Computergraphik

Prof. Dr.-Ing. Kerstin Müller

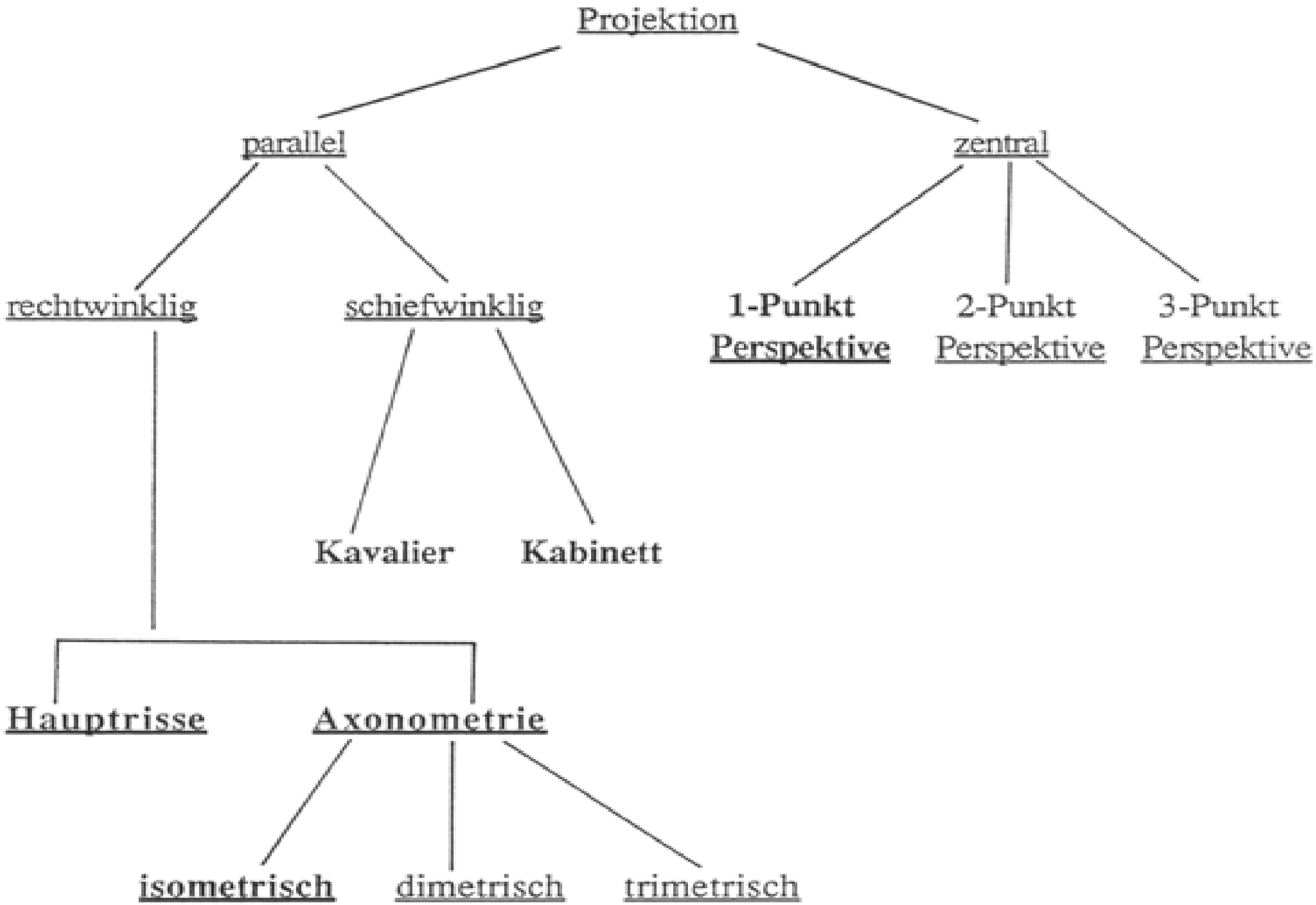
Kapitel 4 (Teil 2)

Transformationen und Projektionen

4.4 Projektionen

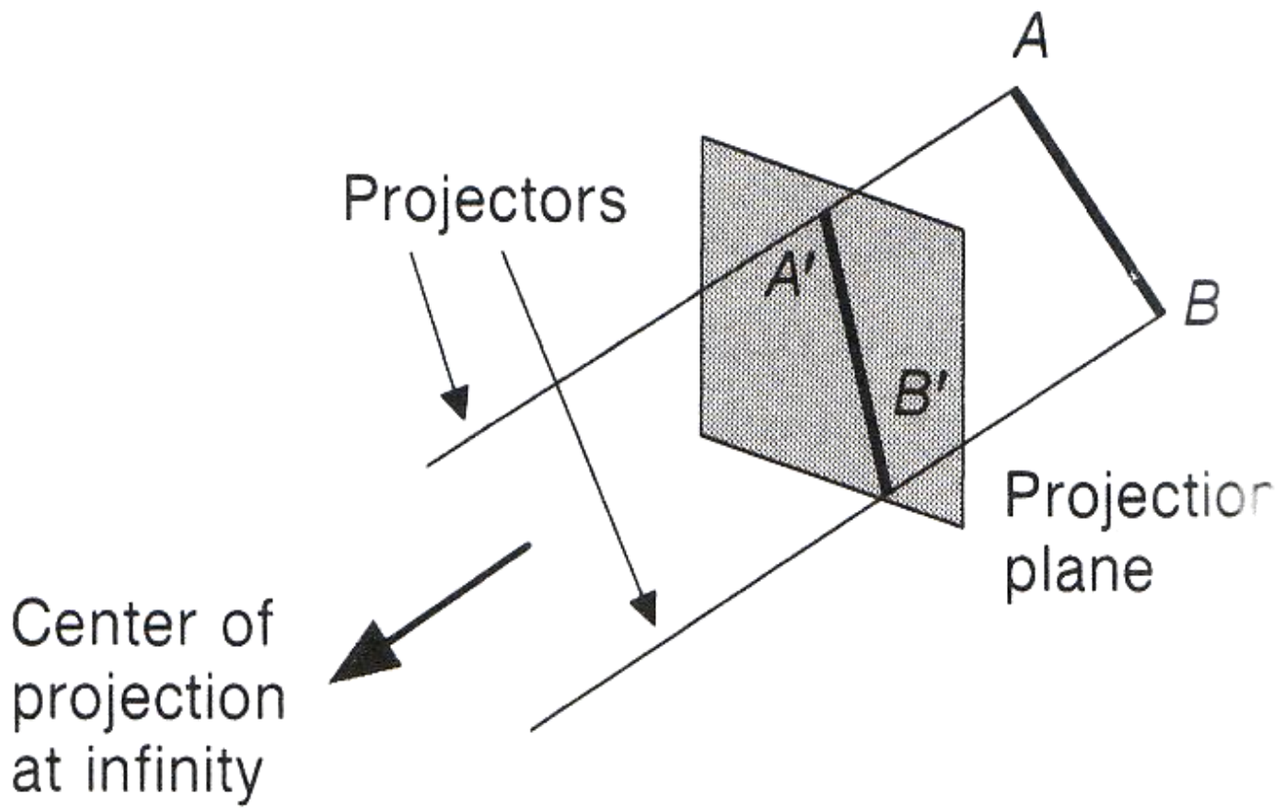
- Hier: Eine Projektion ist eine Abbildung, die einen Raum der Dimension n auf einen Raum mit einer Dimension <n abbildet (speziell: 3D→2D).
- Da ein Bildschirm ein 2-dimensionales Ausgabemedium ist, müssen 3-dimensionale Objekt in 2-dimensionalen Ansichten dargestellt werden. Hierzu wird ein Raumpunkt entlang eines Projektionsstrahls (projector) auf eine vorgegebene Projektionsebene (projection plane) abgebildet.
- Der Projektionsstrahl wird durch das Projektionszentrum und den Raumpunkt festgelegt. Der Schnittpunkt des Projektionsstrahls mit der Projektionsebene bestimmt den projizierten Raumpunkt.

Klassifikation von Projektionen



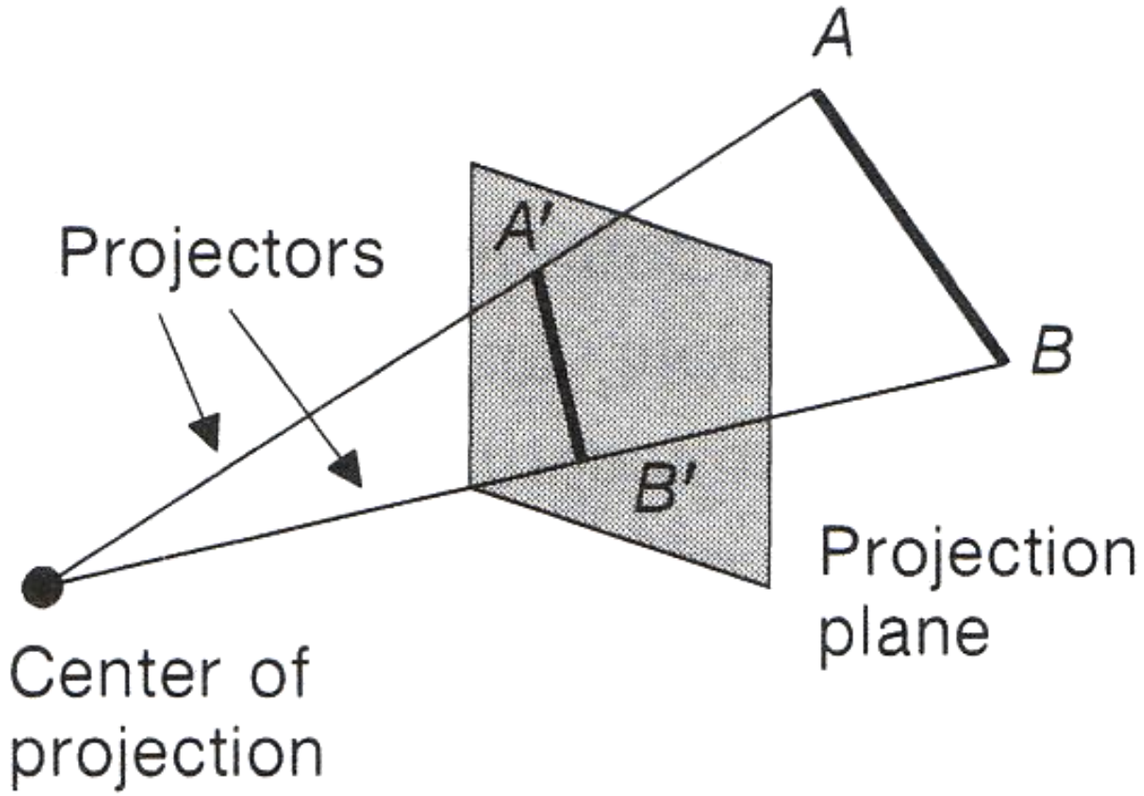
Parallelprojektionen

- Alle Projektionsstrahlen verlaufen parallel in eine Richtung.
- Projektionszentrum liegt in einem unendlich fernen Punkt.
- In der projektiven Geometrie stellt die Parallelprojektion somit einen Spezialfall der Zentralprojektion dar.
- Weniger realistisch, erlaubt aber die Bestimmung exakter Maße aus dem Bild.



Perspektivische Projektionen

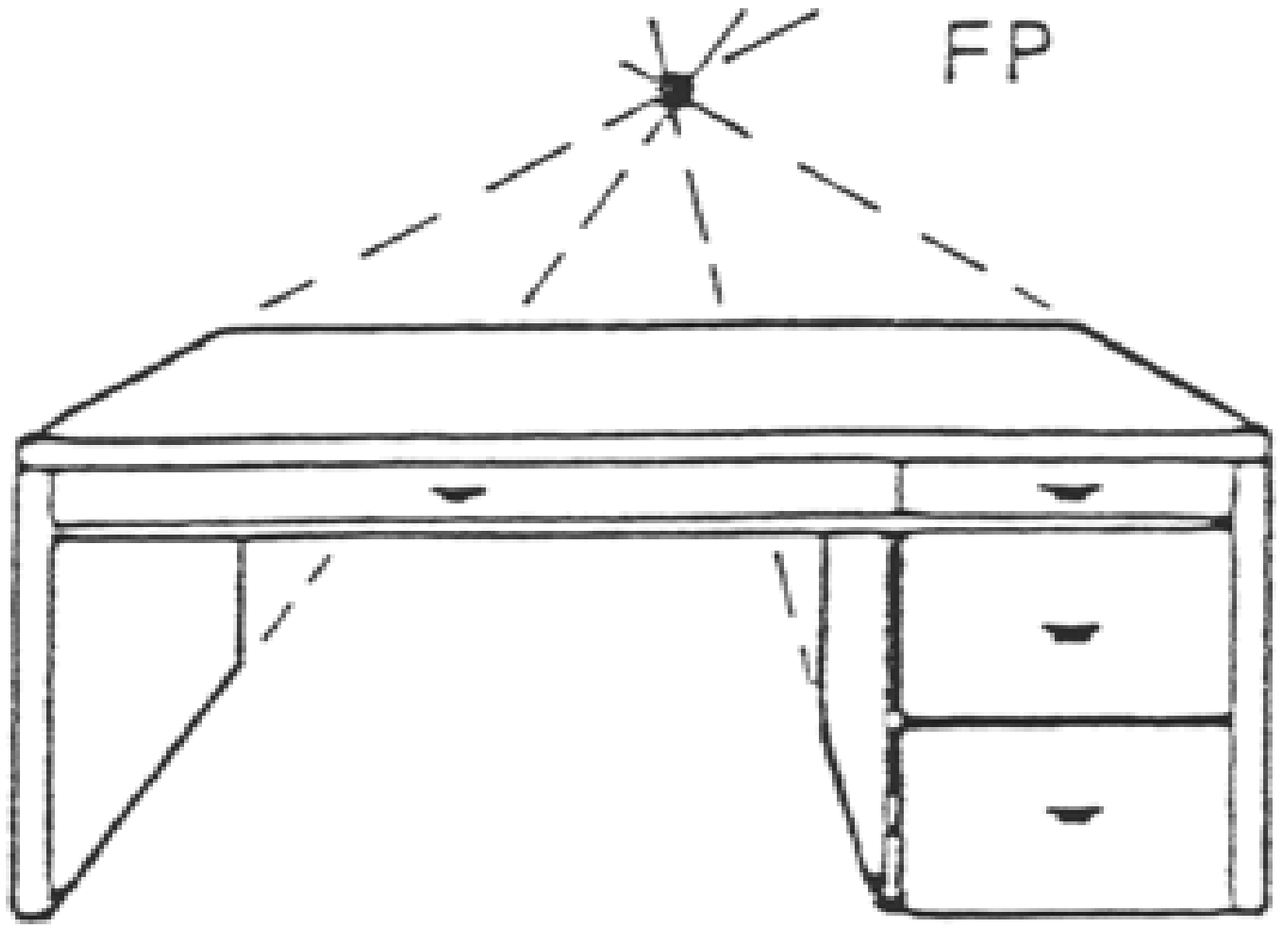
- Bei den perspektivischen Projektionen (Zentralprojektion-en) gehen alle Projektionsstrahlen durch das Projektions-zentrum, das mit dem Auge des Beobachters zusammen-fällt.
- Das Verfahren erzeugt eine optische Tiefenwirkung und geht in seinen Anfängen bis in die Malerei der Antike zurück.



Eigenschaften von Zentralprojektionen

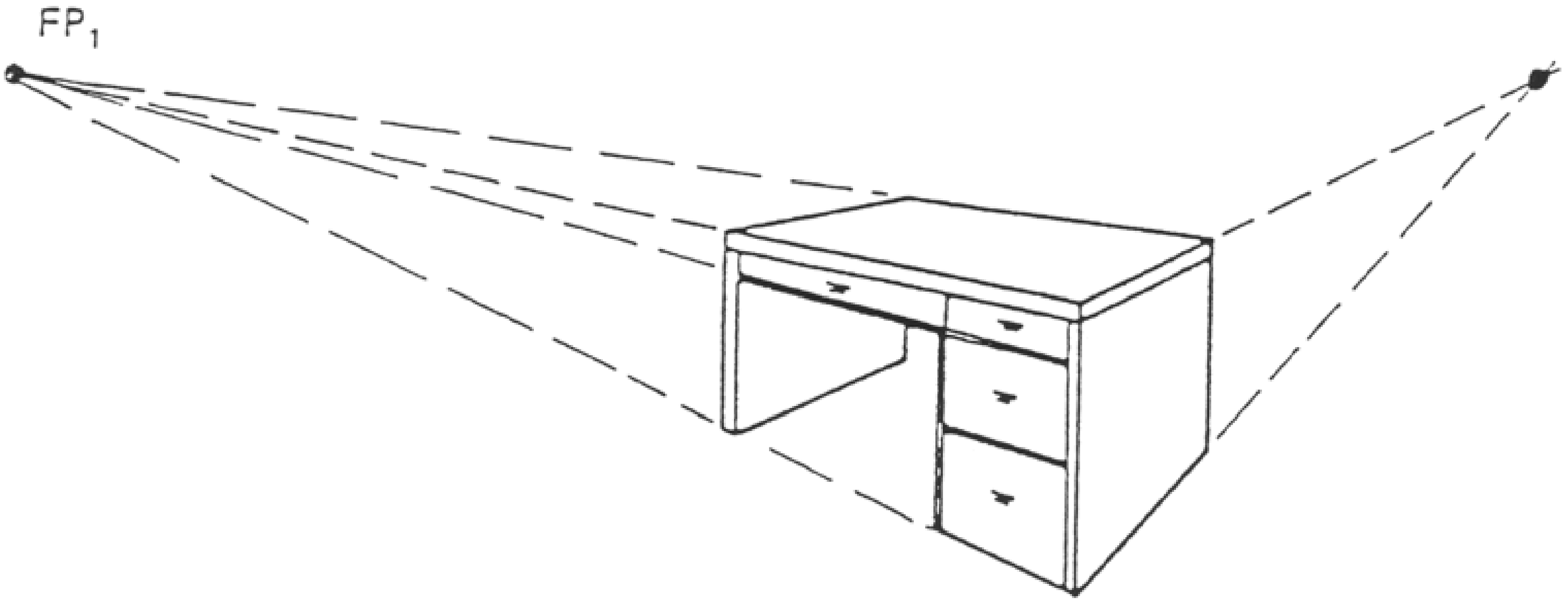
- Eigenschaften:
  - je zwei parallele Geraden, die nicht parallel zur Projektionsebene sind, treffen sich in einem Punkt, dem Fluchtpunkt.
  - Es gibt unendlich viele Fluchtpunkte, je einen pro Richtung nicht parallel zur Projektionsebene.
    - Hervorgehoben werden die Fluchtpunkte der Hauptachsen: z.B. Geraden, die parallel zur x-Achse verlaufen, treffen sich in der Projektionsebene im x-Fluchtpunkt.
- Perspektivische Projektionen werden nach der Anzahl der Hauptachsen, die von der Projektionsebene geschnitten werden, klassifiziert. So entstehen 1-Punkt-, 2-Punkt- und 3-Punkt-Perspektiven.

1-Punkt-Perspektive



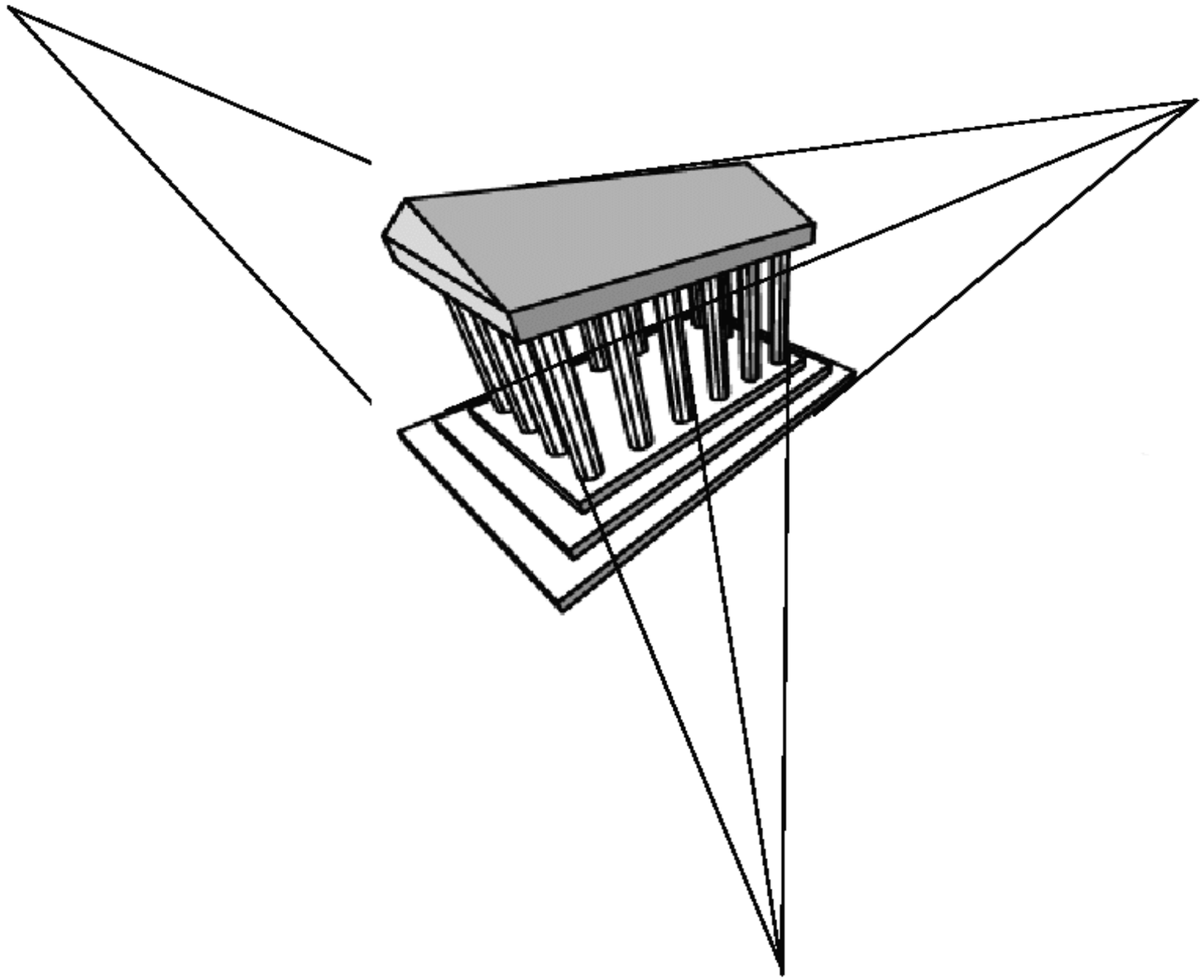
1-Punkt-Perspektive

2-Punkt-Perspektive

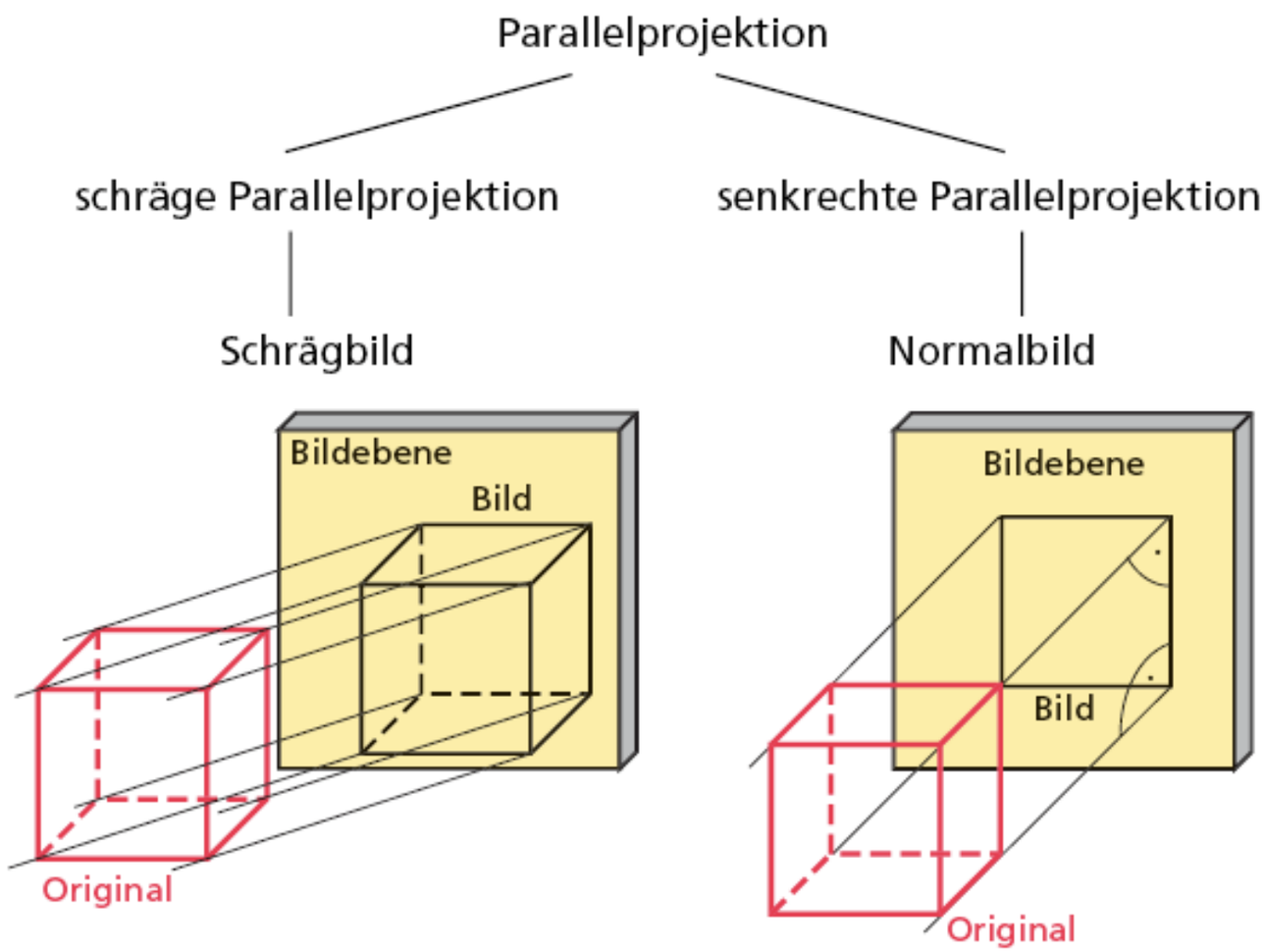


2-Punkt-Perspektive

3-Punkt-Perspektive



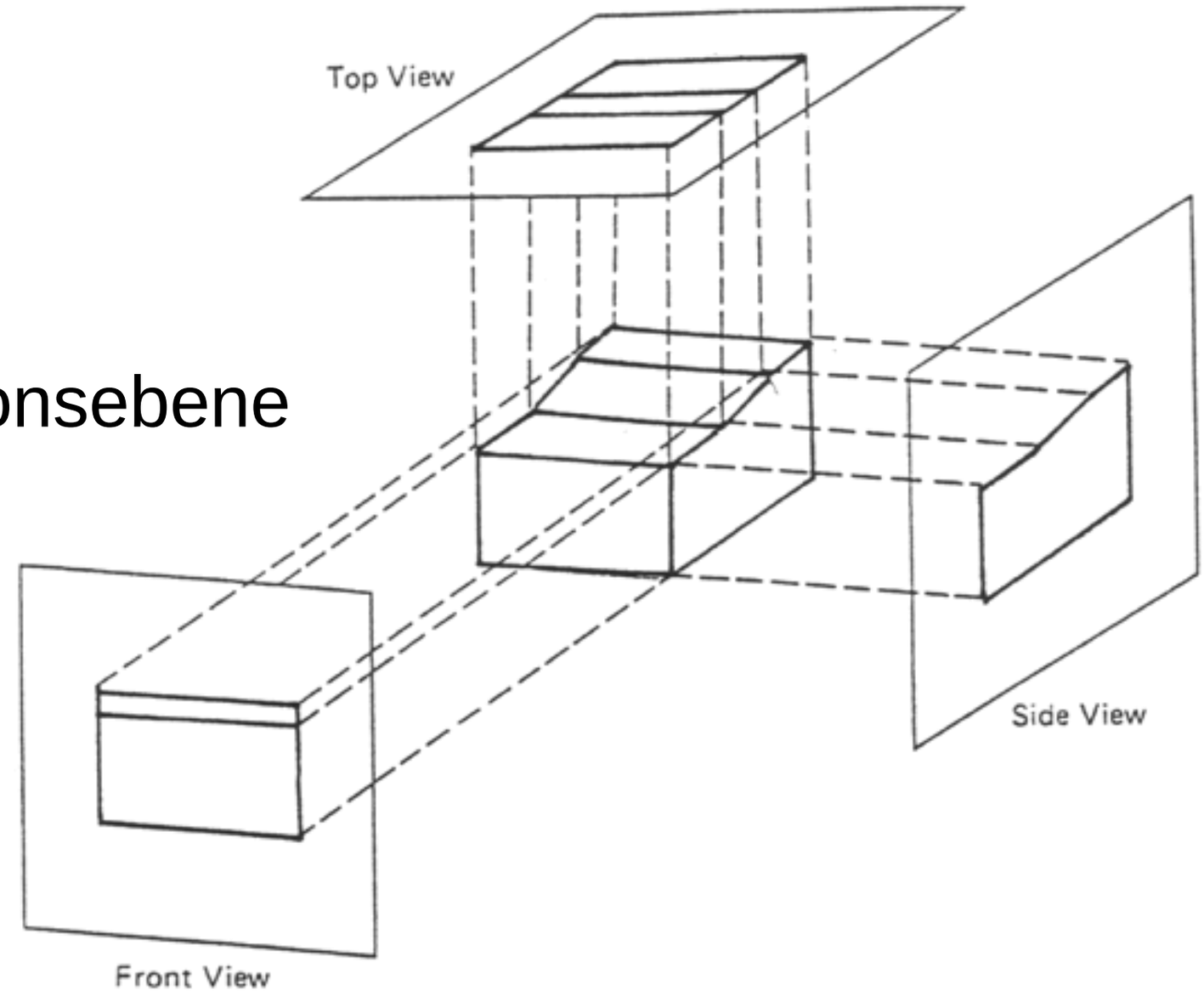
Unterscheidung Parallelprojektionen



Hauptrisse

- Bei den Hauptrissen
  - Grundriss (top view)
  - Aufriss (front view)
  - Kreuzriss (side view)

  schneidet die Projektionsebene nur eine Hauptachse.
- Die Normale der Projektionsebene ist also parallel zu einer der Hauptachsen.



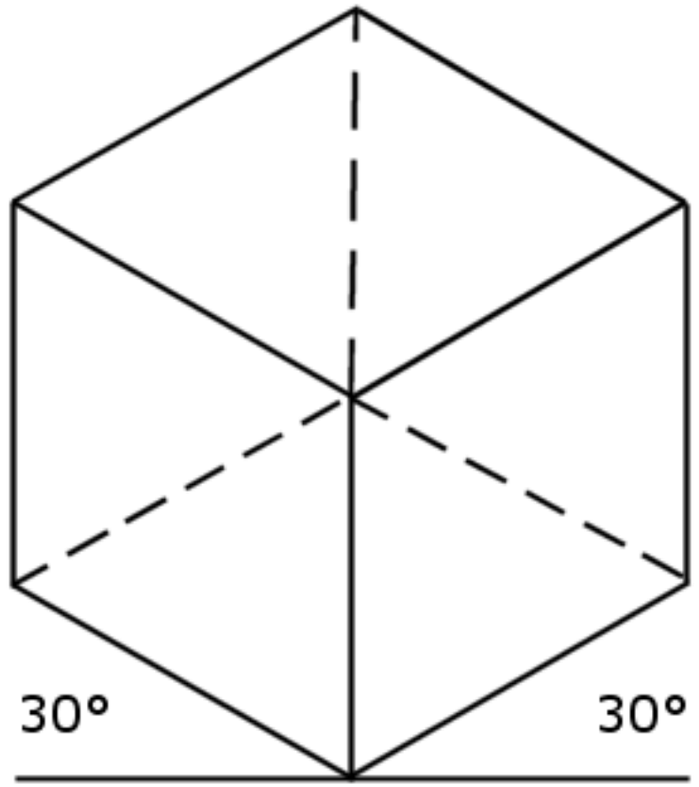
- Typischer Aufbau bei CAD-Systemen: Drei Hauptrisse und eine perspektivische Ansicht.

Axonometrie

- Bei der Axonometrie ist die Projektionsebene nicht orthogonal zu einer der Koordinatenachsen.
  - Parallele Linien werden auf parallele Linien abgebildet.
  - Winkel bleiben nicht erhalten.
  - Abstände können längs der Hauptachsen gemessen werden, allerdings i.a. in jeweils anderem Maßstab.
- Es gibt drei verschiedene Axonometrische Projektionen:
  - Isometrie
  - Dimetrie
  - Trimetrie (eher unüblich)
- Axonometrien werden oft für Handzeichnungen verwendet.

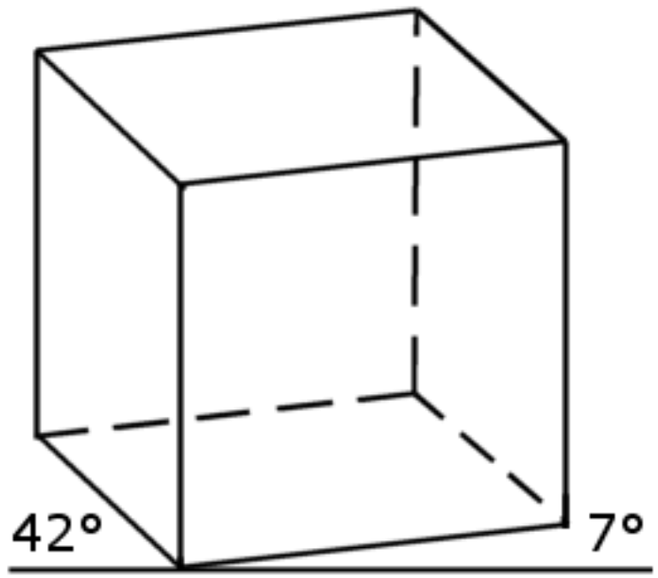
Isometrische Projektion

- Bei der isometrischen Axonometrie bildet die Projektionsebene mit allen Hauptachsen den gleichen Winkel. Hier hat man eine gleichmässige Verkürzung aller Koordinatenachsen (Achsenmaßstab 1 : 1 : 1)
- Es gibt 3D→2D nur 8 mögliche isometrische Projektionen.
- Alle drei Seiten werden unverkürzt wiedergegeben, die Isometrie eignet sich jedoch nicht für zentralsymmetrische Objekte.



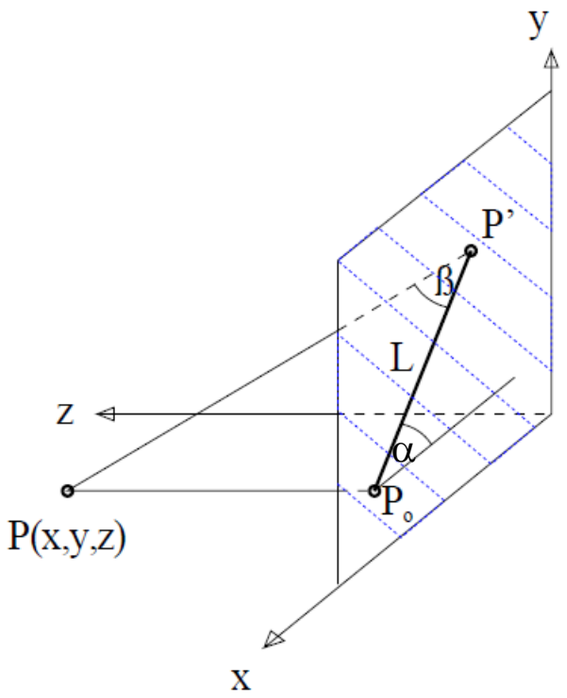
Dimetrische Projektion

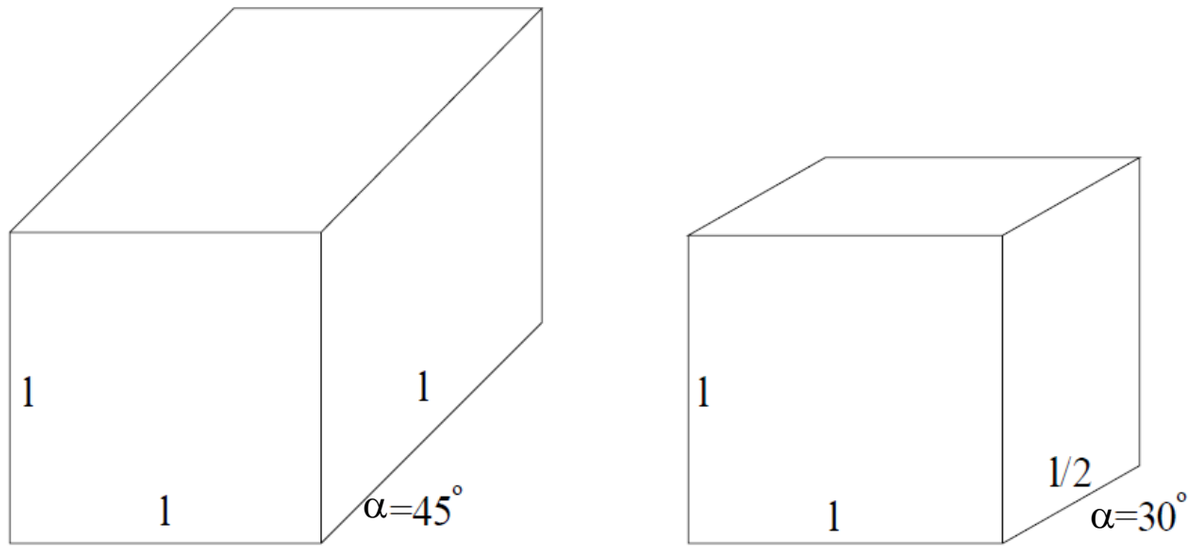
- Bei der dimetrischen Projektion bildet die Projektions-ebene mit zwei Hauptachsen den gleichen Winkel, die Skalierung ist in zwei Achsenrichtungen gleich.
- Typische Winkel: Die Vertikalen bleiben unverkürzt. Von den beiden in die Tiefe laufenden Linien wird die eine im 42° Winkel zur Waagerechten, die andere im 7° Winkel gezeichnet (Achsenmaßstab 1 : 0,5 : 1).
- Die Dimetrie wird vor allem benutzt, wenn die (nur leicht verzerrte) Vorder-ansicht betont werden soll.
- Trimetrie: Achsenmaßstab (a : b : c)



Kavalier- und Kabinettprojektionen

- Kavalier- und Kabinettprojektionen gehören zu den schiefwinkligen Parallelprojektionen. Sie entstehen, wenn sich die Projektionsrichtung von der Projektionsebenennormalen unterscheidet.





Beispiele einer Kavalierprojektion ($\tan(\beta) = 1$) mit $\alpha = 45°$ und einer Kabinettprojektion ($\tan(\beta) = \frac{1}{2}$) mit $\alpha = 30°$ eines Würfels

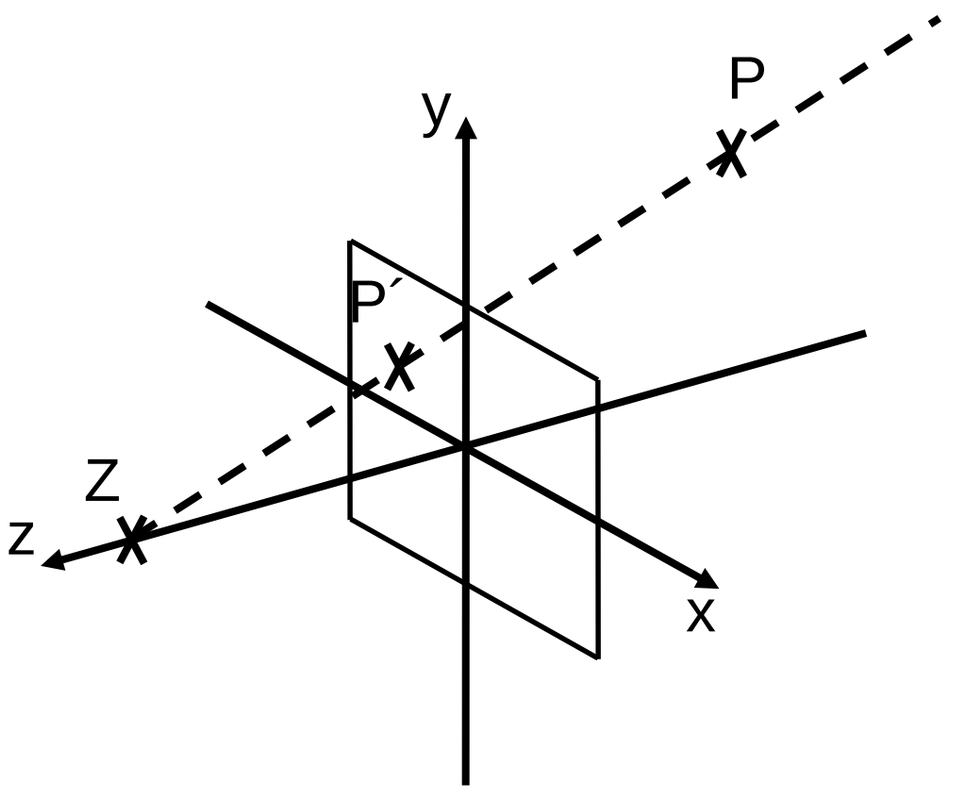
Zusammenfassung Parallelprojektionen

- Parallelprojektionen werden sehr häufig in 3D-CAD Systemen verwendet.
- Längen bleiben erhalten oder werden unabhängig vom Abstand der Objekte zur Bildebene mit einem festen Faktor skaliert.
- Winkel bleiben erhalten oder werden durch eine festgelegte Art und Weise abgebildet.
- Messungen im projizierten Bild sind möglich, Tiefeninformationen gehen verloren.
- Darstellung als Projektionsmatrix ist möglich.

Umsetzung der Zentralprojektion

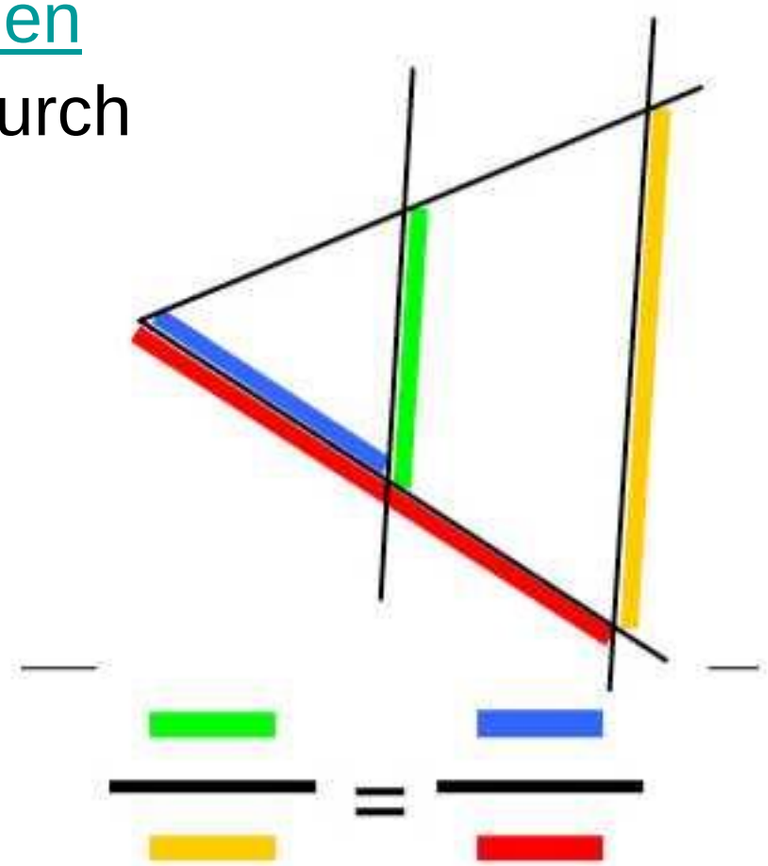
- Die praktische Umsetzung der perspektivischen Projektion erfolgt je nach Anwendung in unterschiedlichen Konfigurationen, die mittels geeigneter Transformation des Koordinatensystems erreicht werden können.

- Exemplarisches Setup:
  - Projektionszentrum Z und Aug-punkt fallen zusammen, liegen auf der positiven z-Achse mit Ab-stand d>0 zum Ursprung, also Z=(0,0,d).
  - Blickrichtung ist die negative z-Achse
  - Bildebene liegt in der (x,y)-Ebene



Berechnung der Zentralprojektion

- 2. Strahlensatz:
  - Wenn zwei durch einen Punkt verlaufende Halbgeraden (Strahlen) von zwei parallelen Geraden geschnitten werden, die nicht durch den Scheitel gehen, dann gilt:

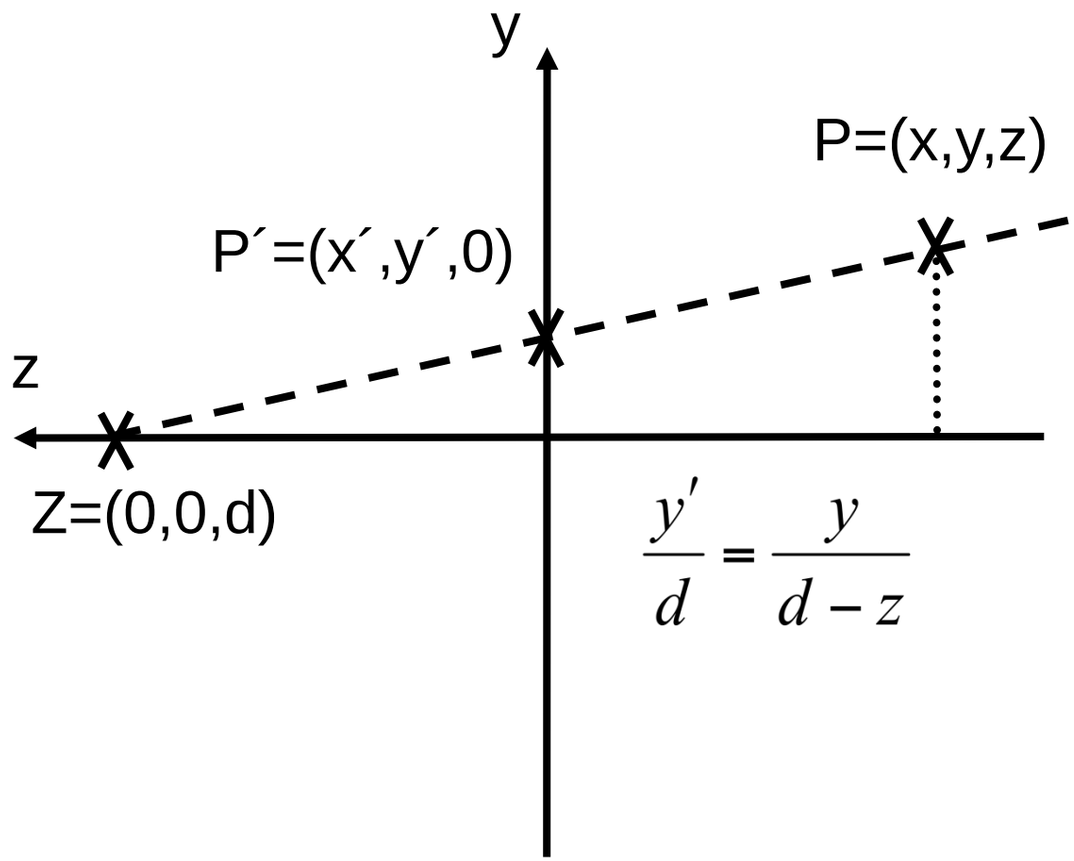


  - Es verhalten sich die ausgeschnittenen Strecken auf den Parallelen, wie die ihnen entsprechenden, vom Scheitel aus gemessenen Strecken auf den Strahlen.

Berechnung der Zentralprojektion

- Aus dem Strahlensatz folgt:

$$\frac{y'}{d} = \frac{y}{d-z} \text{ und } \frac{x'}{d} = \frac{x}{d-z}$$



folglich:

$$y' = y \cdot \left(\frac{d}{d-z}\right) = y \cdot \left(1 - \frac{z}{d}\right)^{-1}$$

$$x' = x \cdot \left(\frac{d}{d-z}\right) = x \cdot \left(1 - \frac{z}{d}\right)^{-1}$$

Berechnung der Zentralprojektion

- Die Zentralprojektion wird in diesem Setup beschrieben durch:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -d^{-1} & 1 \end{bmatrix}$$

- Proberechnung:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -d^{-1} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 0 \\ 1 - \frac{z}{d} \end{bmatrix} \Rightarrow \begin{bmatrix} x \cdot (1 - \frac{z}{d})^{-1} \\ y \cdot (1 - \frac{z}{d})^{-1} \\ 0 \\ 1 \end{bmatrix}$$

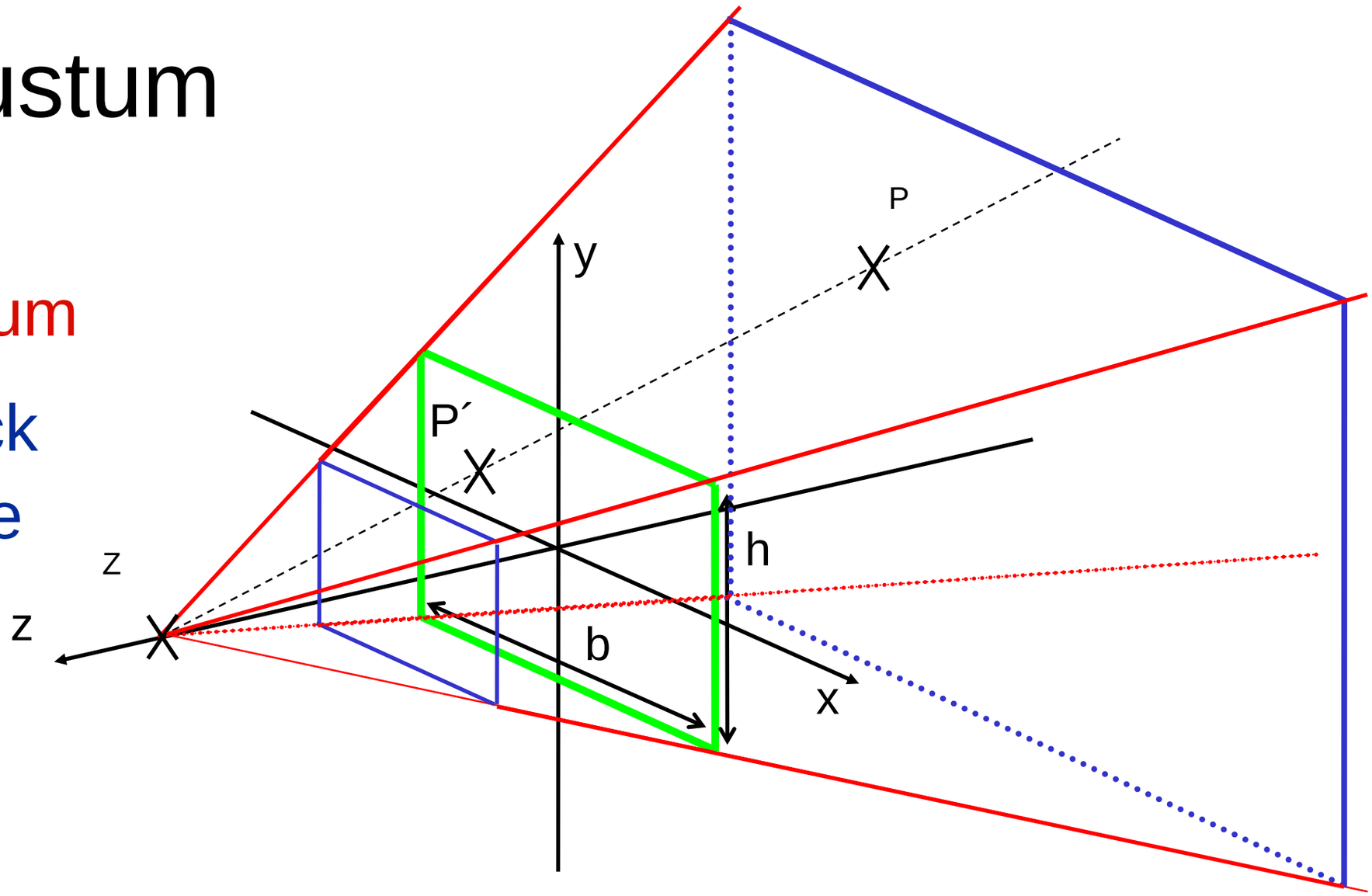
Homogenisierung

Erweitertes Setup

- In der Bildebene wird ein Sichtfenster (view window) spezifiziert (Breite b, Höhe h, Verhältnis Breite zu Höhe); es ist symmetrisch um den Ursprung angeordnet.
- Die Projektoren durch die Ecken des Sichtfensters definieren das sogenannte Sichtvolumen (viewing frustum).
- Zusätzlich begrenzen zwei zur Bildebene parallele Ebenen (front und back clipping plane) das Sichtvolumen in z-Richtung
- Das Sichtvolumen begrenzt den Raum, der dargestellt werden soll (siehe 4.6 Clipping).

Viewing frustum

- view window
- viewing frustum
- front und back clipping plane

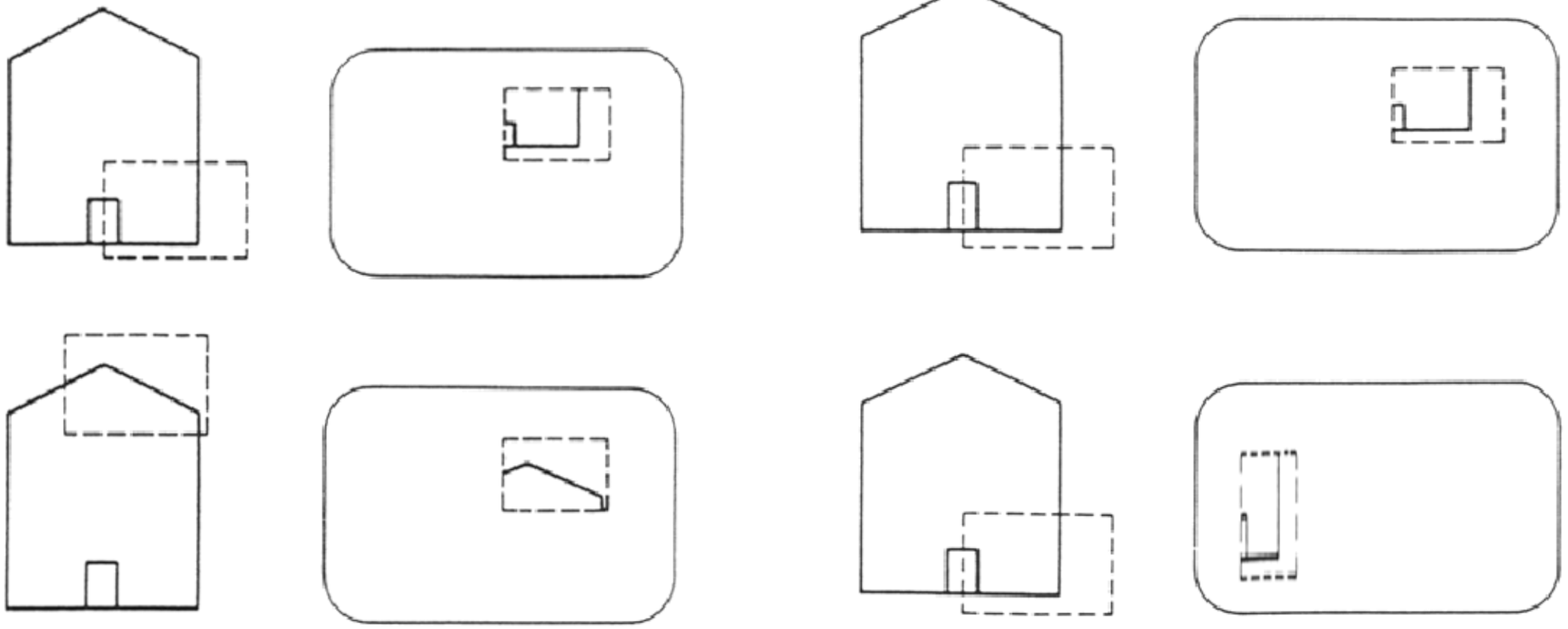


3.5 Windowing

- Window
  - Auch „view window“ (→erweitertes Setup Zentralprojektion)
  - Definiert Sichtfenster in der Bildebene.
  - Definiert, welcher Teilbereich der Szene abgebildet werden soll.
- Viewport
  - Definiert den Bildschirmbereich, wo der Inhalt des view-windows dargestellt werden soll.
- In der Regel sind sowohl Window als auch Viewport an den Koordinatenachsen ausgerichtete rechteckige Gebiete.

Windowing

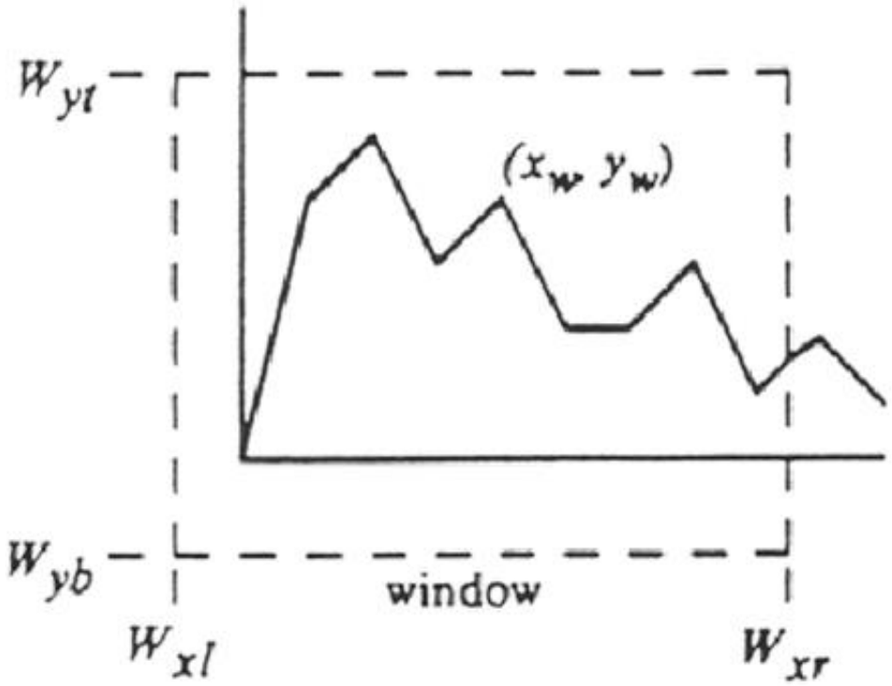
- Die Window-Operation (Window-Viewport Transformation) setzt sich aus elementaren Translationen und Skalierungen zusammen.

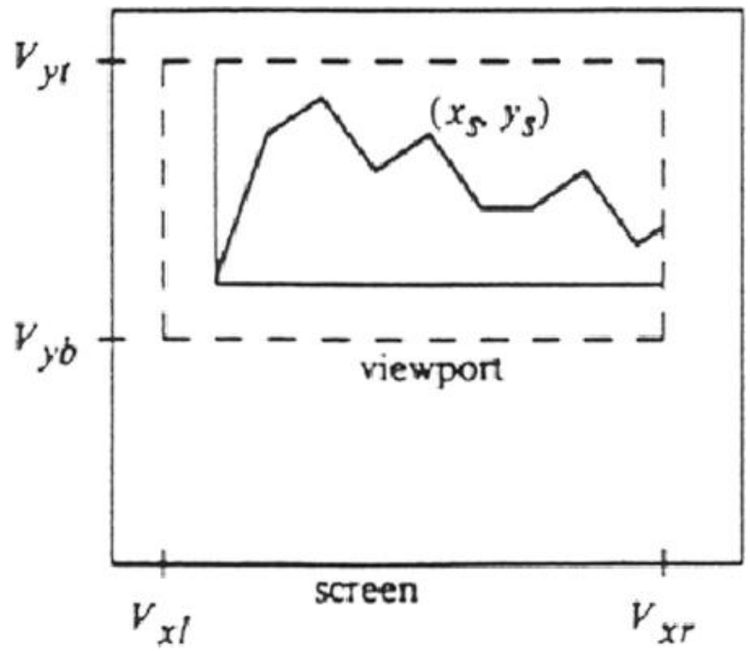


Verschiedene Fenster, dieselben Viewports

Dieselben Fenster, verschiedene Viewports

Window-Viewport Transformation





- x_w, y_w Punktkoordinaten im Window.
- x_s, y_s Punktkoordinaten auf dem Bildschirm.
- W_{xl}, W_{xr}, W_{yb}, W_{yt} Koordinaten des Windows.
- V_{xl}, V_{xr}, V_{yb}, V_{yt} Koordinaten des Viewports im Bild-schirmKoordinatensystem

Window-Viewport Transformation

- Transformation in 3 Schritten
  - Translation in den Koordinatenursprung:

$$\begin{aligned} x' &= x_W - W_{xl} \\ y' &= y_W - W_{yb} \end{aligned}$$

  - Skalierung auf gewünschte Größe:

$$\begin{aligned} x'' &= \frac{V_{xr} - V_{xl}}{W_{xr} - W_{xl}} \cdot x' \\ y'' &= \frac{V_{yt} - V_{yb}}{W_{yt} - W_{yb}} \cdot y' \end{aligned}$$

  - Translation an gewünschte Stelle:

$$\begin{aligned} x_s &= x'' + V_{xl} \\ y_s &= y'' + V_{yb} \end{aligned}$$

Zusammenfassung Windowing

- Zusammenfassung der vorherigen Rechnung:

$$x_s = a \cdot x_W + b$$
$$y_s = c \cdot y_W + d$$

mit

$$a = \frac{V_{xr} - V_{xl}}{W_{xr} - W_{xl}}, \quad b = V_{xl} - a \cdot W_{xl}$$
$$c = \frac{V_{yt} - V_{yb}}{W_{yt} - W_{yb}}, \quad d = V_{yb} - c \cdot W_{yb}$$

- a, b, c, d sind fest und können vorberechnet werden.
- → Punkttransformation durch 2 Multiplikationen und zwei Additionen

4.6 Clipping

- Sollen Objekte in der Bildebene innerhalb eines Fensters dargestellt werden, so wird ein Verfahren benötigt, um alle außerhalb des Fensters liegenden Objektteile abzuschneiden.
- Die innerhalb liegenden Teile sind dann dann potentiell sichtbar (potentiell, da noch keine Objekt-Verdeckungen untereinander berücksichtigt wurden).
- Es existieren Clipping-Verfahren für 2D und 3D.

Clipping von Linien

- Beim Clipping an einem rechteckigen achsenparallelen Fenster gibt es offensichtlich 3 Fälle, von denen zwei schnell gelöst sind:
  - Beide Endunkte liegen innerhalb des Fensters

    → Linie zeichnen

  - Beide Endpunkte der Linie liegen oberhalb, unterhalb, links oder rechts des Fensters:

    → Linie nicht zeichnen

  - Sonst: Schnittpunkte der Linie mit dem Fensterrand berechnen und daraus die sichtbare Strecke bestimmen

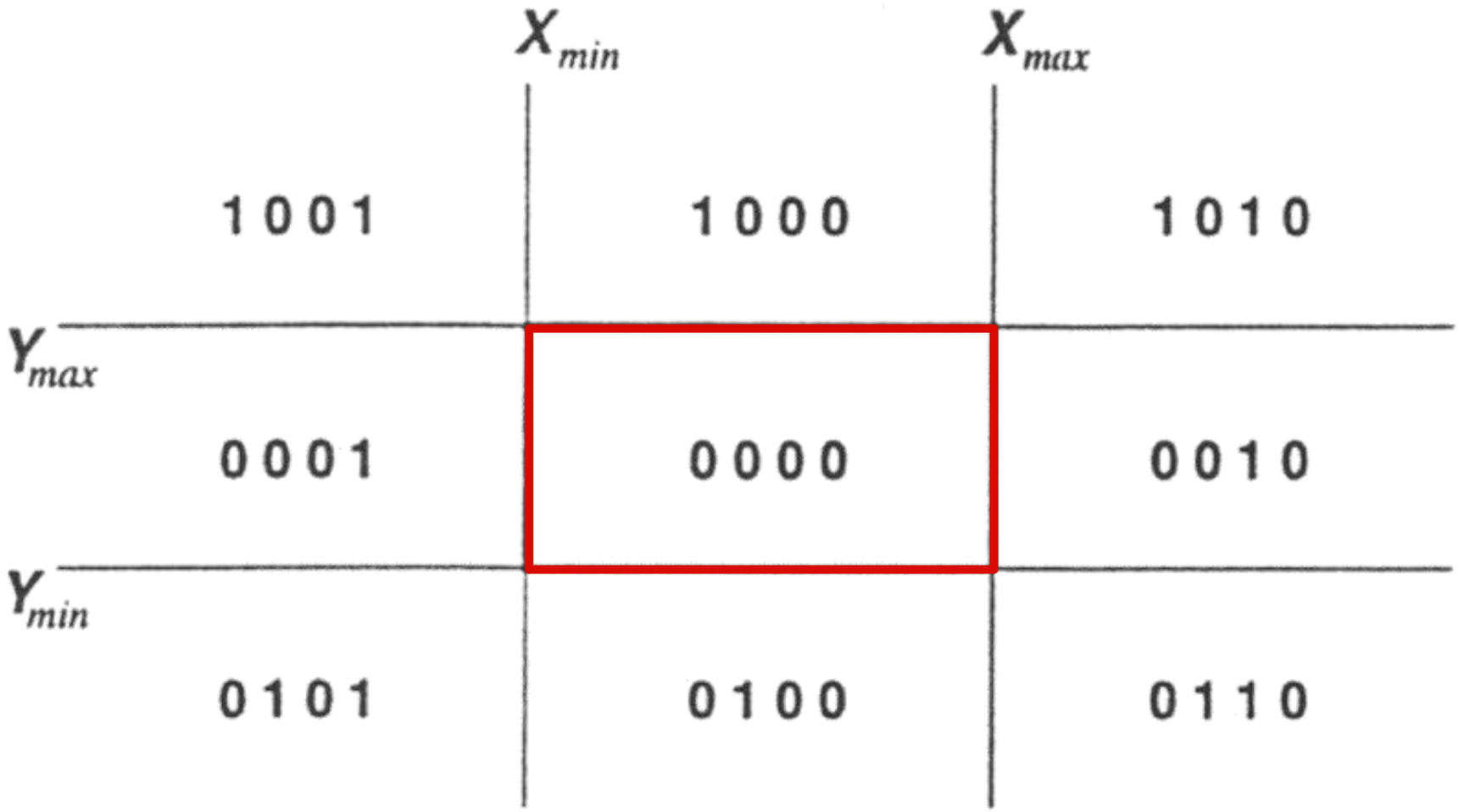
Cohen-Sutherland Line-Clipping (1)

- Idee: schnelles Verfahren zur Klassifizierung der Linien nach innerhalb, außerhalb, schneidend benutzen.
- Gegeben: Fenster (x_{min}, y_{min}, x_{max}, y_{max})
- Die begrenzenden Geraden zerlegen die Bildebene in neun Regionen. Ein 4-Bit-Code gibt Auskunft über die Lage in Bezug auf das Fenster:

	...gesetzt falls Region...	
Bit 0	...links des Fensters	$x<x_{min}$
Bit 1	...rechts des Fensters	$x>x_{max}$
Bit 2	...unterhalb des Fensters	$y<y_{min}$
Bit 3	...oberhalb des Fensters	$y>y_{max}$

Cohen-Sutherland Line-Clipping (2)

- Codes für Fenster und umgebende Regionen:

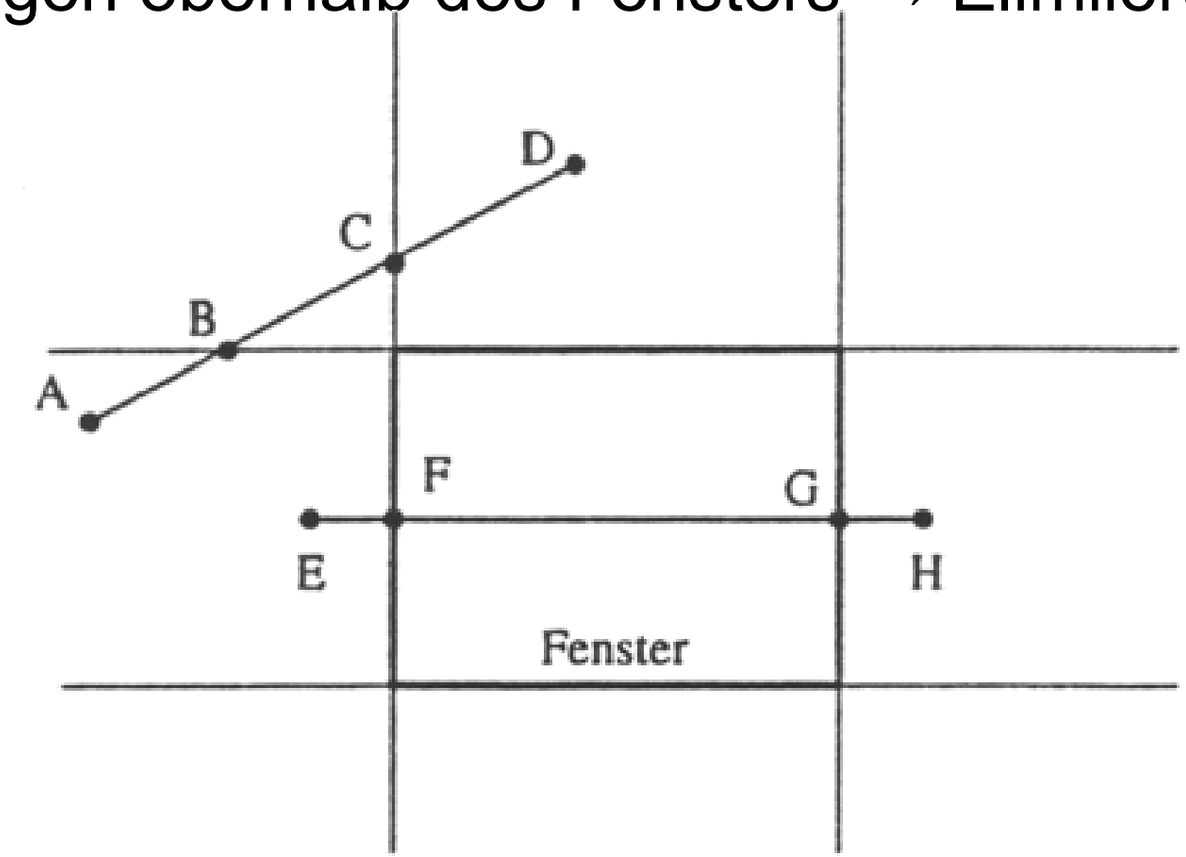


Cohen-Sutherland Line-Clipping (3)

- Für die Endpunkte einer Linie bestimme die 4-Bit Codes:
  - Die Linie liegt vollständig außerhalb des Fensters, falls der Durchschnitt (AND-Verknüpfung) der Codes beider Endpunkte von Null verschieden ist.
  - Die Linie liegt komplett im Fenster, wenn beide Endpunkte den 4-Bit Code 0000 besitzen (OR-Verknüpfung ist Null).
  - Sonst:
    - Schneide alle Linien nacheinander mit den das Fenster begrenzenden Geraden.
    - Zerlege jede Linie in zwei Teile, die gemäß obiger Vorgehensweise kategorisiert werden.
    - Der außen liegende Teil wird sofort eliminiert.

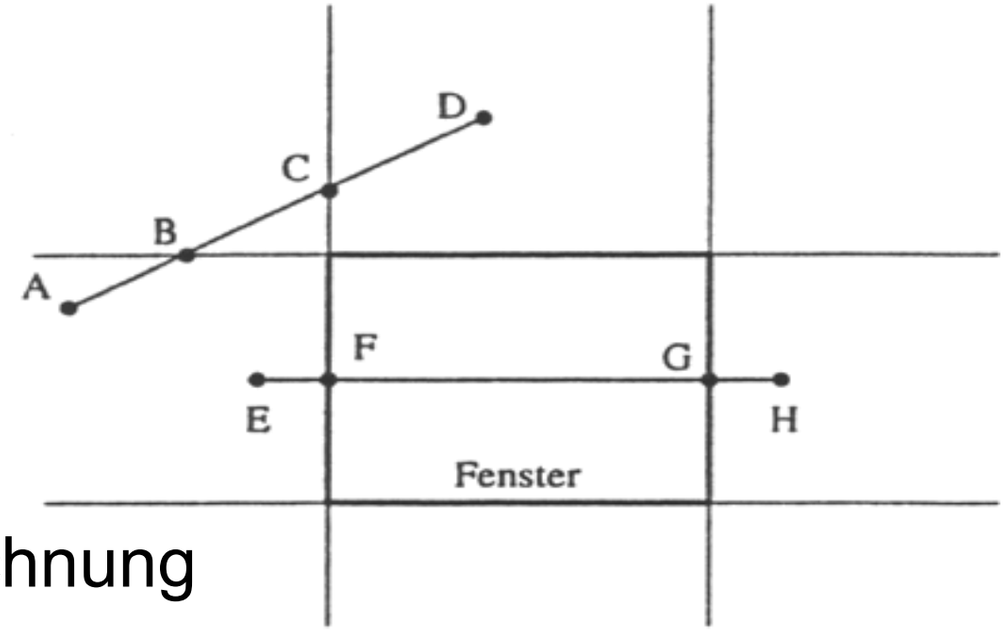
Cohen-Sutherland Line-Clipping (4)

- Beispiel: Linie AD
  - Codes 0001 und 1000 → Schnittpunktberechnung
  - Schnitt mit linker Fenstergrenze liefert C → Eliminiere AC
  - C und D liegen oberhalb des Fensters → Elimiiere CD



Cohen-Sutherland Line-Clipping (5)

- Beispiel: Linie EH



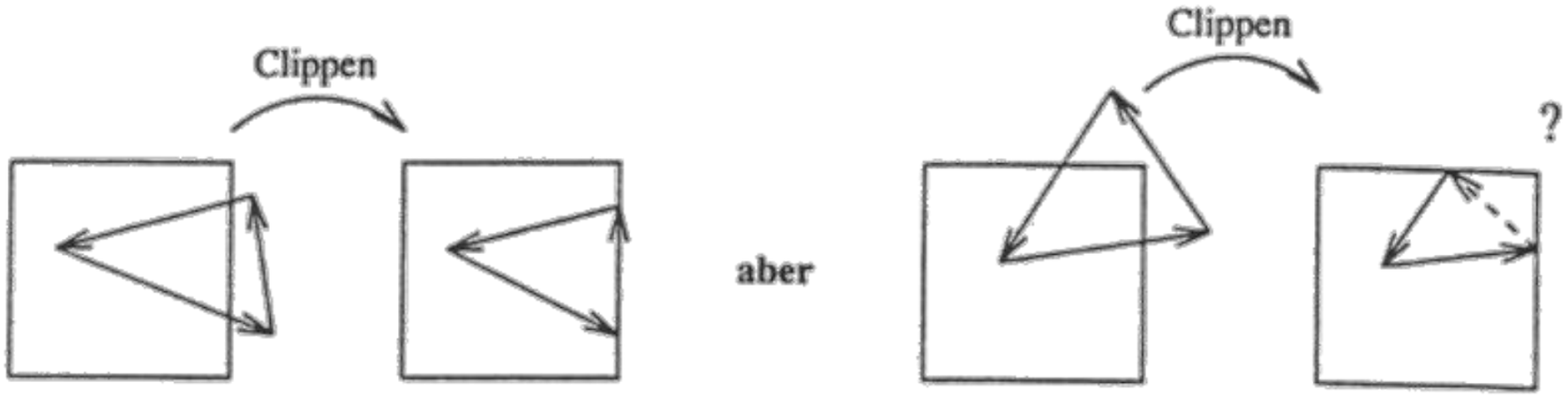
- Linie EH:
  - Codes 0001 und 0010 → Schnittberechnung
  - Schnitt mit linker Fenstergrenze liefert F → eliminiere EF
- Linie FH:
  - Codes 0000 und 0010 → Schnittberechnung
  - Schnitt mit rechter Fenstergrenze liefert G → eliminiere GH
- Linie FG:
  - Codes 0000 und 0000 → FG wird gezeichnet

Spezialfälle und Bescheunigungen

- Senkrechte / waagerechte Linien: nur y- / x- Grenzen testen und schneiden.
- Genau ein Endpunkt im Fenster: nur ein Schnitt mit den Fensterrand.
- Überflüssige Schnittoperationen aus den Bitcodes ermitten:
  - Jedes Bit korrespondiert genau zu einem Fensterrand
  - Nur die Fensterränder beachten, deren zugehörige Bits in den Endpunkt-Codes verschieden sind.

2D Clipping von Polygonen

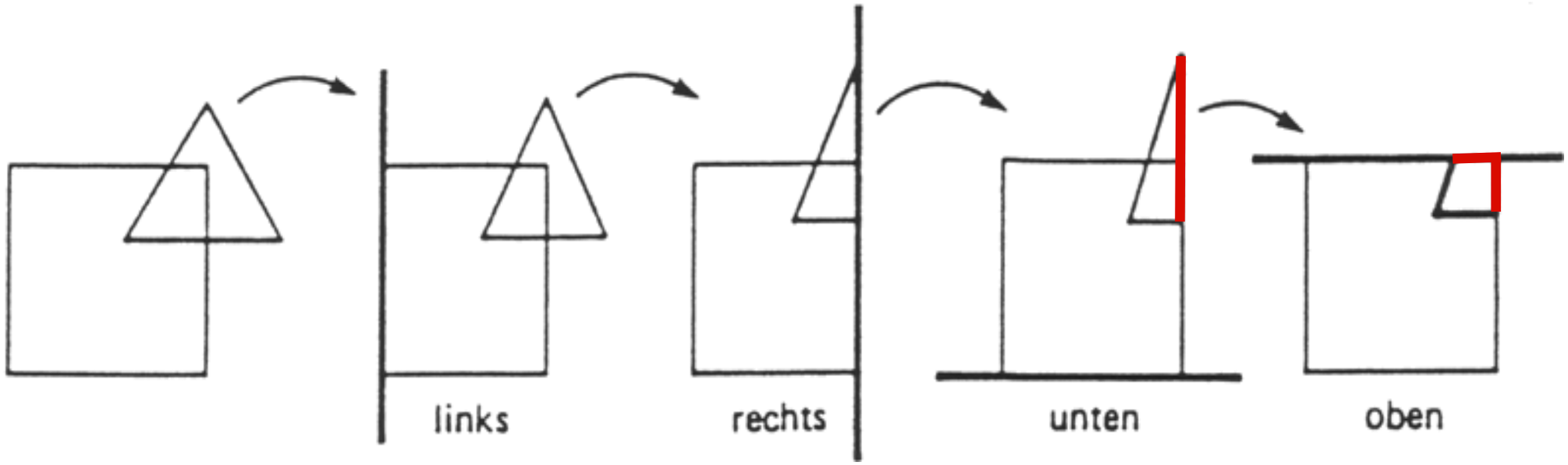
- Polygone sind in der Computergraphik als Begrenzung von Körpern wichtig.
- Nach der Projektion in die Bildebene müssen sie vor ihrer Darstellung am Fenster geclippt werden.
- Naiver Ansatz: Line-Clipping für jede Seite des Polygons.



- Problem: Clipping muss wieder geschlossene Polygone zurückgeben, also ggf. Teile des Fensterrandes enthalten.

Sutherland-Hodgman Polygon-Clipping

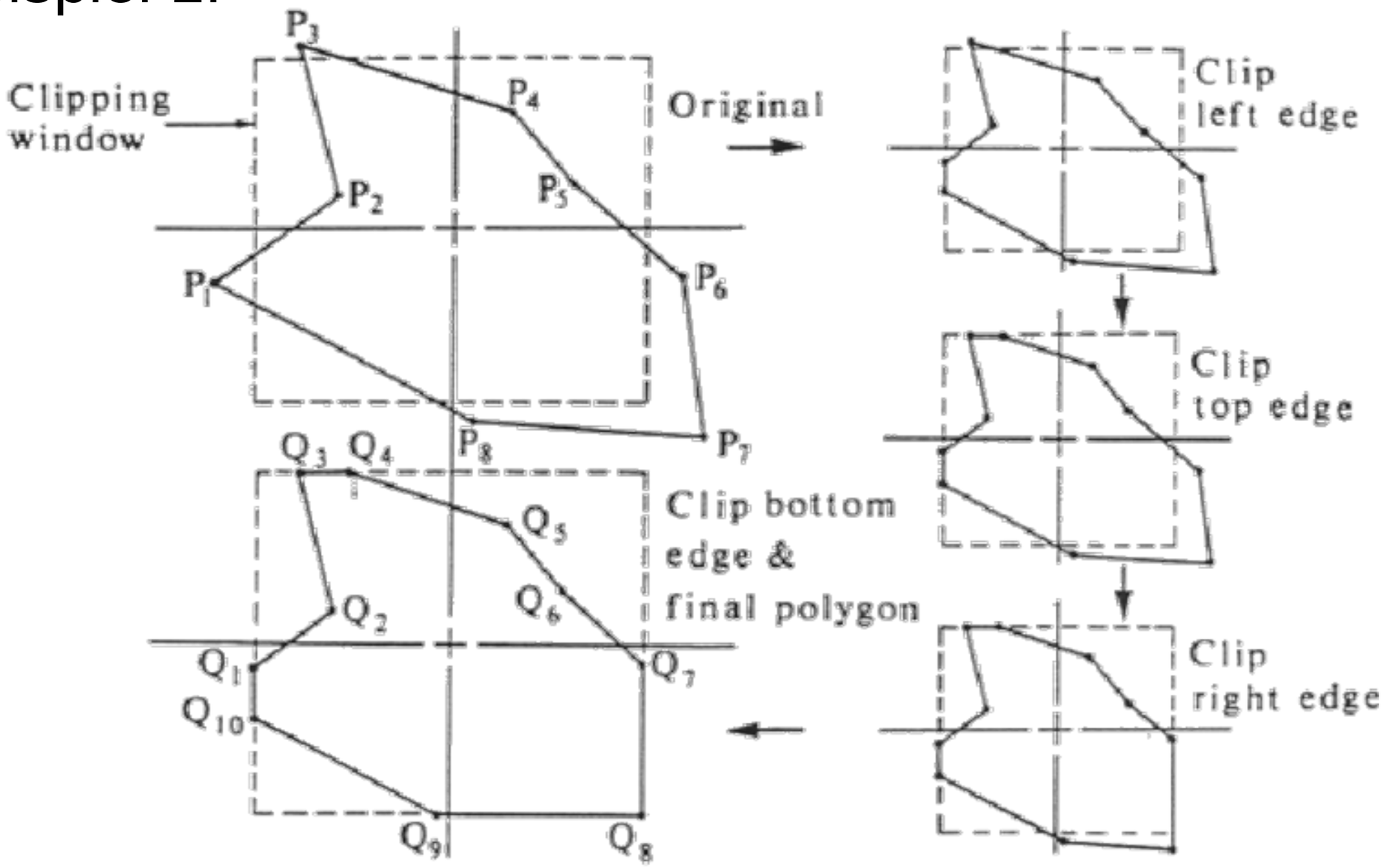
- Vollständiges Clippen des Polygons gegen eine Fensterseite.



- Die Zwischenergebnisse müssen gespeichert werden.
- Im 2. Bild von rechts wird eine Linie (rot) mit dem Fenster verschnitten, die es im ursprünglichen Polygon nicht gab.

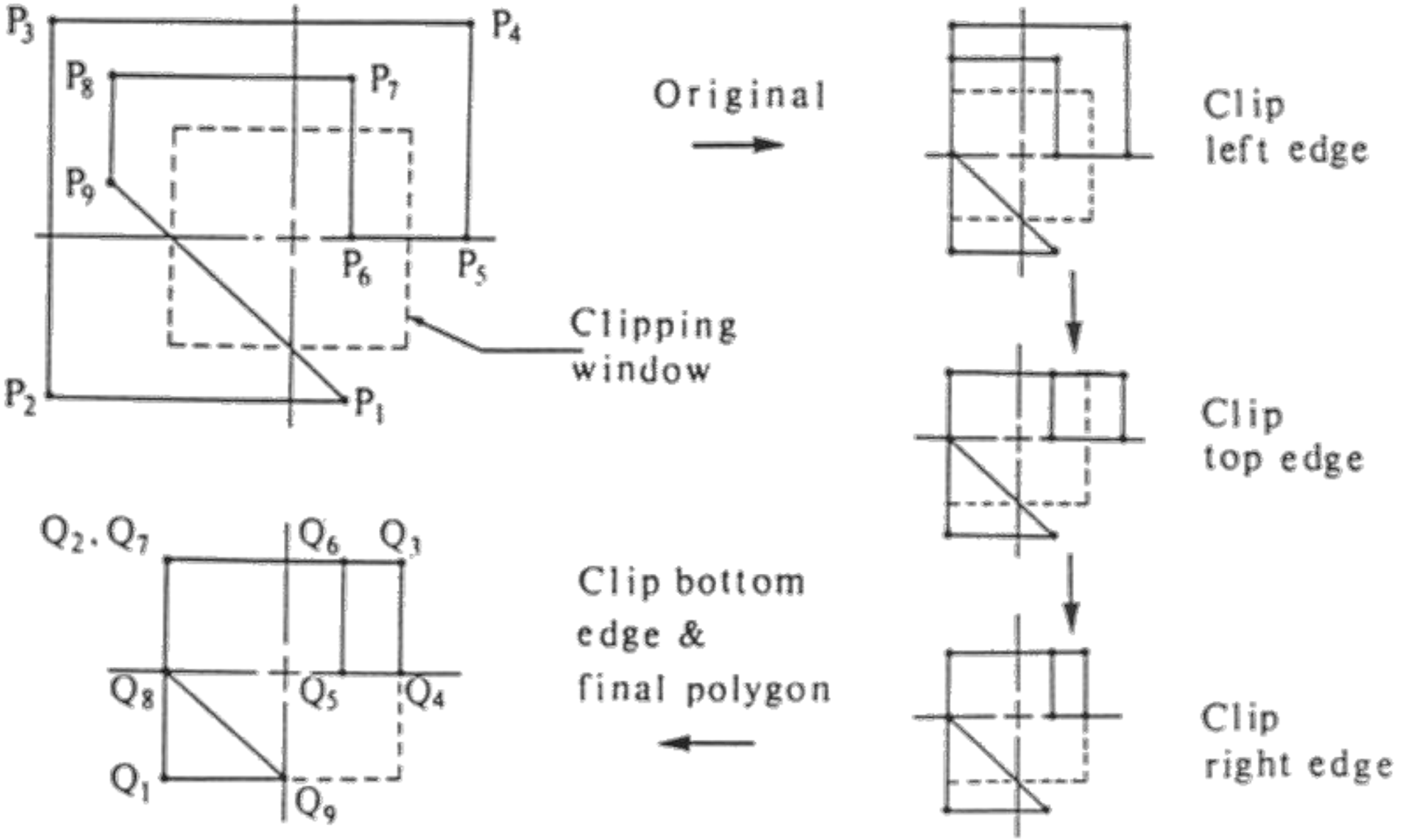
Sutherland-Hodgman Polygon-Clipping

- Beispiel 1:



Sutherland-Hodgman Polygon-Clipping

- Beispiel 2:



Lernziele

- Was ist der Unterschied zwischen Objekt-, Welt-, und Sicht-koordinaten?
- Wie können Translationen, Rotationen und Skalierungen im 2D und 3D beschrieben werden?
- Was sind affine Transformationen?
- Was sind homogene Koordinaten?
- Was ist eine perspektivische Projektion?
- Was ist eine Parallelprojektion?
- Was ist das Frustrum?
- Was ist Windowing?
- Wie arbeiten Clipping-Algorithmen für Linien und Polygone?