ränder nicht in einer Farbe gehalten sind, kann man diese Flächen färben, indem man eine festgelegte innere Farbe ersetzt, anstatt auf die Randfarbe zu achten.

Man nimmt wieder einen Punkt im Inneren und ersetzt die Farbe der Pixel, die eine bestimmte Farbe haben.

Hat die Fläche mehr als eine Farbe, so werden die Pixelwerte zuerst auf eine Farbe gesetzt. Dies geschieht mittels 4- oder 8-Nachbar-system.

16) Welche "inside-outside tests" kennen Sie, und wie funktionieren sie? (S. 127)



- Odd-even rule (auch odd-parity rule oder even-odd rule):
Man zeichnet einen Punkt P und davon ausgehend eine Linie zu einem Punkt außerhalb der Polygongrenzen (nur in eine Richtung). Schneidet diese Linie dabei eine ungerade Anzahl an Kanten, so liegt der Punkt innerhalb, andernfalls außerhalb.
Man muss aufpassen, dass die gezeichnete Linie keinen Eckpunkt kreuzt, da dieser 2x gezählt wird.
- Nonzero Winding Number
Es wird gezählt, wie oft sich die Objektgrenze um einen bestimmten Punkt dreht (gegen den Uhrzeigersinn). Diese Anzahl wird winding-number genannt. Ein innerer Punkt hat eine winding-number $\neq 0$, also eine nonzero winding number. Man nimmt einen beliebigen Punkt P und zeichnet eine Linie zu einem Punkt außerhalb des Objekts (darf keinen Eckpunkt schneiden). Man geht entlang der Linie von Punkt P zum entfernten Punkt und zählt dabei die Objektgrenzen, die diese Linie schneiden (in jede Richtung). Man zählt +1 wenn es eine Linie von rechts nach links ist und -1, wenn eine Linie von links nach rechts schneidet. bestimmt manche Flächen als innerhalb, die die odd-even-Rule als ausserhalb erkennt.

17) Was ist die „nonzero winding number rule“? (S. 128)

Siehe vorige Frage

18) Welche Möglichkeiten gibt es zum Füllen von Polygonen? (S. 126)

Siehe Frage 15)

19) Welche Möglichkeiten (Verfahren) der character generation (Buchstabenerzeugung) gibt es, welche Vor- und Nachteile ergeben sich daraus? (S. 147)

- Bitmap font:
Zeichen = Bitmuster auf einem rechteckigen Raster
Vorteile: Einfach zu definieren und darzustellen
Nachteile: Benötigt viel Speicherplatz, da jede Variation im Cache gespeichert werden muss
Variationen (verschiedene Formate) können oft nur unter Verlust der Qualität dargestellt werden
- Outline font:
Zeichen werden durch gerade Linien und Kurven dargestellt
Vorteile: Benötigt weniger Speicherplatz, Zeichengröße und Format (fett, kursiv, usw.) können einfacher verändert werden
Nachteil: Darstellung dauert länger, weil die Zeichen in den frame buffer geschrieben werden müssen

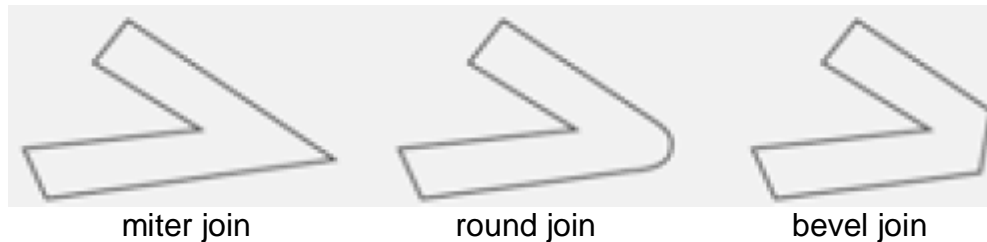
Kapitel 4 – Attributes of Output Primitives (S. 172)

20) Welche Linien-Attribute kennen Sie? (S. 183 ff.)

Eine Linie kann mit 3 Basiselementen dargestellt werden: Farbe, Breite und Style. Die Farbe wird wie auch für alle anderen Primitive gesetzt, wobei Breite und Style in anderen Funktionen ermittelt werden.

- **Linien-Breite (S. 183):**
die Implementierung hängt davon ab, wo es dargestellt werden soll. Bei Videomonitoren kann man eine dicke Linie mittels paralleler Linien darstellen, wobei ein Tintenplotter eine breite Feder benötigt.
Bei einem Raster wird als Standard eine 1-Pixel-breite Linie gezeichnet. Dickere Linien werden als eine Vervielfältigung der Standardlinie dargestellt, indem einfach Pixel dazuaddiert werden und dadurch eine parallele Linie gezeichnet wird. Hat man eine Steigung ≤ 1 , kann man dadurch Linien zeichnen, dass man Pixel in vertikale Richtung plottet und diese geplotteten Pixel dann für jede Spalte (x-Wert) darstellt
soll die Linie 2 Pixel breit sein, so plottet man die Pixel an Stelle (x,y) und (x,y+1). Bei drei und mehr Pixel breiten Linien, werden die Pixel über und unter der Linie geplottet.
Ist die Steigung > 1 , wird einfach in horizontale Richtung geplottet, und das für jeden y-Wert. Pixel werden links und rechts zur Standardlinie hinzugefügt.
Eine weitere Möglichkeit um dicke Linien zu zeichnen ist, selbige als ausgefüllte Rechtecke darzustellen. Um das Rechteck zu zeichnen, bestimmen wir die Vertices in senkrechter Richtung zur Linienrichtung, so dass die Rechteckvertices von den Originallinienendpunkten um die halbe Linienbreite entfernt positioniert werden.
- **Line caps (S. 184):**
Problem: durch Verbreiterung einer Linie können horizontale und vertikale Enden entstehen \Rightarrow bei sehr dicken Linien fällt das auf. Dies kann man durch line caps lösen.
Butt cap: hat quadratische Enden, die senkrecht zur Linienrichtung stehen. Hat die Linie die Steigung m, so haben die Enden eine Steigung von $-1/m$. Jede parallele Linie wird dann zw. den 2 Endlinien dargestellt.
Round cap: wird dargestellt, indem zur Linie ein ausgefüllter Halbkreis gehängt wird. Der Kreismittelpunkt wird genau in der Mitte der dicken Linie festgelegt und der Kreis hat einen Durchmesser = der Breite der Linie
Projecting Square cap: dabei wird die Linie um butt caps erweitert.
- **Linien-Typ: durchgezogen, gestrichelt, punktiert, usw**

- Linien-Verbindungen (S. 149):



Bei Polylines ist das Problem, dass bei unterschiedlichen Steigungen der Linien Pixel an den Rändern zw. zwei Linien überbleiben, vor allem beim Übergang von horizontalen Pixelspans zu vertikalen. Man kann dicke Polylines zeichnen, indem man an den Endpunkten etwas hinzufügt.

Miter join: dabei werden die Enden solange erweitert, bis sie einander treffen (Spitze)

Round Join: dabei wird die Verbindung zwischen den beiden Linien mit einem Kreis abgerundet. Der Kreisdurchmesser entspricht der Linienbreite.

Bevel Join: dabei werden die Liniensegmente mit butt caps dargestellt, und die 3-eckigen Verbindungsstücke aufgefüllt.

21) Auf welche zwei grundsätzlichen Arten können Farben im Framebuffer gespeichert werden?

- Direktspeicherung:
RGB-Farbwerte werden direkt im Buffer gespeichert
wann immer eine spezielle Farbe im Programm gewünscht wird, wird diese Farbinformation im Framebuffer auf die zugehörige Pixelposition (die dargestellt werden soll) gesetzt.
Ein Minimum an Farbe kann durch 3 Bits Speicher pro Pixel erreicht werden. Jede der 3 Bit-Positionen kontrolliert die Intensität (entweder on oder off). Je mehr Bits/Pixel umso mehr Farben hat man zur Auswahl.
6Bits/Pixel: 2 Bits können für jeden Kanal verwendet werden, das erlaubt 4 versch. Intensitäten für jeden der 3 Farbkanäle (64 Farbwerte für jedes Pixel möglich). Bei einer 1024x1024 Auflösung benötigt man 3 MB Speicher.
- Color lookup tables (CLUTs):
die Farbwerte werden in einer separaten Tabelle gespeichert und an jeder Pixelposition referenziert ein Index auf den speziellen Farbwert. Im Framebuffer werden nur die Indizes auf die CLUT gespeichert.

22) Erklären Sie den Begriff soft-fill (tint-fill) (S. 195)

Ein Muster kann mit einer Hintergrundfarbe konstruiert werden, indem man den Transparenzfaktor ermittelt, der angibt, in welchem Verhältnis der Hintergrund mit der Objektfarbe gemischt werden soll, oder man verwendet logische bzw. Entscheidungs-Operatoren. (AND, OR, XOR...)

Eine Verwendung ist z.B. um die Füllfarbe weicher zu gestalten, wenn die Kanten unscharf sind, eine Andere die erneute einfärben einer Fläche, die zuvor mit einer semitransparenten Farbe gefüllt wurde, die aus einer Mischung von Vorder- und Hintergrund bestand.

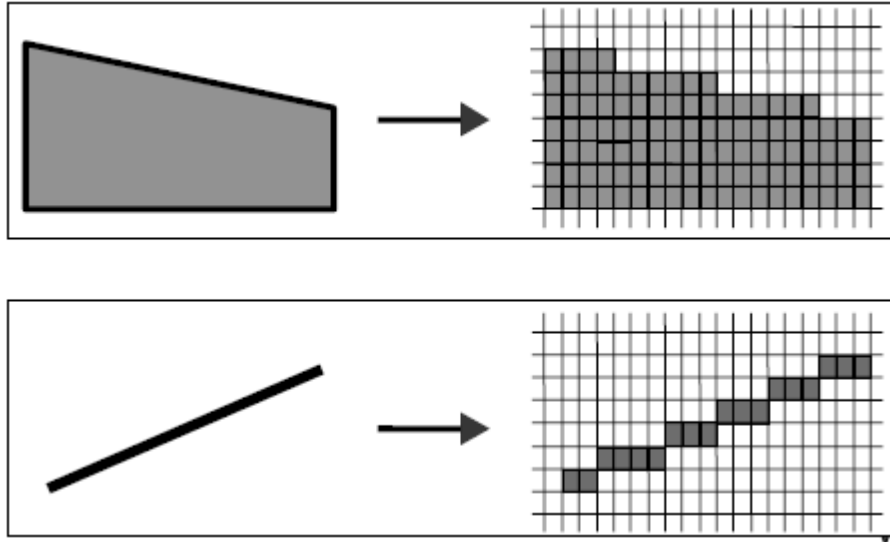
23) Welche Text-Attribute kennen Sie? (S. 211)

- Serif (mit „Schnörkseln“): ganze Textblöcke sind leichter lesbar
- Sans serif: einzelne Buchstaben sind leichter lesbar \Rightarrow besser für Beschriftungen und kurze Überschriften (zb. Arial, Times New Roman, ...)
- Variationen: Unterstrichen, **fett**, *kursiv*, mit Outline, mit Schatten
- Textgröße: durch Veränderung von Höhe und/oder Breite (in points angegeben), Verschiedene Schriften mit der gleichen Pointsgröße, können verschiedene Textgröße haben. Die Distanz zwischen bottomline und topline ist für alle gleich, aber die Breite ist unterschiedlich.
- Abstand: Proportionally spaced fonts: Kleinbuchstaben schmaler als Großbuchstaben.
- Charakterhöhe: wird definiert durch baseline und capline
- Kerned Characters: stehen über das Limit hinaus. zb f ist breiter als normale Kleinbuchstaben, oder g, j, p, q stehen unten aus der baseline
- Ausrichtung: Die Schrift kann vertikal, horizontal oder diagonal dargestellt werden. Dazu benötigt man den character up vector. Die Buchstaben werden dann so dargestellt dass die Orientierung von baseline zu capline der Vektorrichtung entspricht.
- Ausrichtung im Absatz: rechts-, linksbündig, zentriert oder Blocksatz

24) Was versteht man unter (Anti-)Aliasing? (S. 214)

<http://en.wikipedia.org/wiki/Aliasing>, <http://en.wikipedia.org/wiki/Anti-aliasing>

Durch die Darstellung von Linien in einem Raster, müssen die Koordinaten der Punkte auf Integerwerte gerundet werden. Der Stufeneffekt der dabei entsteht wird auch „jaggies“ genannt. Diese Informationsverzerrung aufgrund des low-frequency-samplings (undersampling) heißt aliasing. Antialiasing wird verwendet, um das undersampling auszugleichen.



http://en.wikipedia.org/wiki/Nyquist-Shannon_sampling_theorem

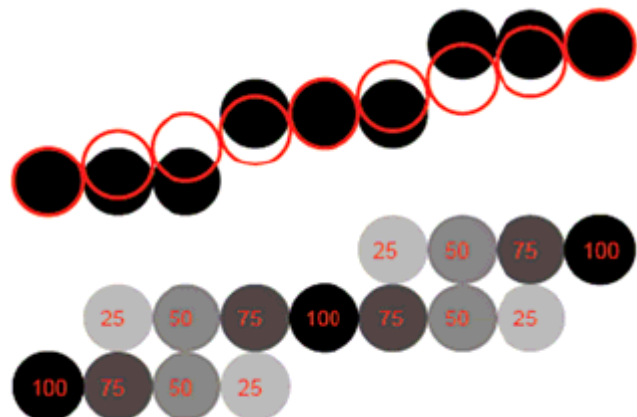
Nach dem Abtasttheorem von Shannon, kann ein Signal nur wiederhergestellt werden, wenn die Abtastfrequenz mindestens doppelt so hoch ist, wie die Frequenz des Signals, das abgetastet werden soll. $f_s = 2 \cdot f_{\max}$ (auch Nyquist Limit bzw. Nyquist sampling frequency genannt).

Eine andere Art um dies auszudrücken ist, dass das Sampling-Intervall nicht größer als die Hälfte des Cycle-Intervalls sein darf.

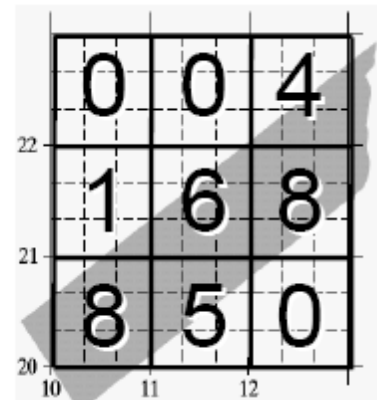
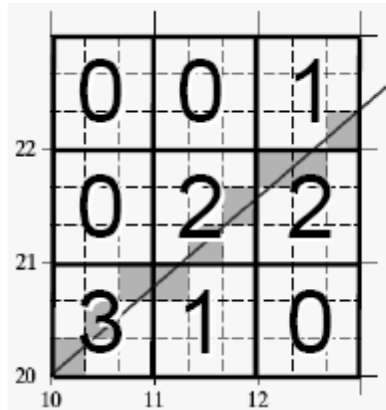
$$\Delta x_s = \frac{\Delta x_{\text{cycle}}}{2} \quad \text{with} \quad \Delta x_{\text{cycle}} = 1 / f_{\max}$$

25) Welche Anti-Aliasing Strategien gibt es und wie funktionieren Sie? (S. 215 ff.)

Bei Rastersystemen, die mehr als 2 Intensitätslevels darstellen können, können wir Antialiasingmethoden anwenden, um die Intensitäten der Pixel zu verändern. Hierbei werden die Farbintensitäten der Pixel an den Rändern der darzustellenden Objekte vermindert und dadurch weichere Kanten und eine weniger treppenförmige Darstellung erreicht.



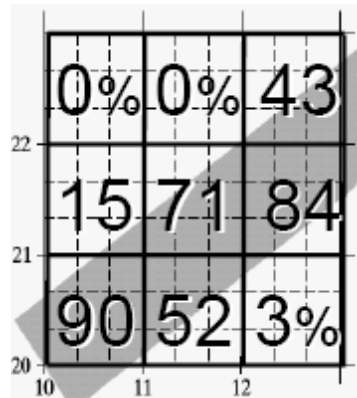
- Supersampling: hierbei wird die Samplingrate erhöht und die Anzeige behandelt, als ob sie ein viel feineres Raster hätte. Nun kann man die Vielzahl von Samplingpoints entlang des feineren Rasters dazu nutzen, um einen entsprechenden Intensitätslevel für jeden Pixel der Anzeige zu berechnen. Objekte werden also in einer hohen Auflösung gesampled und in einer niedrigeren Auflösung dargestellt.



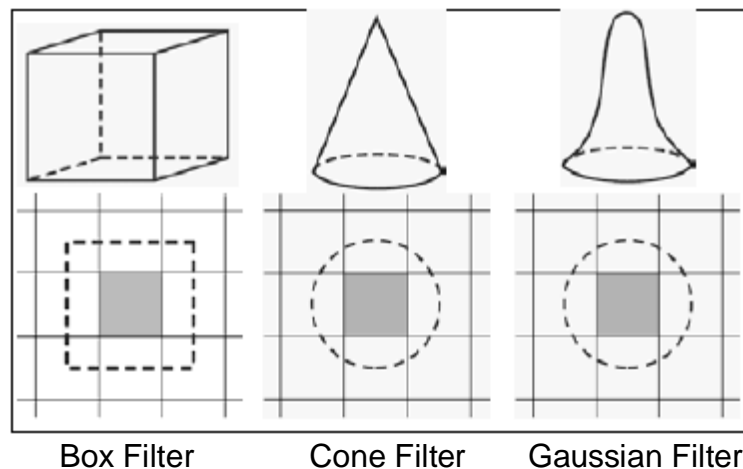
- Pixel-Weighting Masks: Supersampling-Algorithmen werden oft so implementiert, dass sie Subpixeln, die näher am Zentrum des Pixelfeldes liegen, mehr Gewichtung geben, da man annehmen kann, dass diese Subpixel wichtiger für die Bestimmung der Gesamtintensität eines Pixels sind. Dies wird auch als Maske für Subpixel-Gewichte bezeichnet. Diese Masken werden oft erweitert, um die Subpixel der Nachbarpixel mit einzubeziehen, damit die Intensitäten über angrenzenden Pixel gemittelt werden können.

1	2	1
2	4	2
1	2	1

- Area Sampling: man berechnet die Pixelintensität, indem man die Überschneidungsflächen von jedem Pixel mit den Objekten, die angezeigt werden sollen, ermittelt. Diese Überschneidungsflächen bekommt man, indem man bestimmt, wo sich die Objektgrenzen mit den Pixelgrenzen schneiden.

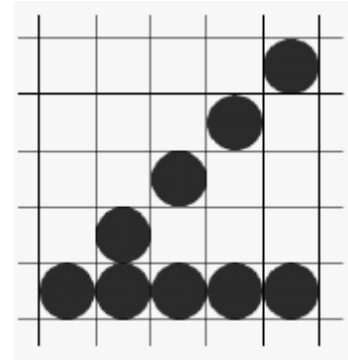


- Filtering Techniques: Eine exaktere Methode zum Antialiasen von Linien ist die Anwendung von Filtertechniken. Diese Methode ist ähnlich der Anwendung von gewichteten Pixelmasken, aber jetzt stellen wir uns eine durchgehende „Gewichtungsoberfläche“ vor, die die Pixel bedeckt. Es wird über die Pixeloberfläche integriert, um die gewichtete Durchschnittsintensität zu erhalten. Um die Berechnungen zu reduzieren, werden üblicherweise Table lookups zur Evaluierung des Integrals verwendet.



- Pixel Phasing: Bei Rastersystemen die Subpixelpositionen innerhalb des Anzeigegitters adressieren können, kann Pixelphasing zum Antialiasen eines Objekts verwendet werden. Stufen entlang eines Linienpfades oder Objekts werden geglättet, indem der Elektronenstrahl zu einer besser passenden Position („Mikropositionierung“), die durch die Objektgeometrie spezifiziert wird, verschoben wird. Ein Elektronenstrahl wird normaler Weise um $\frac{1}{4}$, $\frac{1}{2}$, oder $\frac{3}{4}$ eines Pixeldiameter verschoben, um Punkte näher am echten Pfad einer Linie oder Objektkante auszugeben.

- **Compensating for Line Intensity Differences:** Das Glätten kompensiert auch einen anderen Rastereffekt: Linien mit unterschiedlichen Steigungen erscheinen unterschiedlich intensiv, das z.B. bei einer horizontalen Linie mehr Pixel ausgegeben werden, als bei einer diagonalen. Dadurch hat die Diagonale eine geringere Intensität pro Einheit als die Horizontale. Dies kann behoben werden, indem der Zeichenalgorithmus die Intensität abhängig von der Steigung der Linie anpasst. Hierbei würden horizontale und vertikale Linien am schwächsten, Linien mit 45° Steigung am stärksten dargestellt werden. Wenn aber Antialiasing-techniken verwendet werden, werden die Intensitäten automatisch kompensiert.



- **Antialiasing Area Boundaries:** Die schon diskutierten Antialiasing-Methoden können auch auf Objekt- und Flächengrenzen angewandt werden. Wir können diese Prozeduren in einen Scanline-Algorithmus integrieren, der die Kanten der Objekte beim Zeichnen glättet. Wenn das System die Repositionierung von Pixeln unterstützt, kann das Antialiasing mit der Neupositionierung der Randpixel erreicht werden. Andere Methoden passen die Intensität jedes Pixels entsprechend der Prozent der Pixelfläche, die innerhalb der Grenzen liegen, an. Supersamplingmethoden können angewandt werden, indem die gesamte Fläche sub-unterteilt wird und die Subpixel innerhalb der Flächengrenzen bestimmt werden.

26) Wofür verwendet man super-sampling und area-sampling, und was ist der Unterschied?

Beides sind Konzepte für das Antialiasing

- **Supersampling (S. 215)**
Für eine Grauwertdarstellung eines Liniensegments können wir jedes Pixel in eine bestimmte Anzahl von Subpixeln aufteilen und die Subpixel zählen, die entlang des Linienpfades liegen. Der Intensitätslevel des Pixels wird dann auf einen Wert gesetzt, der proportional zu dieser Subpixelzählung ist. So wird die Intensität der Linie auf eine größere Anzahl an Pixel aufgeteilt und der Treppeneffekt wird durch die etwas verschwommene Darstellung des Linienpfades geglättet. Für mehr unterschiedliche Intensitätslevels zum Antialiasing, erhöht man die Anzahl von Samplingpositionen entlang jedes Pixels. Diese Berechnungen basieren auf der Annahme, dass die Linie keine Dicke aufweist. Üblicherweise haben aber Linien bei der Darstellung ungefähr eine Dicke von 1 Pixel.
Wenn wir nun eine endliche Dicke der Linie mit einbeziehen, können wir die jeweilige Pixelintensität proportional zu der Anzahl der Subpixel innerhalb des Polygons, das die Linienfläche repräsentiert, setzen. Ein Subpixel kann dabei als innerhalb der Linie angesehen werden, wenn die untere, linke Ecke sich innerhalb des Polygons befindet. Der Vorteil dieser Methode ist, dass die Anzahl

der Intensitätslevels für jedes Pixel gleich der Anzahl der Subpixel innerhalb der Pixelfläche sind. Ein weiterer Vorteil von Supersampling mit einer Liniendicke ist, dass die totale Linienintensität über mehr Pixel verteilt wird.

Für ein Farbdisplay können wir diese Methode so modifizieren, dass sie die Hintergrundfarbe mit einbezieht. Eine Linie könnte mehrere verschiedene Hintergrundfarben schneiden und wir können die den Durchschnitt der Subpixelintensitäten verwenden um die korrekte Farbe zu berechnen.

Ein Nachteil ist der hohe Berechnungsaufwand für die Identifizierung der Subpixel entlang des Linienpfades. Dies wird zusätzlich durch die Positionierung der Liniengrenzen in Relation zum Linienpfad erschwert. Diese Positionierung hängt von der Steigung der Linie ab. Bei einer Linie mit 45° ist der Linienpfad zentriert zum Polygonfeld; aber bei jeder horizontalen/ vertikalen Linie soll der Pfad die Polygongrenze sein. Für Linie mit einer Steigung $|m| < 1$ ist der mathematische proportional näher an der unteren Polygongrenze positioniert, für Linien mit $|m| > 1$ näher an der Oberen.

- Area Sampling (S. 217)

Sampling wird durch das Setzen der Pixelintensität proportional zur Fläche der Überlappung des Pixels mit der begrenzt-dicken Linie durchgeführt. Die Linie kann dabei als Rechteck behandelt werden, und die Sektion der Linienfläche zwischen 2 adjazenten vertikalen/horizontalen Bildschirmgitterlinien als Trapezoid. Nun können die Überlappungsflächen der Pixel dadurch berechnet werden, wieviel jedes Pixel in der vertikalen/horizontalen Linie vom Trapezoid überdeckt wird.

Eine Methode um die Überlappungsfläche zu schätzen ist, mit Supersampling die Anzahl der Subpixel innerhalb der Liniengrenzen zu zählen, da diese ca. der Fläche entsprechen. Diese Schätzung wird durch Verfeinern des Pixelgitters verbessert.

Bei Farbanzeigen werden die Flächen der Pixelüberlappung mit anderen Farbregionen berechnet und für die endgültige Pixelfarbe das Mittel über die verschiedenen Überlappungsflächen genommen.

Kapitel 5 – Two-Dimensional Geometric Transformations (S. 230)

27) Welche Arten von Transformationen kennen sie (2D Transformationsmatrizen)?

- Translation: $\begin{pmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{pmatrix}$
- Drehung: $\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- Skalierung: $\begin{pmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{pmatrix}$

- Scherung: $\begin{pmatrix} 1 & shx & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
- Reflexion: $\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ um x Achse, $\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ um y Achse

27) Was sind homogene Koordinaten und warum verwendet man sie? (S. 238)

http://de.wikipedia.org/wiki/Homogene_Koordinaten

http://en.wikipedia.org/wiki/Homogenous_coordinates

Homogene Koordinaten dienen häufig zur Umrechnung von Koordinatensystemen. Um Rundungsfehler zu vermeiden und Berechnungsaufwand massiv zu reduzieren, ist es vorteilhaft, mehrere aufeinander folgende Matrixtransformationen (Translation, Drehung, Skalierung, Scherung) zu einer einzigen Transformationsmatrix zusammenzufassen (möglich auf Grund der Assoziativität von Matrizen). Hinderlich ist jedoch, dass die drei erstgenannten Operationen eine Matrizenmultiplikation erfordern, die Translation jedoch eine Addition.

Um eine Translation ebenfalls als Multiplikation berechnen zu können, wird der Raum um eine weitere Dimension erweitert. Eine Translation im dreidimensionalen Raum lässt sich nun durch eine Matrizenmultiplikation mit einer 4x4-Matrix beschreiben.

Zu beachten ist, dass die Matrizen zwar assoziativ sind, aber nicht kommutativ. D.h. es ist nicht egal, ob ich ein Objekt zuerst drehe und dann verschiebe, oder umgekehrt.

28) Was ist eine affine Transformation, und welche Eigenschaften hat sie? (S. 283)

http://de.wikipedia.org/wiki/Affine_Transformation

Affine Transformation wird auch als affine Abbildung oder lineare Transformation bezeichnet. Es ist dies eine Transformation zwischen zwei Vektorräumen die Kollinearitäten und Abstandsverhältnisse bewahrt. Bewahrung der Kollinearität bedeutet, dass die Bilder von Punkten, die auf einer Geraden liegen (d.h. kollinear sind), wieder auf einer Geraden liegen. Ebenso sind die Bilder paralleler Geraden wieder parallel. Beispiele für affine Transformationen sind Translation, Rotation, Skalierung, Reflexion und Scherung. Jede generelle Transformation lässt sich durch Komposition aus diesen fünf Grund-Transformationen zusammenstellen. Affine Transformationen, die nur aus Translation, Rotation und Reflexion bestehen, bewahren auch die Winkel und die Längen.

Kapitel 6 – Two-Dimensional Viewing (S. 296)

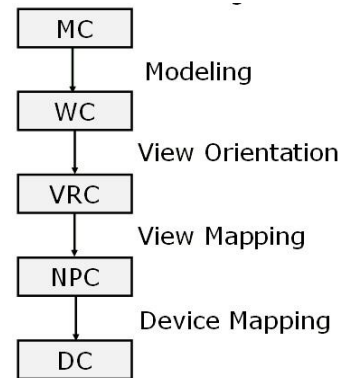
29) Was versteht man unter der Grafikpipeline (viewing pipeline) und aus welchen Komponenten besteht sie? (S. 299)

http://en.wikipedia.org/wiki/Graphics_pipeline

<http://de.wikipedia.org/wiki/Rendering-Pipeline>

Die Viewing Pipeline ist die Abbildung des Vorganges, der ein Modell einer realen Welt auf einem Gerät (Bildschirm) abbildet.

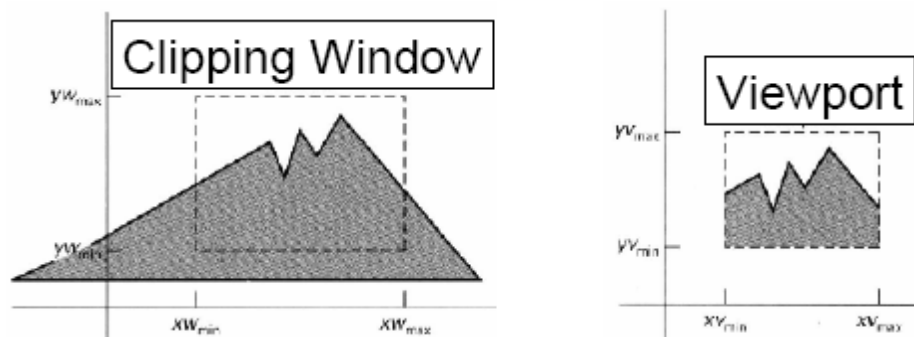
- Modeling: MC \Rightarrow WC: Konstruiere eine Weltkoordinaten-Szene unter Verwendung von Modell-Koordinaten-Transformation – Beschreiben der Szene in Weltkoordinaten
- View Orientation: WC \Rightarrow VC: Wandle Welt-Koordinaten in Viewing-Koordinaten um – Szene aus der Sicht der Kamera
- View Mapping: VC \Rightarrow NVC: Wandle VCs in normalisierte VCs mittels Window-Viewport-Transformation – Überführe die Szene in einen Einheitswürfel
- Device Mapping: NVC \Rightarrow DC: Wandle die NVCs in Device Koordinaten um – Projiziere die Szene auf den Bildschirm



Bei der 3D-Graphik-Pipeline kommt noch hinzu, dass man das „Verstecken“ von Objekten (nicht sichtbare Objekte können außer Acht gelassen werden) sowie die Licht- und Schatteneffekte (unter Berücksichtigung der vorhandenen Lichtquellen) miteinbeziehen muss.

30) Was ist die window-viewport-Transformation? (S. 217)

Bei der window-viewport-Transformation wird ein Teil (Ausschnitt) der in Welt-Koordinaten vorhandenen Objekte in die viewport-Koordinaten transformiert und dargestellt. Dabei bestimmt das clipping-window welcher Teil angezeigt wird und der viewport, wo der ausgewählte Teil betrachtet wird.



Die Transformation wird in zwei Schritten erreicht

1. Translation des view-Ursprung in den Welt-Ursprung
2. Rotation, sodass die Achsen sich überlappen

Bei der Transformation der Punkte bleiben die relativen Positionen erhalten.

31) Welche grundsätzlichen Möglichkeiten für das Clipping gibt es?

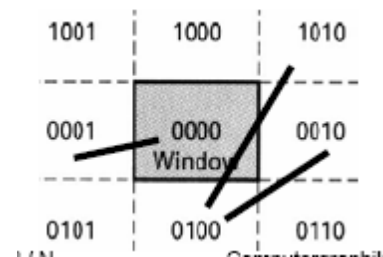
http://en.wikipedia.org/wiki/Clipping_%28computer_graphics%29

Clipping stellt sicher, dass nur die Teile der Objekte dargestellt werden müssen, die nach sichtbar sind. Dazu gibt es grundsätzlich 3 Möglichkeiten:

- analytisch = in Welt-Koordinaten (reduziert den Aufwand bei WC \Rightarrow DC Transformation)
- bei der Rasterkonvertierung = als Teil des Rasterisierungs-Algorithmus (effizient für komplexe Primitive)
- auf Pixel-Ebene = primitivster Algorithmus mit größtem Aufwand

32) Was sind die wesentlichen Schritte beim Cohen-Sutherland line clipping Algorithmus? (S. 317)

- Jedem Endpunkt des clip windows wird eine 4-stelliger region code zugeordnet
- Entscheiden, welche Linien komplett außerhalb des clip windows und welche komplett innerhalb liegen \Rightarrow können vernachlässigt werden. Entscheidung kann anhand einer „OR“ und „AND“-Verknüpfung der Bitmuster der region codes in denen die Endpunkte der Linie liegen, durchgeführt werden
- Alle verbleibenden Linien müssen mit den Randlinien des clip windows (links, rechts, unten, oben) geschnitten werden. Dabei wird der Teil der Linie außerhalb des clip windows vernachlässigt.
- intersection test wird bis zu 4x durchgeführt



33) Wie passiert das clippen an beliebigen Polygonen? (S. 316)

Die Linien des Polygons werden nacheinander gegen das clip window geschnitten (links, rechts, unten, oben) und die entstehenden Schnittpunkte als neue Punkte in das Polygon aufgenommen. Dabei entstehen auch neue Kanten, die die neuen Punkte entlang des Randes des clip windows miteinander verbinden.

Ein Weg hierfür ist jeweils an jeder Clipping-grenze eine neue Vektorenliste zu generieren, und dann diese an die nächste Clipping-grenze weiterzugeben. Hierbei sollte zum Schluss die Vektorenliste des geclippten Polygons heraus kommen.

34) Was sind die wesentlichen Schritte beim Algorithmus von Sutherland und Hodgman? (S. 238)

Das Polygon wird gegen die 4 clip-window-Kanten (links, rechts, unten, oben) getrennt geschnitten. Um Zwischenergebnisse zu verhindern, werden die Schnittpunkte rekursiv durch die Tests geschickt (boundary clipping pipeline) und somit entsteht nur eine Ergebnisliste.



Beim Clipping eines Vektorenpaars gibt es 4 Mögliche Situationen, die unterschieden werden müssen.

- I. Beide Punkte sind innerhalb der Fenstergrenze: Nur der 2. Eckpunkt wird zum nächsten Clipper weitergegeben.
- II. Beide Punkte sind außerhalb der Fenstergrenze: Kein Eckpunkt wird zum nächsten Clipper weitergegeben.
- III. Der 1. Punkt ist innen, der 2. Punkt außen: Nur der Schnittpunkt des Polygons mit der Fenstergrenze wird weitergegeben.
- IV. Der 1. Punkt ist außen, der 2. Punkt innen: Sowohl der Schnittpunkt des Polygons mit der Fenstergrenze, als auch der 2. Eckpunkt werden weitergegeben.

Der letzte Clipper in dieser Serie generiert anschließend die neue Liste der Eckpunkte, die das geclippte Polygon beschreiben.

Ein Problem (S. 335) bei dieser Methode stellt das clippen konkaver Polygone dar, da die durch das clippen entstehenden separaten Polygone durch eine Linie verbunden werden. Es gibt zwei Möglichkeiten um dies zu verhindern:

- I. Aufteilung der Polygone vor dem Clipping die Polygone in konvexe Polygone
- II. Überprüfung der finalen Eckpunktliste auf multiple Schnittpunkte entlang irgendeiner Fenstergrenze. Wenn mehr als 2 Schnittpunkte entdeckt werden, können wir die Liste in 2 oder mehrere einzelne Listen aufteilen, um so eine korrekte Beschreibung des geclippten Polygons zu erreichen.

35) Welche Möglichkeiten des text-clipping kennen Sie, und was sind die Unterschiede?

Es gibt mehrere Methoden um Buchstaben zu clippen. Diese hängen vor allem davon ab, wie die Buchstaben in der jeweiligen Applikation generiert werden und was für Anforderungen wir für die Anzeige selbiger haben.

- „all-or-none string-clipping“-Methode: Einfachste Methode die eine bounding box um den Text benutzt. Wenn der String zur Gänze im sichtbaren Bereich liegt wird er dargestellt, ansonsten nicht. Dies wird durch Überprüfung der Koordinatenmaxima der bounding box erreicht.

- „all-or-none character-clipping“-Methode: Sie ist gleich wie die obige, nur dass die bounding box um einzelne Buchstaben gelegt wird.
- „individual character“-Method: Bei dieser Methode werden die Buchstaben ähnlich wie Polygone behandelt. Buchstaben die durch ihre Außenlinien definiert sind (outline font) werden gleich wie Polygone geclippt. Bei Buchstaben die als Bitmaps vorliegen (bitmap font), werden die relativen Positionen der Pixel in den Buchstaben mit den Clipping-grenzen verglichen.

Kapitel 7 – Three Dimensional Viewing (S. 344)

36) Was ist eine Parallelprojektion und welche Eigenschaften hat sie? (S. 346)

http://de.wikipedia.org/wiki/Projektion_%28Geometrie%29

- Bei der Parallelprojektion sind eine Projektionsebene und eine Projektionsrichtung gegeben. Den Bildpunkt eines beliebigen Punktes im Raum erhält man dadurch, dass man die Parallele zur Projektionsrichtung durch diesen Punkt mit der Projektionsebene zum Schnitt bringt.
- Geraden werden durch eine Parallelprojektion im Allgemeinen wieder auf Geraden abgebildet. Das gilt jedoch nicht für Parallelen zur Projektionsrichtung, da diese in Punkte übergehen. Die Bildgeraden von parallelen Geraden sind - soweit definiert - ebenfalls parallel zueinander. Die Länge einer Strecke bleibt nur dann erhalten, wenn diese parallel zur Projektionsebene verläuft; in allen anderen Fällen erscheinen Strecken in der Projektion verkürzt. Auch die Größe eines projizierten Winkels stimmt normalerweise nicht mit der Größe des ursprünglichen Winkels überein. Aus diesem Grund wird ein Rechteck im Allgemeinen auf ein Parallelogramm abgebildet, aber nur in Ausnahmefällen auf ein Rechteck. Ähnliches gilt für Kreise, die im Allgemeinen in Ellipsen übergehen.
- Ein wichtiger Spezialfall ist die orthogonale (senkrechte) Parallelprojektion. Sie ist dadurch gekennzeichnet, dass Projektionsrichtung und Projektionsebene zueinander senkrecht sind. Die orthogonale Parallelprojektion entspricht einer Fotografie mit einem starken Teleobjektiv.

37) Was versteht man unter "depth cueing", und wofür wird es verwendet? (S. 346)

Die Information für die „Tiefe“ ist essentiell für das räumliche Erfassen einer Szene. Aufgrund der Blickrichtung kann man berechnen welche Seite dem Blickpunkt zugewandt ist (front face) und welche abgewandt (back face). Objekte die näher am Blickpunkt liegen, verdecken Objekte die weiter entfernt sind. Genauso verdecken einzelne nahe Polygon-Flächen solche, die weiter entfernt sind.

Kapitel 8 – Three-Dimensional Object Representations (S. 402)

38) Was versteht man unter Polygonflächen (B-Rep-Listen)? (S. 403)

http://en.wikipedia.org/wiki/Polygon_modeling

- Polygonfläche: ist eine Boundary Representation („B-Rep“) = ein Set von Polygonen, die ein Objekt-Inneres einschließen
- Die Speicherung kann in polygon tables (sog. „B-Rep“ Listen) erfolgen.
 - Haben geometrische Listen (beinhalten vertex-Koordinaten, sowie Parameter zur räumlichen Identifizierung des Polygons und werden in 3 Listen gespeichert \Rightarrow vertex table, edge table und surface-facet table) und Attribut-Listen (Parameter für den Grad der Transparenz, Reflexion der Oberfläche)
 - Ermöglichen Konsistenz- und Vollständigkeit-Checks
- Polygonflächen können durch eine Ebenengleichung dargestellt werden
 - $Ax+By+Cz+D = 0$
 - Kartesische Koeffizienten: A,B,C,D
 - Normale auf die Ebene = N (A,B,C)
 - Distanz zum Ursprung = D
- Polygon-Meshes:
 - http://en.wikipedia.org/wiki/Polygon_mesh
 - Effiziente Datenstruktur für zusammenhängende Polygone (Einsparung von Kanten und Ecken)
 - Kann entweder aus Drei- oder Vierecken bestehen (triangle strip bzw. quadrilateral mesh)

39) Was versteht man unter einem front- und back-face eines Polygons? Wie kann man diese mathematisch berechnen?

http://en.wikipedia.org/wiki/Back-face_culling

- Die Objektseite, die in das Innere eines Objektes schaut, nennt man back-face und die sichtbare, nach aussen zeigende Seite heißt front-face.
- Alles, was vor dem Polygon liegt, ist für das front-face sichtbar; alles, was hinter der Polygon-Fläche liegt, ist für das back-face sichtbar
- Man kann dies für jeden Punkt mittels der Ebenengleichung berechnen.
 - $Ax+By+Cz+D = 0 \Rightarrow$ Punkt (x,y,z) liegt in der Ebene
 - $Ax+By+Cz+D < 0 \Rightarrow$ Punkt liegt hinter der Ebene
 - $Ax+By+Cz+D > 0 \Rightarrow$ Punkt vor der Ebene
 - Die Orientierung einer Polygonsoberfläche im Raum kann durch den Normalvektor bestimmt werden. Der Normalvektor zeigt in die Richtung vom Inneren eines Objekts nach aussen. \Rightarrow vom back-face in Richtung front-face

40) Was sind Splines und welche Eigenschaften haben sie? (S. 420)

<http://de.wikipedia.org/wiki/Spline>

Ein Spline ist ein Begriff aus der numerischen Mathematik und bezeichnet ein stückweises Polynom, das für Polynome n -ten Grades an den Punkten, an denen jeweils zwei Polynome aufeinandertreffen, $n-1$ Mal stetig differenzierbar ist. Sind die einzelnen Polynome alle linear, so nennt man den Spline linear, analog gibt es quadratische, kubische usw. Splines. Der Begriff stammt aus dem Schiffbau: eine lange dünne Latte (Straklatte), die an einzelnen Punkten durch Nägel fixiert wird, biegt sich genau wie ein kubischer Spline mit natürlicher Randbedingung.

41) Was ist der Unterschied zwischen Interpolation und Approximation bei Splines? (S. 420)

- Kontroll-Punkte:
 - Set von Punkten, die die generelle Form der Spline (Kurve) vorgeben
 - Parametrisierung der Spline durch Veränderung der Position der Kontroll-Punkte
 - Transformation der Spline durch Transformation der Kontroll-Punkte
- Interpolation:
 - Die Kontroll-Punkte liegen auf der Spline (Kurve)
- Approximation:
 - Die Kontroll-Punkte geben die ungefähre Form der Spline (Kurve) vor liegen aber nicht (unbedingt) auf ihr

42) Welche Arten der Kontinuität bei Splines kennen Sie? Wie sieht der Unterschied zwischen C1- und G1-Stetigkeit aus? (S. 421 ff.)

- Parametrische Kontinuitäts-Bedingungen (C_n):
 - Ableitungen an den Verbindungspunkten sind gleich
 - C_0 (zero order): die Kurven müssen sich treffen
 - C_1 (first order): die erste Ableitung (Tangente) muss im Verbindungspunkt gleich sein
 - C_2 (second order): die zweite Ableitung muss gleich sein
- Geometrische Kontinuitäts-Bedingungen (G_n):
 - Ableitungen an den Verbindungspunkten sind proportional
 - G_0 (zero order) = C_0
 - G_1 (first order): die Richtung der Tangenten aber nicht unbedingt ihre Steigungen müssen gleich sein
 - G_2 (second order): auch die zweite Ableitung muss gleich sein

43) Was sind kubische Splines und welche Eigenschaften haben sie? (S. 425)

- Grad des Kontrollpolynoms = 3
 - nach der Interpolation und Erfüllung der restlichen Bedingungen bleiben noch zwei Freiheitsgrade, die für gewöhnlich dafür genutzt werden, Randbedingungen anzugeben. Je nach Art der Bedingungen heißt der Spline dann
 - natürlich (freie Ränder, Krümmung gleich Null),
 - periodisch (linker und rechter Rand gehen stetig ineinander über),
 - vollständig (Steigung an den Rändern wie bei interpolierter Funktion) oder
 - allgemein.
- Interpolation wird zur Generierung der Kurve verwendet
- C2-Kontinuität zwischen den Teilstücken muss gegeben sein
- Nachteil: Wenn sich die Position einer der Kontroll-Punkte verändert, verändert sich automatisch die ganze Kurve (keine „lokale Kontrolle“)

44) Was sind Bézier-Kurven und welche Eigenschaften haben sie? (S. 432)

- Grad des Kontrollpolynoms = $n-1$, wobei n die Anzahl der Kontroll-Punkte angibt
- Kontroll-Punkte werden approximiert, wobei der erste und der letzte Kontroll-Punkt immer auf der Kurve liegt (= Eckpunkte werden interpoliert)
- C1-Kontinuität zwischen den Teilstücken muss gegeben sein
- Kurve liegt innerhalb der convex hull, die durch die Kontroll-Punkte aufgespannt wird
- Bézier-Kurven können geschlossen sein ($P_0 = P_n$)
- Die Kurve kann näher an einen Punkte „gezogen“ werden indem man an dieser Stelle einen multiplen Punkt erstellt ($P_i = P_j$)
- Kontroll-Punkte haben (wie bei den kubischen Splines) globale Auswirkungen

45) Welche Eigenschaften haben B-Splines und was sind die Unterschiede zu Bézier-Kurven? (S. 432)

- Werden am öftesten zur Approximierung von Splines verwendet
- Vorteile gegenüber Bézier-Kurven:
 - Der Grad des Trägerpolynoms ist unabhängig von der Anzahl der Kontroll-Punkte
 - B-Splines erlauben „lokale Kontrolle“ über die Form der Kurve bzw. der Fläche
 - Aus der „lokalen Kontrolle“ ergibt sich, dass der Aufwand um eine B-Spline zu zeichnen linear von n abhängig ist \Rightarrow es ist nicht notwendig Splines mit großer Anzahl von Kontroll-Punkten zu teilen
- Nachteil: B-Splines sind komplexer als Bézier-Kurven

46) Was sind NURBS (= Non-Uniform Rational B-Splines)? (S. 454)

<http://en.wikipedia.org/wiki/NURBS>

- Non-uniform: der Abstand zwischen den Kontroll-Punkten ist nicht gleich
- Rational: die Punkte sind gewichtet (mittels h_i)
- Vorteile rationaler (gegenüber non-rationalen) Splines:
 - Ermöglichen eine genaue Repräsentation von quadratischen Kurven (conics). Non-rationale Splines können solche Kurven nur approximieren
 - Sie sind invariant im Bezug auf perspektivische viewing-Transformationen
⇒ allein aus der Transformation der Kontroll-Punkte ergibt sich die korrekte transformierte Ansicht der Kurve

47) Was sind CSG-bäume? (S. 474)

http://en.wikipedia.org/wiki/Constructive_solid_geometry

- CSG = Constructive Solid Geometry
- Beruht auf binären Mengenoperationen (Vereinigung, Schnitt, Differenz) von 3D-Objekten
- Jedes Objekt kann mittels einfacher geometrischer Strukturen durch die Mengenoperationen dargestellt werden
- Datenstruktur zur Speicherung: Binär-Baum, der rekursiv abgearbeitet wird
- Vorgehensweise: Mittels ray-casting werden von einer sogenannten „firing plane“ strahlen auf die Objekte „abgeschossen“, die modelliert werden sollen. Daraus werden die Schnittpunkte zwischen den einzelnen Flächen bestimmt und die Objekte nach ihrer Entfernung sortiert.
- Um das Volumen eines (zusammengesetzten) Objekts zu bestimmen, wird die „firing plane“ in Teilflächen A_{ij} unterteilt. Das Volumen des Objekts kann anschließend aus der Überlappung des Objekts mit den Teilflächen und dem Abstand der Schnittpunkte der Strahlen mit dem Objekt geschätzt werden.
- Vorteile: exakte Repräsentation, niedriger Speicherverbrauch und triviale Kombinationen und Transformationen
- Nachteile: hoher Rendering-Aufwand

48) Was ist ein Quadtree? (S. 477)

<http://en.wikipedia.org/wiki/Quadtree>

- Ist eine Datenstruktur, die für das quadtree encoding verwendet wird
- Stellt eine strukturierte Fläche (2-dim.) in einem Baum vom Grad = 4 dar
- Der Baum wird solange unterteilt, bis alle 4 Quadranten den gleichen wert haben
- Vorteil: effiziente Speicherung von Daten

49) Was ist ein Octree? (S. 477)

<http://en.wikipedia.org/wiki/Octree>

- Stellt einen strukturierten Raum (3-dim.) in einem Baum vom Grad = 8 dar (analog zum Quadtree für 2 Dimensionen)
- Vorteile:
 - Mengenoperationen sind einfach durchzuführen
 - Schnelles Rendering
 - Räumliches Suchen ist möglich
- Nachteil:
 - Transformationen sind (bis auf einige wenige) schwierig durchzuführen
 - Niedrige Bildqualität
 - Hohe Memory-Kosten

Kapitel 9 – Visible Surface Detection Methods (S. 528)

50) Welche zwei Kategorien von visible surface detection methods kennen Sie (S. 529)

http://en.wikipedia.org/wiki/Visibility_problem

- object-space Methoden: vergleichen Objekte und Teile von Objekten miteinander im Rahmen der Szenen-Definition als Ganzes um festzustellen welche Flächen (als Ganzes) als sichtbar eingestuft werden sollen.
- image-space Methoden: Hierbei wird die Entscheidung über die Sichtbarkeit pixel für pixel getroffen für jeden Bildpunkt auf der Projektions-Ebene.

51) Was ist „back-face detection“ und auf was für Objekte darf man es anwenden? (S. 530)

- Back-face detection ist eine schnelle und simple object-space Methode.
- Man berechnet den Normalevektor N für die Polygonebene. Ist V_{view} der Vektor in Blickrichtung von unserer Kameraposition ausgehend, dann ist das Polygon ein back-face, wenn $V_{view} * N > 0$ ist.
- Also wenn der Vektor N in die gleiche Richtung wie unser Sichtvektor zeigt, dann ist die Fläche ein back-face, d.h. der Normalvektor der Fläche muss in unsere Richtung zeigen, damit wir eine Fläche sehen können.
- Darf nur auf konvexe Polyhedra angewendet werden (bei konkaven ist entweder eine Umwandlung notwendig, oder zusätzliche Methoden)
- Eliminiert ca. 50% aller Flächen von der weiteren Verarbeitung

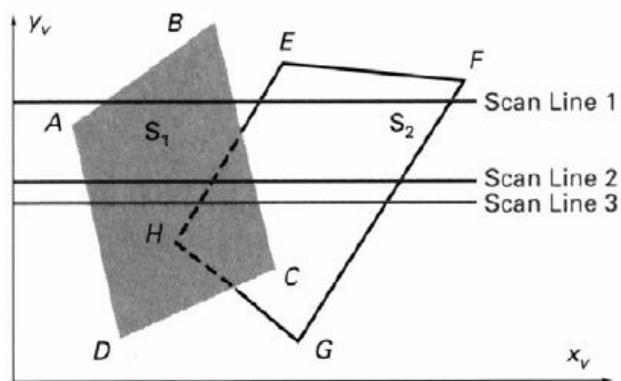
52) Was sind die wesentlichen Schritte beim "depth-buffer" (z-buffer)-Algorithmus? Was ist ein z-Puffer und wozu dient er? (S. 532)

<http://en.wikipedia.org/wiki/Z-buffer>

- Der z-Buffer-Algorithmus ist eine image-space Methode, die mittels einer Hardware-Implementierung umgesetzt wird. Es gibt keine Sortierung.
- Beim z-Buffer-Algorithmus werden die z-Werte für jeden Pixel einer Oberfläche mit den z-Werten der anderen Oberflächen verglichen, und dadurch wird festgestellt, welche Oberfläche der viewing-Plane am nächsten ist und welche Pixel dargestellt werden.
- Wenn eine Oberfläche näher als alle zuvor berechneten Oberflächen ist, so werden für diese Oberfläche die Farbe und die Tiefe gespeichert. Also je größer der z-Wert, umso näher ist die Oberfläche der viewing-plane, da wir mit unserer Kamera in negative z-Richtung schauen.
- Depth-Buffer Algorithmen rechnen mit normalisierten Koordinaten, somit kann der Wert für die Tiefe sich nur zwischen 0.0 (view plane) und 1.0 befinden.
- Der depth-Buffer wird dazu verwendet, die Tiefeninformation für jede (x,y) Position einer Oberfläche zu speichern und der Frame-Buffer speichert die Farbinformation für jeden Pixel einer Oberfläche. Die Größe der Buffer hängt von der Bildschirmauflösung ab.
- Vorgehensweise mittels Scan-Fill:
 - Man startet beim linken Pixel und geht entlang der Scanlinie nach rechts vor.
 - Zuerst berechnet man den z-Wert der linken Polygonkante, dort wo sie die Scanlinie schneidet, daraufhin berechnet man den z-Wert für jeden Pixel entlang der Scanlinie.
 - Man kann den depth-Buffer-Algorithmus so implementieren, dass er beim obersten Vertex des Polygons startet und dann die x-Koordinatenwerte abwärts der linken Polygonskante rekursiv berechnet.
 - Der x-Wert für die Startposition einer jeden Scanlinie kann durch den vorherigen Startpunkt berechnet werden. $x' = x - 1/m$ (m ist die Steigung der Kante)

53) Wie funktioniert die Scan-Line Methode? (S. 535)

- Ist eine image-space Methode, die den scan-line Füll-Algorithmus für Polygone erweitert
- Braucht eine Tabelle mit Kanten (y-sortiert) und mit Polygonen (die auf die Kanten-Tabelle verweist) sowie eine active-edge Liste
- berechnet die z-Werte entlang der Scanlinien für jeden Pixel, um festzustellen, welcher Pixel dargestellt wird, und zwar dann, wenn eine Scanlinie mehrere Flächen schneidet.



- Entlang der Scanlinie werden Tiefeninformationen berechnet, um festzustellen, welche Oberfläche der View-Plane am nächsten ist. Hat man die sichtbare Fläche berechnet, dann wird für jedes Pixel die Oberflächenfarbe im Framebuffer gespeichert.
- Man definiert eine Flag, die entweder „on“ oder „off“ sein kann, um festzustellen ob ein bestimmter Punkt der Scanlinie innerhalb oder ausserhalb des Polygons liegt. Bei dieser Methode berechnet man die Pixel entlang der Scanlinie von links nach rechts, beim linken Schnittpunkt der Scanlinie mit der Kante eines konvexen Polygons, wird die Flag auf „on“ gesetzt, beim rechten Schnittpunkt entlang der Linie auf „off“. Bei konkaven Polygonen werden die Scanlinienschnittpunkte von links nach rechts sortiert und die Flag wird zwischen jedem Schnittpunktpaar auf „on“ gesetzt. An den Stellen, wo die Flächen nicht überlappen, ist keine Tiefeninformation notwendig.
- Überlappen sich zwei Flächen, dann wird zuerst bei einmal die Flag für den Schnittpunkt mit der ersten Oberfläche auf „on“ gesetzt, sobald die Scanlinie auch die zweite Fläche schneidet wird auch für diese Fläche die Flag auf „on“ gesetzt. Die Tiefeninformation wird nur für den Bereich berechnet, indem beide Flags auf „on“ sind.
- Dieser Algorithmus funktioniert nur für Ebenen, die einander nicht durchschneiden oder sonstwie zyklisch überlappen. Tritt dieser Fall dennoch ein, so muss man die Polygone (im Buch mittels einer strichlierten Linie) teilen.

54) Wie funktioniert die Depth-Sorting Methode („Painter“-Methode)? (S. 537)

http://en.wikipedia.org/wiki/Painter%27s_algorithm

- Ist eine Mischung aus einer image-space Methode (fine tuning) und einer object-space Methode (grobe Sortierung)
- die Oberflächen werden in absteigender Reihenfolge bzgl der Tiefe sortiert. Die Oberflächen werden scan-konvertiert, wobei man mit der Oberfläche mit der größten Tiefe startet. Die Sortieroperationen werden sowohl im image- und im objekt-space ausgeführt, aber die Scankonvertierung der Polygonsoberflächen findet im image-space statt.
- Sind die Objekte nebeneinander (aus Betrachtersicht), oder überdecken sich komplett, so muss kein „fine tuning“ mehr gemacht werden.
- Falls alle Tests fehlschlagen, so müssen die Objekte umgeordnet werden.
- Problem: Falls die Flächen sich zyklisch überlappen, müssen sie aufgetrennt werden (siehe zyklische Überlappung bei der scan-line Methode)
- Verwendung des Buffers: die Farbwerte für die am weitesten entfernte Fläche werden im refresh-Buffer eingetragen. Als nächstes kommt die nächste Oberfläche an die Reihe, und beim Speichern der Farbwerte für diese Fläche werden die alten Farbwerte im Framebuffer einfach überschrieben.
- Diese Methode wird auch oft als „Painter’s Algorithm“ bezeichnet. Denn wenn ein Maler ein Bild mit z.B. Ölfarben malt, beginnt auch er zuerst mit dem Hintergrund und arbeitet sich dann von hinten nach vorne schrittweise vor.

55) Wie funktioniert die Area-Subdivision Methode? (S. 541)

- Ist eine image-space Methode, die die viewing area immer weiter unterteilt bis die Sub-Fläche so klein ist, dass das Lösen des Sichtbarkeitsproblems leicht funktioniert. Die Vorgehensweise ist ähnlich, wie der Aufbau eines Quadrees
- Es gibt 4 mögliche Beziehungen, die Flächen miteinander haben können.
 1. Surrounding Surface: Eine Fläche, die die andere komplett einschließt
 2. Overlapping Surface: Eine Fläche, die eine andere überlappt. bzw. ein Teil einer Fläche ist innerhalb und ein Teil ist außerhalb einer Fläche.
 3. Inside Surface: Eine Fläche, die komplett innerhalb einer anderen Fläche liegt.
 4. Outside Surface: Eine Fläche, die komplett außerhalb einer anderen Fläche liegt.
- Die Tests für die Oberflächensichtbarkeit können anhand dieser vier Klassifikationen durchgeführt werden. Man kann die Tests beenden, wenn eine der drei folgenden Konditionen eintritt:
 1. Eine Fläche hat weder eine innere, überlappende oder einschließende Fläche.
 2. Eine Fläche hat nur eine innere, überlappende bzw. einschließende Fläche
 3. Eine Fläche hat eine umschließende Fläche, die alle anderen Flächen innerhalb ihres Randes verdunkelt.
- Ist man auf pixel-Ebene angelangt und keine der obigen 3 Konditionen um den Test zu beenden, ist zutreffend, so muss man die Flächen sortieren und die Intensität der nächsten Fläche auswählen

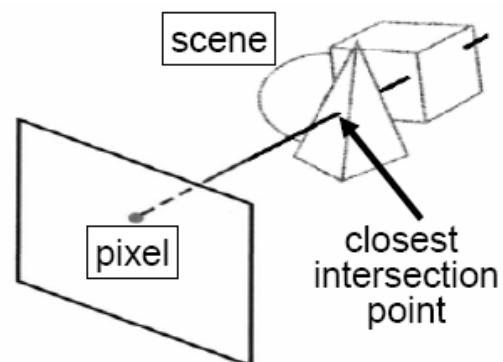
56) Wie funktioniert die Octree Methode? (S. 543)

- Beruht auf dem rekursiven Abarbeiten von Informationen eines Objekts, die in einem Octree gespeichert sind. Die Reihenfolge der Abarbeitung der einzelnen Sub-Oktanten beruht auf der Blickrichtung. Gegebenfalls muss der Octree in die richtige Position transformiert werden.
- Ein Wert wird in den Frame-Buffer eingetragen, wenn noch kein Wert zuvor an dieser Position eingetragen worden ist \Rightarrow jedes pixel(x,y) wird nur einmal eingetragen
- Ist ein Oktant total ausgefüllt, so können alle dahinterliegenden Oktanten vernachlässigt werden, weil sie nicht sichtbar sind.

57) Wie funktioniert die Ray-casting Methode? (S. 487)

http://en.wikipedia.org/wiki/Ray_casting

- Ist eine spezielle Form der Ray-tracing Methode, bei der ein Strahl von der view plane auf die davor liegenden Objekte „abgeschossen“ wird.
- Nachdem dieser Vorgang von jedem Punkt der Ebene abgeschlossen wurde, wird analysiert, welche Objekte „getroffen“ wurde und wiederum welche von diesen am Nächsten an der view



- plane liegen.
- Der Tiefenwert für den z-Buffer wird nur für die Objektpunkte berechnet, in denen eine Objekt-Fläche vom „Pixelstrahl“ geschnitten wird. So gesehen ist diese Methode auch mit der z-Buffer-Methode „verwandt“.
- Raycasting ist eine effektive Methode zum Erkennen von sichtbaren Flächen in einer Szene mit gebogenen Flächen, speziell Kugeln.

Kapitel 10 – Illumination Models and Surface-Rendering Methods (S. 556)

58) Was verstehen Sie unter einem Illumination model? (S. 557)

- = „Lighting Model“ oder „Shading Model“
- wird verwendet, um die beleuchtete Position auf der Oberfläche eines Objektes zu bestimmen.
- Eine Surface-Rendering-Methode verwendet die Farbberechnung des Illumination Models, um die Farbe für jeden Pixel in einer Szene zu berechnen. Man kann entweder das Illumination Model auf jede Position anwenden, oder man vollendet das Surface-Rendering, indem man die Oberflächenfarbe mittels eines kleinen Sets von Illumination-Model-Kalkulationen interpoliert. Scanlinienalgorithmen z.B. verwenden hauptsächlich Interpolierungsschemata, Raytracingalgorithmen hingegen berechnen das Illumination Model für jeden Pixel.
- Die Oberflächenbeleuchtungseffekte beinhalten Reflexion, Transparenz, Muster und Schatten.
- Sind die Parameter für die optischen Eigenschaften der Oberfläche, die relative Position der Fläche in einer Szene, die Farbe und Position der Lichtquelle, sowie die Charakteristika der Lichtquelle und die Orientierung und Position der Viewing Plane gegeben, berechnet das Illumination Model die Lichtintensität, die von einer bestimmten Oberflächenposition in eine bestimmte Richtung projiziert wird.
- Genauere Systeme, wie z.B. der Radiosity Algorithmus, berechnen Lichtintensitäten, indem sie die Ausbreitung der Strahlen zw. der Lichtquelle und den verschiedenen Oberflächen in einer Szene in Betracht ziehen.

59) Welche Arten von Lichtquellen kennen Sie? (S. 558)

- Punktlichtquelle: das einfachste Modell, das Licht erzeugen kann. Das Licht hat nur eine Farbe. Man verwendet die Position dieser Lichtquelle im Illumination Model um festzustellen, welche Objekte einer Szene von dieser Lichtquelle beleuchtet werden und um die Leuchtrichtung bzgl. einer bestimmten Objektoberfläche zu berechnen. Die Stärke der Lichtquelle nimmt mit der Entfernung ab.
- Unendlichweitentfernte Lichtquelle: wie z.B. die Sonne. Kann ebenfalls als Punktlichtquelle verstanden werden, nur mit dem Unterschied, dass solch eine Lichtquelle die Szene nur von einer bestimmten Richtung aus beleuchtet.
- Richtungsorientierte Lichtquelle (spot light): haben bestimmte Position, Richtung und sowie einen Abstrahlwinkel und einen Abschwächungswinkel, die den Kegel des Lichts modellieren

- Reflektierende Lichtquellen: erzeugen selber kein Licht, sondern reflektieren es nur. So ist es möglich, dass auch Objekte, die nicht direkt einer Lichtquelle ausgesetzt sind, sichtbar sind.

60) Welche grundsätzlichen Illumination Models kennen Sie? (S. 563 ff.)

Alle hier aufgezählten Modelle sind empirische Modelle. Faktoren, die zur Berechnung der Farbe in einem Punkt eingehen: Materialeigenschaften des Objekts, Augenpunkt des Betrachters, Normalenvektor des Objekts, Positionierung der Lichtquellen und die Art der Lichtquellen

- Umgebungslicht (ambient light) I_a :
 - Keine Position, keine Richtung
 - Grundhelligkeit eines Objekts
 - Intensität: I_a
 - Reflexions-Koeffizient: k_a ($0 \leq k_a \leq 1$)
- Diffuse Reflexion:
 - Je flacher der Lichteinfall auf eine Fläche \Rightarrow desto dunkler erscheint die Fläche (Lambert'sche Regel)
 - Konstant für eine Fläche, aber abhängig davon wie die Fläche modelliert werden soll (Material-Eigenschaft)
 - Reflexions-Koeffizient: k_d
- Spiegelnde Reflexion (specular reflection) und das Phong Model:
 - = Phänomen, dass in einem bestimmten Toleranzwinkel um die spiegelnden Reflexionswinkel das Licht (fast) total reflektiert wird (direkt in das Auge des Betrachters)
 - wie groß dieser Winkel ist, ist wiederum eine Materialeigenschaft
 - glänzende Materialien haben einen kleinen Bereich, matte einen größeren
 - das Phong-Modell ist ein empirisches Modell, das versucht diese Eigenschaft zu wiederzugeben
 - Reflexions-Koeffizient n_s : groß für glänzende Materialien, kleine für matte
- Weitere Aspekte:
 - Anisotrope Lichtquellen (Warn Model)
 - Intensitätsabfall mit steigender Entfernung
 - Transparenz (Snell's Law)
 - Schatten

61) Wie funktioniert die Gouraud-Schattierung? (S. 592)

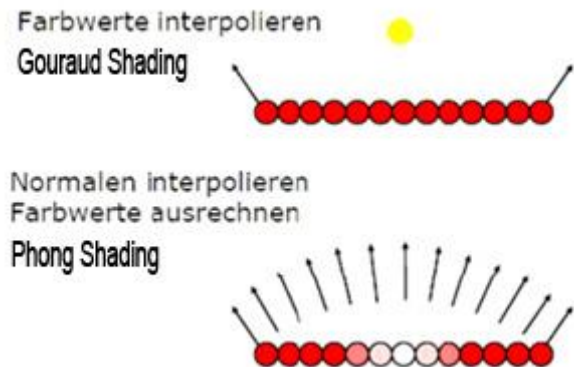
http://en.wikipedia.org/wiki/Gouraud_shading

- Berechnet die Normalvektoren in den Eckpunkten der Polygone
- Berechnet die Lichtintensitäten in den Eckpunkten
- Interpoliert die Licht-Intensitäten entlang der Polygon-Kanten
- Interpoliert die Licht-Intensitäten für jeden Punkt auf der Scan-line innerhalb des Polygons

62) Wie funktioniert die Phong-Schattierung? (S. 595)

http://en.wikipedia.org/wiki/Phong_shading

- Berechnet die Normalvektoren in den Eckpunkten der Polygone
- Interpoliert die Normalvektoren entlang der Polygon-Kanten
- Interpoliert für jeden Punkt auf der Scan-line innerhalb des Polygons den Normalvektor
- Berechnet aus dem interpolierten Normalvektor die Lichtintensität



63) Wozu verwendet man Gouraud- und Phong-Schattierung und was sind die Unterschiede und Gemeinsamkeiten(S. 523 ff.)

Siehe Frage 61) und 62)

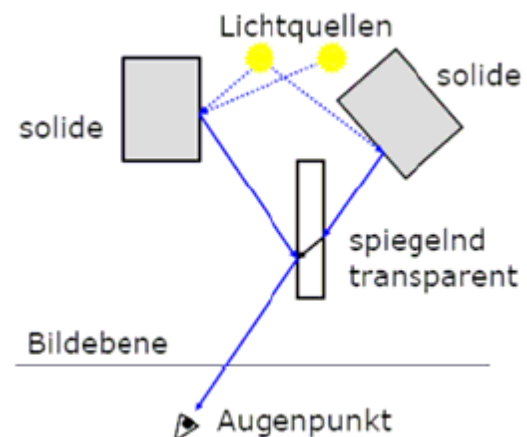
64) Wie sieht das Beleuchtungsmodell aus, welches diffuse und spiegelnde (specular) Reflexionen mit mehreren Lichtquellen berücksichtigt? (S. 571)

Das ambiente Licht wird gleich berechnet wie bei nur einer Lichtquelle. Zusätzlich wird für jede Lichtquelle der diffuse und spiegelnde Anteil berechnet und für alle Lichtquellen aufsummiert.

65) Was verstehen Sie unter Ray-Tracing? (S. 597 ff.)

http://de.wikipedia.org/wiki/Ray_tracing, http://en.wikipedia.org/wiki/Ray_tracing

- ist ein Verfahren in der Bildsynthese, bei dem die Ausbreitung von Lichtstrahlen simuliert wird.
- Von einer Ebene vor der Kamera wird ein Strahl (ray) in die Szene „hineingeschossen“ und zurückverfolgt. Man spricht deshalb auch von backward ray tracing. Die Farbwerte der von den jeweiligen Strahlen getroffenen Objekte bestimmen die Farbe des entsprechenden Pixels. Backward ray tracing beschäftigt sich so mit der Frage, woher das Licht kommt.
- Wenn der Strahl ein Objekt im Raum trifft, so wird er
 - refraktiert, also gebrochen (Glas, Luft)
 - reflektiert (Spiegelungen und diffuse Abstrahlung)
 - reflektiert und refraktiert, Anteile gemäß der Fresnel'schen Formeln



- Um eine Spiegelung zu berechnen, ist der Strahl an der getroffenen Oberfläche zu reflektieren (Einfallswinkel ist gleich Ausfallswinkel). Dazu muss lediglich der Raytracing-Algorithmus rekursiv aufgerufen werden, weshalb man auch von rekursivem Raytracing spricht. Die Rekursion kann abgebrochen werden, wenn keine Objekte mehr getroffen werden, eine vorbestimmte Rekursionstiefe (eine Tiefe von 5-7 reicht oft aus) erreicht ist oder die Reflexion in der Rekursion keine nennenswerten Ergebnisse mehr beisteuert (engl. adaptive depth control).
- Die Farbwerte der getroffenen Objekte werden dann nach ihrem Reflexionsgrad gewichtet, aufaddiert und mit der Lichtfarbe verrechnet. Analog verfährt Raytracing mit Strahlen, die nicht reflektiert, sondern in ein durchsichtiges Objekt gebrochen (Refraktion) werden.
- Die Farbe der Oberfläche ergibt sich analog zum Labert'schen und Phong Modell aus der Diffusions-, der Reflexions- und der Transparenzkomponente, wobei zusätzlich gelten muss, dass $k_d + k_r + k_t \leq 1$ sein muss.

66) Was sind die Vorteile und Nachteile von Ray tracing?

- Vorteile:
 - Einfache Implementierung mit überschaubarer Komplexität
 - Kein starres Schema wie bei der Render-Pipeline
 - Leichtere Austauschbarkeit der Shader und dadurch erleichterte Implementierung neuer Shader
 - Raytracing erzeugt, obwohl es in weiten Teilen nur von einer vereinfachten Realität ausgeht, dennoch überraschend realitätsnahe Licht- und Schatten- sowie Reflexionseffekte.
 - Kann mit mathematischen Oberflächen- bzw. Volumenmodellen von zu modellierenden Objekten arbeiten. Die Oberflächen der Objekte müssen so nicht durch einzelne Flächen modelliert werden und liegen als auflösungsunabhängige Daten vor. Die Modellierung von Objekten mit CSG lässt sich leicht realisieren
 - Ebenso ist die Berechnung von Schatten oder die Darstellung von Objekten, in denen sich das Licht bricht leicht realisierbar
- Nachteile:
 - Es wird nur die direkte Beleuchtung simuliert und berücksichtigt daher nicht das indirekte Licht, dass von anderen Objekten reflektiert wird (wird nur in Monte-Carlo-Raytracing berücksichtigt)
 - Es wird nicht berücksichtigt, dass bei der Refraktion Licht unterschiedlicher Wellenlängen stärker oder schwächer gebrochen wird (Dispersion) und das Licht so auffächert (Prismeneffekt). Dies liegt vornehmlich daran, dass Raytracer aus Gründen der Benutzerfreundlichkeit und Effizienz mit dem RGB-Farbmodell und nicht mit gesampelten Spektren arbeiten.
 - Raytracing-Berechnungen gelten als sehr zeitintensiv – Die Schnittpunktberechnungen brauchen die meiste Zeit da bei jedem Strahl eine Schnittpunktberechnung durchgeführt werden muss.
 - Seidenmatte Oberflächen kann nicht modelliert werden

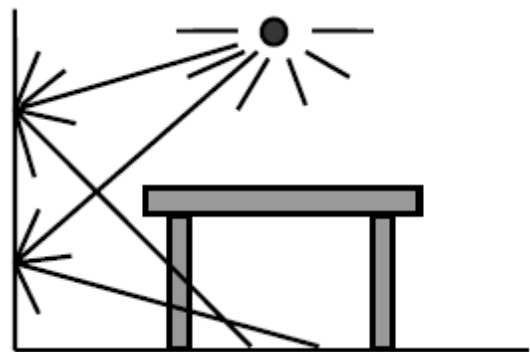
67) Welche Konzepte der Reduzierung von Objekt-Strahl Schnittberechnungen kann man beim Ray-Tracing Algorithmus anwenden? (S. 604)

- Schnitt-Berechnung machen bis zu 95% des Zeitaufwandes eine Ray tracers aus, daher ist es wichtig, Verfahren zu benutzen, die die Anzahl der Schnitt-Berechnung minimieren
- Bounding Volumes:
 - Eine Hülle („bounding sphere“) wird um eine Gruppe von Objekten gelegt
 - Wird die Hülle nicht geschnitten, so kann man auch Schnittberechnungen mit den Objekten innerhalb der Hülle vernachlässigen
 - In weitere Folge kann man Hierarchien von Bounding Volumes aufbauen, indem man mehrere Bounding Volumes zu einer übergeordneten Hülle zusammenfaßt.
- Space-Subdivision Methods:
 - Der Raum wird z.B. mittels eines Rasters oder eines Octrees aufgeteilt, und zwar so, dass in einem Teilraum nur mehr eine bestimmte Anzahl an Flächen ist
 - Formen: Uniforme Subdivision (der Raum wird immer gleich aufgeteilt, z.B. in 8 Teile bei einem Octree) bzw. adaptive Subdivision (es wird nur dann weiter unterteilt, wenn sich in dem Teilraum Objekte befinden)
 - Anhand eines 3D-Bresenham-Algorithmus können die Schnittpunkte des Strahls mit dem Würfel berechnet werden und so potentielle Seitenflächen ermittelt werden

68) Erklären Sie Radiosity bzw. das Radiosity Modell (S. 615)

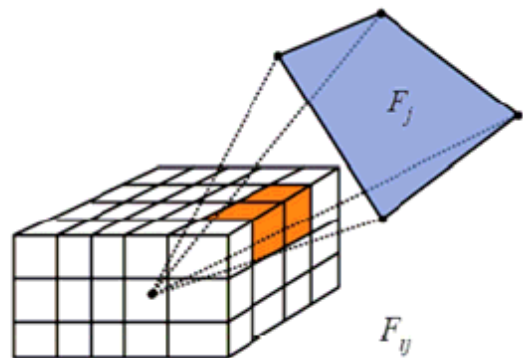
<http://de.wikipedia.org/wiki/Radiosity>, <http://en.wikipedia.org/wiki/Radiosity>

- ist ein globales Beleuchtungsmodell für die 3D-Computergrafik. Es beruht auf dem Energieerhaltungssatz: Alles Licht, das eine Fläche empfängt und nicht absorbiert, muss sie wieder emittieren. Außerdem kann eine Fläche auch selbstleuchtend sein.
- Für jede Fläche wird eine Gleichung aufgestellt, die das emittierte Licht aus dem von den anderen Flächen empfangenen Licht und evtl. ihrer eigenen Leuchtkraft bestimmt. Insgesamt ergibt sich damit ein Gleichungssystem, dessen Lösung die Helligkeit jeder einzelnen Fläche angibt.
- Mathematisch besteht das Verfahren in der Lösung eines Gleichungssystems. Hierfür kann ein iteratives Verfahren angewendet werden, das gegen die exakte Lösung konvergiert. Damit ist ein frei wählbarer Kompromiss zwischen Darstellungsqualität und Rechenzeit möglich.



69) Wofür verwendet man Formfaktoren, wie berechnet man sie einfach und was sind ihre Eigenschaften? (S. 546)

- Formfaktoren sind – unabhängig vom gewählten Algorithmus – der aufwendigste Schritt im Verfahren zur Berechnung der Radiosity (Berechnung eines Doppel-Integrals notwendig).
- Ein Formfaktor gilt immer zwischen zwei Patches (= Flächen-Polygone mit konstanter Radiosity BK) und beschreibt die Menge der ausgetauschten Strahlung, liegt also zwischen null (keine Strahlung wird ausgetauscht) und eins (alle Strahlung wird ausgetauscht).
- Der Formfaktor ist rein geometrischer Natur und wird durch die Stellung der Patches zueinander bestimmt. Des Weiteren spielt die Sichtbarkeit der Patches eine Rolle. Die Sichtbarkeitsberechnung braucht bei weitem die meiste Zeit in der Berechnung.
- Besondere Eigenschaften:
 - Konservierung der Energie: (Summe $k=1..n$) $F_{jk} = 1$
 - Uniforme Lichtreflexion: $A_j F_{jk} = A_k F_{kj}$
 - Kein Selbst-Einfall: $F_{kk} = 0$
- Approximation der Berechnung durch Hemi-Cube Algorithmus:
 - Es wird ein repräsentativer Punkt der Empfängerfläche, der Mittelpunkt, ausgewählt.
 - Die sichtbaren Teile der Senderfläche werden auf den Einheitswürfel (Approximation der Einheitshalbkugel nach Nusselt) um diesen Punkt projiziert.
 - Die Flächen des Einheitswürfels sind in ein diskretes Gitter unterteilt.
 - Jeder Gitterfläche wird ein Gewichtungsfaktor, der Delta-Formfaktor, zugeordnet, welcher von der Position der Gitterfläche abhängig ist. Ein Delta-Formfaktor ist damit der Formfaktor der Gitterfläche nach Nusselt. Die Summe der Delta-Formfaktoren ist 1.
 - Für jede der fünf Halbwürfelflächen wird mit modifizierten Rasteralgorithmen (üblicher Weise Z-Buffer) ein Itembuffer berechnet. Dieser enthält für jede Gitterfläche die Identität der Objektfläche (Item, im Bild rotes Dreieck), die darauf projiziert wurde. Für jedes Item wird die Summe der Delta-Formfaktoren der überdeckten Gitterflächen berechnet (im Bild rote Gitterflächen). Diese Summe wird als Formfaktor zwischen Item und betrachteter Fläche aufgefasst.



70) Was sind die Vorteile und Nachteile des Radiosity Modells?

- Vorteile:
 - Berechnung vom Standort und Blickwinkel des Betrachters unabhängig erfolgt \Rightarrow Berechnungen müssen so für eine Szene nur ein Mal gemacht werden. Danach kann die Szene in Echtzeit gerendert werden, was für Anwendungen wie virtuelle Architekturmodelle interessant ist.
 - Diffuse Lichtreflexionen sind im Radiosity-Verfahren enthalten. Die Helligkeit und Farbe einer Fläche wird nicht allein aufgrund der direkten Beleuchtung einer Lichtquelle, sondern auch durch diffus reflektiertes Licht anderer Flächen bestimmt.
- Nachteile:
 - Hauptnachteil des Radiosity-Verfahrens war, dass es bisher nur diffuse Reflexion kannte. Spiegelnde Flächen ließen sich mit dieser Methode nicht berechnen. Mittlerweile existieren zwar Verfahren, die das können, diese sind aber vorerst nur auf dem akademischen und noch nicht auf dem kommerziellen Level verfügbar.
 - Für die möglichst realistische Darstellung einer Szene muß die Globale Beleuchtung simuliert werden, was aber nur in Spezialfällen mit Radiosity effizient möglich ist. Dies kann man zwar z.B. mit Monte-Carlo-Raytracing wettmachen, doch ist die Kombination mit Radiosity aus Effizienzgründen ist aufwändig, sodass hierzu meist nur entweder die eine oder die andere Technik verwendet wird.
 - Simulation von Materialien mit beliebigen Beleuchtungsmodellen ist nur schwer möglich.

71) Erklären Sie Environment und Texture Mapping (S. 623 ff.)

http://de.wikipedia.org/wiki/Texture_Mapping,
http://en.wikipedia.org/wiki/Texture_mapping

- Environment-Mapping:
 - Ist eine Form des Reflexions-Mapping. Dabei werden Informationen des Universums (sphere, Würfel, Zylinder), das ein Objekt umgibt, abgebildet und auf das Objekt selber aufgetragen \Rightarrow Textur enthält Projektion der Umgebung
 - Der Zugriff ist vom Augenpunkt abhängig
- Texture-Mapping:
 - Textmuster wird auf die Oberfläche des Objekts aufgebracht
 - Texturen erhöhen den Realismus von computergenerierten 3D-Szenarien deutlich, ohne dass die Anzahl der verwendeten Polygone erhöht werden muss.
 - Bei Polygonen, die eine größere Ausdehnung in Sichtrichtung haben muss eine Perpektiven-Korrektur durchgeführt werden
 - Antialiasing wird verwendet um bessere Ergebnisse zu erzielen

72) Erklären Sie Bump Mapping, seine Funktionalität sowie Vorteile und Nachteile (S. 634)

http://de.wikipedia.org/wiki/Bump_Mapping, http://en.wikipedia.org/wiki/Bump_mapping

- Ist eine spezielle Form des Texture-Mappings, bei der durch die Modifikation von Normalvektoren eine Textur aufgebracht wird, die durch die Verwendung von Schattierungen einen 3D-Effekt erzielt
- Bump Mapping ist somit nur ein (sehr wirksamer) Darstellungs-"Trick", der Oberflächenunebenheiten simuliert, die in der Geometrie des Modells gar nicht vorhanden sind.
- Vorteile:
 - Bumpmapping verbessert die Bildqualität und den Realismusgrad ohne sich groß auf die Performance niederzuschlagen.
 - Reflexionen können mit dieser Technik ebenfalls dargestellt werden
- Nachteile:
 - Manche Oberflächen schauen noch immer nicht real aus, weil die Schatten an den Rändern der Bumps nicht stimmen.
 - Der Rand der Objekte (z.B. Kugel) bleibt unverändert (perfekt rund).

Kapitel 12 – Color Models and Color Applications (S. 712)

73) Was ist Licht bzw. Farbe? (S. 713 ff.)

Licht ist nichts anderes als ein schmales Spektrum elektromagnetischer Wellen. Jeder Frequenzwert des elektromagnetischen Spektrums entspricht einer Spektralfarbe. Bei den niedrigen Frequenzen befinden sich die Rottöne, bei den hohen Frequenzen die Blautöne. Der sichtbare Bereich liegt zwischen 780nm und 380nm. Weitere Parameter, die die Farbe beschreiben, sind Helligkeit abhängig von der Leuchtkraft der Quelle (wie viel Licht das Objekt reflektiert) und die Sättigung (wie intensiv die Farbe ist – Abmischung selbiger zwischen weiß und schwarz).

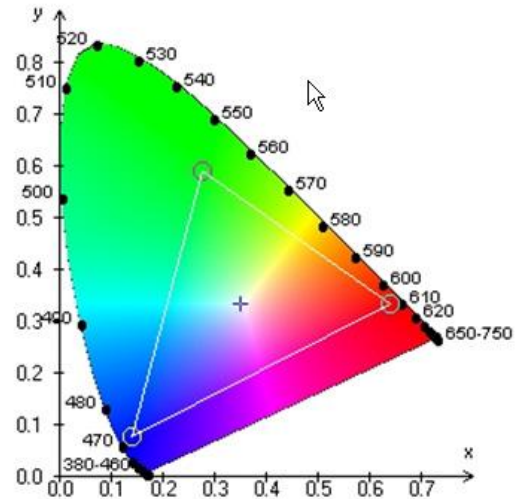
74) Was sind Primärfarben bzw. Komplementärfarben? (S. 716)

- Komplementärfarben: Mischung ergibt weißes Licht (z.B. rot – cyan, grün – magenta, blau – gelb)
- Primärfarben:
 - Meistens 3 Farben, z.B. Rot-Grün-Blau
 - Alle möglichen Farben innerhalb des Farbmodels (= „color gamut“) können damit durch Mischen hergestellt werden
 - Dennoch sind nicht alle sichtbaren Farben möglich (manche liegen außerhalb des „color gamut“)

75) Was versteht man unter „color gamut“ und welcher Zusammenhang zum CIE diagramm besteht? (S. 719)

http://en.wikipedia.org/wiki/CIE_1931_color_space,
http://en.wikipedia.org/wiki/Color_gamut

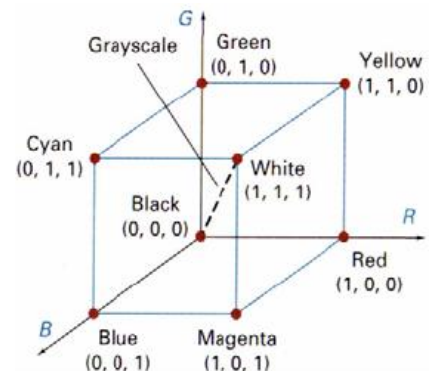
- CIE-Diagramm:
 - Umfasst alle für den Menschen sichtbaren Farben.
 - Die Spektralfarben liegen am Bogen-Rand des Diagramms. Sie stellen das monochromatische Licht dar.
 - Die Linie am unteren Ende des Diagramms heisst „purple line“. Diese Farben liegen zwar auf dem Rand des Diagramms, sind aber keine Spektralfarben.
 - Das Diagramm erlaubt Farbmodelle und ihre „color gamuts“ miteinander zu vergleichen, sowie Komplementärfarben zu finden (liegen auf einer Geraden die durch C verläuft)
- Color gamut:
 - Wenn man (2 oder) 3 Farben auswählt, die die Primärfarben eines Farbmodells darstellen sollen und im Diagramm (eine Linie bzw.) ein Dreieck zwischen ihnen aufspannt, so wird dies als „color gamut“ bezeichnet.
 - Alle Farben die (auf der Linie bzw.) innerhalb des Dreiecks liegen, können mit den Primärfarben dargestellt werden.
 - Man sieht auf den ersten Blick, dass die Fläche immer ein konvexes Dreieck ist und daher nie alle Farben umfassen kann.



76) Was wissen Sie zum RGB-Farbsystem? (S. 720)

<http://en.wikipedia.org/wiki/RGB>

- Primärfarben: Rot-Grün-Blau
- Additives Farbmodell (z.B. für Monitore)
- Basiert auf der menschlichen Farbwahrnehmung
- Farbe wird dargestellt, indem man das Beleuchtungsmodell auf die 3 Primärfarben anwendet
- Darstellung durch einen Würfel



77) Was wissen Sie zum CMY-Farbsystem? (S. 723)

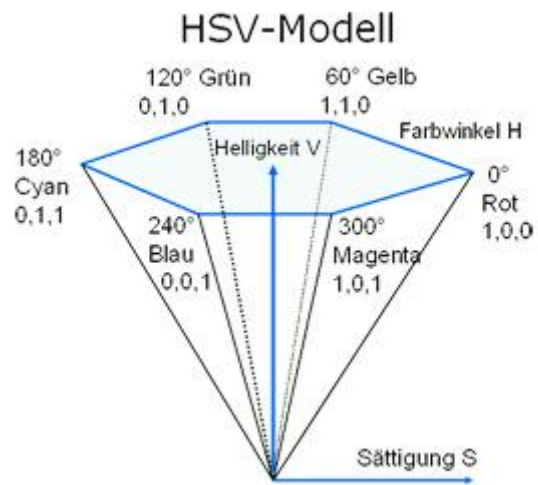
<http://en.wikipedia.org/wiki/RGB>

- Primärfarben: Cyan-Magenta-Gelb (yellow)
- Subtraktives Farbmodell (z.B. für Kopiergeräte)
- Gelb kann dargestellt werden indem man grün und blau mischt
- Komplementär-Modell zu RGB: $(C,M,Y) = (1,1,1) - (R,G,B)$
- Bei Druckern wird oft zusätzlich noch schwarz als 4. Farbe verwendet, da die Mischung aus C+M+Y oft nur ein starkes Dunkelgrau ergibt

78) Was wissen Sie zum HSV-Farbsystem? (S. 724)

http://en.wikipedia.org/wiki/HSV_color_space, <http://de.wikipedia.org/wiki/HSV-Farbraum>

- Intuitivere Spezifikation von Farbe
- Stellt man den Würfel des RGB-Modells auf die Diagonale zwischen schwarz (unten) und weiß (oben), so sieht man von oben ein 6-eck mit den G-Y-R-M-B-C
- Farbkomponenten: hue (Farbton), saturation (Sättigung) und value (Helligkeit)
- Blickt man von der Seite auf den Querschnitt der Farbpypamide, so sieht man unten an der Spitze die Farbe schwarz, oben in der Mitte die Farbe weiß, und ganz rechts oben den Ausgangspunkt mit $S=1$ und $V=1$
- Farbdefinition:
 1. Spektralfarbe auswählen, $S=1$, $V=1$
 2. schwarze Pigmente hinzufügen \Rightarrow Helligkeit verringern (weiter runter im Kegel)
 3. weiße Pigmente hinzufügen \Rightarrow Sättigung verringern (weiter in die Mitte im Kegel)
- Die RGB-Werte und die CYM-Werte sind jeweils 60° von einander entfernt, will man nun die Komplementärfarbe zu einer Farbe wissen, so nimmt die 180° entfernte Farbe.



79) Was wissen Sie zum HLS-Farbsystem? (S. 728)

http://en.wikipedia.org/wiki/HLS_color_space

- Farbkomponenten: hue, lightness (Helligkeit) und saturation
- Abbildung des Farbraumes mittels eines Doppel-Kegels
- Der H-Wert entspricht auch hier einer Spektralfarbe. $H=0^\circ$ entspricht in diesem Modell blau.
- Die Primärfarben liegen bei $L=0,5$. Erhöht bzw. vermindert sich der L-Wert so hat man entweder mehr weiß oder schwarz in der Farbe.