

# Zusammenfassung CG

<https://github.com/chefe/summaries>

3. Januar 2019

## Inhaltsverzeichnis

<b>Einführung</b>	<b>7</b>
Wo werden überall Computergrafik gebraucht? . . . . .	7
Wie heissen die Schritte in einer CG Pipeline? . . . . .	7
<b>Farbe</b>	<b>7</b>
Durch welche Charakteristiken kann eine Farbe beschrieben werden? . . . . .	7
In welche Bereiche kann das sichtbare Licht aufgeteilt werden? . . . . .	7
Wie entsteht die Farbe einer Fläche? . . . . .	7
Aus welchen zwei Bestandteilen setzt sich die Farbe eines Objekts zusammen? . . . . .	7
Wie kann ein Auge mit einer Kamera verglichen werden? . . . . .	8
Wie viele Stäbchen und Zäpfchen hat ein Mensch? . . . . .	8
Was ist die Fovea? . . . . .	8
Welche Arten von Zäpfchen gibt es? . . . . .	8
Wie sehen wir Farbe? . . . . .	8
Was ist der unterschied der Helligkeitswahrnehmung bei Stäbchen und Zäpfchen? . . . . .	8
Können alle Farben durch 3 Farben gemischt werden? . . . . .	8
Wie sehen die Spektralwertkurven im CIE Farbsystem aus? . . . . .	9
Wie setzen sich die Anteile im CIE Farbsystem zusammen? . . . . .	9
Was ist die Normfarbtafel und wie sieht sie aus? . . . . .	9
Welche Farbmodelle gibt es? . . . . .	10
Was sind die Eigenschaften des RGB Farbmodell? . . . . .	10
Welche Werte haben die "einfachen" RGB Farben? . . . . .	10
Was sind die Eigenschaften des CMY Farbmodell? . . . . .	10
Wie ist der mathematische Zusammenhang zwischen RGB und CMY? . . . . .	10
Was bedeutet CMYK? . . . . .	10
Wie lässt sich CMYK aus CMY berechnen? . . . . .	10
Was sind die Eigenschaften des HSV Farbmodell? . . . . .	11
Welche Gradwerte von HSV werden den RGB Grundfarben zugeordnet? . . . . .	11
Was ist YUV? . . . . .	11
Wie kann YUV aus RGB berechnet werden? . . . . .	11
Für was stehen die Begriffe Tints und Shades? . . . . .	11
Was ist das CIE Lab System? . . . . .	11
Was passiert bei einer Gammakorrektur? . . . . .	11

Was ist das spezielle von Helligkeit und Kontrast? . . . . .	11
Was ist Kontrast? . . . . .	12
Was ist HDR? . . . . .	12
Was ist der Grund, dass viele Farbsysteme 3 Grundfarben verwenden? . . . . .	12
Wie viele Spektralfarben können durch Mischen von 2 Spektralfarben erzeugt werden? . . .	12
Welche Farbe ist auch bei sehr wenig Licht noch gut sichtbar? . . . . .	12
Welche Farben kann ein Monitor darstellen? . . . . .	12
In welcher Farbe erscheint ein gelber Tennisball? . . . . .	12
<b>WebGL</b>	<b>12</b>
Was sind die Merkmale von OpenGL? . . . . .	12
Was sind die Vorteile von programmierbaren Shaders? . . . . .	12
Was ist der Unterschied zwischen Vertex & Fragment Processing? . . . . .	13
Wie sieht eine vereinfachte WebGL Pipeline aus? . . . . .	13
Wie sieht der Datenfluss in einer WebGL Pipeline aus? . . . . .	13
<b>Farbe, Halbtontechnik</b>	<b>13</b>
Welche Verfahren gibt es in der Halbtontechnik? . . . . .	13
Was bedeutet der Begriff Quantisierung? . . . . .	13
Was bedeutet der Begriff Dithering? . . . . .	14
Was versteht man unter dem Begriff Halbtontechnik? . . . . .	14
Wie funktioniert die digitale Halbtontechnik? . . . . .	14
Wie kann eine Dithermatrix aussehen? . . . . .	14
Was sind die Regeln für Dithering? . . . . .	14
Wie kann Dithering benutzt werden, wenn die Auflösung gleich bleiben soll? . . . . .	14
Wie wird Dispersed Dot Dithering berechnet? . . . . .	14
Was ist die Idee hinter Error Diffusion? . . . . .	15
Wie wird die Error Diffusion berechnet? . . . . .	15
Wie viele Intensitätsstufen können mit einer Dithermatrix dargestellt werden? . . . . .	15
Wie wird eine RGB Farbe in einen Grauwert umgerechnet? . . . . .	15
<b>Projektive Geometrie 1</b>	<b>15</b>
Wie werden zwei Vektoren addiert? . . . . .	15
Wie wird ein Vektor mit einem Skalar multipliziert? . . . . .	15
Was ist das Inverse eines Vektors? . . . . .	15
Was ist der Nullvektor? . . . . .	15
Wie werden zwei Vektoren subtrahiert? . . . . .	16
Wann sind zwei Vektoren gleich? . . . . .	16
Was ist der Unterschied zwischen einem Rechtssystem und einem Linkssystem? . . . . .	16
Wie kann man sich den Unterschied zwischen den beiden Systemen merken? . . . . .	16
Wie ist ein Vektor im Kartesisches Koordinatensystem definiert? . . . . .	16
Was muss bei der Schreibweise im Kartesisches Koordinatensystem beachtet werden? . . . .	16
Wie wird die Länge eines 2D-Vektors bestimmt? . . . . .	16
Welche Rechenregeln gelten für Vektoren? . . . . .	16
Wie ist das Skalarprodukt definiert? . . . . .	16
Wie kann das Skalarprodukt im kartesischen Koordinatensystem berechnet werden? . . . .	16
Wie wird das Skalarprodukt im allgemeinen Fall berechnet? . . . . .	17
Welche Rechenregeln gelten für das Skalarprodukt? . . . . .	17
Was sind zwei Vektoren orthogonal? . . . . .	17
Wie ist eine Linearkombination definiert? . . . . .	17
Wie ist lineare Unabhängigkeit definiert? . . . . .	17
Auf welche Arten kann eine Gerade beschrieben werden? . . . . .	17

Wie ist die Hessesche Normalform (HNF) der Gerade definiert? . . . . .	17
Wie kann der Abstand einer Gerade zum Ursprung berechnet werden? . . . . .	17
Wie kommt man von der 2D-Koordinatengleichung zur HNF? . . . . .	18
Wie ist der normierter Normalenvektor in 2D definiert? . . . . .	18
Wie kommt man von der 3D-Koordinatengleichung zur HNF? . . . . .	18
Wie ist der normierter Normalenvektor in 3D definiert? . . . . .	18
Wie wird die Projektion eines Vektors auf einen anderen Vektor berechnet? . . . . .	18
Was sind die Eigenschaften des Vektorprodukt? . . . . .	18
Wie wird das Vektorprodukt berechnet? . . . . .	18
Was sind die Rechenregeln für das Vektorprodukt? . . . . .	18
Wie wird das Spatprodukt berechnet? . . . . .	19
Was sind die Rechenregeln für das Spatprodukt? . . . . .	19
<b>Projektive Geometrie 2</b>	<b>19</b>
Wieso wird die Projektive Ebene benutzt? . . . . .	19
Wie wird ein Punkt homogenisiert? . . . . .	19
Wie wird ein Punkt dehomogenisiert / realisiert? . . . . .	19
Was sind die Eigenschaften einer Projektive Transformationen? . . . . .	19
Wie ist die allgemeine Skalierungs-Matrix definiert? . . . . .	20
Wie ist die allgemeine Translations-Matrix definiert? . . . . .	20
Wie ist die allgemeine Rotations-Matrix definiert (Rotation um den Nullpunkt)? . . . . .	20
Wie ist die allgemeine Spiegelungs-Matrix definiert (Spiegelung durch den Ursprung)? . . . . .	20
Wie ist die Matrix für die Rotation um einen beliebigen Punkt definiert? . . . . .	20
Welche Arten von Transformationen gibt es? . . . . .	20
<b>Projektive Geometrie 3</b>	<b>21</b>
Wie ändern sich die Koordinaten bei einer 2D-Transformation des Koordinatensystems? . . . . .	21
Wie ändern sich die Koordinaten bei einer 2D-Rotation des Koordinatensystems? . . . . .	21
Wie kann diese Rotation auch gemacht werden? . . . . .	21
Wie kann ein Raumpunkt in homogene Koordinaten umgerechnet werden? . . . . .	21
Wie wird eine Ebenen im Raum definiert? . . . . .	21
Wie sieht die allgemeine Translation-Matrix in 3D aus? . . . . .	21
Wie sieht die Rotationsmatrix um die z-Achse aus? . . . . .	21
Wie sieht die Rotationsmatrix um die y-Achse aus? . . . . .	21
Wie sieht die Rotationsmatrix um die x-Achse aus? . . . . .	22
Welche Schritte sind für die Rotationen um eine beliebige Achse nötig? . . . . .	22
Wie ist die Matrix für eine Rotationen um eine beliebige Achse definiert? . . . . .	22
Wie wird eine parallele Projektion beschrieben? . . . . .	22
Wie wird eine perspektivische Projektion beschrieben? . . . . .	22
Wie wird eine Clipping-Transformation beschrieben? . . . . .	22
Wie sieht die allgemeine Skalierungs-Matrix aus? . . . . .	23
<b>Scan Konvertierung</b>	<b>23</b>
Wie könnte ein Algorithmus zum Zeichnen einer Linie aussehen? . . . . .	23
Was ist ein DDA? . . . . .	23
Was sind die Nachteile von DDA? . . . . .	23
Welche Arten gibt es um eine Gerade zu definieren? . . . . .	23
Was ist die Definition des Mittelpunktschema? . . . . .	23
Wie wird beim Runden entschieden wo ein Punkt gezeichnet werden soll? . . . . .	23
Wie könnte der Mittelpunktschema-Algorithmus implementiert werden? . . . . .	24
Wie kann das Mittelpunktschema für Kreise angewandt werden? . . . . .	24
Wie könnte der Mittelpunktschema-Algorithmus für Kreise implementiert werden? . . . . .	24

Wie könnte ein Algorithmus zum Füllen von Regionen implementiert werden? . . . . .	24
Wie werden gefüllten Polygonen gezeichnet? . . . . .	24
Welche Tabellen benötigt der Scanlinien-Algorithmus? . . . . .	25
Wie könnte ein Scanlinienalgorithmus implementiert werden? . . . . .	25
Welche Ansätze gibt es für das Zeichnen von Dreiecken? . . . . .	25
Was ist Anti-Aliasing? . . . . .	25
Wie funktioniert Prefiltering? . . . . .	26
Wie funktioniert Supersampling? . . . . .	26
<b>Viewing</b>	<b>26</b>
Was sind die Eigenschaften der Parallel-Projektion? . . . . .	26
Welche Arten von Projektionen gibt es? . . . . .	26
Wie sieht eine Orthographische Projektion aus? . . . . .	27
Wie sieht eine Isometrische Projektion aus? . . . . .	27
Wie sieht eine Kavaliersprojektion aus? . . . . .	27
Wie sieht eine Kabinettsprojektion aus? . . . . .	27
Wie sieht eine Perspektivische Projektion aus? . . . . .	28
Was sind die Eigenschaften der Paralell-Projektion? . . . . .	28
Was sind die Eigenschaften der perspektivischen Projektion? . . . . .	28
Wie kann die Kamera Position spezifiziert werden? . . . . .	28
Wie wird eine Orthographische Projektion mit WebGL implementiert? . . . . .	28
Wie wird eine perspektivische Projektion mit WebGL implementiert? . . . . .	29
<b>Visibility</b>	<b>29</b>
Was sind Algorithmen für verdeckte Linien und Flächen? . . . . .	29
Wie funktioniert Backface Culling? . . . . .	29
Was ist die Idee hinter der Tiefensortierung? . . . . .	29
Wie wird bei der Sortierung für die Tiefensortierung vorgegangen? . . . . .	30
Welche Bedingungen werden bei der Tiefensuche überprüft? . . . . .	30
Was ist wenn es keine gültige Reihenfolge für die Tiefensortierung gibt? . . . . .	30
Was sind die Vor- und Nachteile der Tiefensortierung? . . . . .	30
Was ist die Idee des Z-Buffer Algorithmus? . . . . .	30
Wie könnte der Z-Buffer Algorithmus implementiert werden? . . . . .	30
Wie wird z berechnet? . . . . .	30
Was sind die Vorteile des z-Buffer Algorithmus? . . . . .	30
Was sind die Nachteile des z-Buffer Algorithmus? . . . . .	31
Was ist die Idee des Warnock Algorithmus? . . . . .	31
Was sind die Kriterien des Warnock Algorithmus für Einfachheit? . . . . .	31
Was ist der Unterschied zwischen Objektraum und Bildraum? . . . . .	31
<b>Clipping</b>	<b>31</b>
Was ist die Definition von 2D Clipping? . . . . .	31
Was ist die Definition von 3D Clipping? . . . . .	31
Was kann durch 3D Clipping verhindert werden? . . . . .	31
Welche Clipping Verfahren gibt es? . . . . .	31
Wie funktioniert das Clipping von Linien nach Cohen-Sutherland? . . . . .	32
Wie funktioniert die Bereichsunterteilung beim Clipping? . . . . .	32
Was ist der Vorteil des Algorithmus von Sutherland-Hodgeman? . . . . .	32
Wie funktioniert der Sutherland-Hodgman Algorithmus für Polygone? . . . . .	32
Was ist die vier möglichen Ausgabe des Sutherland-Hodgman-Algorithmus? . . . . .	32
Was liegt der Punkt I beim Sutherland-Hodgman Algorithmus? . . . . .	32

<b>Beleuchtung und Schattierung</b>	<b>32</b>
Welche Materialeigenschaften haben Einfluss auf die Farbe eines Objektes? . . . . .	32
Wie funktioniert das Standardbeleuchtungsmodell? . . . . .	33
Wie funktioniert eine diffuse Reflektion (Lambert Modell)? . . . . .	33
Wie verändert sich die Energie auf eine Fläche, abhängig vom Winkel? . . . . .	33
Wie wird die diffuse Reflektion berechnet? . . . . .	33
Wie funktioniert eine spiegelnde Reflektion (Phong Modell)? . . . . .	33
Wie wird die spiegelnde Reflektion berechnet? . . . . .	33
Wie setzt sich ein Spiegelungsmodell meistens zusammen? . . . . .	33
Wie setzt sich die Farbe der Reflexion zusammen? . . . . .	33
Wie könnten die Reflexionskonstanten für ein rote Kugel und ein weisses Licht sein? . . . .	33
Wie wird die Abschwächung des Lichts gehandhabt? . . . . .	34
Welche Arten von Lichtquellen gibt es? . . . . .	34
Was ist eine Schattierung? . . . . .	34
Welche Arten Schattierung unterscheidet man? . . . . .	34
<b>Curves</b>	<b>34</b>
Welche Darstellungsmöglichkeiten gibt es für eine Kurve in der Ebene? . . . . .	34
Wie wird ein Kreis auf einer Ebene dargestellt? . . . . .	35
Wie wird eine Kurve im Raum beschrieben? . . . . .	35
In welche Gruppen werden Splines unterteilt? . . . . .	35
Was sind die Eigenschaften der beiden Spline-Arten? . . . . .	35
Wieso werden Kurve gebraucht? . . . . .	36
Wie funktioniert die Methode der unbestimmten Koeffizienten? . . . . .	36
Wie funktioniert die Methode nach Lagrange? . . . . .	36
Welche Arten von Bézier Splines gibt es? . . . . .	36
Wie sind die Bernsteinpolynome definiert? . . . . .	36
Was sind die Eigenschaften von Bézier-Kurven? . . . . .	37
Was sind die Eigenschaften von B-Spline-Kurven? . . . . .	37
Wie werden B-Spline-Kurven berechnet? . . . . .	37
Wie werden rationale B-Splines-Kurven (NURBS) berechnet? . . . . .	37
Was sind NURBS-Kurven? . . . . .	37
<b>Flächen</b>	<b>38</b>
Wie wird eine Ellipsoid in 3D beschreiben? . . . . .	38
Wie funktioniert die Umrechnung in Zylinderkoordinaten? . . . . .	38
Wie funktionieren Kugel- oder sphärische Koordinaten? . . . . .	38
Wie wird eine Rotationsflächen um die $x$ -Achse beschrieben? . . . . .	39
Wie werden die Vektoren des Dreiein einer Rotationsflächen berechnen? . . . . .	39
Was ist eine Regelfläche? . . . . .	39
Was ist eine doppelte Regelfläche? . . . . .	39
Wie wird eine Freiformflächen beschrieben? . . . . .	39
Wie werden Bézierflächen berechnet? . . . . .	39
Wie wird die Flächennormale einer Bézier-Fläche berechnet? . . . . .	39
<b>Raytracing</b>	<b>40</b>
Für was wird Raytracing gebraucht? . . . . .	40
Wie funktioniert ein einfacher Raytracing Algorithmus? . . . . .	40
<b>Rendering Equation</b>	<b>40</b>
Was bedeutet die Abkürzung BRDF? . . . . .	40
Wie ist die Rendering Equation definiert? . . . . .	40

Was ist das Ziel von Radiosity? . . . . .	40
Radiosity Konzept . . . . .	40
Radiosity Gleichung . . . . .	40
Welche Möglichkeiten gibt es ein Gleichungssystem zu lösen? . . . . .	41
Was ist das Problem bei der Berechnung der Formfaktoren? . . . . .	41
Welche Möglichkeiten gibt es für die Berechnung der Radiosity? . . . . .	41

## Einführung

### Wo werden überall Computergrafik gebraucht?

GUI, Visualisierungen, Filme, Werbung, Architektur, Spiele, Virtual Reality

### Wie heissen die Schritte in einer CG Pipeline?

1. Objekte
2. Modelltransformation
3. 3D-Szene
4. Viewingtransformation
5. 3D-Clipping
6. Entfernen verdeckter Flächen
7. Beleuchtung
8. Perspektive
9. 2D-Szene
10. 2D-Clipping
11. Rasterung
12. Darstellung

## Farbe

### Durch welche Charakteristiken kann eine Farbe beschrieben werden?

- Farbton / Farbe
- Farbstich / Sättigung
- Helligkeit

### In welche Bereiche kann das sichtbare Licht aufgeteilt werden?

- Ultraviolett (weniger als 380nm)
- Violett (380–420nm)
- Blau (420–490nm)
- Grün (490–575)
- Gelb (575–585nm)
- Orange (585–650nm)
- Rot (650–750nm)
- Infrarot (mehr als 750nm)

### Wie entsteht die Farbe einer Fläche?

- Lichtquelle bescheint ein Objekt
- Oberfläche des Objekt reflektiert das Licht in Abhängigkeit des Normalenvektor
- Kamera fängt die Strahlen der Reflexion ein

### Aus welchen zwei Bestandteilen setzt sich die Farbe eines Objekts zusammen?

- Beleuchtung
- Reflexion

### Wie kann ein Auge mit einer Kamera verglichen werden?

- **Iris:** Kreisring mit radialen Muskeln
- **Pupille:** Öffnung, durch Iris kontrolliert
- **Linse:** erlaubt Fokussieren
- **Photorezeptoren:** Nehmen das Licht wahr

### Wie viele Stäbchen und Zäpfchen hat ein Mensch?

- 75 - 100\*10<sup>6</sup> - Stäbchen für Intensität
- 6 - 7\*10<sup>6</sup> - Zäpfchen für Farbe

### Was ist die Fovea?

Gebiet im Zentrum des Sichtfeldes mit grosser Zäpfchendichte, aber ohne Stäbchen

### Welche Arten von Zäpfchen gibt es?

- 3 Arten von Zäpfchen:
  - S - blau (440nm)
  - M - grün (530nm)
  - L - rot (560nm)
- Verhältnis: L : M : S = 10 : 5 : 1

### Wie sehen wir Farbe?

- Spektrale Strahlungsverteilung wird durch das Auge eingefangen
- Stäbchen und Zäpfchen reagieren auf diese Strahlungsverteilung
- Im Kopf wird aus diesen Komponenten dann die "Farbe" gebildet

### Was ist der unterschied der Helligkeitswahrnehmung bei Stäbchen und Zäpfchen?

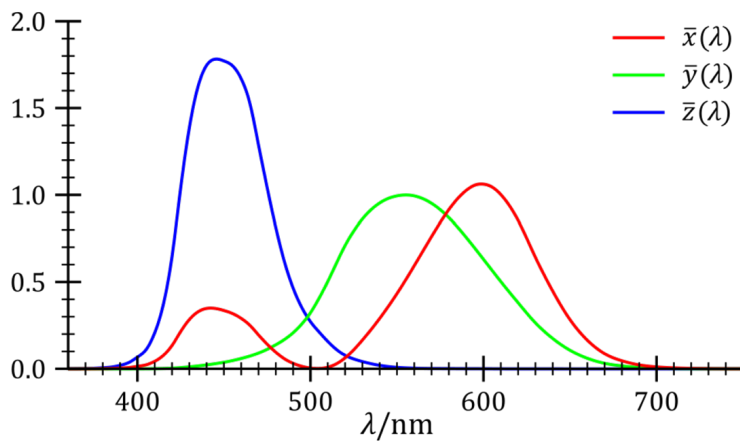
- Stäbchen
  - Scotopic Vision
  - 400nm - 600nm
  - besser bei weniger Licht
- Zäpfchen
  - Photopic Vision
  - 430nm - 680nm
  - besser bei mehr Licht

### Können alle Farben durch 3 Farben gemischt werden?

- Nein, weil teilweise negative Teile benötigt werden



## Wie sehen die Spektralwertkurven im CIE Farbsystem aus?

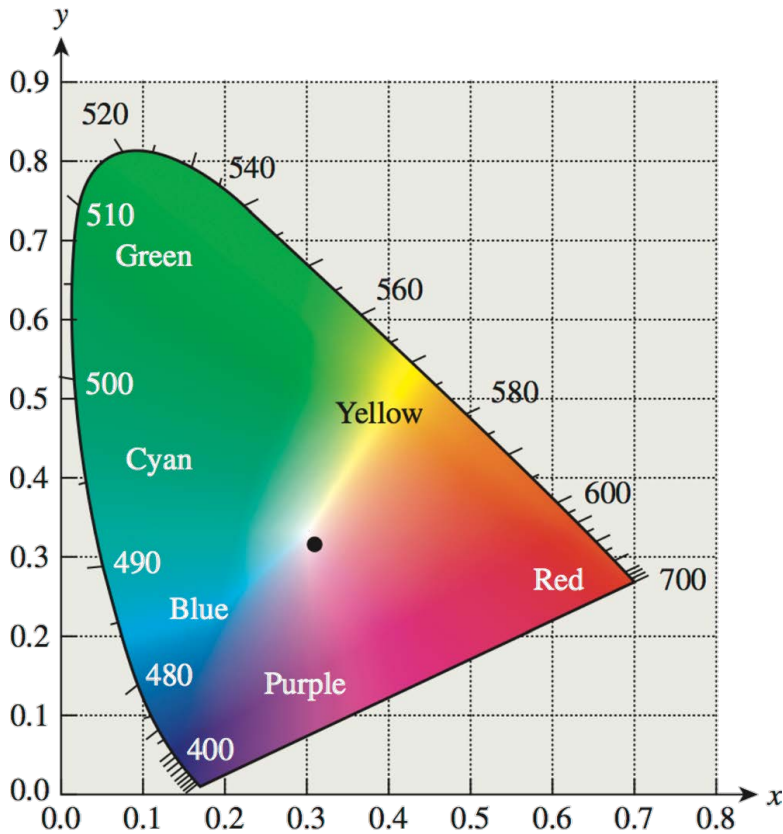


## Wie setzen sich die Anteile im CIE Farbsystem zusammen?

- $X = k \int P(\lambda) \bar{x}_\lambda d\lambda$
- $Y = k \int P(\lambda) \bar{y}_\lambda d\lambda$
- $Z = k \int P(\lambda) \bar{z}_\lambda d\lambda$

## Was ist die Normfarbtafel und wie sieht sie aus?

Ebene im X, Y, Z Farbraum die durch (1,0,0), (0,1,0) und (0,0,1) geht



## Welche Farbmodelle gibt es?

- Hardware orientierte Farbmodelle:
  - RGB für Monitore
  - CMY und CMYK für die Drucktechnik
  - YIQ für Fernsehen
- Intuitive Farbmodelle:
  - HSV
  - Munsellsystem
  - CIE Lab

## Was sind die Eigenschaften des RGB Farbmodell?

- Additives Farbmodell
- 3 Grundfarben (Rot, Grün, Blau)
- Wertebereich der Komponenten von 0 bis 1

## Welche Werte haben die “einfachen” RGB Farben?

(0, 0, 0) = Schwarz	(0, 1, 1) = Cyan
(1, 0, 0) = Rot	(0, 0, 1) = Blau
(1, 1, 0) = Gelb	(1, 0, 1) = Magenta
(0, 1, 0) = Grün	(1, 1, 1) = Weiss

## Was sind die Eigenschaften des CMY Farbmodell?

- Farbsystem für die subtraktive Farbmischung
- 3 Grundfarben (Cyan, Magenta, Gelb)
- Grundfarben sind Komplementärfarben von RGB
- Wertebereich der Komponenten von 0 bis 1

## Wie ist der mathematische Zusammenhang zwischen RGB und CMY?

$$C = 1 - R$$
$$M = 1 - G$$
$$Y = 1 - B$$

## Was bedeutet CMYK?

C = cyan, M = magenta, Y = yellow, K = black

## Wie lässt sich CMYK aus CMY berechnen?

$$CMY \Rightarrow C' M' Y' K$$
$$K = \min(C, M, Y)$$
$$C' = C - K$$
$$M' = M - K$$
$$Y' = Y - K$$

## Was sind die Eigenschaften des HSV Farbmodell?

- Empfindungsmässig bedeutsame Dimensionen
- Setzt sich ebenfalls aus 3 Grundkomponenten zusammen
  - Hue(Farbton, Farbe),
  - Saturation (Sättigung, Reinheit)
  - Value (Intensität)

## Welche Gradwerte von HSV werden den RGB Grundfarben zugeordnet?

xxx = Schwarz	xxx = Weiss
000 = Rot	060 = Gelb
120 = Grün	180 = Cyan
240 = Blau	300 = Magenta

## Was ist YUV?

- wird gebraucht für Composite Video
- Y: Helligkeit
- UV: Farbkodierung, häufig mit 1/4 der Auflösung

## Wie kann YUV aus RGB berechnet werden?

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$
$$U = 0.436 * (B - Y) / (1 - 0.114)$$
$$V = 0.615 * (R - Y) / (1 - 0.299)$$

## Für was stehen die Begriffe Tints und Shades?

- Tints:
  - Verringern der Sättigung
  - Von “reiner” Farbe zu Weiss
- Shades:
  - Verringern der Helligkeit
  - Von “reiner” Farbe zu Schwarz

## Was ist das CIE Lab System?

- L: Helligkeit
- a: rot - grün Achse
- b: gelb - blau Achse
- absolutes Farbsystem
- gleiche Abstände für gleich empfundene Farbunterschiede

## Was passiert bei einer Gammakorrektur?

- Anpassen einer oder mehrer Farbkurven im Bild
- Ziel ist die Helligkeit im Bild besser zu verteilen

## Was ist das spezielle von Helligkeit und Kontrast?

werden beide logarithmisch wahrgenommen

**Was ist Kontrast?**

Unterschied zwischen zwei Helligkeitswerte

**Was ist HDR?**

- High Dynamic Range
- Hoher Kontrast

**Was ist der Grund, dass viele Farbsysteme 3 Grundfarben verwenden?**

Farben werden durch 3 Typen von Zellen wahrgenommen.

**Wie viele Spektralfarben können durch Mischen von 2 Spektralfarben erzeugt werden?**

Keine

**Welche Farbe ist auch bei sehr wenig Licht noch gut sichtbar?**

grün-blau

**Welche Farben kann ein Monitor darstellen?**

Alle Farben die im Innern des Dreiecks RGB liegen

**In welcher Farbe erscheint ein gelber Tennisball?**

- wenn er von einer blauen Lichtquelle beschienen wird?
  - => schwarz
- wenn er von einer roten Lichtquelle beleuchtet wird?
  - => rot

**WebGL****Was sind die Merkmale von OpenGL?**

- Low Level Graphics API
- Unterstützung für verschiedene Plattformen
- gute Hardware Unterstützung auf allen Grafikkarten
- Definierte Extensions
- Programmierbare Shaders seit OpenGL 2.0
- Subset für Embedded Systems

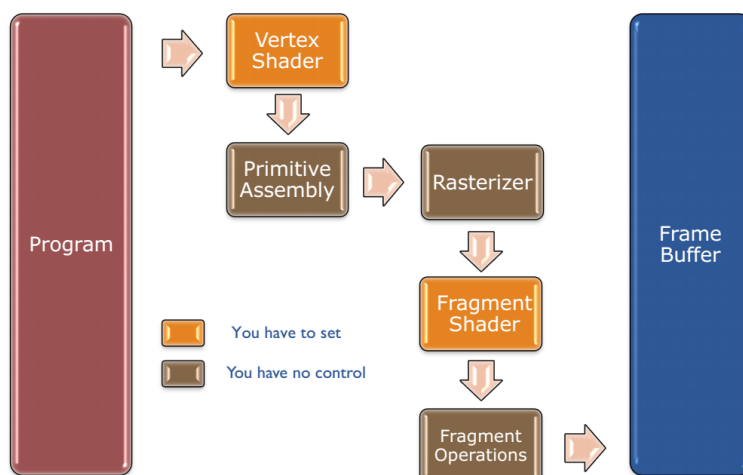
**Was sind die Vorteile von programmierbaren Shaders?**

Direkte Ausführung auf der Grafikkarte

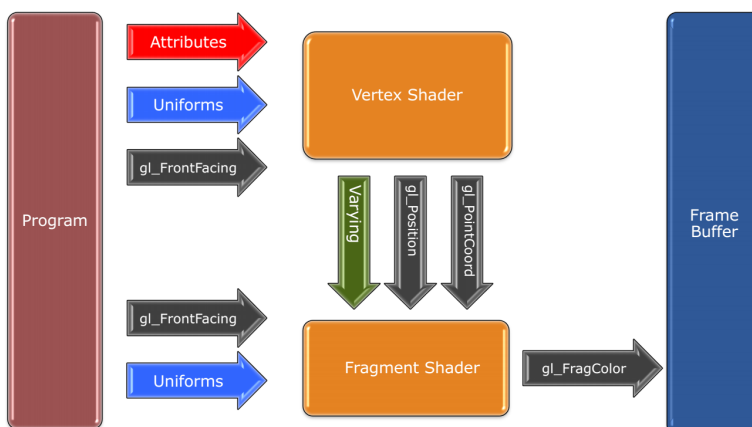
## Was ist der Unterschied zwischen Vertex & Fragment Processing?

- Vertex Processing:
  - Berechnet die Position eines Vertex
  - Vorberechnungen von weiteren Daten für Fragment Processing möglich
- Fragment Processing:
  - Berechnet die Farbe des Pixels

## Wie sieht eine vereinfachte WebGL Pipeline aus?



## Wie sieht der Datenfluss in einer WebGL Pipeline aus?



## Farbe, Halbtontechnik

### Welche Verfahren gibt es in der Halbtontechnik?

- Quantisierung
- Dithering
- Error Diffusion

### Was bedeutet der Begriff Quantisierung?

- Rundung auf die verfügbaren Farben

- nur möglich, falls noch genügend Farben zur Verfügung stehen

### Was bedeutet der Begriff Dithering?

- Drucker haben eine grössere Auflösung
- können aber weniger Farbstufen darstellen als der Bildschirm
- mittels Dithering werden Farben mit Auflösung simuliert

### Was versteht man unter dem Begriff Halbtontechnik?

- Grautöne werden durch schwarze Kreise auf weissem Hintergrund simuliert
- Anteil der schwarzen und weissen Flächen entspricht dem Grauwert

### Wie funktioniert die digitale Halbtontechnik?

- Ersetzen eines Grauwertpixels durch eine Pixelmatrix mit fester Grösse
- Je nach Grauwert sind verschieden viele Pixel gesetzt
- Weniger Farben werden benötigt, das Bild wird jedoch grösser

### Wie kann eine Dithermatrix aussehen?

2x2	3x3	4x4
---	-----	-----
0 2	6 8 4	0 8 2 10
3 1	1 0 3	12 4 14 6
	5 2 7	3 11 1 9
		15 7 13 5

### Was sind die Regeln für Dithering?

- einmal gesetzte Pixel bleiben gesetzt
- Strukturen sollen vermieden werden
- es soll möglichst ein Kreis approximiert werden

### Wie kann Dithering benutzt werden, wenn die Auflösung gleich bleiben soll?

- Berechnen des Mittelwert einer  $n \times n$  Region
- Ersetzen der Region durch die Dithermatrix  $\Rightarrow$  Clustered dot dithering
- Vergleich der einzelnen Pixel mit den Werten der Dithermatrix  $\Rightarrow$  Dispersed dot dithering

### Wie wird Dispersed Dot Dithering berechnet?

Beispiel für eine  $2 \times 2$  Dithermatrix:

```
i = x modulo n
j = y module n
```

der Pixel (x,y) wird gesetzt, falls  $I(x,y) > D_{ij}$

```
Dij = [ 0 2
        3 1 ]
```

```
Ineu = Ialt * 5 / 256
```

```
I_alt = [ 255 200 150 100 080      I_neu = [ 4 3 2 1 1
          200 150 100 080 040 ]          3 2 1 1 0 ]
```

### Was ist die Idee hinter Error Diffusion?

- Anstatt Kreise verschiedener Grösse, können auch Punkte verschieden dicht angeordnet werden
- Fehler durch das Setzen eines Pixels wird auf die umliegenden Pixel verteilt
- Bild wird sequentiell von oben nach unten und von links nach rechts durchlaufen
- Fehler wird anhand der folgenden Gewichte auf die noch nicht besuchten Pixel verteilt

```
Gewichtung = [ ---- xxxx 7/16
               1/16 5/16 3/16 ]
```

### Wie wird die Error Diffusion berechnet?

Beispiel für 2 gültige Farben (255 & 0):

```
Z1 = [          Z2 = [          Z3 = [
      xxx 191 140 113      xxx 255 112 113      xxx xxx 000 162
      244 221 105 100      240 201 093 100      240 208 128 121
]                               ]
```

### Wie viele Intensitätsstufen können mit einer Dithermatrix dargestellt werden?

- $2 \times 2$  Dithermatrix  $\Rightarrow$  5 Stufen
- $3 \times 3$  Dithermatrix  $\Rightarrow$  10 Stufen
- $4 \times 4$  Dithermatrix  $\Rightarrow$  17 Stufen

### Wie wird eine RGB Farbe in einen Grauwert umgerechnet?

Grauwert =  $0.299 * R + 0.587 * G + 0.114 * B$

## Projektive Geometrie 1

### Wie werden zwei Vektoren addiert?

$$\vec{a} + \vec{b} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \end{bmatrix}$$

### Wie wird ein Vektor mit einem Skalar multipliziert?

$$\lambda \vec{a} = \lambda \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \lambda a_1 \\ \lambda a_2 \end{bmatrix}$$

### Was ist das Inverse eines Vektors?

$$-\vec{a} = - \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \end{bmatrix}$$

### Was ist der Nullvektor?

$$\vec{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

**Wie werden zwei Vektoren subtrahiert?**

$$\vec{a} - \vec{b} = \vec{a} + (-\vec{b})$$

**Wann sind zwei Vektoren gleich?**

wenn ihre Komponenten gleich sind

**Was ist der Unterschied zwischen einem Rechtssystem und einem Linkssystem?**

- Rechtssystem  $\Rightarrow$  von X nach Y im Gegenuhrzeigersinn
- Linkssystem  $\Rightarrow$  von X nach Y im Uhrzeigersinn

**Wie kann man sich den Unterschied zwischen den beiden Systemen merken?**

- Daumen = Z, Zeigefinger = X, Mittelfinger = Y
- Rechte Hand  $\Rightarrow$  Rechtssystem, Linke Hand  $\Rightarrow$  Linkssystem

**Wie ist ein Vektor im Kartesisches Koordinatensystem definiert?**

$$\vec{a} = a_x \vec{e}_x + a_y \vec{e}_y = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

**Was muss bei der Schreibweise im Kartesisches Koordinatensystem beachtet werden?**

- $\vec{e}_x$  wird manchmal auch als  $\vec{i}$  bezeichnet
- $\vec{e}_y$  wird manchmal auch als  $\vec{j}$  bezeichnet

**Wie wird die Länge eines 2D-Vektors bestimmt?**

$$a = |\vec{a}| = \sqrt{a_x^2 + a_y^2}$$

**Welche Rechenregeln gelten für Vektoren?**

- $\vec{a} + \vec{b} = \vec{b} + \vec{a}$  (*Kommutativgesetz*)
- $\vec{a} + (\vec{b} + \vec{c}) = (\vec{a} + \vec{b}) + \vec{c}$  (*Assoziativgesetz*)
- $\lambda(\vec{a} + \vec{b}) = \lambda\vec{a} + \lambda\vec{b}$
- $(\lambda + u)\vec{a} = \lambda\vec{a} + u\vec{a}$
- $(\lambda u)\vec{a} = \lambda(u\vec{a}) = u(\lambda\vec{a})$

**Wie ist das Skalarprodukt definiert?**

$$\vec{a} \odot \vec{b} = |\vec{a}| |\vec{b}| \cos \phi$$

**Wie kann das Skalarprodukt im kartesischen Koordinatensystem berechnet werden?**

$$\vec{a} \odot \vec{b} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \odot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = a_1 b_1 + a_2 b_2$$



**Wie wird das Skalarprodukt im allgemeinen Fall berechnet?**

- $\vec{a} = a_1\vec{e}_1 + a_2\vec{e}_2 + a_3\vec{e}_3 = [a_1 a_2 a_3]^T$
- $\vec{b} = b_1\vec{e}_1 + b_2\vec{e}_2 + b_3\vec{e}_3 = [b_1 b_2 b_3]^T$
- $\vec{a} \odot \vec{b} = \sum_{i=1}^3 \sum_{j=1}^3 g_{ij} a_i b_j = [a_1 a_2 a_3] \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = a^T G b$

**Welche Rechenregeln gelten für das Skalarprodukt?**

- $\vec{a} \odot \vec{b} = \vec{b} \odot \vec{a}$  (Kommutativgesetz)
- $\vec{a} \odot (\vec{b} + \vec{c}) = \vec{a} \odot \vec{b} + \vec{a} \odot \vec{c}$  (Distributivgesetz)
- $\lambda(\vec{a} + \vec{b}) = (\lambda\vec{a}) \odot \vec{b} = \vec{a} \odot (\lambda\vec{b})$

**Was sind zwei Vektoren orthogonal?**

$$\vec{a} \odot \vec{b} = 0 \iff \vec{a} \perp \vec{b}$$

**Wie ist eine Linearkombination definiert?**

$$\vec{c} = \lambda\vec{a} + \mu\vec{b} \text{ wobei } \lambda, \mu \in \mathbb{R}$$

**Wie ist lineare Unabhängigkeit definiert?**

$$\lambda_1\vec{a}_1 + \lambda_2\vec{a}_2 + \dots + \lambda_n\vec{a}_n = \vec{0} \iff \lambda_1 = \lambda_2 = \dots = \lambda_n = 0$$

**Auf welche Arten kann eine Gerade beschrieben werden?**

- Punkt-Punktform mit Vektoren
  - $\vec{r} = \vec{r}_0 + t(\vec{r}_1 - \vec{r}_0), t \in \mathbb{R}$
- Punkt-Richtungsform mit Vektoren
  - $\vec{r} = \vec{r}_0 + t\vec{r}_1, t \in \mathbb{R}$
- Achsenabschnitt-Steigungsform
  - $y = mx + b$
- Punkt-Richtungsform
  - $(y - y_0) = m(x - x_0)$
- Allgemeine Geradengleichung
  - $ax + by + c = 0, a \in \mathbb{R}, b \in \mathbb{R}, c \in \mathbb{R}$

**Wie ist die Hessesche Normalform (HNF) der Gerade definiert?**

$$\vec{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} \quad \vec{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \vec{x}_0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad \vec{n} \odot (\vec{x} - \vec{x}_0) = \begin{bmatrix} n_x \\ n_y \end{bmatrix} \odot \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = 0 \text{ da } \vec{n} \perp (\vec{x} - \vec{x}_0)$$

$$\vec{n} \odot (\vec{x} - \vec{x}_0) = n_x(x - x_0) + n_y(y - y_0) = n_x x + n_y y - (n_x x_0 + n_y y_0)$$

**Wie kann der Abstand einer Gerade zum Ursprung berechnet werden?**

- $d = n_x x_0 + n_y y_0 = \vec{n} \odot \vec{x}_0$
- $d > 0 \iff (0, 0) \in -HE$  (negative Halbebene)
- $d < 0 \iff (0, 0) \in +HE$  (positive Halbebene)

**Wie kommt man von der 2D-Koordinatengleichung zur HNF?**

$$ax + by + c = 0$$

$$n_x = \frac{a}{\sqrt{a^2 + b^2}} \quad n_y = \frac{b}{\sqrt{a^2 + b^2}} \quad d = -\frac{c}{\sqrt{a^2 + b^2}}$$

$$n_x x + n_y y - d = 0$$

**Wie ist der normierter Normalenvektor in 2D definiert?**

$$\vec{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \frac{1}{\sqrt{a^2 + b^2}} \begin{bmatrix} a \\ b \end{bmatrix}$$

**Wie kommt man von der 3D-Koordinatengleichung zur HNF?**

$$ax + by + cz + d = 0$$

$$n_x = \frac{a}{\sqrt{a^2 + b^2 + c^2}} \quad n_y = \frac{b}{\sqrt{a^2 + b^2 + c^2}} \quad n_z = \frac{c}{\sqrt{a^2 + b^2 + c^2}} \quad D = -\frac{d}{\sqrt{a^2 + b^2 + c^2}}$$

$$n_x x + n_y y + n_z z - D = 0$$

**Wie ist der normierter Normalenvektor in 3D definiert?**

$$\vec{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \frac{1}{\sqrt{a^2 + b^2 + c^2}} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

**Wie wird die Projektion eines Vektors auf einen anderen Vektor berechnet?**

Komponente von  $\vec{b}$  in Richtung von  $\vec{a}$ :  $\vec{b}_{\vec{a}} = \frac{\vec{a} \odot \vec{b}}{|\vec{a}|^2} \vec{a}$

**Was sind die Eigenschaften des Vektorprodukt?**

- $a \times b$  steht senkrecht auf beiden Vektoren
- $|a \times b|$  entspricht der Fläche des aufgespannten Parallelogramms
- die Vektoren  $a$ ,  $b$  und  $a \times b$  bilden in dieser Reihenfolge ein Rechtssystem

**Wie wird das Vektorprodukt berechnet?**

$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad a \times b = (a_2 b_3 - a_3 b_2) e_1 + (a_3 b_1 - a_1 b_3) e_2 + (a_1 b_2 - a_2 b_1) e_3$$

Merkhilfe:

**Was sind die Rechenregeln für das Vektorprodukt?**

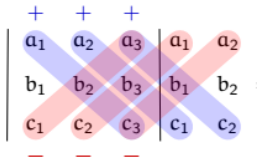
- $\vec{a} \times \vec{b} = -\vec{b} \times \vec{a}$  (Anti-Kommutativgesetz)
- $\vec{a} \times (\vec{b} + \vec{c}) = \vec{a} \times \vec{b} + \vec{a} \times \vec{c}$  (Distributivgesetz)
- $\lambda(\vec{a} \times \vec{b}) = (\lambda\vec{a}) \times \vec{b} = \vec{a} \times (\lambda\vec{b})$

**Wie wird das Spatprodukt berechnet?**

$$[\vec{a}, \vec{b}, \vec{c}] = \vec{a} \odot (\vec{b} \times \vec{c}) = \det \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix}$$

$$= a_1 b_2 c_3 + a_2 b_3 c_1 + a_3 b_1 c_2 - a_3 b_2 c_1 - a_1 b_3 c_2 - a_2 b_1 c_3$$

Merkhilfe:

**Was sind die Rechenregeln für das Spatprodukt?**

- Vertauschen von zwei bewirkt Vorzeichenwechsel

$$- [\vec{a}, \vec{b}, \vec{c}] = -[\vec{b}, \vec{a}, \vec{c}]$$

- Zyklisches Vertauschen ändert nichts

$$- [\vec{a}, \vec{b}, \vec{c}] = [\vec{b}, \vec{c}, \vec{a}]$$

- Multiplikation mit reellen Zahlen

$$- [\lambda \vec{a}, u \vec{b}, v \vec{c}] = \lambda uv [\vec{a}, \vec{b}, \vec{c}]$$

- Addition zweier Vektoren

$$- [\vec{a} + \vec{b}, \vec{c}, \vec{d}] = [\vec{a}, \vec{c}, \vec{d}] + [\vec{b}, \vec{c}, \vec{d}]$$

**Projektive Geometrie 2****Wieso wird die Projektive Ebene benutzt?**

- In der Projektiven Ebene haben zwei Geraden immer einen Schnittpunkt
- Dadurch lassen sich viele geometrische Phänomene knapper beschreiben
- Jeder Punkt hat eine Darstellung als Vektor mit drei homogenen Komponenten
- Diesem Vektor entspricht natürlich ein Punkt im dreidimensionalen Euklidischen Raum  $\mathbb{R}^3$

**Wie wird ein Punkt homogenisiert?**

$$\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \approx \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

**Wie wird ein Punkt dehomogenisiert / realisiert?**

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \begin{bmatrix} x_1/x_3 \\ x_2/x_3 \end{bmatrix}$$

**Was sind die Eigenschaften einer Projektive Transformationen?**

- ist eineindeutig (bijektive) und daher umkehrbar
- bildet Geraden auf Geraden ab (ist also geradentreu)

**Wie ist die allgemeine Skalierungs-Matrix definiert?**

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Wie ist die allgemeine Translations-Matrix definiert?**

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

**Wie ist die allgemeine Rotations-Matrix definiert (Rotation um den Nullpunkt)?**

$$R = \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Wie ist die allgemeine Spiegelungs-Matrix definiert (Spiegelung durch den Ursprung)?**

$$R = \begin{bmatrix} \cos 2\delta & \sin 2\delta & 0 \\ \sin 2\delta & -\cos 2\delta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Wie ist die Matrix für die Rotation um einen beliebigen Punkt definiert?**

$$R_a = T^{-1} R_0 T = \begin{bmatrix} \cos \phi & -\sin \phi & -t_x \cos \phi + t_x \sin \phi + t_x \\ \sin \phi & \cos \phi & -t_y \cos \phi - t_x \sin \phi + t_x \\ 0 & 0 & 1 \end{bmatrix}$$

**Welche Arten von Transformationen gibt es?**

- Euklidischer Transformation (*siehe R*)
  - Zusammensetzungen von Translation, Drehung und Spiegelung
  - Abstand zwischen zwei Punkten und alle Winkel bleiben gleich
- Ähnlichkeitstransformation (*siehe S*)
  - Euklid + Skalierung
  - Winkel zwischen zwei Geraden und Verhältnis von Abständen bleiben gleich
- affiner Transformation (*siehe A*)
  - Parallelität von Geraden und Verhältnis zwischen Flächeninhalten bleiben erhalten
  - Form der Figuren wird verzerrt
- allg. Transformation (*siehe H*)
  - alles erlaubt
  - Geraden bleiben Geraden

$$R = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \quad S = \begin{bmatrix} k * M & t \\ 0^T & 1 \end{bmatrix} \quad A = \begin{bmatrix} C & t \\ 0^T & 1 \end{bmatrix} \quad H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

## Projektive Geometrie 3

**Wie ändern sich die Koordinaten bei einer 2D-Transformation des Koordinatensystems?**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

**Wie ändern sich die Koordinaten bei einer 2D-Rotation des Koordinatensystems?**

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

**Wie kann diese Rotation auch gemacht werden?**

Statt das Koordinatensystem um  $\phi$  zu drehen, kann man das Objekt um  $-\phi$  drehen und vice versa!

**Wie kann ein Raumpunkt in homogene Koordinaten umgerechnet werden?**

$$\vec{r} = [x_1, x_2, x_3, x_4]^T$$

$$(x, y, z) = \left( \frac{x_1}{x_4}, \frac{x_2}{x_4}, \frac{x_3}{x_4} \right)$$

**Wie wird eine Ebenen im Raum definiert?**

$$\vec{q} \odot \vec{r} = w^T * r = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = ax + by + cz + d = 0$$

**Wie sieht die allgemeine Translation-Matrix in 3D aus?**

$$\vec{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, T(\vec{t}) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, T^{-1}(\vec{t}) = T(-\vec{t}) = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Wie sieht die Rotationsmatrix um die z-Achse aus?**

*Rotation um den Winkel  $\phi_z$  um die z-Achse im Gegenuhrzeigersinn*

$$R_z(\phi_z) = \begin{bmatrix} \cos(\phi_z) & -\sin(\phi_z) & 0 & 0 \\ \sin(\phi_z) & \cos(\phi_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_z^{-1}(\phi_z) = R_z(-\phi_z) = \begin{bmatrix} \cos(\phi_z) & \sin(\phi_z) & 0 & 0 \\ -\sin(\phi_z) & \cos(\phi_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Wie sieht die Rotationsmatrix um die y-Achse aus?**

*Rotation um den Winkel  $\phi_y$  um die y-Achse im Gegenuhrzeigersinn*

$$R_y(\phi_y) = \begin{bmatrix} \cos(\phi_y) & 0 & \sin(\phi_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi_y) & 0 & \cos(\phi_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_y^{-1}(\phi_y) = R_y(-\phi_y) = \begin{bmatrix} \cos(\phi_y) & 0 & -\sin(\phi_y) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\phi_y) & 0 & \cos(\phi_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Wie sieht die Rotationsmatrix um die x-Achse aus?**

Rotation um den Winkel  $\phi_x$  um die x-Achse im Gegenuhrzeigersinn

$$R_x(\phi_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi_x) & -\sin(\phi_x) & 0 \\ 0 & \sin(\phi_x) & \cos(\phi_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, R_x^{-1}(\phi_x) = R_x(-\phi_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi_x) & \sin(\phi_x) & 0 \\ 0 & -\sin(\phi_x) & \cos(\phi_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Welche Schritte sind für die Rotationen um eine beliebige Achse nötig?**

1. Rotation um den Winkel  $\phi$  um die z-Achse (mit der Matrix D)
2. Rotation um den Winkel  $\theta \in [0, \pi]$  um die frühere x-, jetzt x'-Achse (mit der Matrix C)
3. Rotation um den Winkel  $\psi$  um die frühere z-, jetzt z'-Achse (mit der Matrix B)

**Wie ist die Matrix für eine Rotationen um eine beliebige Achse definiert?**

$c_\alpha = \cos \alpha, s_\alpha = \sin \alpha$  wobei  $\alpha \in \{\phi, \theta, \psi\}$

$$D = \begin{bmatrix} c_\phi & s_\phi & 0 & 0 \\ -s_\phi & c_\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\theta & s_\theta & 0 \\ 0 & -s_\theta & c_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} c_\psi & s_\psi & 0 & 0 \\ -s_\psi & c_\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A = BCD = \begin{bmatrix} c_\psi c_\phi - c_\theta s_\phi s_\psi & c_\psi s_\phi + c_\theta c_\phi s_\psi & s_\psi s_\theta & 0 \\ -s_\psi c_\phi - c_\theta s_\phi s_\psi & -s_\psi s_\phi + c_\theta c_\phi s_\psi & c_\psi s_\theta & 0 \\ s_\theta s_\phi & -s_\theta c_\phi & c_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Wie wird eine parallele Projektion beschrieben?**

Projektion in Richtung  $v = [v_x, v_y, v_z]^T$  auf die Ebene  $ax + by + cz + d = 0$  wenn  $a^2 + b^2 + c^2 = 1$

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = H \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \frac{1}{\cos \psi} \begin{bmatrix} (\cos \psi - av_x) & -bv_x & -cv_x & -dv_x \\ -av_y & (\cos \psi - bv_y) & -cv_y & -dv_y \\ -av_z & -bv_z & (\cos \psi - cv_z) & -dv_z \\ 0 & 0 & 0 & \cos \psi \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$$

**Wie wird eine perspektivische Projektion beschrieben?**

perspektivische Projektion mit Zentrum im Nullpunkt und mit Bildebene  $ax + by + cz + d = 0$

$$\begin{bmatrix} x^* \\ y^* \\ z^* \\ 1 \end{bmatrix} = H \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & -d & 0 \\ a & b & c & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}$$

**Wie wird eine Clipping-Transformation beschrieben?**

$$T = \begin{bmatrix} \frac{2a}{x_Q - x_P} & 0 & \frac{x_Q + x_P}{x_Q - x_P} & 0 \\ 0 & \frac{2a}{y_Q - y_P} & \frac{y_Q + y_P}{y_Q - y_P} & 0 \\ 0 & 0 & -\frac{b+a}{b-a} & -2\frac{ba}{b-a} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$a$  = Abstand vom Nullpunkt,  $P(x_P, y_P, -a)$  = Punkt unten links,  $Q(x_Q, y_Q, -a)$  = Punkt oben rechts

**Wie sieht die allgemeine Skalierungs-Matrix aus?**

$$\vec{s} = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}, S(\vec{s}) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, S^{-1}(\vec{s}) = \begin{bmatrix} -s_x & 0 & 0 & 0 \\ 0 & -s_y & 0 & 0 \\ 0 & 0 & -s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Scan Konvertierung****Wie könnte ein Algorithmus zum Zeichnen einer Linie aussehen?**

Brute Force Version

```
-----
for x = x0 to x1 do
  y = m * x + B
  WritePixel(x, Round(y))
end
```

Inkrementell Version

```
-----
y = B
for x = x0 to x1 do
  y += m
  WritePixel(x, Round(y))
end
```

**Was ist ein DDA?**

Digital Differential Analyzer

**Was sind die Nachteile von DDA?**

- es wird mit Gleitkommazahlen gerechnet
- die Rundungsoperation ist aufwendig

**Welche Arten gibt es um eine Gerade zu definieren?**

- explizit (Geradengleichung)
- implizit (Mittelpunktschema)
- Parameterdarstellung

**Was ist die Definition des Mittelpunktschema?**

- Implizite Darstellung einer Linie:
  - $F(x, y) = ax + by + c$
- Ist der Punkt (x,y) auf der Linie, so gilt:
  - $F(x, y) = 0$ ,
- Ist der Punkt (x,y) unterhalb oder oberhalb der Linie so ist:
  - $F(x, y) = d \neq 0$

**Wie wird beim Runden entschieden wo ein Punkt gezeichnet werden soll?**

- Bei einer Liniensteigung gibt es zwei Möglichkeiten
  - $E = (x_i + 1, y_i)$
  - $NE = (x_i + 1, y_i + 1)$
- Um zu entscheiden betrachten wir den Mittelpunkt
  - $M = (x_i + 1, y_i + 1/2)$

- Liegt M unterhalb der Linie, so nehmen wir NE, andernfalls E
- Um zu entscheiden auf welcher Seite M liegt, betrachten wir

$$- d = F(M) = a(x_i + 1) + b(y_i + 1/2) + c = \Delta y(x_i + 1) - \Delta x(y_i + 1/2) + c$$

### Wie könnte der Mittelpunktschema-Algorithmus implementiert werden?

```
// Initialisierung      // Berechnung
Dx = x1 - x0;           WritePixel(x0, y0)
Dy = y1 - y0;           for x = x0+1 to x1 do
DE = 2 * Dy;             if d <= 0 then
DNE = 2 * (Dy - Dx);     d = d + DE;
d = 2 * Dy - Dx;         else
y = y0;                 d = d + DNE;
                        y = y + 1;
                        end
                        WritePixel(x,y);
                        end
end
```

### Wie kann das Mittelpunktschema für Kreise angewandt werden?

$$d = F(M) = (x_i + 1)^2(y_i - 1/2)^2 - R^2$$

### Wie könnte der Mittelpunktschema-Algorithmus für Kreise implementiert werden?

```
// Initialisierung      // Berechnung
x = 0;                 WritePixel(x,y);
y = R;                 while y > x do
d = 1 - R;             if d < 0 then
                        d = d + 2*x + 3;
                        else
                        d = d + 2*(x-y) + 5;
                        y = y - 1;
                        end
                        x = x + 1;
                        WritePixel(x,y);
                        end
end
```

### Wie könnte ein Algorithmus zum Füllen von Regionen implementiert werden?

```
proc FloodFill(int x, int y, Color oldColor, Color newColor)
  if ReadPixel(x, y) == oldColor then
    WritePixel(x, y, newColor);
    FloodFill(x, y-1, oldColor, newColor);
    FloodFill(x-1, y, oldColor, newColor);
    FloodFill(x, y+1, oldColor, newColor);
    FloodFill(x+1, y, oldColor, newColor);
  end
end
```

### Wie werden gefüllten Polygonen gezeichnet?

- Effizient mit dem Scanlinien-Algorithmus
- Dabei werden Spans entlang der y Koordinate des Fensters gezeichnet



- Die Kanten werden dabei vorsortiert und die aktuellen zu zeichnenden Spans verwaltet

### Welche Tabellen benötigt der Scanlinien-Algorithmus?

- Kantentabelle (ET):
  - enthält alle Kanten nach minimaler  $y$  Koordinate sortiert
  - innerhalb gleicher  $y_{min}$  wird nach  $x$  sortiert
- Tabelle aktiver Kanten (AET):
  - enthält alle Kanten die von der aktuellen Scanlinie geschnitten werden
  - sortiert nach  $x$
- Informationen pro Kante:
  - $x, 1/m, y_{min}, y_{max}$

### Wie könnte ein Scanlinienalgorithmus implementiert werden?

```
Erzeuge ET
Initialisiere AET = empty
y = 0

Repeat
    Addiere alle Kanten ET(y) zu AET
    Sortiere AET nach x
    Zeichne Spans
    y = y + 1
    Entferne Kanten mit ymax = y aus AET
    Aktualisiere den x Wert aller Kanten in AET
Until AET == empty
```

### Welche Ansätze gibt es für das Zeichnen von Dreiecken?

- “Brute Force” Algorithm:
  - für alle Pixel-Kandidaten (innerhalb Bounding-Box)
  - berechne ob sich der Pixel innerhalb des Dreiecks befindet
- Baryzentrische Koordinaten
  - Darstellung eines Punktes P im Dreieck ABC
  - $P = \alpha A + \beta B + \gamma C, \alpha + \beta + \gamma = 1$
  - $\alpha, \beta$  und  $\gamma$  heißen baryzentrische Koordinaten
  - Werte zwischen 0 und 1 falls Punkt innerhalb ABC
- Unterteilung der Ebene mit 3 Geraden (Dreieckskanten)
  - jeweils prüfen ob Punkt links/recht oder oben/unten der Gerade ist

### Was ist Anti-Aliasing?

- beim Zeichnen von Linien und Polygonen entsteht ein Treppemuster
- zum Verhindern von Aliasing werden Anti-Aliasing Techniken verwendet
  - beispielsweise Prefiltering oder Supersampling

## Wie funktioniert Prefiltering?

- arbeitet beispielsweise mit Graustufen
- bestimme die Fläche der Linie auf dem Pixel
- wähle die Farbintensität proportional zur Fläche

## Wie funktioniert Supersampling?

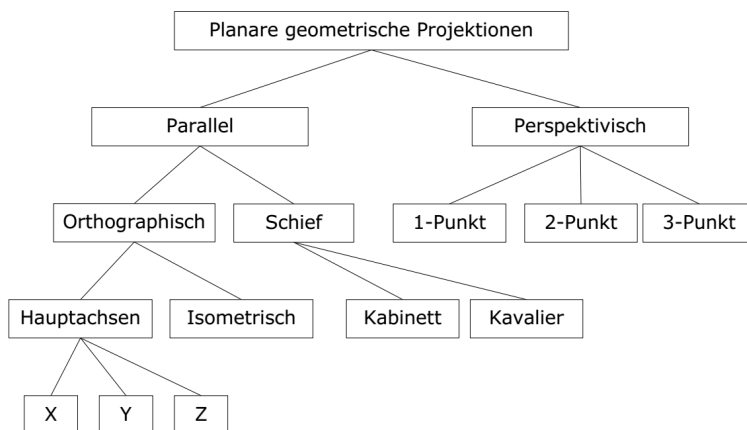
- Bild in  $n$  Mal grössere Auflösung berechnen
- immer  $n \times n$  Pixel zu einem Zusammenfassen
- als Farbwert den Mittelwert der Pixel nehmen

## Viewing

### Was sind die Eigenschaften der Parallel-Projektion?

- Geraden bleiben erhalten
- Parallelen bleiben erhalten
- Konstante Verkürzung in eine gegebene Richtung
- keine Winkeltreue

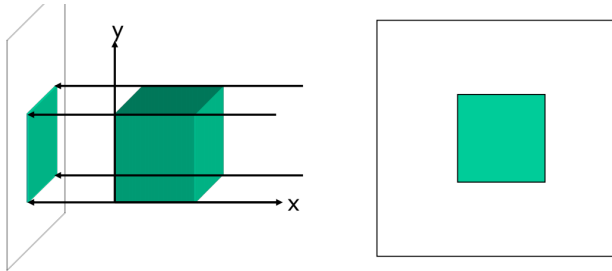
### Welche Arten von Projektionen gibt es?



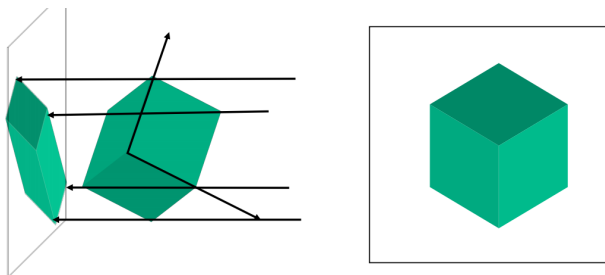
- Paralell-Projektion
  - Spezifikation durch Ebene und Projektionsrichtung
- Orthographische Projektion
  - Rechter Winkel zwischen Projektionsebene und Projektionsrichtung
- Isometrische Projektion
  - Orthographische Projektion auf Ebene mit Normalvektor  $(1, 1, 1)$
- Kavaliersprojektion
  - Parallelprojektion mit 45 Grad Winkel zwischen Ebene und Projektionsrichtung
  - Linien rechtwinklig zur Ebene haben natürliche Länge
- Kabinettsprojektion
  - Parallelprojektion mit 63.4 Grad zwischen Ebene und Projektionsrichtung

- Linien rechtwinklig zur Ebene haben halbe Länge
- Perspektivische Projektion
  - Simuliert Kamera oder Auge
  - Spezifiziert durch Projektionszentrum und Projektionsebene

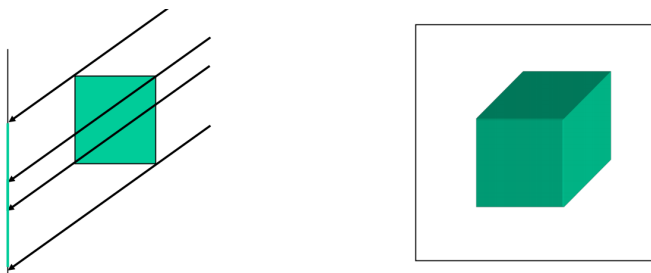
### Wie sieht eine Orthographische Projektion aus?



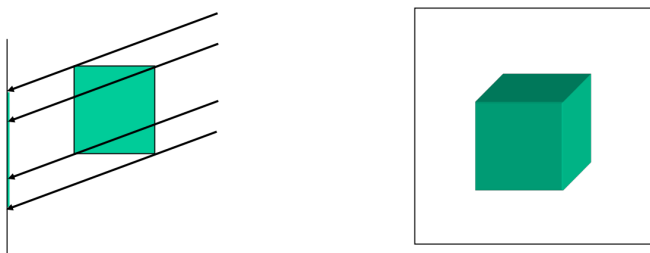
### Wie sieht eine Isometrische Projektion aus?



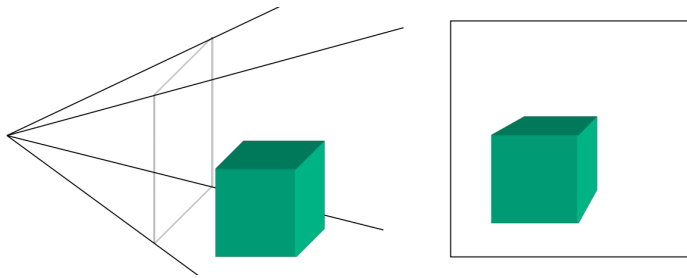
### Wie sieht eine Kavaliersprojektion aus?



### Wie sieht eine Kabinettsprojektion aus?



## Wie sieht eine Perspektivische Projektion aus?



## Was sind die Eigenschaften der Paralell-Projektion?

- Geraden bleiben erhalten
- Parallelen bleiben erhalten
- Konstante Verkürzung in eine gegebene Richtung
- keine Winkeltreu

## Was sind die Eigenschaften der perspektivischen Projektion?

- Geraden bleiben erhalten
- Parallelen schneiden sich in einem Punkt
- Objektgröße nimmt proportional zum Abstand vom Projektionszentrum ab

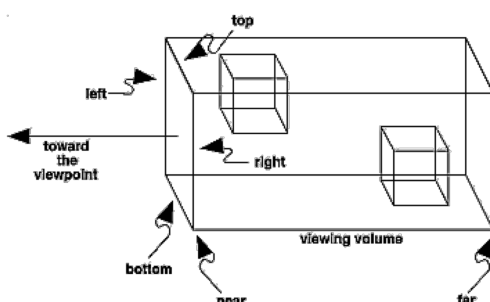
## Wie kann die Kamera Position spezifiziert werden?

```
var matrix = mat4.create();
mat4.lookAt(
    matrix,
    [eyeX, eyeY, eyeZ],
    [centerX, centerY, centerZ],
    [upX, upY, upZ]
);

// Parameter beschreibung
// [eyeX,   eyeY,   eyeZ   ] Koordinaten der Kamera in Weltkoordinaten
// [centerX, centerY, centerZ] Wohin schaut die Kamera
// [upX,    upY,    upZ    ] Richtungsvektor der bei der Kamera nach oben zeigt
```

## Wie wird eine Orthographische Projektion mit WebGL implementiert?

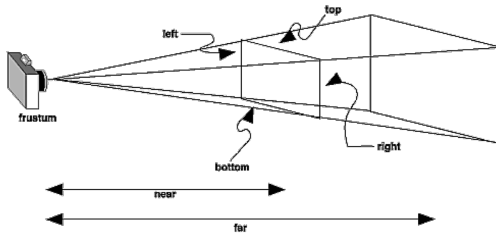
```
var projectionMatrix = mat4.create();
mat4.ortho(projectionMatrix, left, right, bottom, top, near, far);
```



## Wie wird eine perspektivische Projektion mit WebGL implementiert?

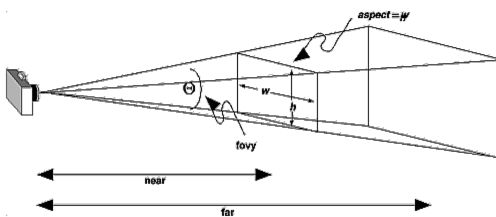
### Variante 1:

```
var projectionMatrix = mat4.create();
mat4.frustum(projectionMatrix, left, right, bottom, top, near, far);
```



### Variante 2:

```
var projectionMatrix = mat4.create();
mat4.perspective(projectionMatrix, fovy, aspect, near, far);
```



## Visibility

### Was sind Algorithmen für verdeckte Linien und Flächen?

- Backface Culling (WebGL)
- Tiefensortierung (Painters Algorithms)
- Z-Buffer (WebGL)
- Warnock Algorithmus
- Raycasting / Raytracing

### Wie funktioniert Backface Culling?

- bei geschlossenen Objekten sind die Rückseiten nie sichtbar
- dh. sie müssen nicht gezeichnet werden
- der Test erfolgt durch die Betrachtung des Normalenvektors
- falls dieser negative ist wird die Seite ignoriert

### Was ist die Idee hinter der Tiefensortierung?

- Sortiere Polygone in geeigneter Reihenfolge
- Zeichne Polygone in dieser Reihenfolge,
- die zuletzt gezeichneten Polygone überdecken die früheren

## Wie wird bei der Sortierung für die Tiefensortierung vorgegangen?

1. sortiere Polygone nach minimaler z-Koordinate
2. falls sich die Polygone in z überlagern, betrachte zusätzliche Bedingungen
3. falls keine der Bedingungen erfüllt ist,  
vertausche die Polygone und überprüfe die Bedingungen nochmals

## Welche Bedingungen werden bei der Tiefensuche überprüft?

*Voraussetzung: Polygon P ist in der Sortierung vor Q*

1. Überlagern sich die x-Ausdehnungen nicht?
2. Überlagern sich die y-Ausdehnungen nicht?
3. Liegt P ganz auf der vom Betrachter abgewandten Seite von Q?
4. Liegt Q ganz auf der Betrachterseite von P?
5. überlappen sich die Polygone nicht auf der Projektion in die xy Ebene?

## Was ist wenn es keine gültige Reihenfolge für die Tiefensortierung gibt?

dann muss ein Polygon entlang der Ebene des anderen geschnitten werden

## Was sind die Vor- und Nachteile der Tiefensortierung?

- **Vorteile:** auch für transparente Objekte möglich, einfach für Spezialfälle (2.5D)
- **Nachteile:** ineffizient für viele Objekte  $O(n^2)$ , nicht Hardware unterstützt

## Was ist die Idee des Z-Buffer Algorithmus?

- pro Pixel wird ausser der Farbe auch die Tiefe (z-Wert) gespeichert
- beim Zeichnen jedes Pixels wird geprüft ob der neue Pixel näher liegt

## Wie könnte der Z-Buffer Algorithmus implementiert werden?

```
// Initialisierung
for y = 0 to YMAX do
  for x = 0 to XMAX do
    color[x,y] = Background
    depth[x,y] = MAXDEPTH
  end
end

// Polygone zeichnen
for alle Polygone q do
  for alle pixel p(x,y) in q do
    z = CalculateDepth(q,x,y);
    if z > depth[x,y] then
      depth[x,y] = z;
      color[x,y] = Color of q
    end
  end
end
```

## Wie wird z berechnet?

$$z = \frac{-D - Ax - By}{C}, \quad Ax + By + Cz + D = 0$$

$$z_{neu} = \frac{-D - A(x_{alt} + 1) - By}{C} = z_{alt} - \frac{A}{C}$$

## Was sind die Vorteile des z-Buffer Algorithmus?

- Hardware unterstützt
- Polygone können in beliebiger Reihenfolge gezeichnet werden
- Berechenzeit  $O(n)$ , häufig konstant ab einer gewissen Anzahl Polygone

### Was sind die Nachteile des z-Buffer Algorithmus?

- Rundungsprobleme
- Gleiche z-Werte problematisch
- grosser Speicherbedarf

### Was ist die Idee des Warnock Algorithmus?

- rekursive Unterteilung des Bildbereichs
- falls ein Bereich "einfach" zu zeichnen ist, wird er gezeichnet
- sonst wird er in vier Bereiche geteilt.

### Was sind die Kriterien des Warnock Algorithmus für Einfachheit?

1. der Bereich kein Polygon enthält
2. der Bereich nur ein Polygon enthält
3. der Bereich ein Polygon enthält, das am nächsten liegt und den Bereich vollständig ausfüllt
4. der Bereich nur aus einem Pixel besteht

### Was ist der Unterschied zwischen Objektraum und Bildraum?

- Objektraum Algorithmus
  - iteriert über die zu zeichnenden Objekte und zeichnet diese
- Bildraum Algorithmus
  - iteriert über die zu zeichnenden Pixel und findet die relevanten Objekte

## Clipping

### Was ist die Definition von 2D Clipping?

Die zu zeichnenden Grafikobjekte werden typischerweise in einem Bereich spezifiziert der grösser als das dargestellte Fenster ist

### Was ist die Definition von 3D Clipping?

Objekte die sich ausserhalb des Viewing Bereichs befinden, müssen nicht mehr weiterverarbeitet werden:

### Was kann durch 3D Clipping verhindert werden?

- unnötige Rasterisierung
- Rechenproblemen bei sehr grossen z-Koordinaten

### Welche Clipping Verfahren gibt es?

- Scissoring:
  - es werden alle Pixel berechnet
  - aber nur diejenigen gezeichnet die im Fenster liegen
- Temporärer Buffer
  - das ganze Objekt wird in einen temporären Buffer gezeichnet
  - der Buffer wird kopiert

- Analytische Berechnung
  - derjenigen Teile, die im Innern des Fensters liegen

### Wie funktioniert das Clipping von Linien nach Cohen-Sutherland?

1. Die ganze Ebene wird in Bereiche unterteilt
2. Durch Vergleichen der Bereiche in denen die Endpunkte liegen wird entschieden ob die Linie einfach akzeptiert oder abgelehnt werden kann
3. Ist dies nicht der Fall wird die Linie in zwei Teile geteilt, wovon eines abgelehnt wird und mit dem anderen wird zu Schritt 2 zurückgekehrt

### Wie funktioniert die Bereichsunterteilung beim Clipping?

1001   1000   1010	oberhalb (Bit 1)
-----+-----+-----	unterhalb (Bit 2)
0001   0000   0010	rechts (Bit 3)
-----+-----+-----	links (Bit 4)
0101   0100   0110	

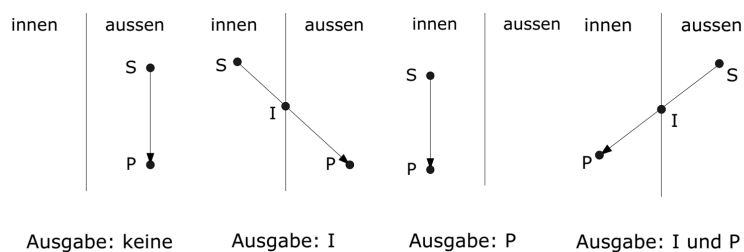
### Was ist der Vorteil des Algorithmus von Sutherland-Hodgeman?

Parallele Implementation möglich

### Wie funktioniert der Sutherland-Hodgman Algorithmus für Polygone?

- Betrachtet wird jeweils der aktuelle Punkt S, sowie der nächste Punkt P
- 4 Konfigurationen von S und P sind möglich
- Je nach Konfiguration werden 0-2 Eckpunkte des neuen Polygons ausgegeben

### Was ist die vier möglichen Ausgabe des Sutherland-Hodgman-Algorithmus?



### Was liegt der Punkt I beim Sutherland-Hodgman Algorithmus?

Schnittpunkt der Linie zwischen S und P mit der Teilungsgerade zwischen aussen und innen

## Beleuchtung und Schattierung

### Welche Materialeigenschaften haben Einfluss auf die Farbe eines Objektes?

- Struktur / Textur / Oberfläche
- Spiegelung / Lichtbrechung
- Glanz / Matt
- Transparenz
- Farbe



## Wie funktioniert das Standardbeleuchtungsmodell?

- üblicherweise wird ein vereinfachtes Beleuchtungsmodell verwendet
- da ein physikalisch korrektes Beleuchtungsmodell relativ aufwendig zu berechnen ist
- Das Modell simuliert diffuse und spiegelnde Reflektion
- Raytracing und Radiosity werden für aufwendigere Beleuchtungsmodelle verwendet

## Wie funktioniert eine diffuse Reflektion (Lambert Modell)?

- gleichmässige Abstrahlung des Lichts in alle Richtungen
- Eigenschaften eines matten, nicht glänzenden Materials

## Wie verändert sich die Energie auf eine Fläche, abhängig vom Winkel?

die Energie ist proportional zum Cosinus zwischen Lichtrichtung und Flächennormalen

## Wie wird die diffuse Reflektion berechnet?

$$I_d = I_L * k_d * \cos\phi \text{ mit } \cos\phi = \vec{N} * \vec{L}$$

- $N$  die Flächennormale
- $L$  die Richtung zur Lichtquelle
- $I_d$  die reflektierte Intensität
- $I_L$  die Intensität der Lichtquelle
- $k_d$  eine Reflexionskonstante ( $\Rightarrow$  siehe ein paar Fragen weiter)

## Wie funktioniert eine spiegelnde Reflektion (Phong Modell)?

- Simulation der Spiegelung auf glänzenden Oberflächen, wie Plastik, Metall oder lackiertes Holz
- die Intensität des Lichts nimmt mit  $\cos^n\phi$  ab,  
wobei  $\phi$  der Winkel zwischen der idealen Reflektionsrichtung und der betrachteten Richtung ist

## Wie wird die spiegelnde Reflektion berechnet?

$$I_s = I_L * k_s * \cos^n\phi$$

- $n$  steuert wie scharf die Abbildung ist (je grösser desto schärfer)
- $k_s$  eine Reflexionskonstante ( $\Rightarrow$  siehe ein paar Fragen weiter)

## Wie setzt sich ein Spiegelungsmodell meistens zusammen?

- einen spiegelnde (Lambert) Teil - Bsp. 30%
- einen diffusen (Phong) Teil - Bsp. 50%

## Wie setzt sich die Farbe der Reflexion zusammen?

- die Reflektionskonstanten  $k_d$  und  $k_s$  hängen von der Wellenlänge ab
- im einfachen Modell werden sie durch Konstanten für die Farben rot, grün und blau bestimmt

## Wie könnten die Reflexionskonstanten für ein rote Kugel und ein weisses Licht sein?

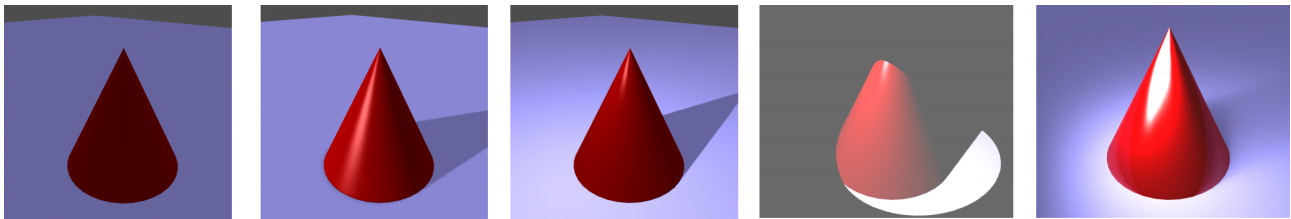
- $k_d = (0.7, 0.0, 0.0)$
- $k_s = (0.3, 0.3, 0.3)$
- $L = (1.0, 1.0, 1.0)$

## Wie wird die Abschwächung des Lichts gehandhabt?

- die Energie des Lichts nimmt mit dem Quadrat des Abstands ab, dadurch werden die Objekte allerdings schnell zu dunkel
- häufig findet deshalb das folgende, flexiblere Modell Verwendung

$$f_{Att} = \frac{1}{c_1 + c_2d + c_3d^2}$$

## Welche Arten von Lichtquellen gibt es?



Ambiente  
Lichtquelle

Direktionale  
Lichtquelle

Punkt  
Lichtquelle

Spot  
Lichtquelle

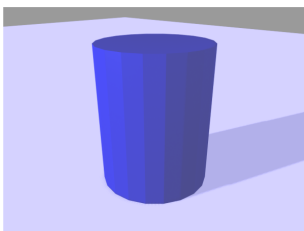
Verteilte  
Lichtquelle

## Was ist eine Schattierung?

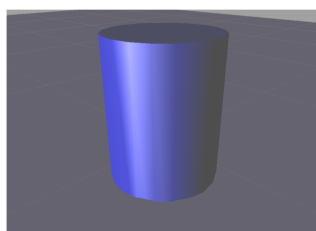
bestimmt an welchen Orten die Beleuchtung berechnet wird

## Welche Arten Schattierung unterscheidet man?

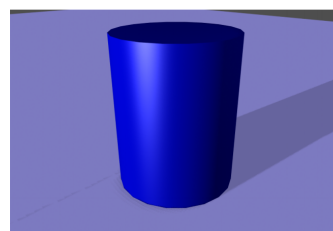
- Konstante Schattierung
  - pro Polygon wird nur eine Farbe berechnet
  - Eignet sich nicht für gekrümmte Objekte
- Gouraud Schattierung
  - Berechnung der Farbe an jedem Eckpunkt des Polygons
  - Lineare Interpolation der Farbe im Innern des Polygons
- Phong Schattierung
  - Interpolation des Normalenvektors im Innern des Polygons
  - Beleuchtungsberechnung für jeden Pixel



Konstante Schattierung



Gouraud Schattierung



Phong Schattierung

## Curves

### Welche Darstellungsmöglichkeiten gibt es für eine Kurve in der Ebene?

- Explizite Darstellung:

- $\gamma : [a, b] \rightarrow \mathbb{R}, x \mapsto y = f(x)$
- Implizite Darstellung:
  - $F(x, y) = 0$
- Parameterdarstellung
  - $\gamma : [a, b] \rightarrow \mathbb{R}^2, t \mapsto X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$

### Wie wird ein Kreis auf einer Ebene dargestellt?

- Explizite Darstellung:
  - $\gamma : [-r, r] \rightarrow \mathbb{R}, x \mapsto y = \sqrt{r^2 - x^2}$  (*oberer Halbkreis*)
  - $\gamma : [-r, r] \rightarrow \mathbb{R}, x \mapsto y = -\sqrt{r^2 - x^2}$  (*unterer Halbkreis*)
- Implizite Darstellung:
  - $x^2 + y^2 - r^2 = 0$
- Parameterdarstellung
  - $\gamma : [0, 2\pi] \rightarrow \mathbb{R}^2, t \mapsto X(t) = \begin{bmatrix} r \cos t \\ r \sin t \end{bmatrix}$

### Wie wird eine Kurve im Raum beschrieben?

$$\gamma : [a, b] \rightarrow \mathbb{R}^3, t \mapsto X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

### In welche Gruppen werden Splines unterteilt?

- interpolierende Splines
  - kubische Splines
- approximierende Splines (B-Splines)
  - nicht rationale B-Splines
    - uniforme B-Splines
    - nicht uniforme B-Splines (klassische B-Splines)
  - rationale B-Splines
    - uniforme B-Splines
    - nicht uniforme B-Splines (NURBS)

### Was sind die Eigenschaften der beiden Spline-Arten?

- interpolierende Splines
  - spezifiziert werden Punkte, durch welche die Kurve gehen soll
  - gehen exakt durch vorgegebene Punkt
  - Verhalten ist nur schwer vorauszusagen (Überschiessen, Oszillationen)
  - Schwierig, schöne glatte Kurven zu erhalten
  - am Rand wird es immer ungenauer
- approximierende Splines (B-Splines)
  - gehen nicht direkt durch alle Punkte

- einige Punkte sind Kontrollpunkte
- Kontrollpunkte beeinflussen die Form der Kurve
- wird häufig in der Praxis verwendet

### Wieso werden Kurve gebraucht?

- Begrenzung einer Fonts mit Hilfe von Bézier-Kurven
- Schwerpunkte von Objekten bewegen sich entlang von Kurven
- viele Objekte (wie Strassen, Räder, etc.) werden durch Kurven begrenzt
- Rotationsflächen werden durch eine Kurve erzeugt
- allgemeine Flächen werden durch zwei Kurvenscharen erzeugt

### Wie funktioniert die Methode der unbestimmten Koeffizienten?

Gesucht:  $P_3(x) = c_0 + c_1x + c_2x^2 + c_3x^3$

Punkte:  $(0, 1)$ ,  $(1, 1)$ ,  $(2, 0)$  und  $(3, 1)$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow P_3(x) = 1 + \frac{3}{2}x - 2x^2 + \frac{1}{2}x^3$$

### Wie funktioniert die Methode nach Lagrange?

Gesucht ist ein Polynom 3. Grades durch die Punkte:

$$(x_0, f(x_0)) = (0, 1), (x_1, f(x_1)) = (1, 1), (x_2, f(x_2)) = (2, 0), (x_3, f(x_3)) = (3, 1)$$

$$L_0(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} \quad L_1(x) = \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)}$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} \quad L_3(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}$$

$$P_3(x) = L_0(x)f(x_0) + L_1(x)f(x_1) + L_2(x)f(x_2) + L_3(x)f(x_3)$$

$$= L_0(x) * 1 + L_1(x) * 1 + L_2(x) * 0 + L_3(x) * 1$$

$$= L_0(x) + L_1(x) + L_3(x)$$

### Welche Arten von Bézier Splines gibt es?

- Linear Interpolation (linear Bézier spline)
  - $P(t) = (1 - t)P_0 + tP_1$
- Quadratic Bézier spline
  - $P(t) = (1 - t)^2P_0 + 2(1 - t)tP_1 + t^2P_2$
- Cubic Bézier spline
  - $P(t) = (1 - t)^3P_0 + 3(1 - t)^2tP_1 + 3(1 - t)t^2P_2 + t^3P_3$

### Wie sind die Bernsteinpolynome definiert?

$$B_i^n(t) = \binom{n}{i}(1 - t)^{n-i}t^i \text{ für } (0 \leq i \leq n)$$

**Was sind die Eigenschaften von Bézier-Kurven?**

- Beispiel für eine Kurve mit zwei Kontrollpunkten:
  - beginnt bei  $P_0$  und verläuft dort in Richtung  $P_1$
  - endet in  $P_3$  und ist dort tangential an  $P_3 - P_2$
- Kurve liegt immer innerhalb der konvexen Hülle des Linienzuges
- Anzahl der Kontrollpunkte bestimmt den Grad  $n$  der Kurve
- Änderung eines Kontrollpunktes bewirkt eine Änderung der gesamten Kurve

**Was sind die Eigenschaften von B-Spline-Kurven?**

- verallgemeinerung von Bézier-Kurven
- basieren auf:
  - rekursiv definierten rationalen
  - oder nicht rationalen B-Spline-Funktionen
  - oder Basis-Funktionen
- Änderung der Kontrollpunkte haben nur lokalen Einfluss
- Grad ist abhängig von der Anzahl Kontrollpunkte
- falls Parameterwerte äquidistant sind hat man uniforme, sonst nicht uniforme B-Spline-Kurven

**Wie werden B-Spline-Kurven berechnet?**

$$B_{n,k}(t) = \sum_{j=0}^n N_{j,k}(t)P_j, \quad (0 \leq t \leq n - k + 2) \text{ wobei } 2 \leq k \leq n + 1$$

$$N_{j,1}(t) = \begin{cases} 1 & \text{für } t_j \leq t \leq t_{j+1} \\ 0 & \text{sonst} \end{cases}$$

$$N_{j,m}(t) = \frac{t - t_j}{t_{j+m-1} - t_j} N_{j,m-1}(t) + \frac{t_{j+m} - t}{t_{j+m} - t_{j+1}} N_{j+1,m-1}(t), \quad m = 2, 3, \dots, k$$

$$t_j = \begin{cases} 0, & \text{für } j < k \\ j - k + 1, & \text{für } k \leq j \leq n, \quad 0 \leq j \leq n + k \\ n - k + 2, & \text{für } j > n \end{cases}$$

$$t = (t_0, t_1, \dots, t_{n+k}), \quad j = i, \dots, i + k, \quad \text{für } i = 0, \dots, n$$

**Wie werden rationale B-Splines-Kurven (NURBS) berechnet?**

$$P_j = (x_j, y_j, z_j) \equiv C_j = (x_j w_j, y_j w_j, z_j w_j, w_j)$$

$$P_H(t) = \sum_{j=0}^n N_{j,k}(t) C_j$$

$$P(t) = \frac{\sum_{j=0}^n N_{j,k}(t) w_j P_j}{\sum_{j=0}^n N_{j,k}(t) w_j}$$

**Was sind NURBS-Kurven?**

- Non Uniform Rational Basic Splines
- allgemeinste Form von Freiformkurven
- erweitern die Funktionalität der Bézier- und B-Spline-Kurven
  - durch zusätzliche Gewichte  $w_i$
  - die homogene Koordinate, für die Koordinaten der Kontrollpunkte

- Bézier- und B-Spline-Kurven sind Spezialfälle von NURBS-Kurven
- NURBS sind invariant bei den Transformationen
  - Skalierung, Translation, Rotation, Scherung und den Parallel- und Perspektivprojektionen

## Flächen

### Wie wird eine Ellipsoid in 3D beschreiben?

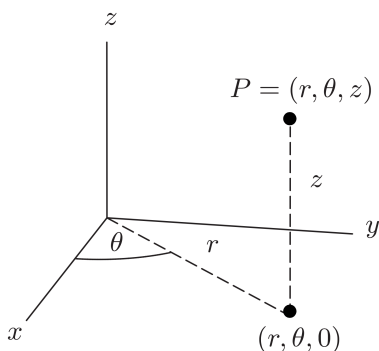
Gegeben: Halbmessern  $a, b, c$  und Zentrum in  $(x_o, y_o, z_o)$

$$\left(\frac{x - x_o}{a}\right)^2 + \left(\frac{y - y_o}{b}\right)^2 + \left(\frac{z - z_o}{c}\right)^2 = 1$$

### Wie funktioniert die Umrechnung in Zylinderkoordinaten?

Zylinderkoordinaten:  $(r, \theta, z) \in [0, \infty) \times [0, 2\pi) \times (-\infty, \infty)$

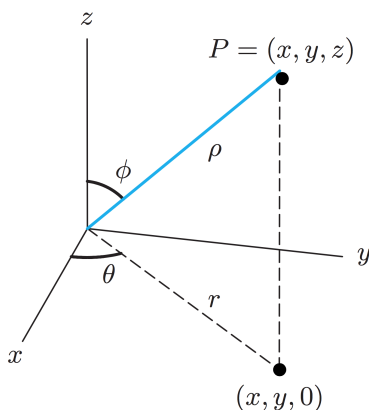
$$\begin{aligned} x &= r \cos \theta & r &= \sqrt{x^2 + y^2} \\ y &= r \sin \theta & \iff \tan \theta &= \frac{y}{x} & \theta &= \arctan\left(\frac{y}{x}\right) \pm 180 \\ z &= z & z &= z \end{aligned}$$



### Wie funktionieren Kugel- oder sphärische Koordinaten?

Kugelkoordinaten:  $(\rho, \phi, \theta) \in [0, \infty) \times [0, 2\pi) \times [0, 2\pi)$

$$\begin{aligned} x &= \rho \sin \phi \cos \theta & \rho &= \sqrt{x^2 + y^2 + z^2} \\ y &= \rho \sin \phi \sin \theta & \iff \tan \theta &= \frac{y}{x} \\ z &= \rho \cos \phi & \cos \phi &= \frac{z}{\rho} \end{aligned}$$



**Wie wird eine Rotationsflächen um die  $x$ -Achse beschrieben?**

$$x = u, \quad y = f(x) \cos v, \quad z = f(x) \sin v$$

wobei  $a \leq u \leq b$  und  $0 \leq v \leq 2\pi$

**Wie werden die Vektoren des Dreiein einer Rotationsflächen berechnen?**

$$t(u) = \frac{\dot{x}(u)}{|\dot{x}(u)|} \text{ Tangentenvektor}$$

$$n(u) = \frac{\dot{t}(u)}{|\dot{t}(u)|} \text{ Hauptnormalenvektor}$$

$$b(u) = t(u) \times n(u) \text{ Binormalenvektor}$$

**Was ist eine Regelfläche?**

Eine Fläche  $S$  ist eine Regelfläche falls durch jeden ihrer Punkte eine Gerade existiert, die vollständig in  $S$  liegt

**Was ist eine doppelte Regelfläche?**

Eine Fläche  $S$  ist eine doppelte Regelfläche falls durch jeden ihrer Punkte zwei unterschiedliche Geraden existieren, die vollständig innerhalb von  $S$  liegen

**Wie wird eine Freiformflächen beschrieben?**

- polynomiale Kurve  $F(u)$  vom Grad  $n$ 
  - $F(u) = \sum_{i=0}^n C_i N_i(u), \quad 0 \leq u \leq 1$
- polynomiale Kurve  $G(v)$  vom Grad  $m$ 
  - $G(v) = \sum_{j=0}^m C_j N_j(v), \quad 0 \leq v \leq 1$
- Tensorproduktfläche  $S(u, v)$ 
  - $S(u, v) = \sum_{i=0}^n \sum_{j=0}^m C_i C_j N_i(u) N_j(v), \quad 0 \leq u, v \leq 1$

**Wie werden Bézierflächen berechnet?**

$$Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(v) B_j^m(u) P_{i,j}$$

$$B_i^n(v) = \binom{n}{i} (1-v)^{n-i} v^i, \quad 0 \leq i \leq n$$

$$B_j^m(u) = \binom{m}{j} (1-u)^{m-j} u^j, \quad 0 \leq j \leq m$$

**Wie wird die Flächennormale einer Bézier-Fläche berechnet?**

$$\frac{\partial P(u, v)}{\partial u} = [0, 1, 2u, 3u^2] M G M^T V^T$$

$$\frac{\partial P(u, v)}{\partial v} = U M G M^T [0, v, 2v, 3v^2]^T$$

$$N(u, v) = \frac{\partial P(u, v)}{\partial u} \times \frac{\partial P(u, v)}{\partial v}$$

## Raytracing

### Für was wird Raytracing gebraucht?

- Erzeugung von photorealistischen Bildern
- Berücksichtigung von Spiegelung und Transparenz
- Pro Pixel wird ein Strahl gelegt und das Objekt ermittelt, das zuerst geschnitten wird
- Weitere Strahlen werden verwendet um zusätzliche Effekte zu erzielen
  - Spiegelung, Transparenz, Schatten, Umgebungslichtes

### Wie funktioniert ein einfacher Raytracing Algorithmus?

```

for alle Pixel do
  Ermittle Strahl durch Pixel
  for alle Objekte do
    Berechne Schnittpunkte mit Strahl
    if Schnittpunkt am nächsten then
      Speichere Schnittpunktinformation
    end
  end
  if Schnittpunkt gefunden then
    Berechne Farbe
  end
end
end

```

## Rendering Equation

### Was bedeutet die Abkürzung BRDF?

Bidirectional Reflectance Distribution Function

### Wie ist die Rendering Equation definiert?

$$L_O(x, \vec{w}_r) = L_e(x, \vec{w}_r) + L_r(x, \vec{w}_r)$$

### Was ist das Ziel von Radiosity?

Berücksichtigung der Farbe des reflektierten Lichtes eines Objektes

### Radiosity Konzept

- Oberflächen der Objekte werden in Patches unterteilt
- Ein Patch ist ein Polygon mit konstanter Licht Intensität
- Die Abhängigkeiten der Patches werden durch ein System von linearen Gleichungen modelliert
- Durch Lösung der Gleichungen wird die Intensität und Farbe pro Patch berechnet

### Radiosity Gleichung

$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^n B_j A_j F_{ij}$$

$B_i$  Radiosity  $A_i$  Flächeninhalt des Patch  $i$   $E_i$  Emission des Patch  $i$   $\rho$  Reflektion  $F_{ij}$  Formfaktor



**Welche Möglichkeiten gibt es ein Gleichungssystem zu lösen?**

- Jacobi Iteration
- Gauss Seidel Relaxation
- Southwell Iteration

**Was ist das Problem bei der Berechnung der Formfaktoren?**

- Analytische Berechnung des Integrals geht nur in Ausnahmefällen
- Sichtbarkeit muss berechnet werden
- Lösung durch Approximationen und geometrische Interpretation

**Welche Möglichkeiten gibt es für die Berechnung der Radiosity?**

- Progressive Radiosity:
  - Berechnung und frühe Darstellung einer approximativen Lösung
- Hierarchical Radiosity:
  - Erhöhung der Anzahl Patches wo es notwendig ist