Computergrafik

am Hochschulinformationstag

– eine Auslese –

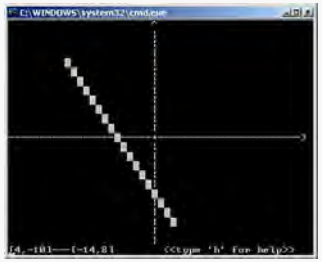
Prof. Dr. Aris Christidis

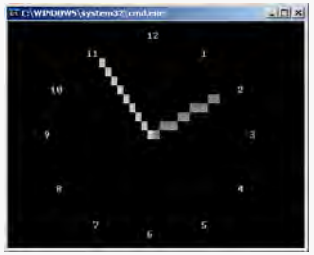
http://homepages.thm.de/christ/

Allgemeines

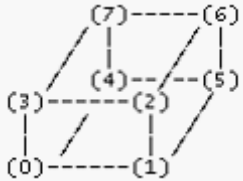
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Inhalt der BSc-Vorlesung:



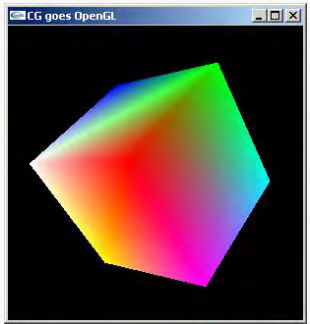


- Vom Pixel zur Linie
- Von der Linie in die Ebene: Transformationen
- Von der (Linien-)Kante zum 3D-Objekt



- Vom 3D-Objekt zum projizierten Drahtmodell

- Vom Drahtmodell zum Flächenmodell

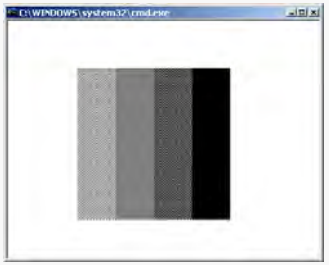


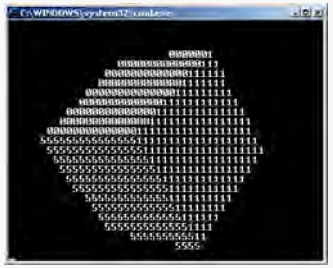
- Design, Notation und Einsatz von OpenGL

Allgemeines

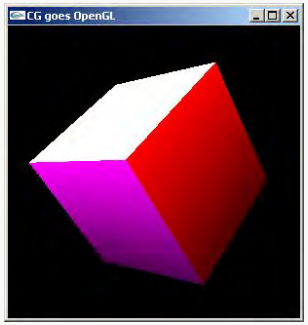
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

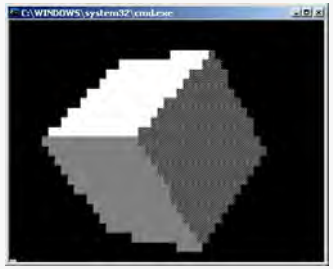
Inhalt der MSc-Vorlesung:

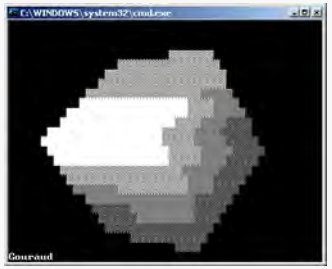


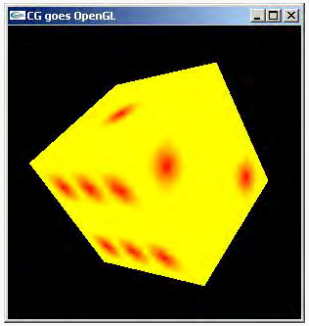


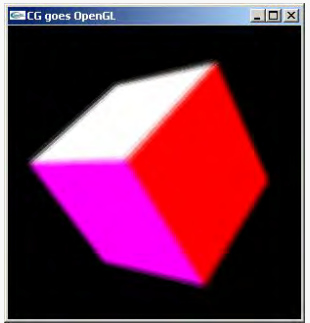
- Kurzes Repetitorium
- Füllen / Färben v. Objektflächen
- Schattierung
  und Beleuchtung
- Texturierung
- Antialiasing











Allgemeines

1921: Erste Übertragung eines gerasterten Bildes N.York ⇨ London (Fernschreiber mit Typen-Aufsätzen)

Hierzu später (Rechner-Kontext): **das** (seltener: der) **Pixel** – aus „picture element“, eher: „pic cell“, dt.: Bildpunkt

Pixel-Grafiken als (Hilfs-)Mittel, zunehmend als Ergebnis techn.-wiss. Tätigkeit: Bild-Markierung ⇨ ... ⇨ Kartographie

Bild-Retusche ⇨ ... ⇨ Zeichentrick ⇨ ... ⇨ Simulation / VR

Allen Anwendungen gemeinsam: Abb. in Rastern (Matrizen)

Begriff aus TV-Technik: Scan Conversion („Abtastumwandlg“)

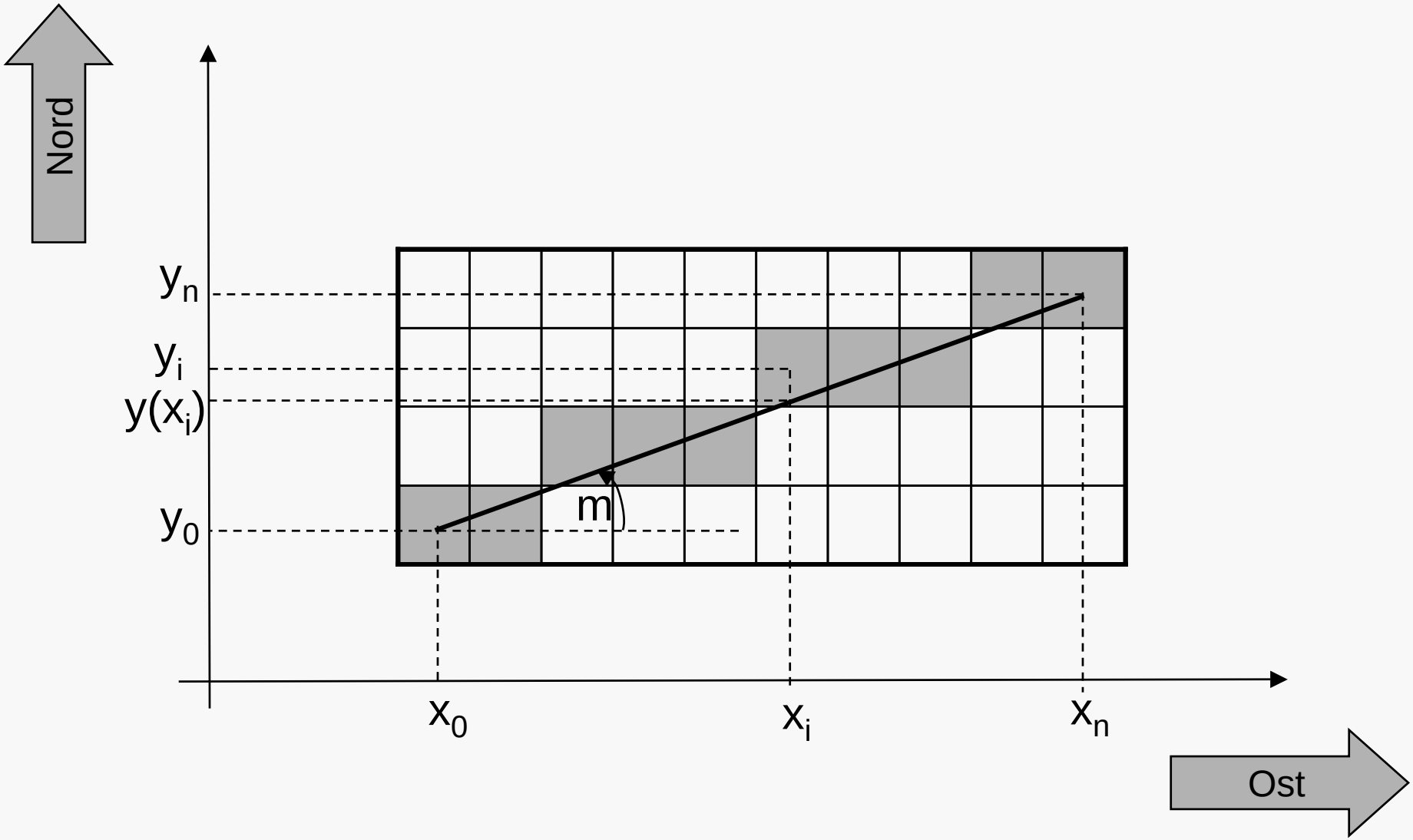
Aus der ersten CG-Klausur (SS 07, Aufgabe 1c):

Aus wievielen Pixeln besteht die Diagonale eines Quadrats mit einer Kantenlänge von 25 Pixeln?

Aus 25.

Gerasterte Linien

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis



Übung

Übung: Implementierung des Bresenham-Algorithmus für alle Oktanten mit ASCII-Zeichen als Bildpunkten:

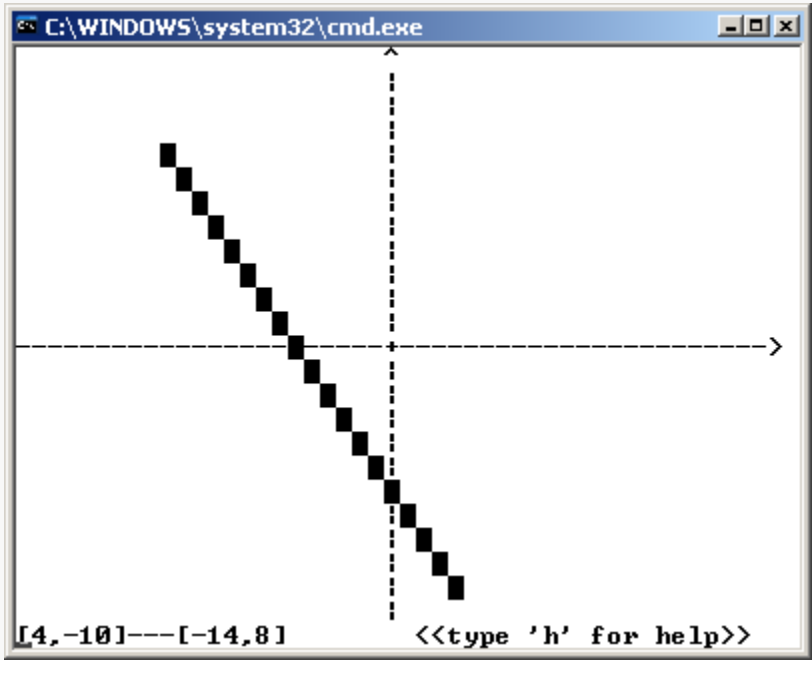
(i) Erweiterung des Algorithmus

(ii) Feststellung / Sicherstellung der Umkehrbarkeit

Zu allen Übungen:

- *.obj-Dateien größtenteils mitgeliefert
  (pro Fehler-Lokalisierung)
- Aufgaben mit großen Anteilen zur Wiederverwendung
  (contra Altlast-Ansammlung)

`BresenLine.exe`



2D-Punkt-Transformationen

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Zur Erinnerung – Drehung eines beliebigen Punktes B‘ um den Winkel θ um den Koordinaten-Ursprung zum Punkt B“:

$x_{B'} = r \cdot \cos\alpha$

$y_{B'} = r \cdot \sin\alpha$ [r, α: Hilfsgrößen]

$x_{B''} = r \cdot \cos(\alpha+\theta)$

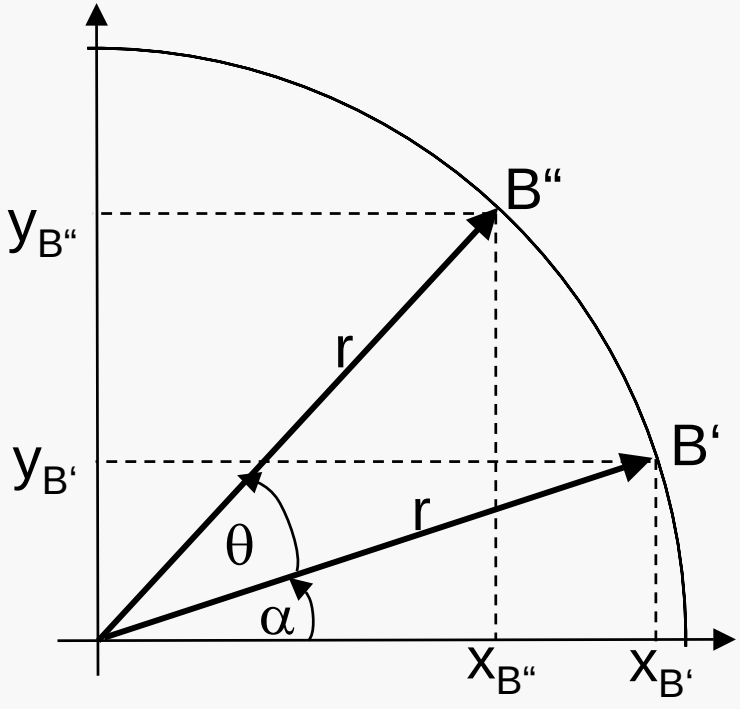
$= r \cdot (\cos\alpha \cos\theta - \sin\alpha \sin\theta)$

$= x_{B'} \cdot \cos\theta - y_{B'} \cdot \sin\theta$

$y_{B''} = r \cdot \sin(\alpha+\theta)$

$= r \cdot (\sin\alpha \cos\theta + \cos\alpha \sin\theta)$

$= x_{B'} \cdot \sin\theta + y_{B'} \cdot \cos\theta$



In Matrizen-Schreibweise:

$$\begin{pmatrix} x_{B''} \\ y_{B''} \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{B'} \\ y_{B'} \end{pmatrix}$$

$\sin(\alpha+\beta) = \sin\alpha \cos\beta + \cos\alpha \sin\beta$

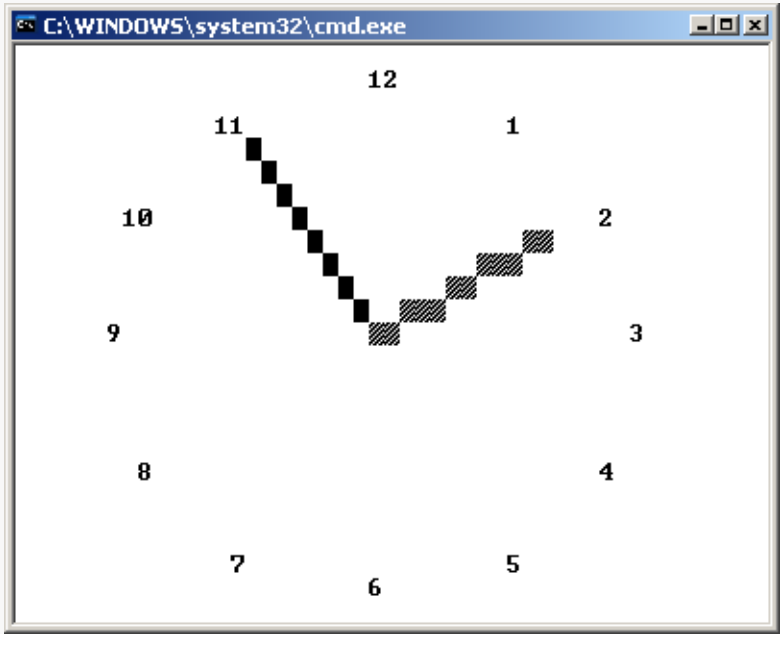
Übung

Übung: Erstellung einer Analog-Uhr aus ASCII-Zeichen der Größe 8x12:

(i) Zeichnung und Positionierung der Zeiger als Linien;

(ii) Realitätsnahe Animation.

Durch entsprechende Skalierung ist die Uhr auf eine kreisrunde Form zu bringen.



`BresenClock.exe`

3D-Punkte, -Vektoren, -Transformationen

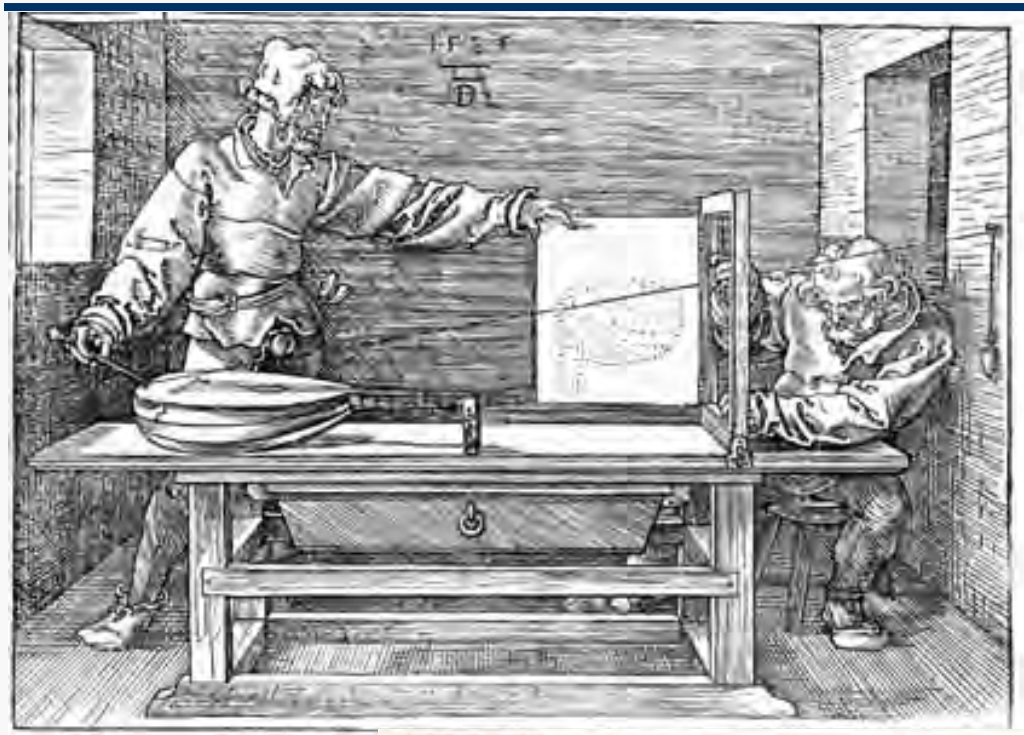
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Rotation um Winkel θ um belieb. Achse durch Koordinaten-Ursprung mit Richtungsvektor $[X, Y, Z, 0]^T$, $X^2+Y^2+Z^2=1$:

$$\begin{pmatrix} X^2+(1-X^2)\cdot\cos\theta & X\cdot Y\cdot(1-\cos\theta)-Z\cdot\sin\theta & X\cdot Z\cdot(1-\cos\theta)+Y\cdot\sin\theta & 0 \\ X\cdot Y\cdot(1-\cos\theta)+Z\cdot\sin\theta & Y^2+(1-Y^2)\cdot\cos\theta & Y\cdot Z\cdot(1-\cos\theta)-X\cdot\sin\theta & 0 \\ X\cdot Z\cdot(1-\cos\theta)-Y\cdot\sin\theta & Y\cdot Z\cdot(1-\cos\theta)+X\cdot\sin\theta & Z^2+(1-Z^2)\cdot\cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(Herleitung)

3D-Sicht, Projektionen



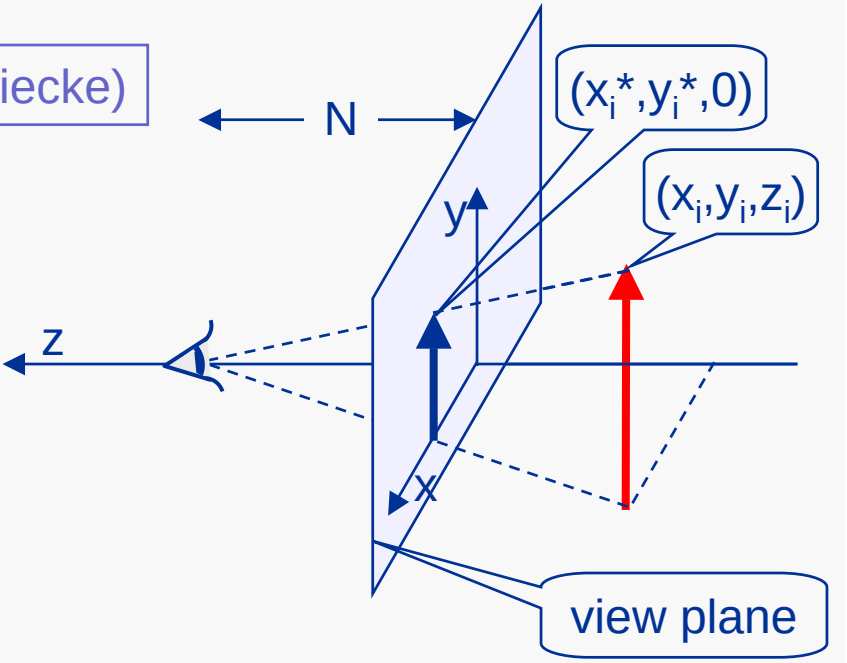
3D-Sicht, Projektionen

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Transformationen, deren Matrix als letzte Zeile nicht die Form: [0 ... 0 1] hat, gehören zur allgemeineren Klasse der **perspektivischen Transformationen**.

Perspektivische Projektion von Punkten (x_i,y_i,z_i) auf $(x_i^*,y_i^*,0)$ in der Projektionsebene z=0 mit Proj.zentrum („Augenpunkt“) bei z=N (N>0) in einem Rechts(koordinaten)system:

$x_i^*/x_i = y_i^*/y_i = N/(N-z_i)$ (ähnliche Dreiecke)



Versuch der Bildung eines Matrizenprodukts:

$$\begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & N \end{pmatrix} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = \begin{pmatrix} Nx_i \\ Ny_i \\ 0 \\ N-z_i \end{pmatrix} = \begin{pmatrix} (N-z_i)\cdot x_i^* \\ (N-z_i)\cdot y_i^* \\ (N-z_i)\cdot 0 \\ (N-z_i)\cdot 1 \end{pmatrix}$$

Grafik-Objekte

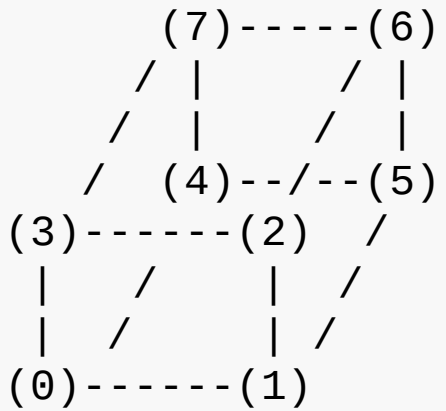
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Speicherung eines Grafik-Objekts (z.B. eines Würfels):

```
cube.cgf ; Objekt-Name - CGF Version 0.0
8 ; Anzahl Punkte
-1., -1.,  1. ; Koord. Pkt[0] etc.
 1., -1.,  1.
 1.,  1.,  1.
-1.,  1.,  1.
-1., -1., -1.
 1., -1., -1.
 1.,  1., -1.
-1.,  1., -1.
6 ; Anzahl Flaechen
4, 4, 4, 4, 4, 4 ; Pkte je Flaeche
0, 3, 7, 4 ; Pkt-Id ab Flaeche[0]
1, 0, 4, 5
2, 3, 0, 1
3, 2, 6, 7
4, 7, 6, 5
5, 6, 2, 1
```

Geometrie der Objektpunkte (ihre räumliche Lage)

Topologie der Objektpunkte (Beziehungen zwischen ihnen)

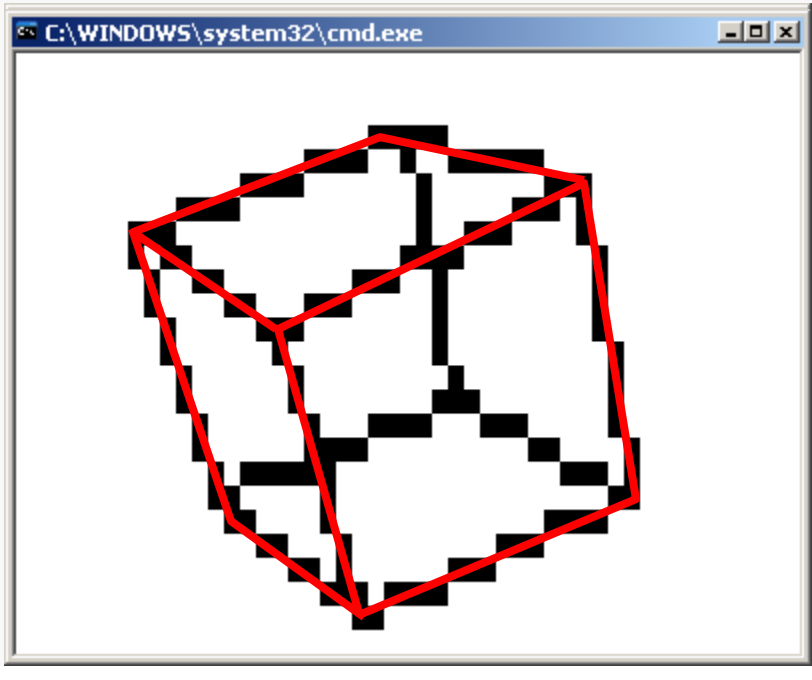


(Xface?)

Übung

Übung:

Im Konsole-Fenster sollen mit ASCII-Zeichen geladene 3D-Drahtmodelle (*wire frame*) dargestellt werden; sie sollen um die x-, y- und z-Achsen drehbar und bei veränderlichem Projektionszentrum darstellbar sein.



`WireCullFill(1).exe`

3D-Sicht, Projektionen

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

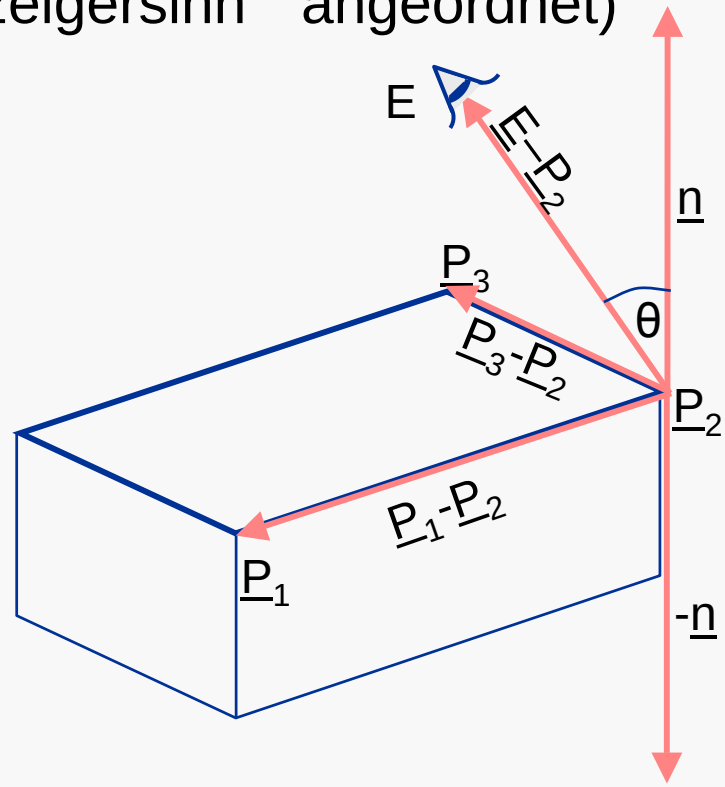
Orientierung einer Objektfläche mit den Eckpunkten $\underline{P}_1,\underline{P}_2,\underline{P}_3$ (bei Draufsicht: gegen den Uhrzeigersinn angeordnet) gegenüber dem Augenpunkt E:

Nach außen gerichtete Normale $\underline{n}$:

$$\underline{n} = (\underline{P}_3 - \underline{P}_2) \times (\underline{P}_1 - \underline{P}_2)$$

Winkel zwischen der Normalen und dem Verbindungsvektor vom Eckpunkt P_2 zum Augenpunkt E:

$$\cos\theta = \underline{n} \bullet (\underline{E} - \underline{P}_2) / (|n| \cdot |\underline{E} - \underline{P}_2|)$$



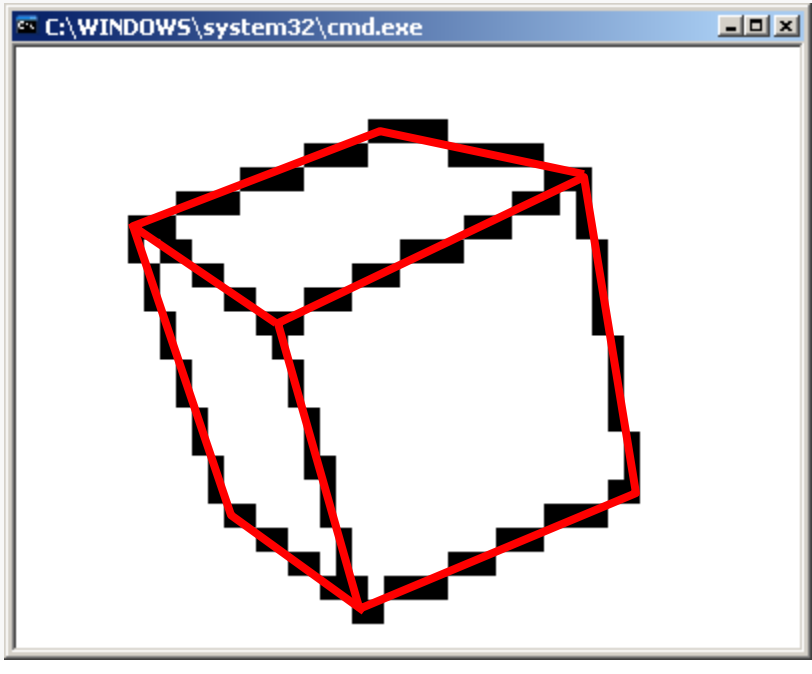
$\underline{n} \bullet (\underline{E} - \underline{P}_2) \geq 0 \Leftrightarrow -90° \leq \theta \leq 90° \Leftrightarrow$ sichtbare Fläche

$\underline{n} \bullet (\underline{E} - \underline{P}_2) < 0 \Leftrightarrow 90° < \theta < 270° \Leftrightarrow$ nicht sichtbare (Rück-)Fläche

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Übung (Forts.):

Erweiterung des Programms zur Konsole-Darstellung eines 3D-Drahtmodells um die wahlweise Ausblendung abgewandter Objektflächen (Flächenmodell – *solid model*).

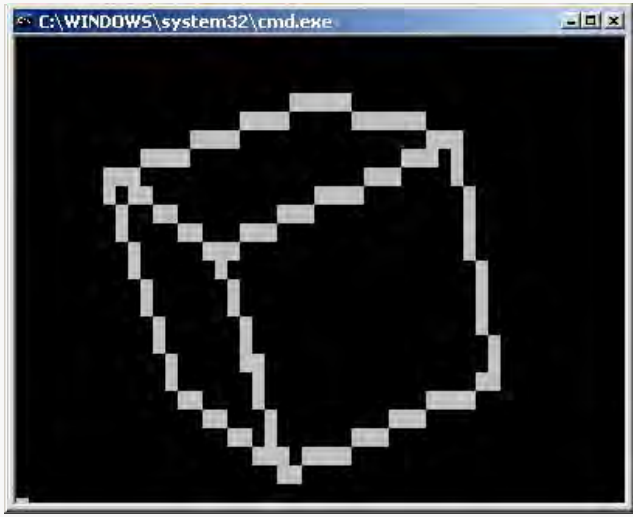


`WireCullFill(2).exe`

3D-Sicht, Projektionen

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

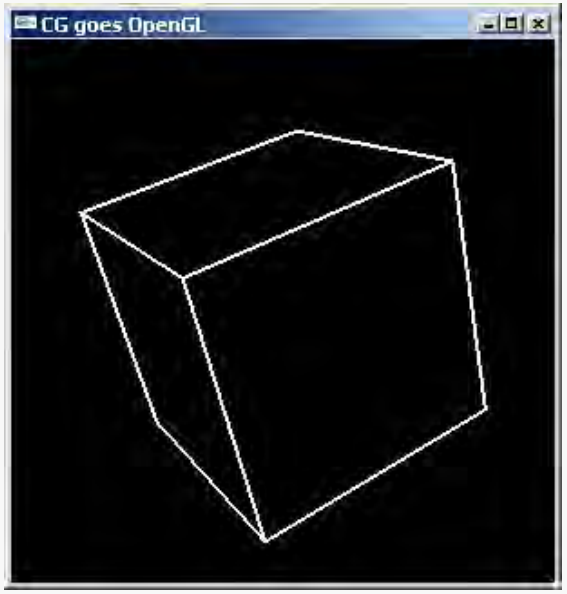
Typische Struktur eines Grafik-Programms, das erst in einem „verborgenen“ Speicher zeichnet, den es nach Fertigstellung der Grafik sichtbar macht (Double Buffering):




```
void draw (void)
{/*Hintergrund-Puffer loeschen:*/
  conClrBuff();
 /*Erstellung der Grafik (im Hintergrund):*/
  conDrawCGFobj(&obj,eyez);
 /*Hintergrund-Puffer sichtbar machen:*/
  conSwapBuff();
  return;
}
```

OpenGL

Typische Struktur eines Grafik-Programms, das nach aktuellem Industrie-Standard mit Double Buffering zeichnet:




```
void draw (void)
{/*Hintergrund-Puffer loeschen:*/
  glClear(GL_COLOR_BUFFER_BIT);
 /*Erstellung der Grafik (im Hintergrund):*/
  drawBox();
 /*Hintergrund-Puffer sichtbar machen:*/
  glutSwapBuffers();
  return;
}
```

CubeGL.exe

OpenGL-Pipeline

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

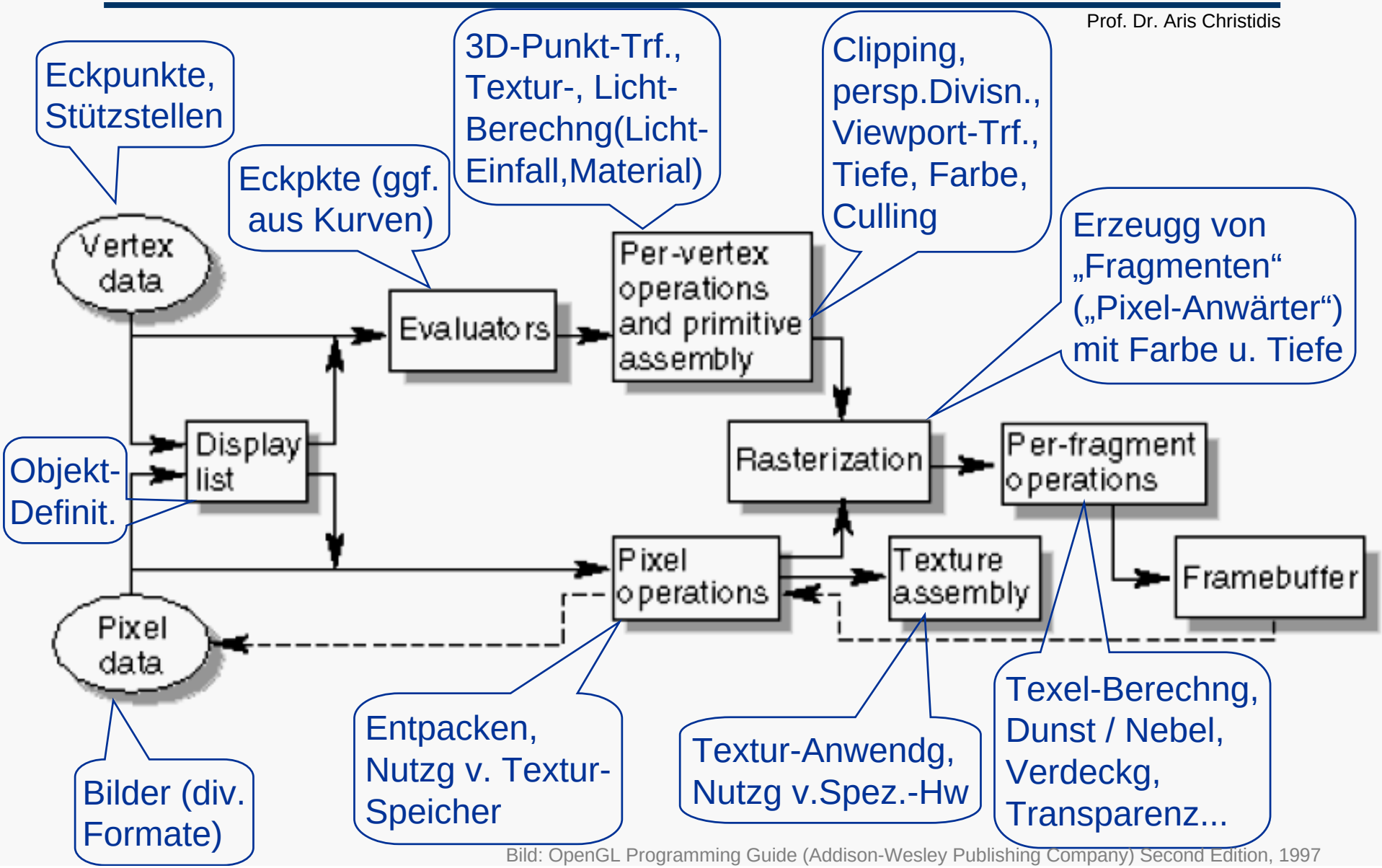
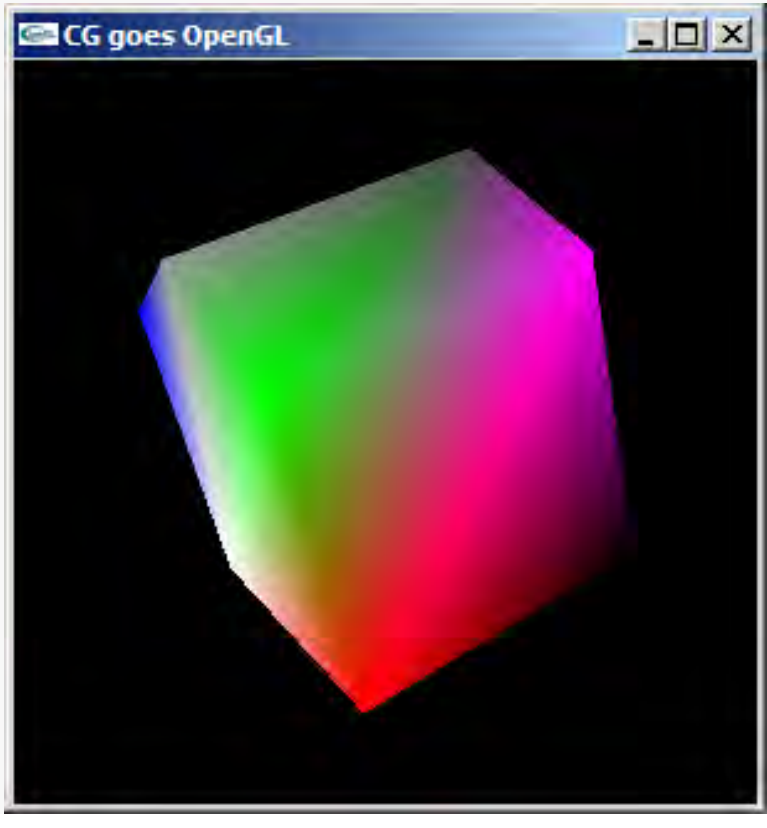


Bild: OpenGL Programming Guide (Addison-Wesley Publishing Company) Second Edition, 1997

Übung

Übung:

In einem GLUT-Fenster sollen mit OpenGL geladene 3D-Modelle dargestellt werden mit den Funktionalitäten, die im Bedienungsmenü vorgesehen sind.

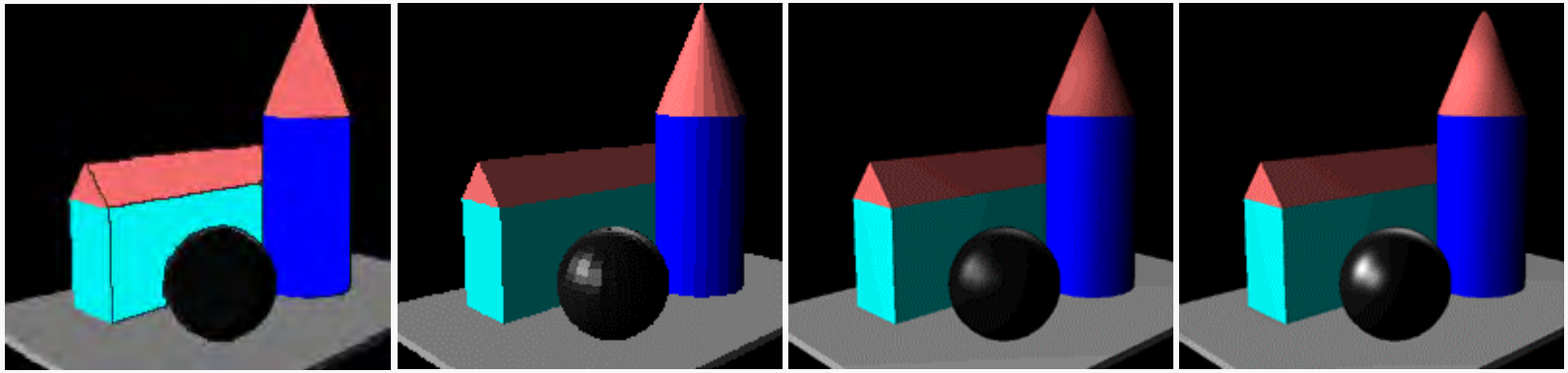


`ObjElabGL.exe`

Exkurs: Fotorealismus, Echtzeit, Arithmetik

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

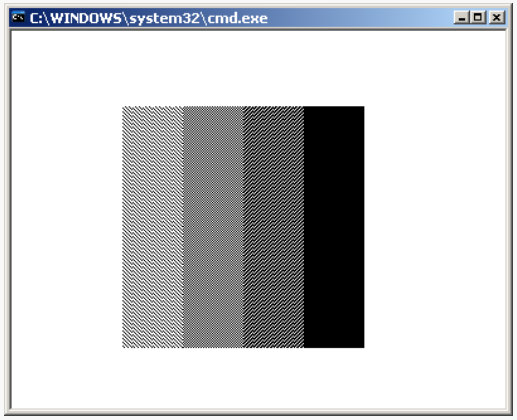
- Ziel vieler CG-Anwendungen: Fotorealismus in Echtzeit; wichtiges Mittel: stetige Helligkeits- / Farbübergänge – z.B.:

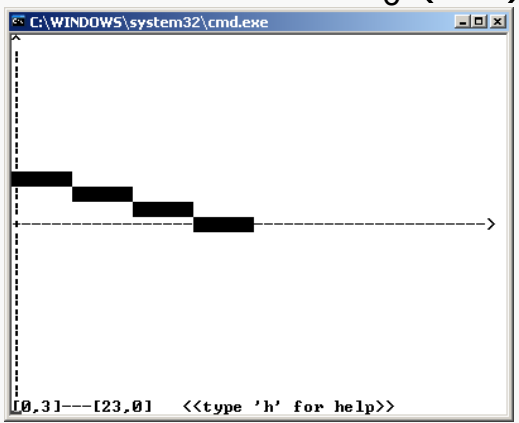


Bilder: www.glossar.de/glossar/z_shading.htm

Meist angewandte math. Methode: lineare Interpolation

Zuordnung d.Werte einer abhängigen Var $y \in \mathbf{N}_0$ (Pixel-Helligkeit,-Farbe) d.Werten einer unabhängigen Variablen $x \in \mathbf{N}_0$ (Ort).





Farbe u. Schattierung

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

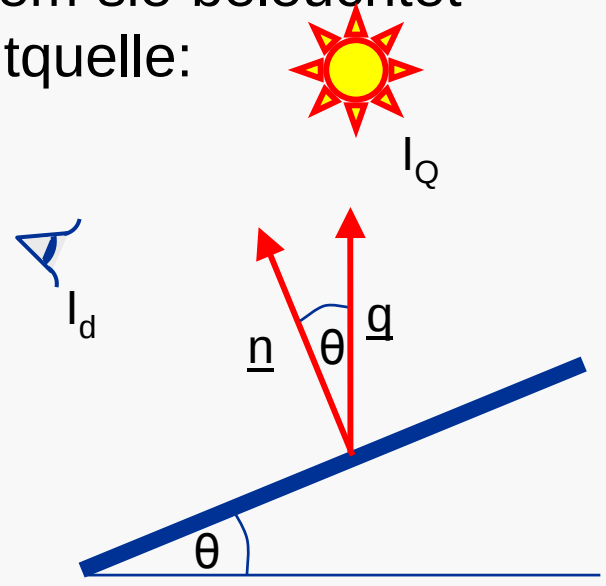
Lambertsches (Kosinus-) **Gesetz** für **Diffuse Reflexion**:
Eine matte weiße Fläche hat eine Helligkeit, die von der Richtung ihrer Beobachtung unabhängig ist. Die von ihr ausgehende Lichtstärke I_d (Lichtenergie pro Raumwinkel und Zeit) ist abhängig nur von dem gegen die Flächennormale gemessenen Winkel θ, unter dem sie beleuchtet wird, und von der Lichtstärke I_Q der Lichtquelle:

$$I_d = I_Q \cdot \rho_d \cdot \cos\theta$$
$$= I_Q \cdot \rho_d \cdot \underline{q} \cdot \underline{n}$$

mit: $|\underline{q}| = |\underline{n}| = 1$

ρ_d: diffuser Reflexionskoeffizient

$\rho_d = \rho_d(\lambda, \theta, \text{Oberfläche}) \approx \text{const.}$
$(0 < \rho_d < 1$; in CG meist: trial & error)



Unabhängigkeit vom Abstand Objekt-Lichtquelle –

Annahme: ambientes Licht (Reflexionen aus der Umgebung)

Farbe u. Schattierung

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Spekulare Reflexion (Spiegelungseffekt) am idealen Spiegel gemäß **Reflexionsgesetz**: Ausfallswinkel = Einfallswinkel

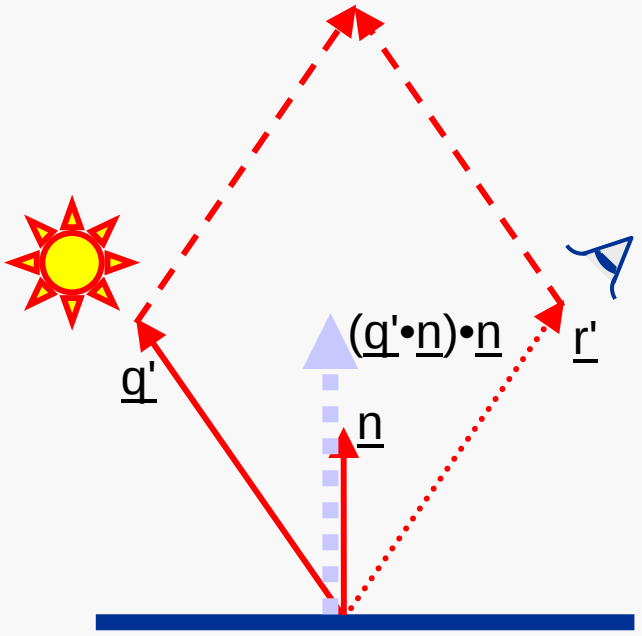
$$\underline{r}' + \underline{q}' = 2 \bullet (\underline{q}' \bullet \underline{n}) \bullet \underline{n}$$

worin:

$\underline{q}'$ Strahl zur Lichtquelle (bel. Vektor)
($\underline{q}' = I_Q \bullet \underline{q}$, $|\underline{q}| = 1$)

$\underline{r}'$ reflektierter Strahl ($|\underline{r}'| = |\underline{q}'|$)

$\underline{n}$ Normalenvektor der reflektierenden Fläche (normiert: $|\underline{n}| = 1$)



Berechnung des Reflexionsstrahls $\underline{r}'$:

$$\underline{r}' = 2 \bullet (\underline{q}' \bullet \underline{n}) \bullet \underline{n} - \underline{q}'$$

Farbe u. Schattierung

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

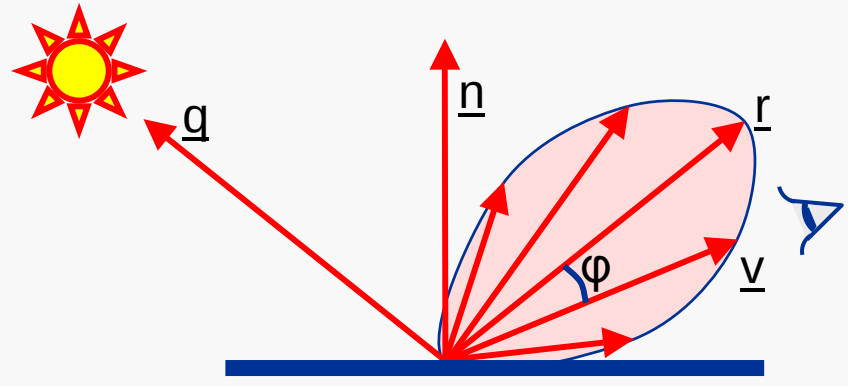
Infinitesimale Rauheit bewirkt Strahlenstreuung um R.-winkel

⇨ Lichtstrom komplex abhängig von Betrachtungswinkel φ.

Ansatz von Bui-Tuong Phong für Spekulare Reflexion („Phong-Shading“, 1975): Lichtstrom abhängig von experimentell ermittelter Potenz f von cos φ (idealer Spiegel: f → ∞):

$I_s = I_Q \bullet \rho_s \bullet (\cos \varphi)^f = I_Q \bullet \rho_s \bullet (\underline{r} \bullet \underline{v})^f$

Darin:



Phong-Shading auch nach weiterer math. Vereinfachung für Echtzeit problematisch!

$\underline{q}$	Strahl zur Lichtquelle ($\lvert\underline{q}\rvert=1$)	I_s	spekularer Anteil Lichtstärke
$\underline{r}$	reflektierter Strahl ($\lvert\underline{r}\rvert=1$)	I_Q	Lichtstärke der Lichtquelle
$\underline{v}$	Strahl zum Beobachter ($\lvert\underline{v}\rvert=1$)	ρ_s:	spekularer Reflexionskoeff.

Texturierung

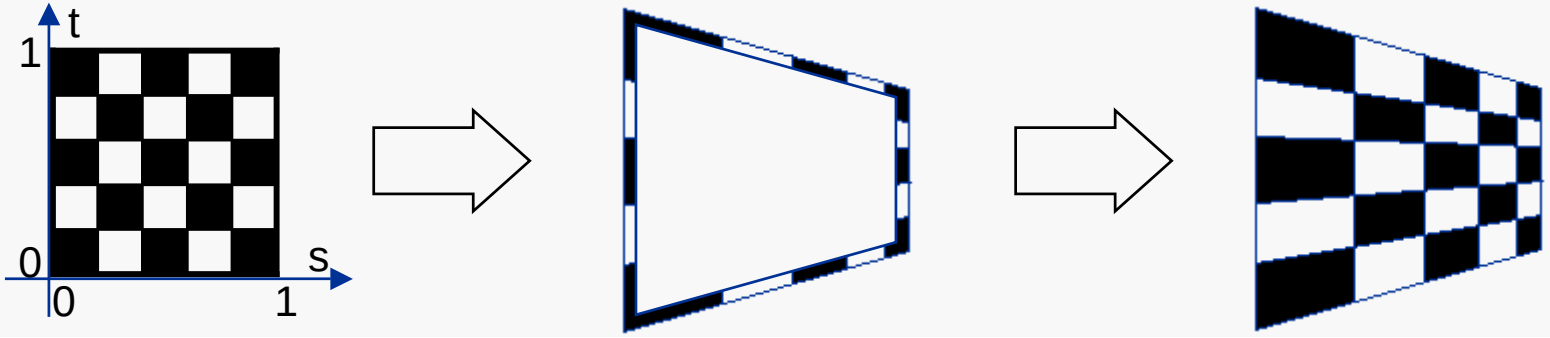
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Vorgehensweise vergleichbar zu Gouraud Shading:

1. Nach Trf. u. Projektion d. Eckpunkte: Zeichnen d. Objekt-kanten unter Verwendung von Texeln der Texturkanten
   (Gouraud:...unter Verwendung v.interpolierten Werten zw.Eckpunkten)
2. Flächenfüllen entlang allen Bildzeilen zwischen Pixeln gegenüberliegender Kanten; dazugehörige Texel werden durch Schnitte in die Textur ermittelt (Linienalgorithmus!).
   (Gouraud:...durch Interpolation zw.gegenüberliegenden Kantenwerten)

Besonders wichtiger / delikater Unterschied zu Gouraud:

- Berücksichtigung der (stark nichtlinearen) perspektiv.Trf.– sowohl beim Kantensetzen als auch beim Flächenfüllen!



Texturierung

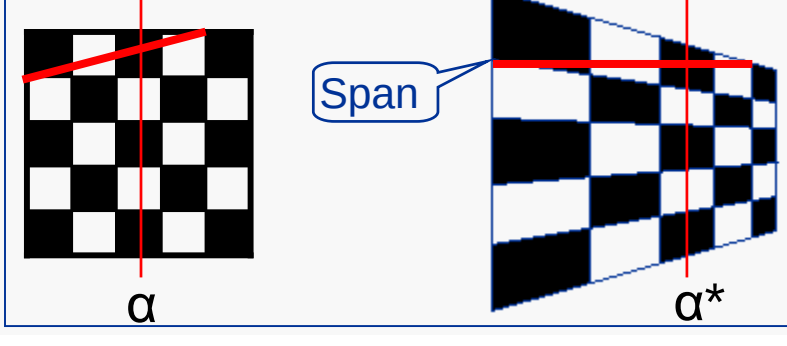
Prof. Dr. Aris Christidis

Schritte zur Texturierung einer perspektivisch dargestellten Fläche:

Zeichnen der Flächenkanten:

Für jedes Kantenpixel

- Ermittle Lage zw.projizierten Kantenenden ($0 \le \alpha^* \le 1$) •
- Ermittle zugehörige Raum-Tiefe (hom. Koord. w) •
- Ermittle Lage korrespondierenden Kantentexels($0 \le \alpha \le 1$) •
- Merke Kantentexel-Index u. w bei Bildzeilen-Ein-/Austritt
- Setze (Texel als) Pixel



Füllen der Fläche:

Für jede Bildzeile durch Fläche

- Führe Schnitt durch Textur zw. registrierten Kanten-texeln (Linienalgorithmus) u. merke getroffene Texel

Für jedes Span-Pixel

- Ermittle Lage des Pixels zwischen Span-Enden (α^*) •
- Ermittle (aus α^* und w) Lage korrespondierenden Texels in der Textur-Draufsicht (α) •
- Setze (Texel als) Pixel

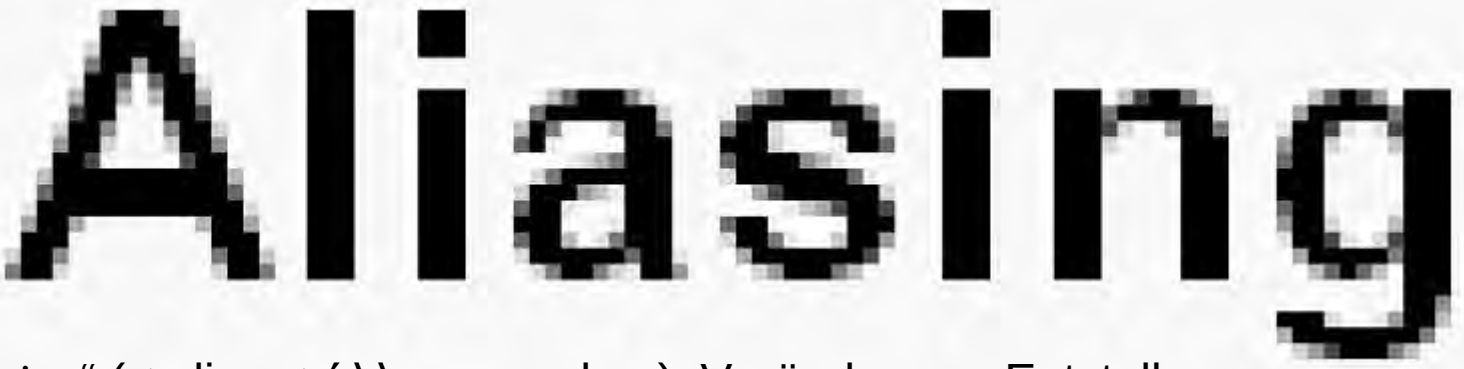
(•) zu behandelnde Aufgaben

Aliasing und Antialiasing

TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Wesen des Phänomens **Aliasing** (*)

Wirkung der Technik Anti-**Aliasing**



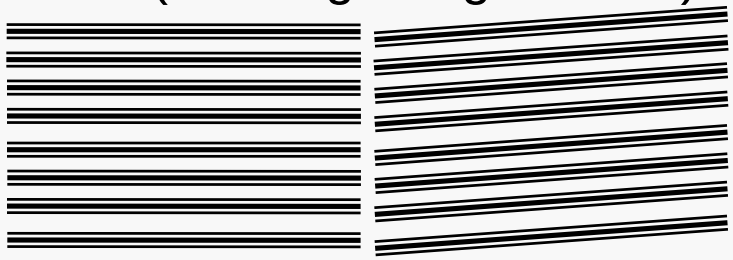
(*) „éiliæsing“ (< alias < άλλως = anders): Veränderung, Entstellung

Aliasing und Antialiasing

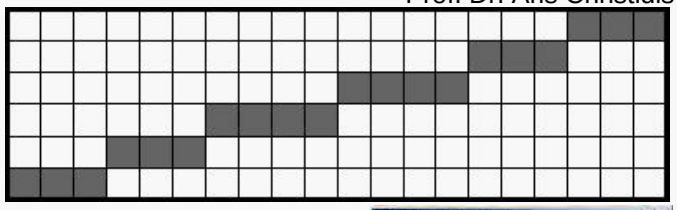
Prof. Dr. Aris Christidis

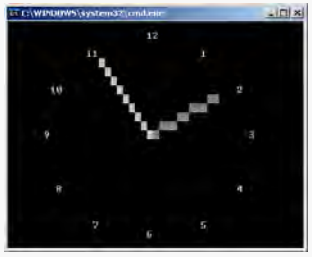
Aliasing-Effekte:

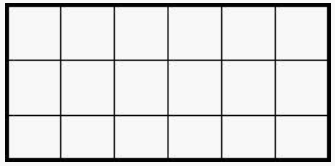
- Brüchige, gestufte, zackige Kanten und Linien („jaggies“)
- Schwingende („atmende“) Kanten animierter Objekte
- Ausblendung / Flimmern ein-pixel-breiter Strukturen
- „Postkutschen-Effekt“ (Objekt-bewegung gegen erwartete Richtung) – ohne Unschärfe!
- Moiré (Überlagerungsmuster)

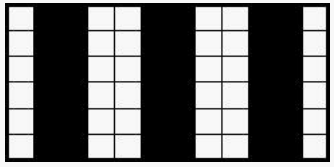


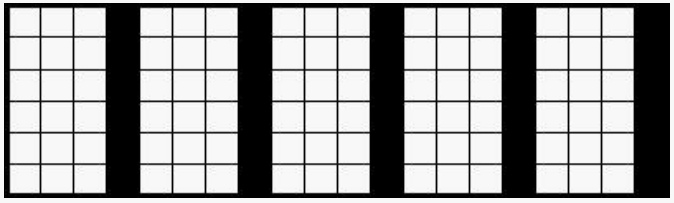
Reelle und synthetische Bilder gleichermaßen betroffen!

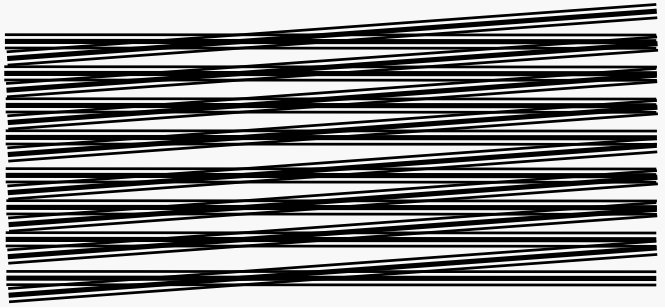










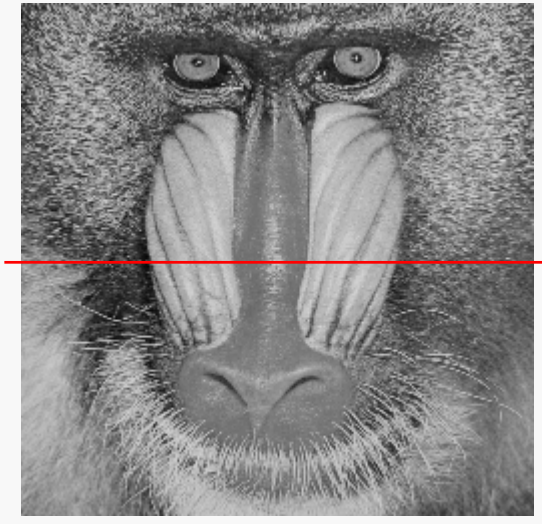


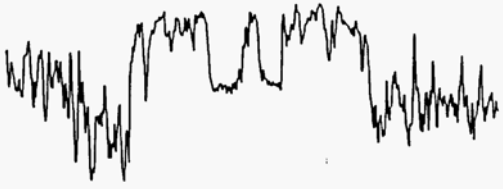
Aliasing und Antialiasing

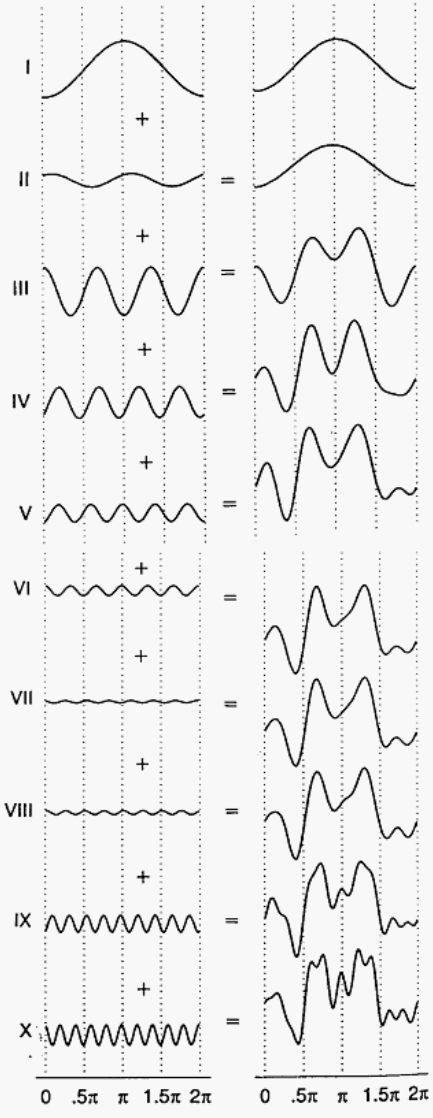
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Abtasttheorem
(H.Nyquist 1928; Beweis: C.E.Shannon 1949):

Ein periodisches Signal läßt sich **aus einer endlichen Anzahl von Abtastwerten exakt** (d.h.: fehlerfrei) **rekonstruieren**. Dabei dürfen die Abtastpunkte nicht weiter auseinanderliegen als eine halbe Periode der höchsten Frequenz, die im Signal enthalten ist. Diese Grenzfrequenz wird „Nyquist-Frequenz“ genannt.







Grafiken: J.D.Foley, A.v.Dam et al.: „Computer Graphics: Principles and Practice in C“, 2/E, Addison-Wesley 1996

Aliasing und Antialiasing

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

Anmerkung:

- Antialiasing mit dem Akk/puffer bedeutet mehrfache, versetzte Überlagerung ein und derselben Szene bei individueller Bild-Rasterisierung nach jeder Versetzung; dazu kann z.B. der Grafik-Pipeline eine abschließende Translation hinzugefügt werden (**`glTranslatef()`** im **`GL_MODELVIEW`**-Modus).

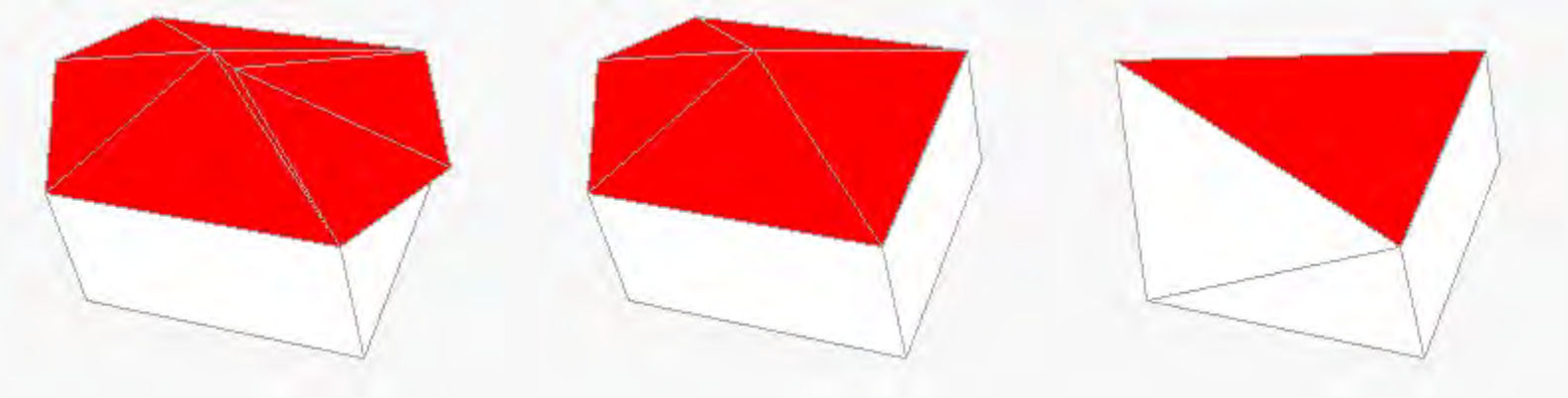
- Alternativ wird der Viewport mehrfach versetzt, die Projektion durchgeführt und das Ergebnisbild rasterisiert (**`glFrustum()`** bzw. **`glOrtho()`** im **`GL_PROJECTION`**-Modus – empfohlen!).

Codierung von Grafik-Modellen

Veränderte Aufgabenstellung in der Praxis:

- Anzahl von Punkten u. Flächen erst beim Laden bekannt (⇨ Speicherplatz-Belegung zur Laufzeit)
- Anzahl von Punkten pro Fläche zudem i.d.R. ungleich (⇨ z.T. verkettete Listen statt Feldern)
- Je nach Anwendung:

  Anzahl von Objektpunkten und Flächen, aber auch Identität von Flächen-Eckpunkten variabel



OpenGL: Punkte, Linien, Polygone

THM
TECHNISCHE HOCHSCHULE MITTELHESSEN
Prof. Dr. Aris Christidis

1.087.716 Dreiecke

Technisch-gestalterische Tips bei Modellierung:

- Einheitlichen Drehsinn für alle Modellflächen einhalten!
- Ebene, konvexe (besser: triangulierte) Polyg. verwenden!
- Bei d. Modelldesign-Entscheidung zwischen facettenreich (langsam) und grob (schnell) sollte die zentrale oder seitliche Lage im Sichtvolumen, die Entfernung vom Augenpunkt, die Vielfalt (Unebenheiten, Buntheit) und der aktuell sichtbare Teil des Originals berücksichtigt werden. Bei Animation sollten Modelle in mehreren Auflösungen (*Levels of Detail*, LODs) verfügbar sein.
- Rechner-Ungenauigkeit einplanen!

3.774 Dreiecke