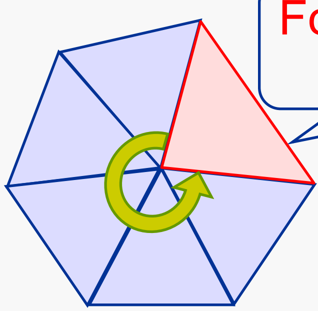
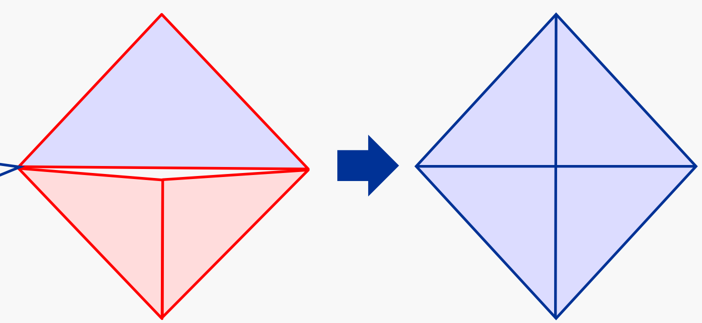
Nachtrag: Allgemeine Betrachtungen

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!





Nachtrag: Allgemeine Betrachtungen

- Bestrebung bei Planung von Grafik-Systemen: Virtuelle Kamera nahe einem fotografischen Normalobjektiv,

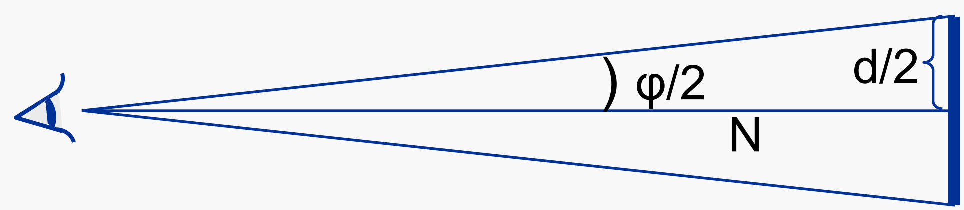
  d.h.: Betrachtung der Bilddiagonale unter ca. 40 – 60°

  z.B. Kleinbild mit Seitenverhältnis 2:3 (24 mm × 36 mm) und Brennweite N=50 mm:

  Bilddiagonale $d = (24^2+36^2)^{1/2}\text{mm} = 1.872^{1/2}\text{mm} \approx 43{,}27\text{mm}$

  Bildwinkel $\varphi=2*\text{arctg}(d/(2*N))=2*\text{arctg}(43{,}27/100) \approx 46{,}79°$

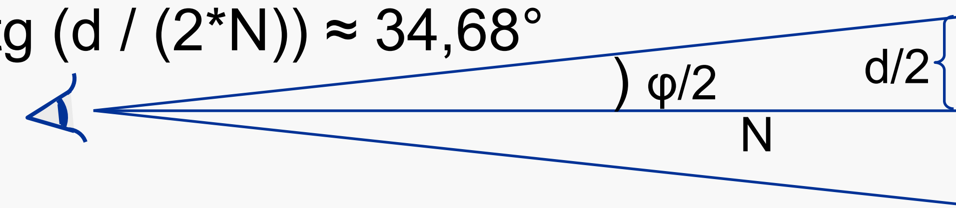
  (vgl.: Weitwinkel: > 75°; Fischauge: ≈ 180°; Tele: < 30°)



Nachtrag: Allgemeine Betrachtungen

- `WireCull()`, `ObjElabGL()` haben variablen Bildwinkel:
  Fensterhöhe = längste Objekt-Ausdehnung (`viewScale`)
  <u>Beispiel `cube.cgf`:</u> Kanten-Koordinaten [-1…+1]; N=10
  Raumdiagonale = $(2^2+2^2+2^2)^{½} = (12)^{½} = 3{,}464$
  Angenommenes Fenster-Seitenverhältnis 2:3 ergibt:
  Fensterdiagonale $d = \left((3{,}464 \cdot 2/2)^2 + (3{,}464 \cdot 3/2)^2 \right)^{½}$
  $= (12 \cdot (4/4+9/4))^{½} = (12 \cdot 13)^{½} / 2 \approx 6{,}24$
  Mit N=10 (`eyez`) folgt:
  Bildwinkel $\varphi = 2 \cdot \arctan(d / (2 \cdot N)) \approx 34{,}68°$

leichtes Tele



- Brennweite N zur Erlangung eines Bildwinkels φ=45°:
  $N = d / (2 \cdot \tan(\varphi/2)) = (12 \cdot 13)^{½}/(2 \cdot 2 \cdot \tan(22{,}5°)) \approx 7{,}538$

Nachtrag: Animation

Grafik-Animation („-Beatmung“, „-Behauchung“):

- Bewegungsillusion durch schnelle Abfolge Bild-Löschung-Bild... (vgl. Kino: Bild-Blende-Bild...); bei sichtbarer Bild-Entstehung: Flimmern des zuletzt berechneten Bildteils

  ➥ Animation meist mit **Double-Buffering**:

  Darbietung erst nach Fertigstellung eines Bildes durch Wechsel (engl. *swap*) des zugeschalteten Bildspeichers
- Aufgrund starker Abhängigkeit von Hardware überläßt es OpenGL dem Fenstersystem: **`glutSwapBuffers();`**
- Typische Bildraten in Hz (Bilder/sec = *fps* [frames per sec]):

Ch.Chaplin-Filme:	16 Hz bzw. 62,5 ms/Bild
modernes Kino:	24 Hz bzw. 41,66... ms/Bild
TV (Interlaced):	50 Hz bzw. 20,0 ms/Halbbild
Flug-Simulation:	60-120 Hz bzw. 16,66...-8,33... ms/Bild

Nachtrag: Animation

- Darstellung auf elektr. Ausgabemedien (Monitor, Beamer) flüchtig: Bild wird in konstanten Zeitabständen t_R [msec] aktualisiert (vgl. TV).

  ⇨ Bildauffrischungsrate (engl. *refresh rate*) [Hz] $f_R = 1/\, t_R$

- Wichtig für die Illusion kontinuierlicher Bewegung: Konstante Bildrate (engl. *frame rate*) [Hz] $f_F = 1 \,/\, t_F$, mit t_F [msec]: Darbietungszeit eines Bildes bzw. Zeit zwischen dem Zeichenbeginn zweier aufeinanderfolgender Bilder.

  ⇨ t_F kann nur ein ganzzahliges Vielfaches von t_R sein:

  $t_F = n \cdot t_R$, mit $n \geq 1$, $n \in \mathbf{N}$

- Es muß je Bild Rechenzeit $t_A \leq t_F$ **= 1/ f_F** gewährleistet sein ⇨ bei Eintritt komplexer Objekte in das Sichtvolumen ggf. Auslassen (engl. *drop*) v. Einzelbildern u. Herabsetzg der Bildrate so, daß jedes Bild als Rechenzeit ein ganzzahl. Vielfaches der Sichtsystem-Bildauffrischungszeit erhält.

Nachtrag: Animation

- Definition: Die **Ganzzahlfunktion** [x] ordnet einer reellen Zahl x eine ganze Zahl z zu:

  „Gauß-Klammer"

  $$z = [x] \quad \Leftrightarrow \quad x-1 < z \leq x \quad (z \in \mathbf{Z},\ x \in \mathbf{R})$$

  Für nicht-positive $x = -|t| \leq 0$, $t \in R$, gilt:

  $$z = [-|t|] \quad \Leftrightarrow \quad -|t| - 1 < z \leq -|t|$$

  bzw.: $\Leftrightarrow \quad |t| + 1 > -z \geq |t|$

  Mit $n = -z \geq 0$ wird daraus:

  $$n = -z = -[-|t|] \quad \Leftrightarrow \quad |t|+1 > n \geq |t| \quad (n \in \mathbf{N}_0,\ t \in \mathbf{R})$$

  – in Worten: Die Funktion $n_{(t)} = -[-|t|]$ ordnet einer nicht-negativen reellen Zahl $|t|$ die nicht-negative ganze Zahl n mit dem gleichen oder dem nächstgrößeren Betrag zu.

- Daraus Forderung für Animation: $t_F = n \cdot t_R$ mit $n = -[-t_A / t_R]$

  $\Rightarrow \quad t_F = -[-t_A / t_R] \cdot t_R$

  bzw. wegen $f_F = 1/t_F$ und $f_R = 1/t_R$ und $n = -[-t_A \cdot f_R]$:

  $$f_F = -f_R / [-t_A \cdot f_R]$$

Nachtrag: Animation

⇨ Berechnung der Bildrate f_F aus der Rechenzeit $\boldsymbol{t_A}$ für ein Bild und der Bildauffrischungsrate f_R:

$f_F = - f_R \,/\, [- t_{\mathbf{A}} \cdot f_R\,]$

<u>Beispiel:</u>

Wenn bei einer „*refresh rate*“ von f_R =120 Hz ein Bild $t_{\mathbf{A}}$ = 1/50 sec Rechenzeit braucht, beträgt die „*frame rate*“:

f_F = 120 / -[-120 / 50] Hz = 120/ -[-2,4] Hz = 120/ -[-3] Hz = 40 Hz

Die verfügbare (Rechen-,Warte-) Zeit je Bild beträgt dann:

t_F = 1/40 sec = 1000/40 ms = 25 ms

D.h., jede 3. Bildauffrischung bekommt einen neuen Inhalt u.d.Grafik-System bleibt untätig (engl. *idle*) für 20% d. Zeit

$(t_{\mathbf{F}} – t_{\mathbf{A}}) / t_{\mathbf{F}} = (1/40–1/50) / 1/40 = 40·(5-4)/200 = 1/5 = 0{,}2$