

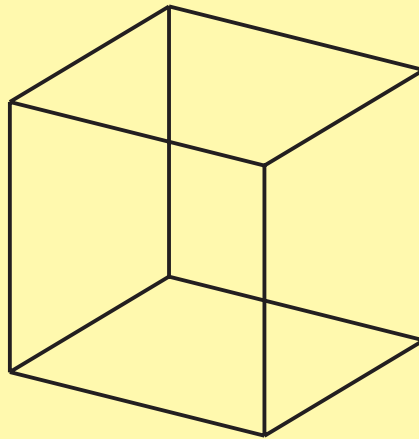
HLHSR-Algorithmen

(Hidden Line Hidden Surface Removal)

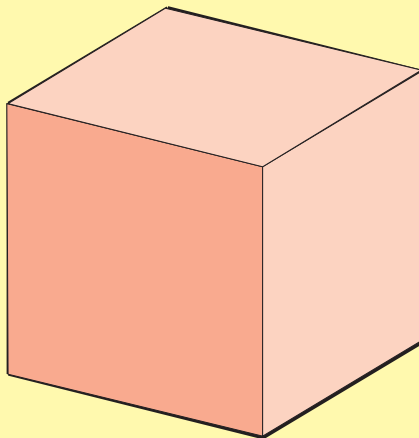
1. Vorbemerkungen
2. Klassifizierung
3. Grundfunktionen für Sichtbarkeitsverfahren
 - 3.1 Perspektivische Projektion
 - 3.2 Entfernen von Rückseiten (Back Face Culling)
 - 3.3 Punktposition
 - 3.4 Ausdehnungsbereiche und Begrenzungsvolumen, Minmax-Test
4. Z-Buffer- und A-Buffer-Algorithmus
5. EXACT-Algorithmus
6. Prioritätslisten-Algorithmus
7. Flächenunterteilungs-Algorithmus
8. Weitere Algorithmen (Auswahl)

Interpretationsmöglichkeiten eines Drahtmodells

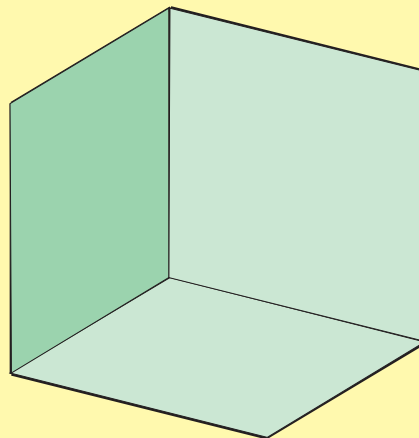
Drahtmodell



Flächenmodell
(Interpretationsmöglichkeiten des Drahtmodells)

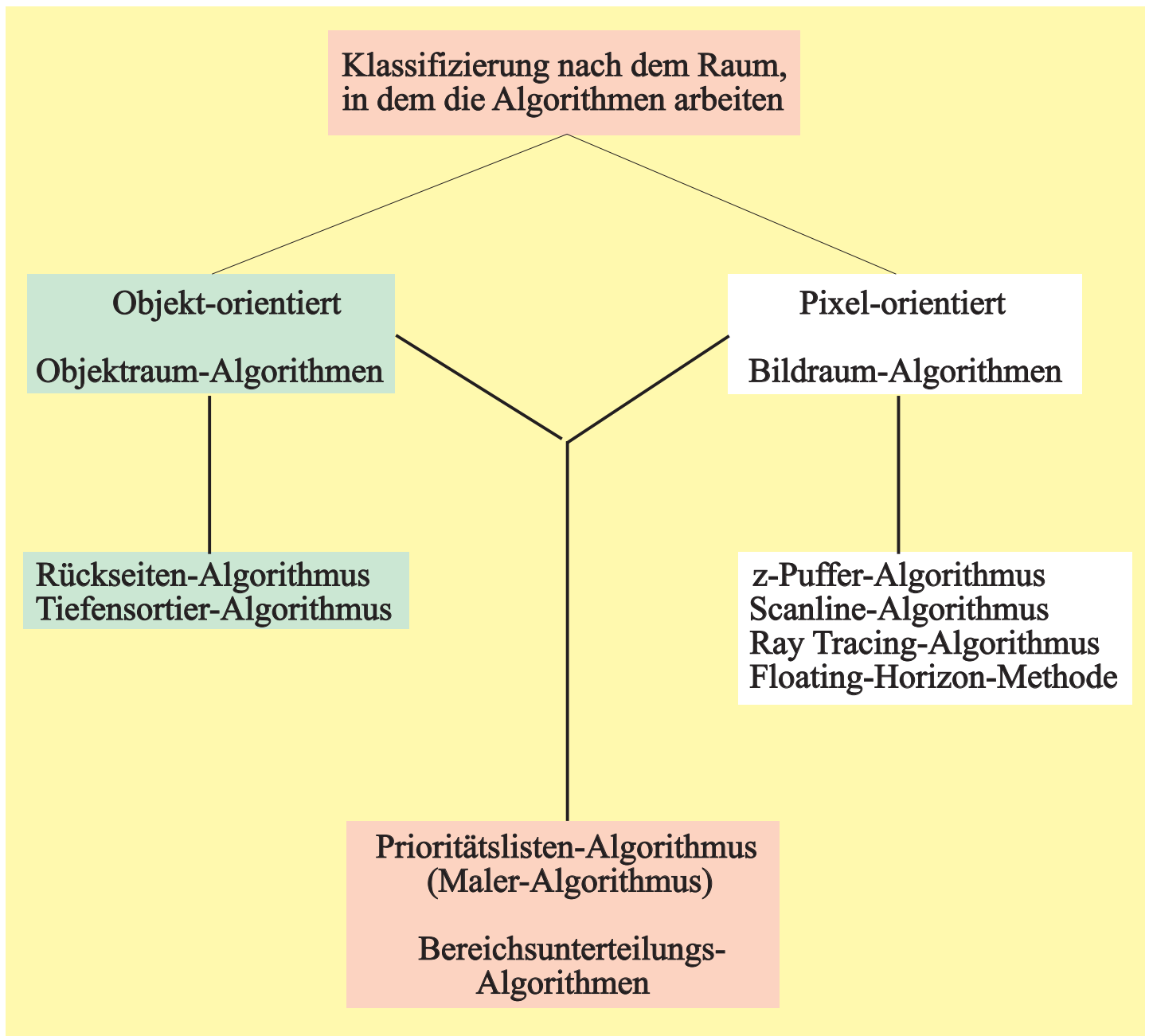


Version 1



Version 2

Klassifizierung, Teil 1



Klassifizierung, Teil 2

Objektraum-Algorithmen

Algorithmus:

```
for jedes Objekt der Szene do  
  begin  
    Bestimmung der nicht verdeckten Objektteile  
    {durch sich selbst oder andere Objekte};  
    Ausgabe dieser Teile in der entsprechenden Intensität {Farbe}  
  end {Objektraum-Algorithmus}
```

Charakteristik:

- Objektraum-Algorithmen arbeiten im Weltkoordinatensystem.
- Die Genauigkeit der Berechnung entspricht der Maschinengenauigkeit. Lokale Skalierungen erfordern deshalb keine erneute Berechnung.

Klassifizierung, Teil 3

Bildraum-Algorithmen

Algorithmus:

```
for jedes Bildschirmpixel do  
  begin  
    Bestimmung des dem "Beobachter" am nächsten gelegenen  
    Objekts {Pixel};  
    Pixelintensität {Farbe} entsprechend setzen  
  end {Bildraum-Algorithmus}
```

Charakteristik:

- Bildraum-Algorithmen arbeiten im Gerätekoordinatensystem.
- Die Genauigkeit der Berechnung entspricht der Bildschirmauflösung. Lokale Skalierungen erfordern eine erneute Berechnung.

Klassifizierung, Teil 4

HLHSR-Algorithmen

Klassifizierung nach der Art der Nutzung von Kohärenz

Kanten-Kohärenz

Seitenflächen-Kohärenz

Objekt-Kohärenz

Abtastzeilen-Kohärenz

Einzelbild-Kohärenz

Geometrische Kohärenz

Klassifizierung, Teil 5

Nutzung von Kohärenz

Kohärenz: Gegenseitige Abhängigkeiten einzelner Bildbestandteile untereinander.

Kanten-Kohärenz: Die Sichtbarkeit einer Kante ändert sich nur, wenn sie andere Kanten kreuzt.

Seitenflächen-Kohärenz: Wegen der im Vergleich zum Gesamtbild geringen Größe polyedrischer Seitenflächen trifft es häufig zu, dass die gesamte Seitenfläche sichtbar ist, wenn nur ein Teil von ihr sichtbar ist.

Objekt-Kohärenz: Die Sichtbarkeit eines Objekts lässt sich häufig durch die Untersuchung des umschreibenden Körpers ermitteln.

Klassifizierung, Teil 6

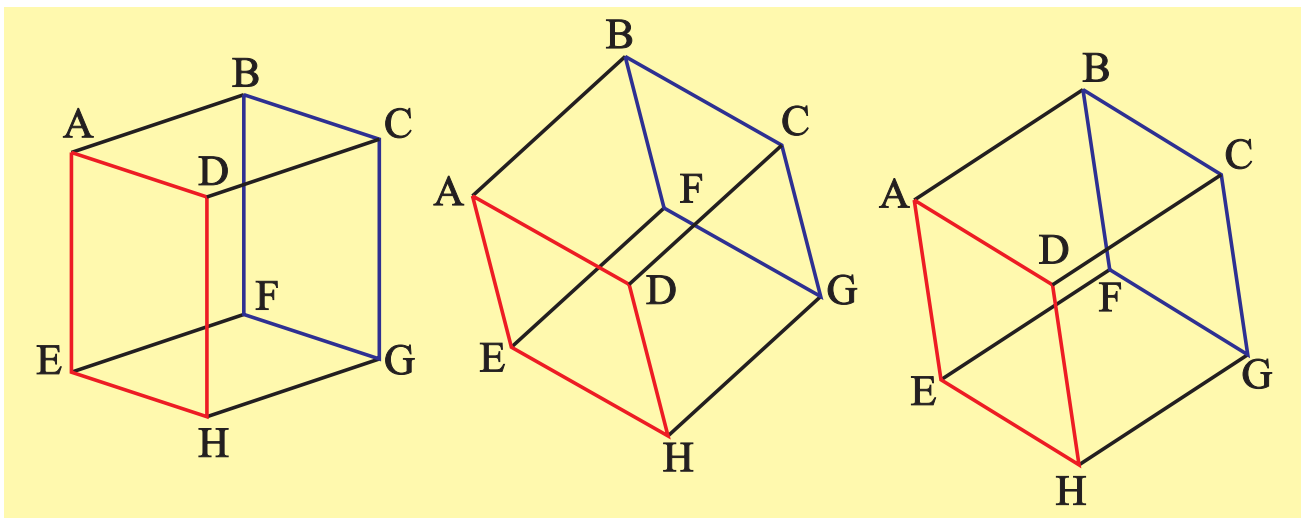
Nutzung von Kohärenz (Fortsetzung)

Einzelbild-Kohärenz: Ein Abbild ändert sich von Einzelbild zu Einzelbild nur wenig.

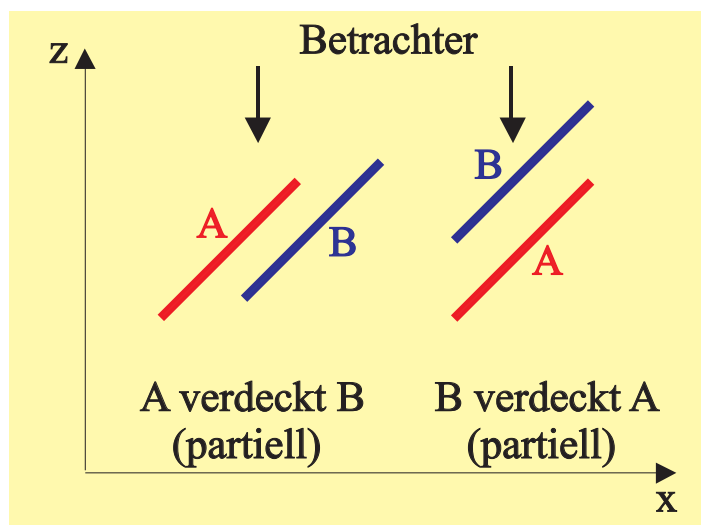
Geometrische Kohärenz: Die Zahl der möglichen sichtbaren bzw. unsichtbaren Konfigurationen für jede Spitze ist beschränkt.

Abtastzeilen-Kohärenz: Segmente, die innerhalb einer Abtastzeile sichtbar sind, sind mit großer Wahrscheinlichkeit auch in der nächsten Abtastzeile sichtbar.

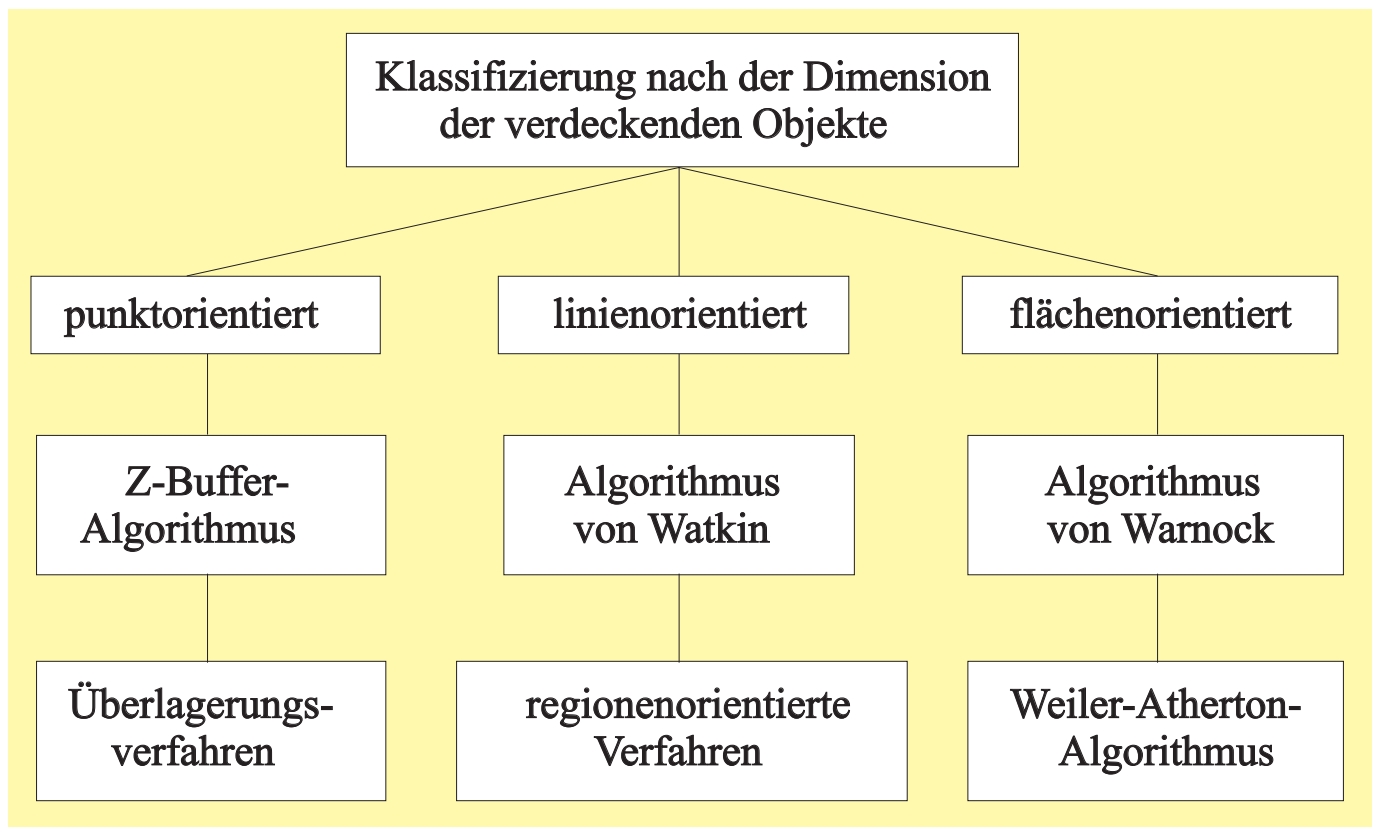
Bildfolge-Kohärenz



Sichtbarkeitsprobleme



Punkt-, linien- und flächenorientierte HLHSR-Verfahren



Sichtbarkeit bei perspektivischen Projektionen

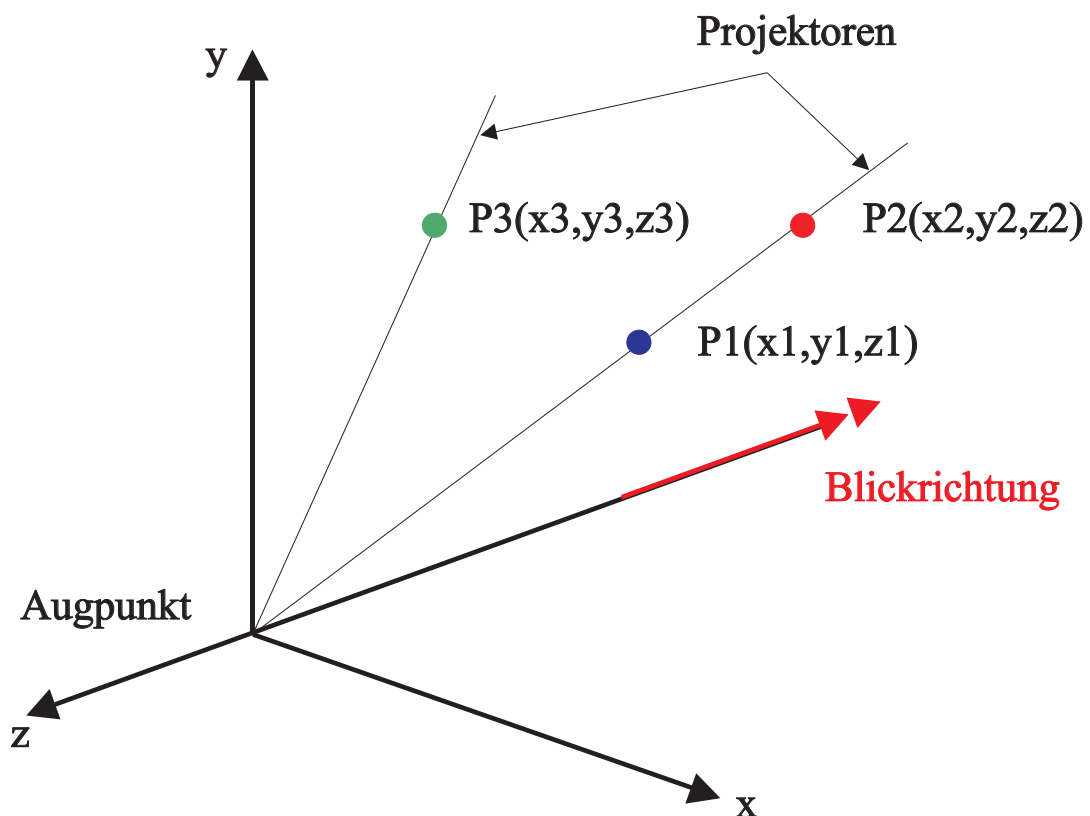
Verdeckt einer der Punkte den anderen?
(Liegen beide Punkte auf dem gleichen Projektor?)

ja

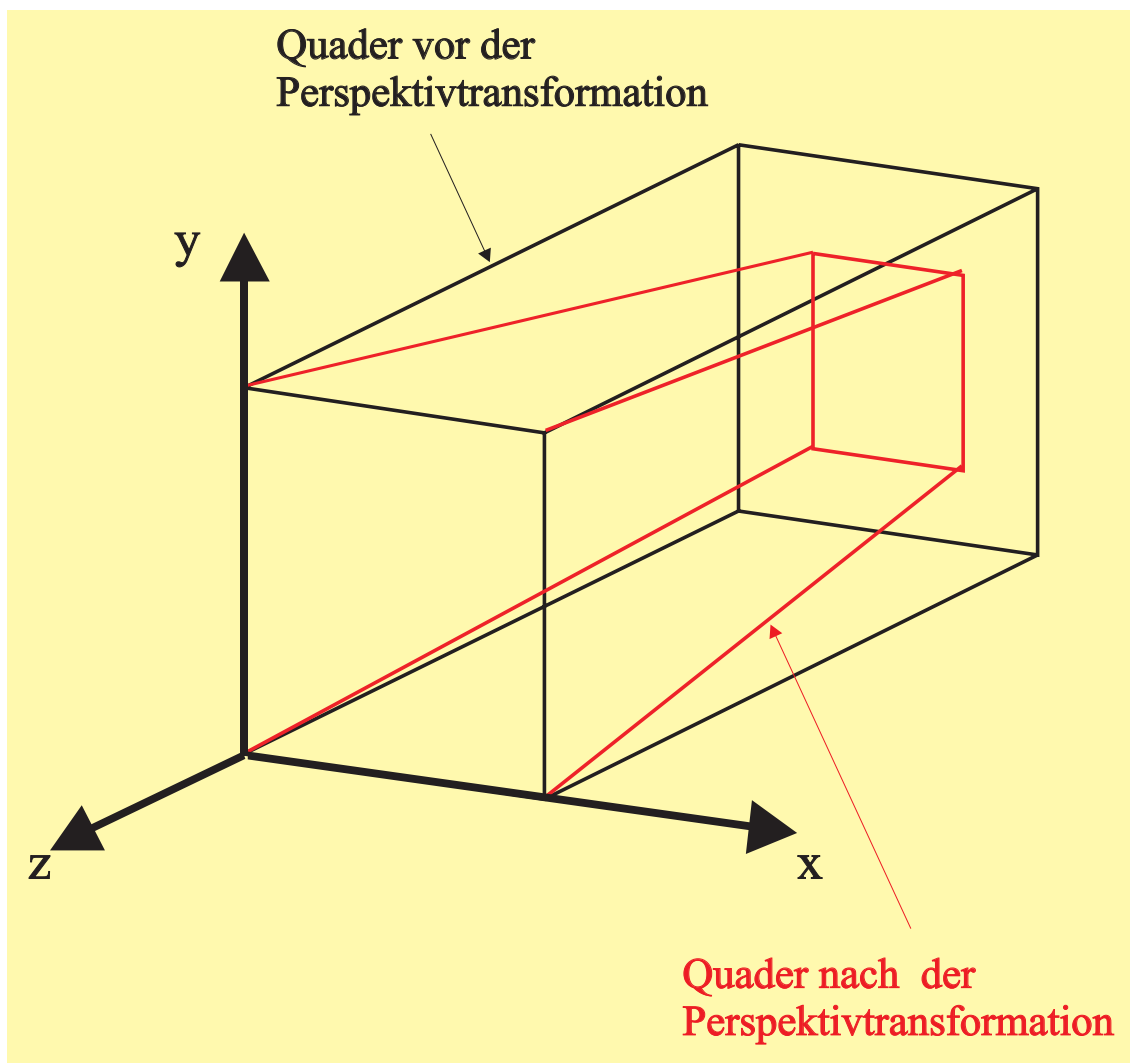
nein

Tiefenvergleich
von z_1 und z_2

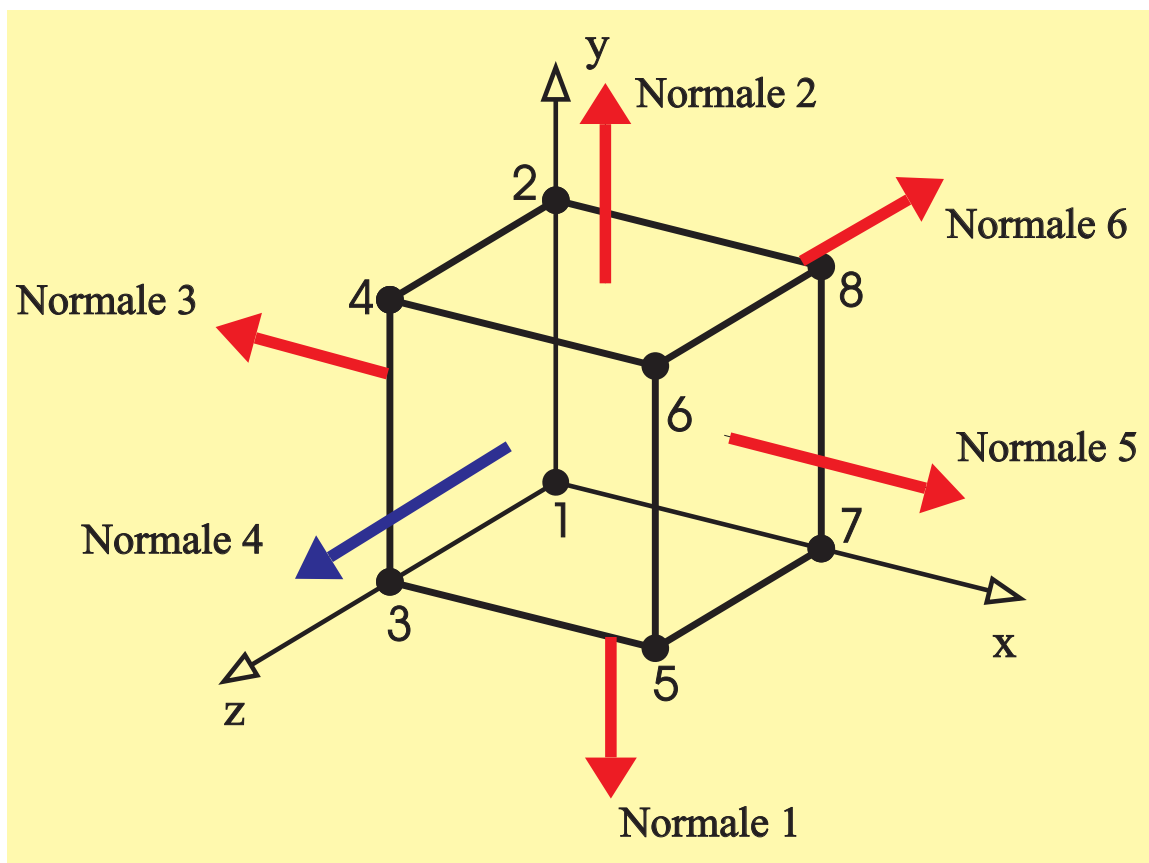
keine
Operation



Perspektivtransformation



Rückseitenalgorithmus, Teil 1



Rückseitenalgorithmus, Teil 2

Normalenvektor $\vec{N} = (N_x, N_y, N_z)^T$

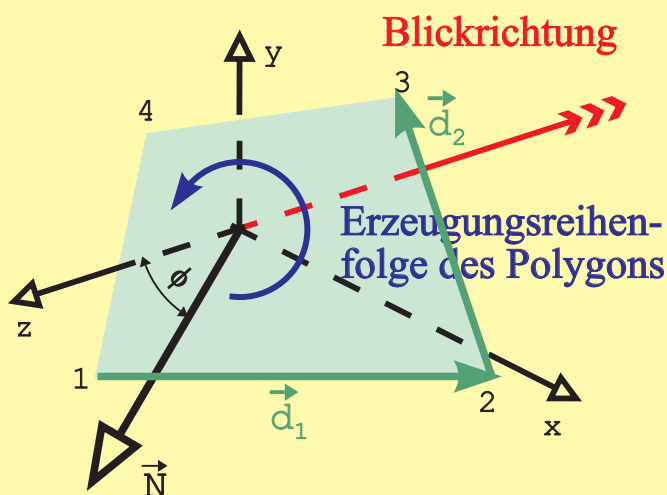
Vektorprodukt

$$N_x = d_{1y} \cdot d_{2z} - d_{1z} \cdot d_{2y}$$

$$N_y = d_{1z} \cdot d_{2x} - d_{1x} \cdot d_{2z}$$

$$N_z = d_{1x} \cdot d_{2y} - d_{1y} \cdot d_{2x}$$

Methode von Nevell



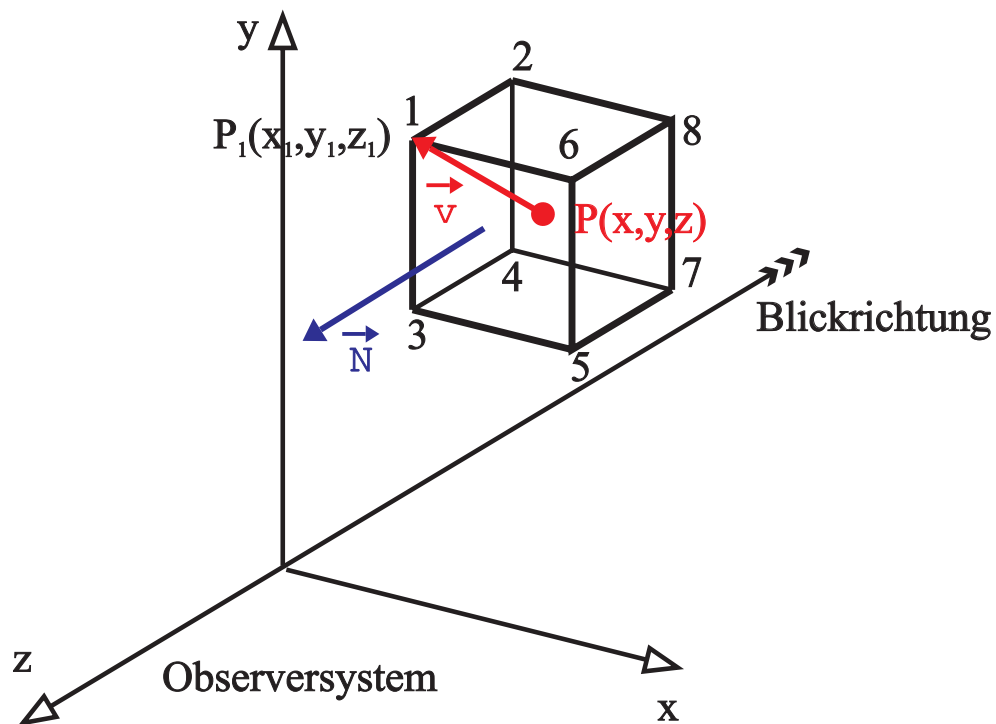
$$N_x = \sum_{i=1}^n (y_i - y_j) \cdot (z_i + z_j)$$

$$N_y = \sum_{i=1}^n (z_i - z_j) \cdot (x_i + x_j)$$

$$N_z = \sum_{i=1}^n (x_i - x_j) \cdot (y_i + y_j)$$

$$j = \begin{cases} 1 & \text{für } i=n \\ i+1 & \text{sonst} \end{cases}$$

Rückseitenalgorithmus, Teil 3



$$P(x, y, z) = \frac{1}{n} \sum_{i=1}^n P_i(x_i, y_i, z_i)$$

n = Anzahl der Körpereckpunkte

$$\vec{v} = (\vec{P}_k - \vec{P}) = (x_k - x, y_k - y, z_k - z)^T$$

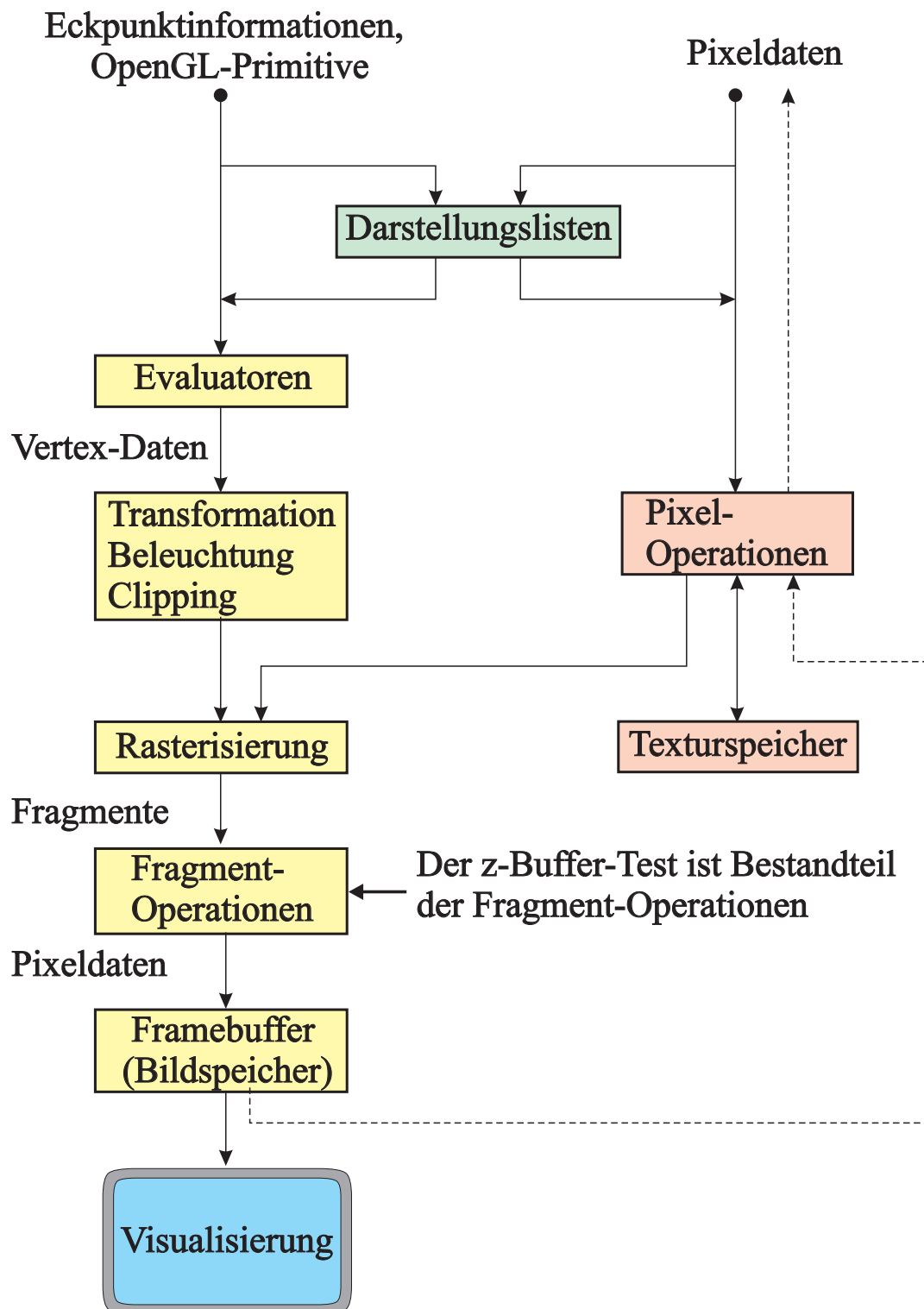
Bedingung für eine Außennormale: $\vec{N} \cdot \vec{v} > 0$

z-Buffer-Algorithmus

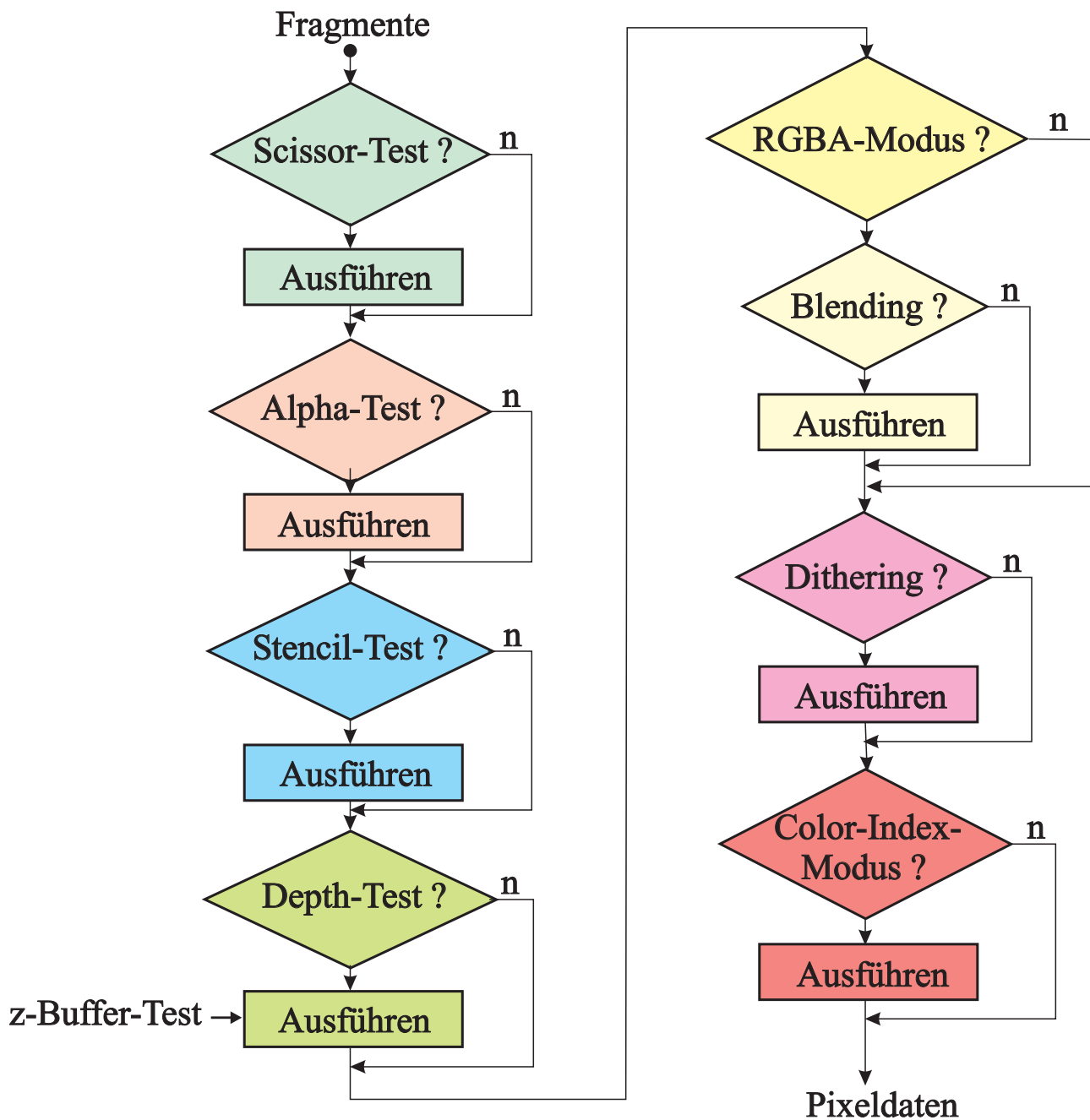
```
for alle Bildschirmpixel do  
  begin  
    Initialisieren des Bildspeichers mit der Hintergrundfarbe;  
    Initialisieren des z-Puffers mit dem am entferntesten Wert  $z_{\max}$ ;  
  end;
```

```
for alle Objekte der Szene do  
  begin  
    Bestimmung des z-Wertes für Pixel P(x,y) des Objektes;  
    if  $|z| < |z_{\max}|$  then  
      begin  
         $z_{\max} := z$ ;  
        Speichern der zugehörigen Farbe in den Bildspeicher  
      end;  
    end;
```

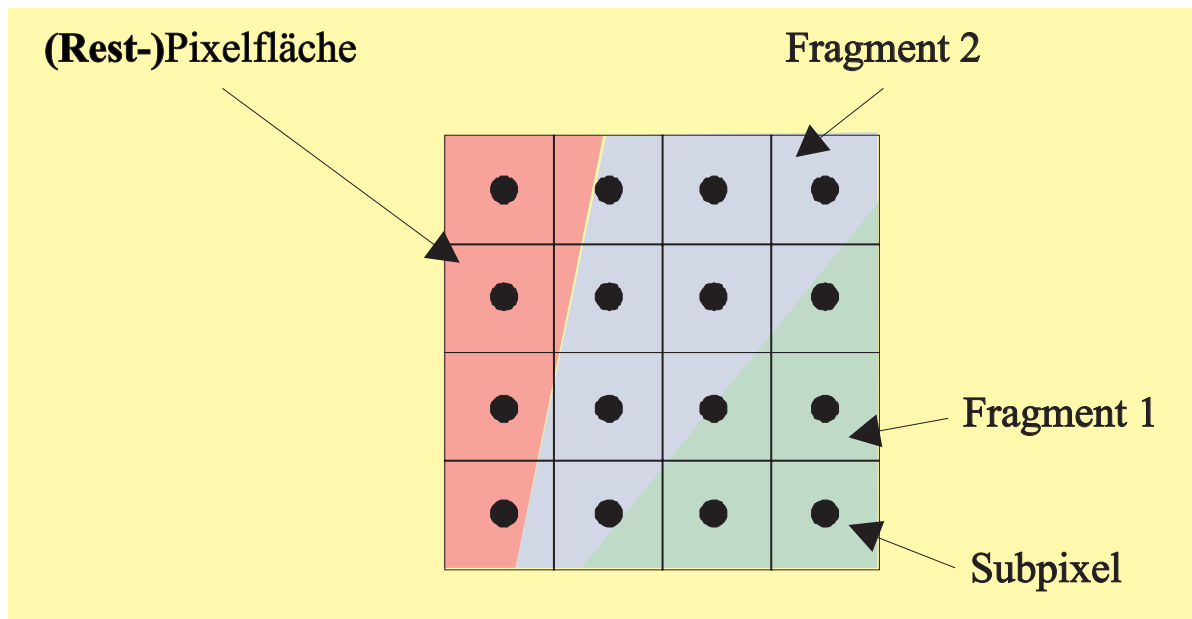
Renderingpipeline der OpenGL



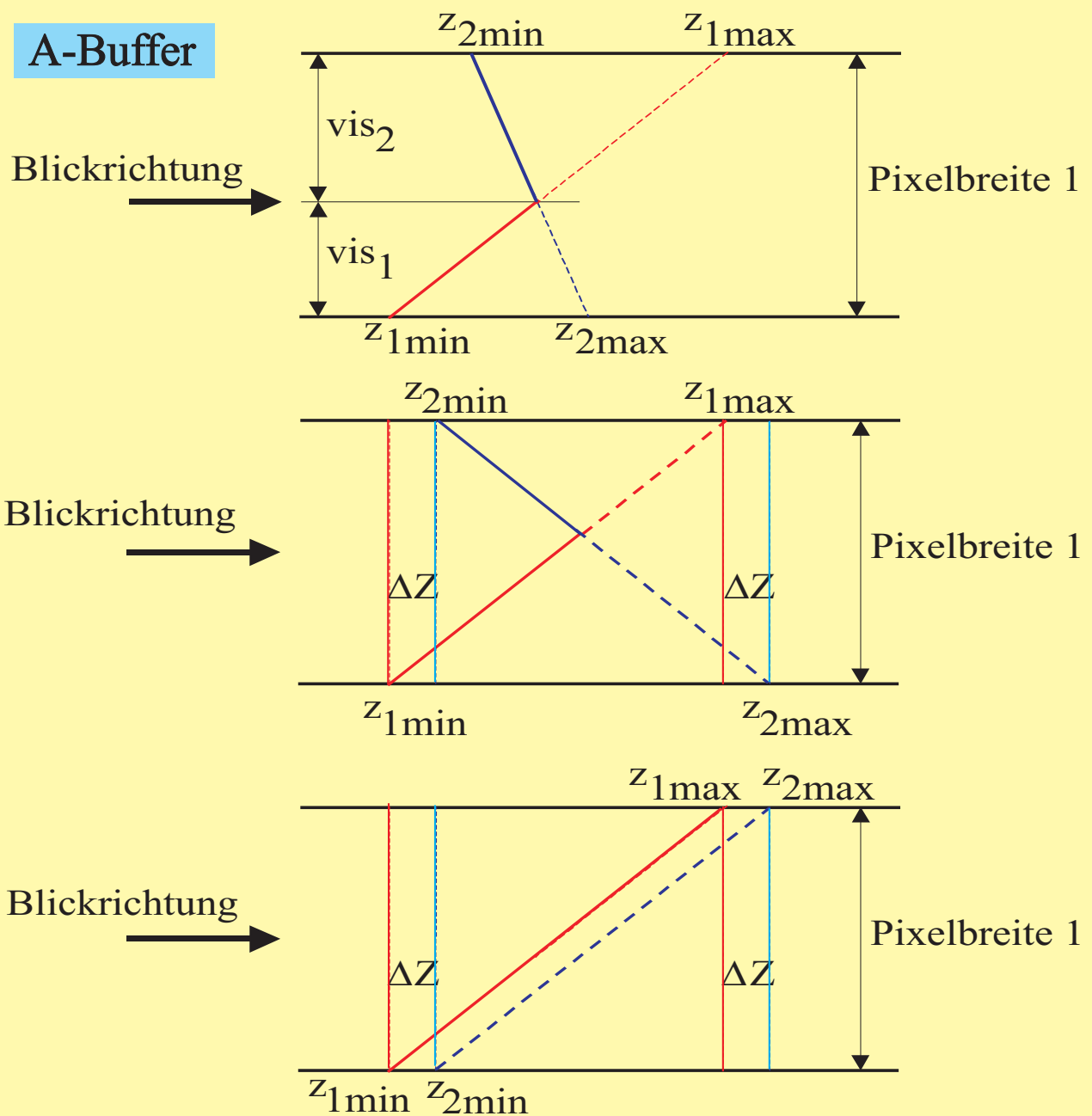
Fragmentoperationen der OpenGL



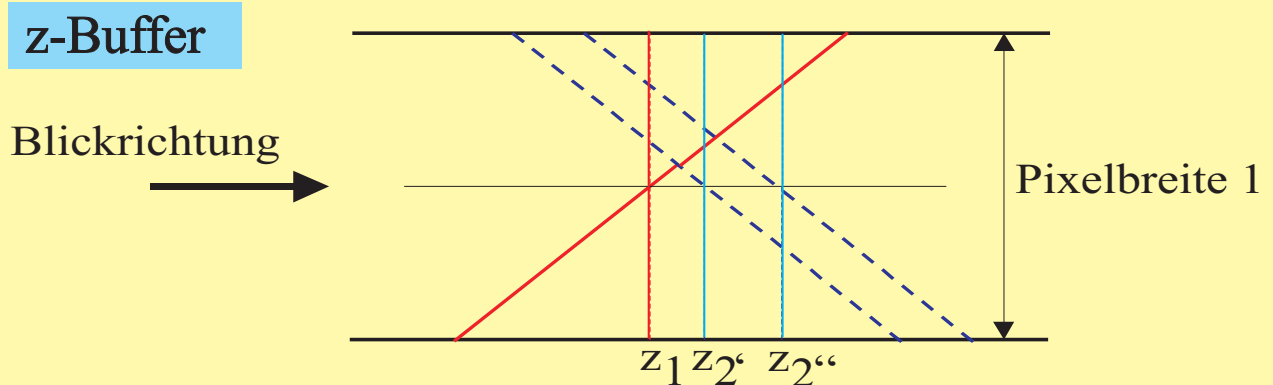
Fragmentierung eines Pixels



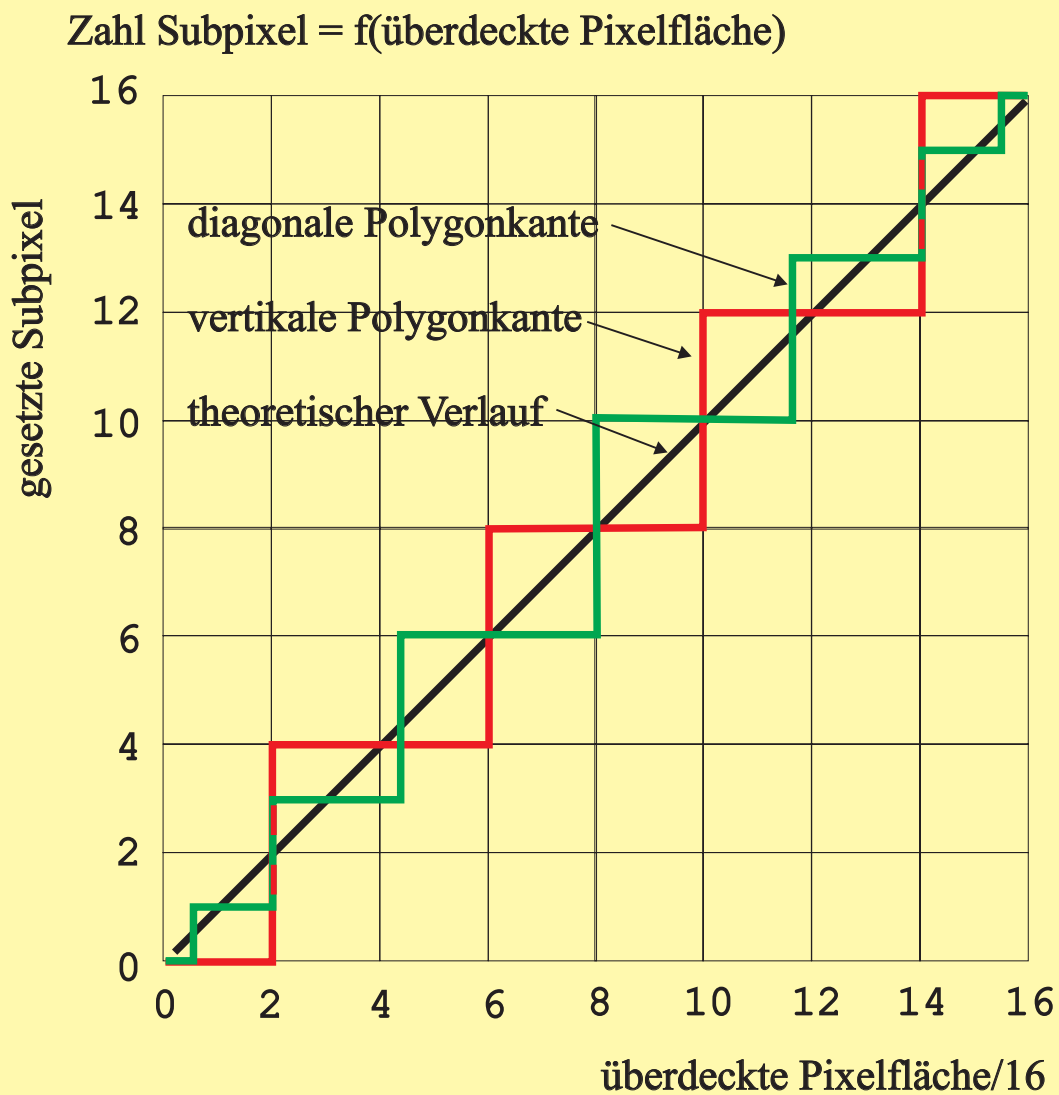
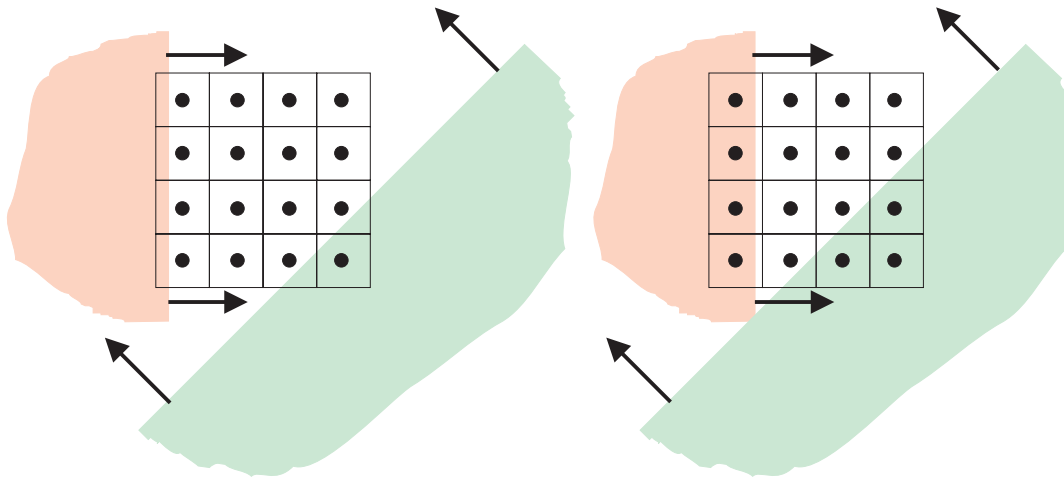
A-Buffer



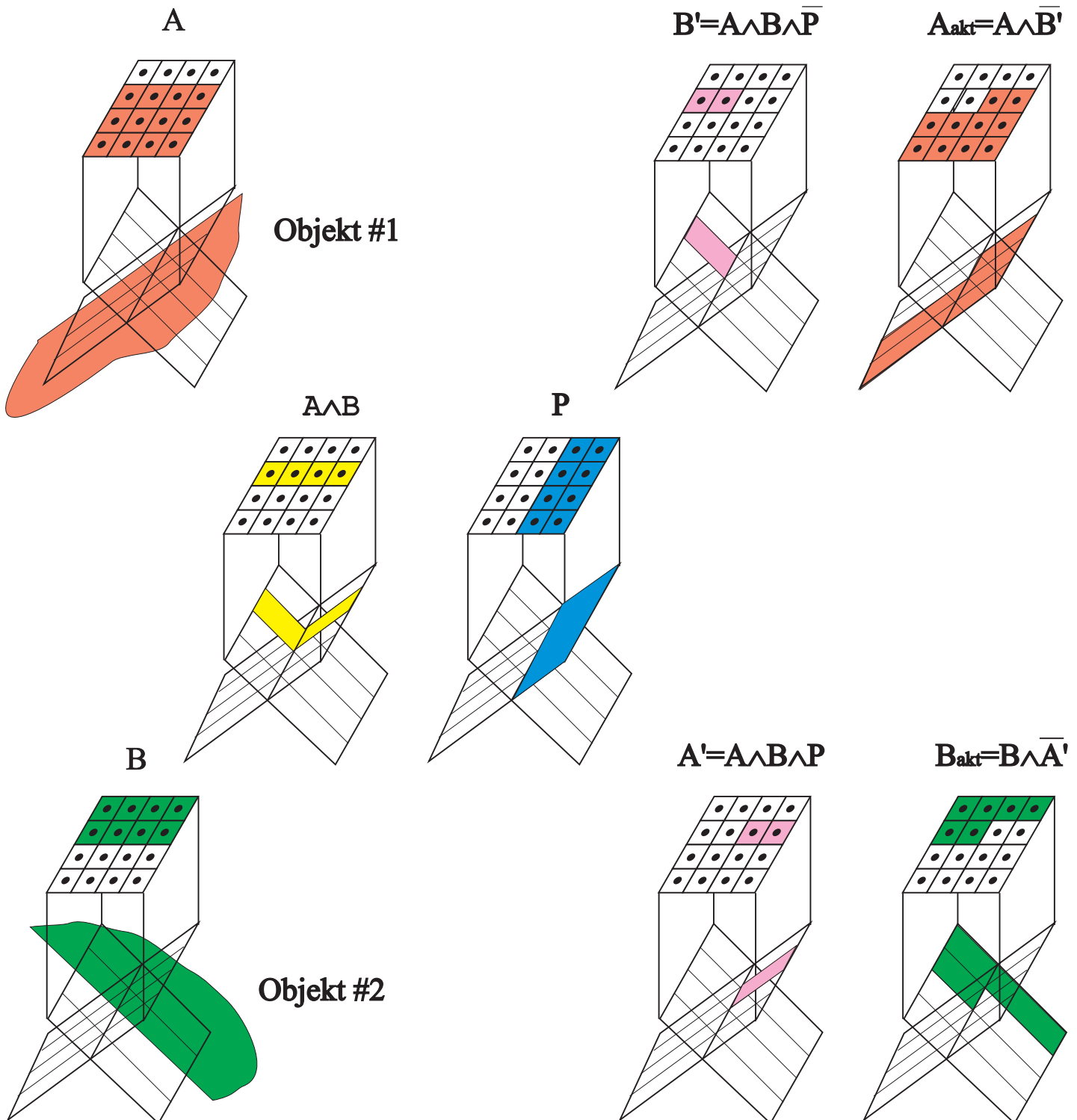
z-Buffer



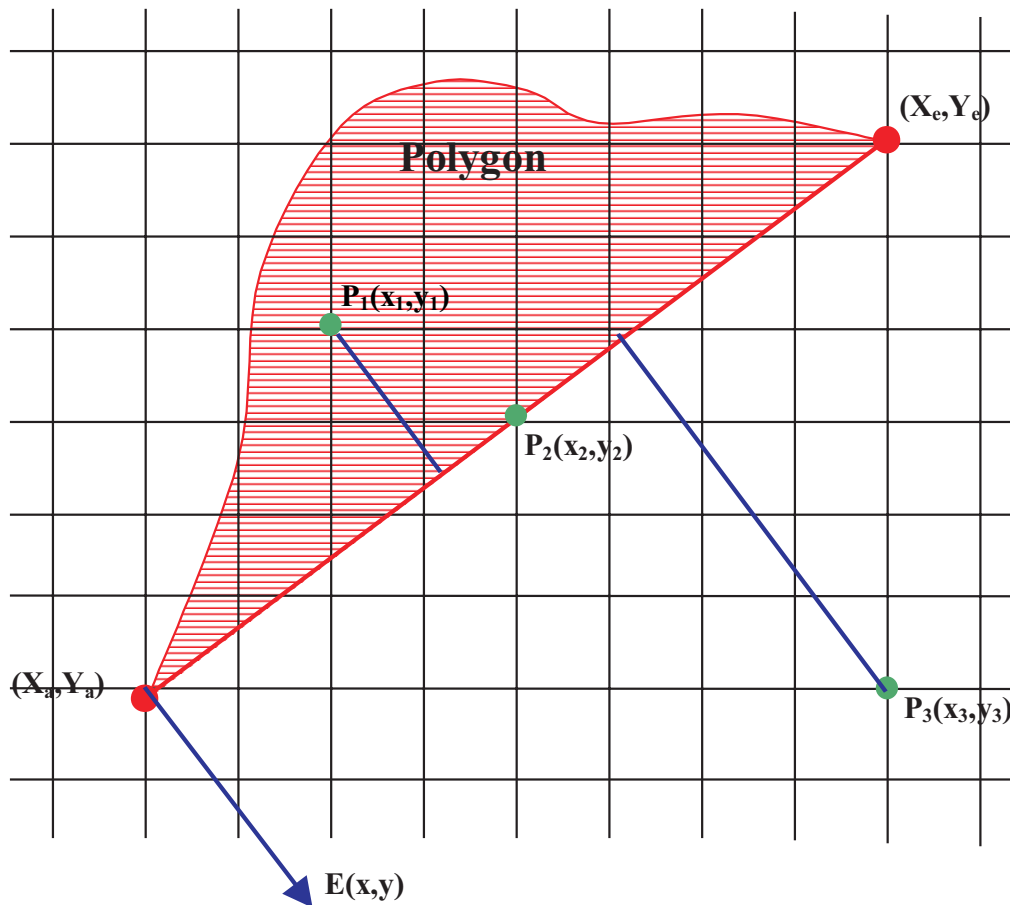
Reihenfolge der Überdeckung der Subpixel



Wirkung der Prioritätsmaske P auf die Subpixelmasken A und B

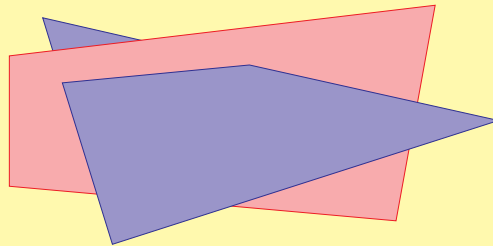


Kantenfunktion $E(x,y)$

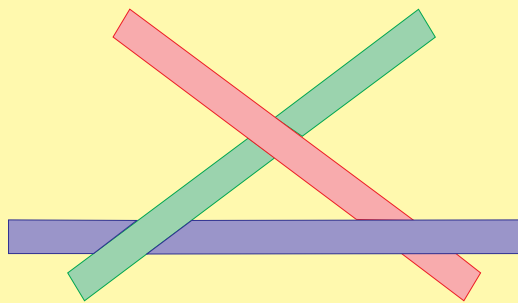


$$E(x,y) = [\Delta Y \cdot (x - X_a) - \Delta X \cdot (y - Y_a)] / \sqrt{\Delta X^2 + \Delta Y^2}$$

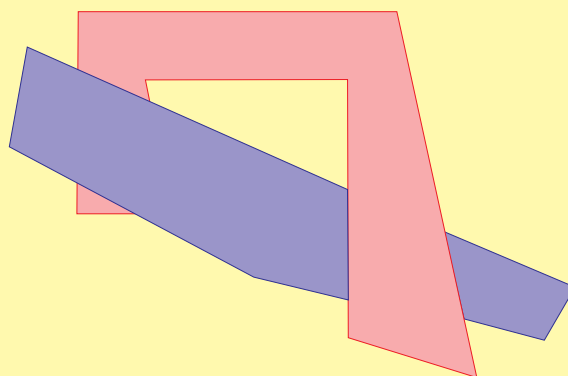
Prioritätslisten-Algorithmus



Durchdringung



Zyklische Überlappung



Wechselnde Überlappung

Prioritätslisten-Algorithmus, Teil 1

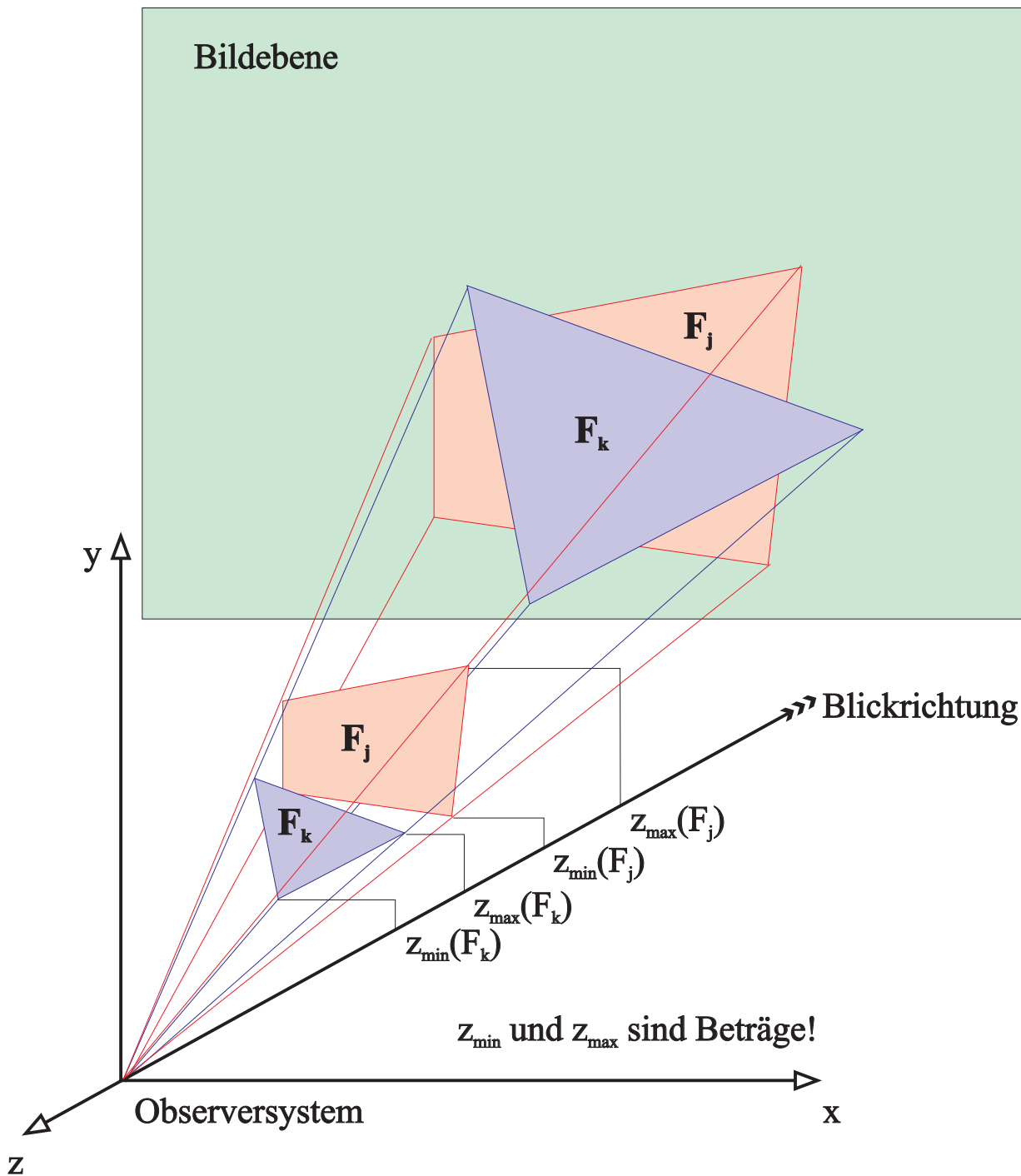
Teststrategie

Voraussetzung: Die Flächen F_i sind nach ihrem betragsmäßig größten Abstand $z_{\max}(F_i)$ sortiert.

1. Überlappen sich zwei Flächen F_j und F_k in z-Richtung?
 2. Wenn ja, so entscheiden die folgenden vier Bedingungen, ob eine Umordnung der Flächen F_j und F_k in der Liste erfolgen muß.
 - 2.a: Die umschließenden achsenparallelen Rechtecke der projizierten Flächen schneiden sich nicht.
 - 2.b: Alle Eckpunkte der Fläche F_k liegen *vor* der durch F_j aufgespannten Ebene.
 - 2.c: Alle Eckpunkte der Fläche F_j liegen *hinter* der durch F_k aufgespannten Ebene.
 - 2.d: Die Projektionen der beiden Flächen F_j und F_k schneiden sich nicht.
- Ist *eine* der Bedingungen 2.a bis 2.d erfüllt, so muß *keine* Umordnung durchgeführt werden.

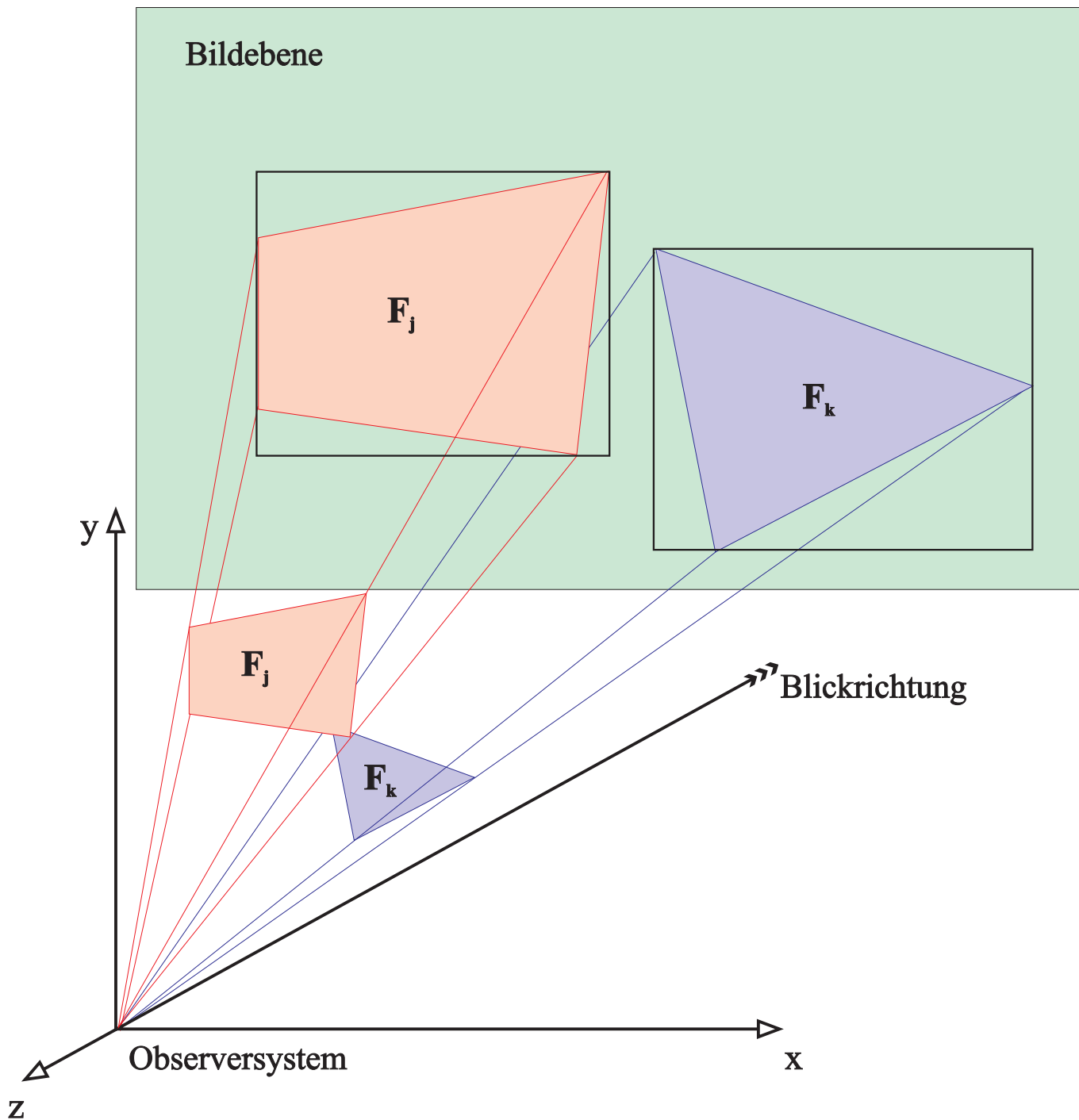
Prioritätslisten-Algorithmus, Teil 2

Test 1: $z_{\min}(F_j) > z_{\max}(F_k)$?



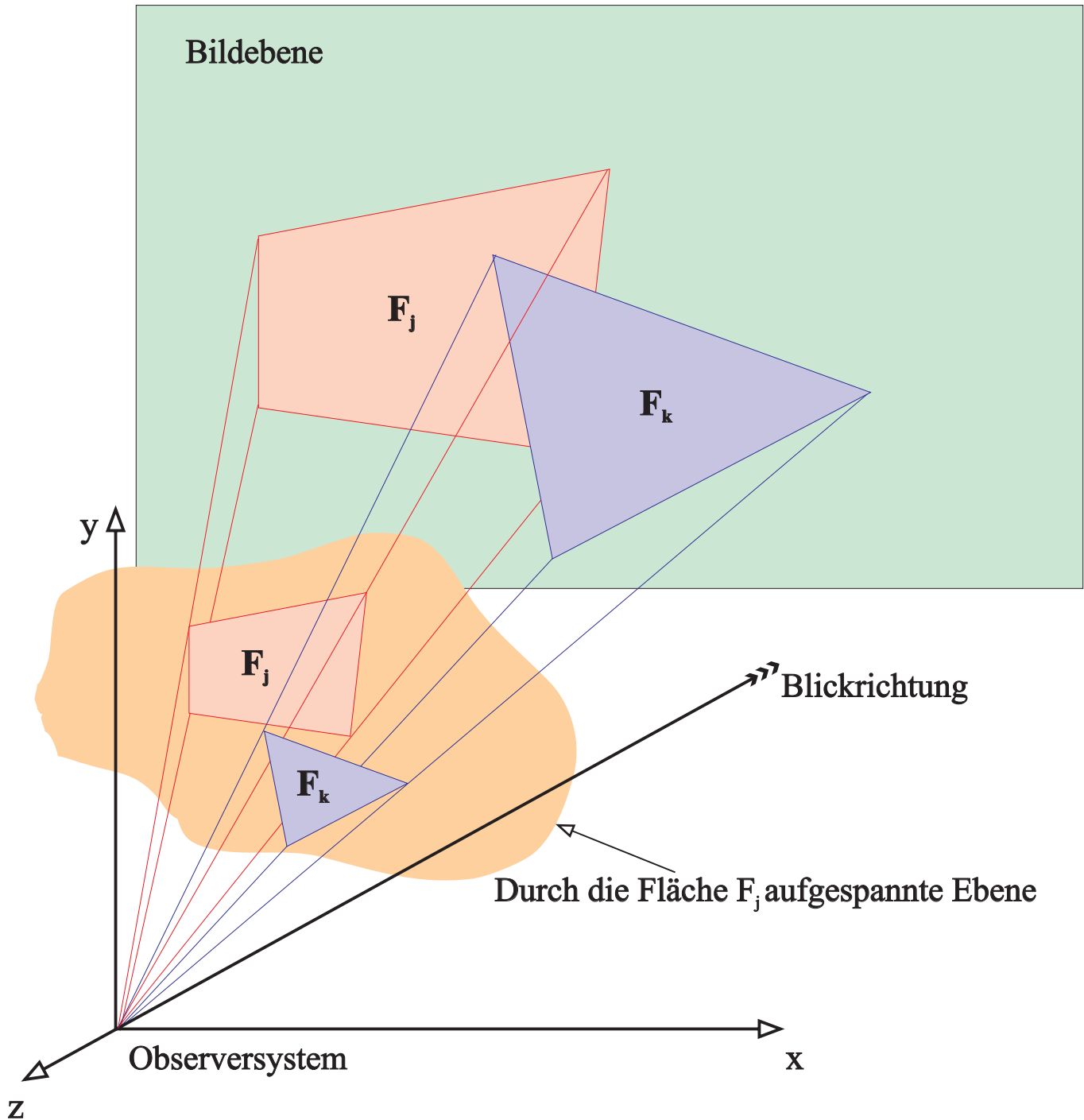
Prioritätslisten-Algorithmus, Teil 3

Test 2a: Schneiden sich die umschließenden Rechtecke?



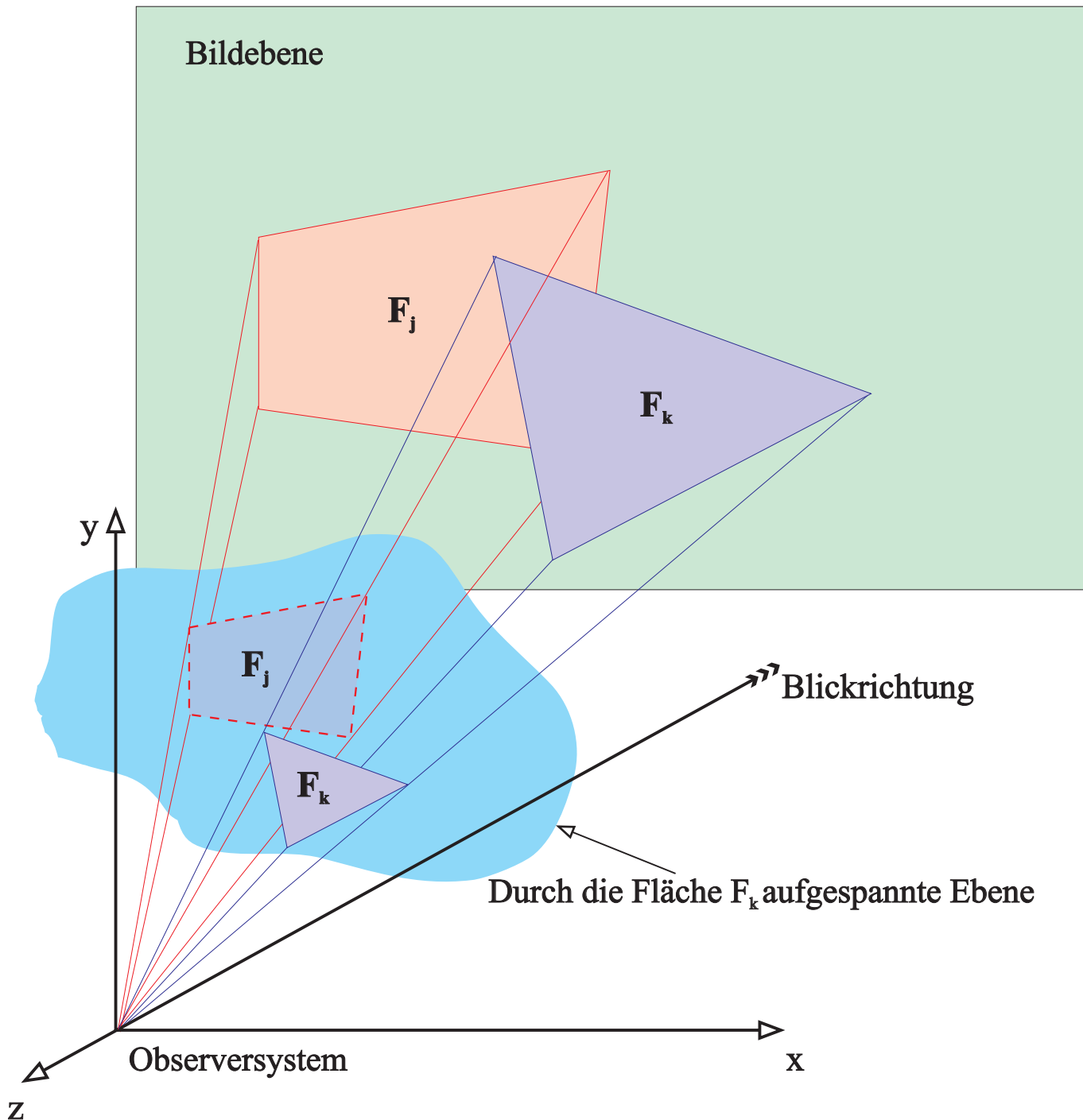
Prioritätslisten-Algorithmus, Teil 4

Test 2b: Liegen alle Eckpunkte der Fläche F_k vor der durch die Fläche F_j aufgespannten Ebene?



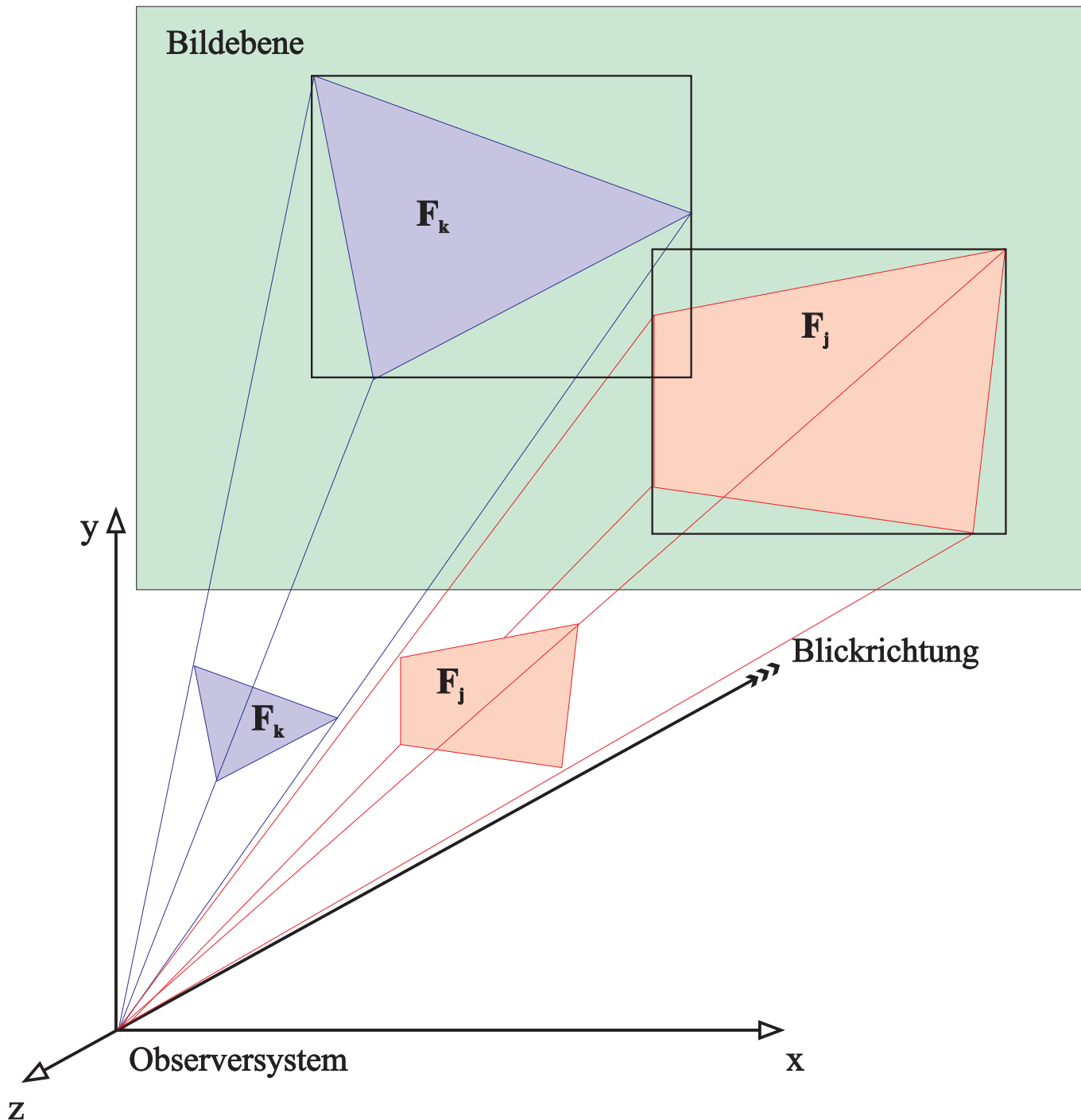
Prioritätslisten-Algorithmus, Teil 5

Test 2c: Liegen alle Eckpunkte der Fläche F_j hinter der durch die Fläche F_k aufgespannten Ebene?



Prioritätslisten-Algorithmus, Teil 6

Test 2d: Schneiden sich die umschließenden Rechtecke?



Prioritätslisten-Algorithmus, Teil 7

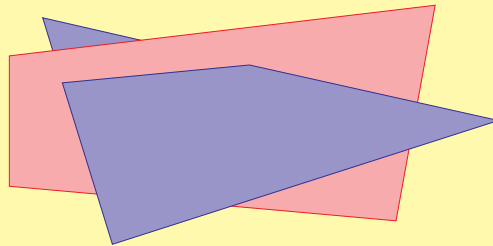
Teststrategie (Fortsetzung)

Ist **keine** der Bedingungen **2.a** bis **2.d** erfüllt, so werden F_j und F_k umgeordnet, falls F_j nicht schon umgeordnet wurde. F_j wird dann als umgeordnet gekennzeichnet. Die Tests beginnen dann mit F_k von neuem.

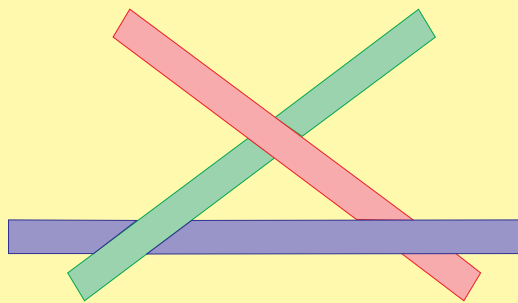
War F_j schon umgeordnet, so liegt eine zyklische Selbstüberlappung vor und F_j muss längs der Überlappungsgrenze geteilt werden. Mit den Teilflächen werden dann die Tests durchgeführt.

Anmerkung: Die Bedingungen **2.a** bis **2.d** sind nach steigendem Berechnungsaufwand geordnet.

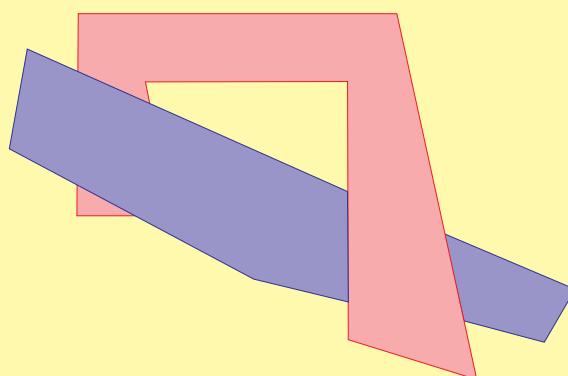
Prioritätslisten-Algorithmus, Teil 8



Durchdringung



Zyklische Überlappung



Wechselnde Überlappung

Bereichsunterteilungs-Algorithmen, Teil 1

Prinzip: Suche nach jenen Szenen ausschneiden, die "einfach genug" sind, um sofort ausgegeben werden zu können.

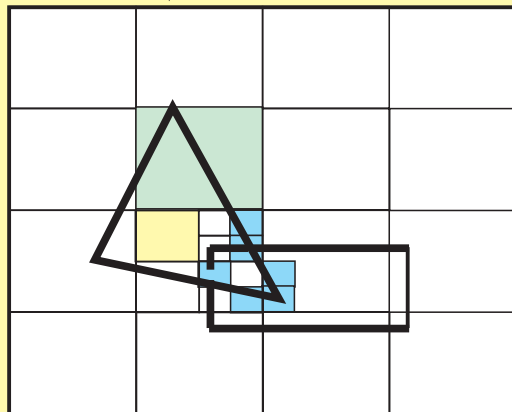
Methode: Rekursive Unterteilung von Fenstern, im Grenzfall bis auf Pixelgröße, wenn deren Inhalt noch zu komplex ist.

Vertreter: Warnock-Algorithmus
Weiler-Atherton-Algorithmus

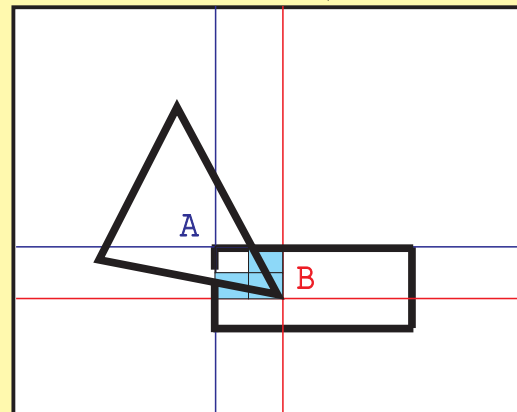
Bereichsunterteilungs-Algorithmen, Teil 2

Warnock-Algorithmus

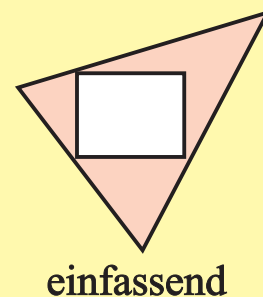
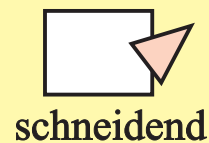
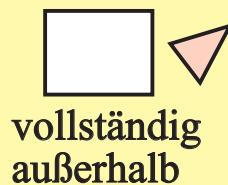
Regelmäßige
Aufteilung



Unregelmäßige
Aufteilung unter
Zuhilfenahme von
Polygonknoten



Relative Lage von Polygon und Fenster



Bereichsunterteilungs-Algorithmen, Teil 3

Warnock-Algorithmus

Abbruchkriterien

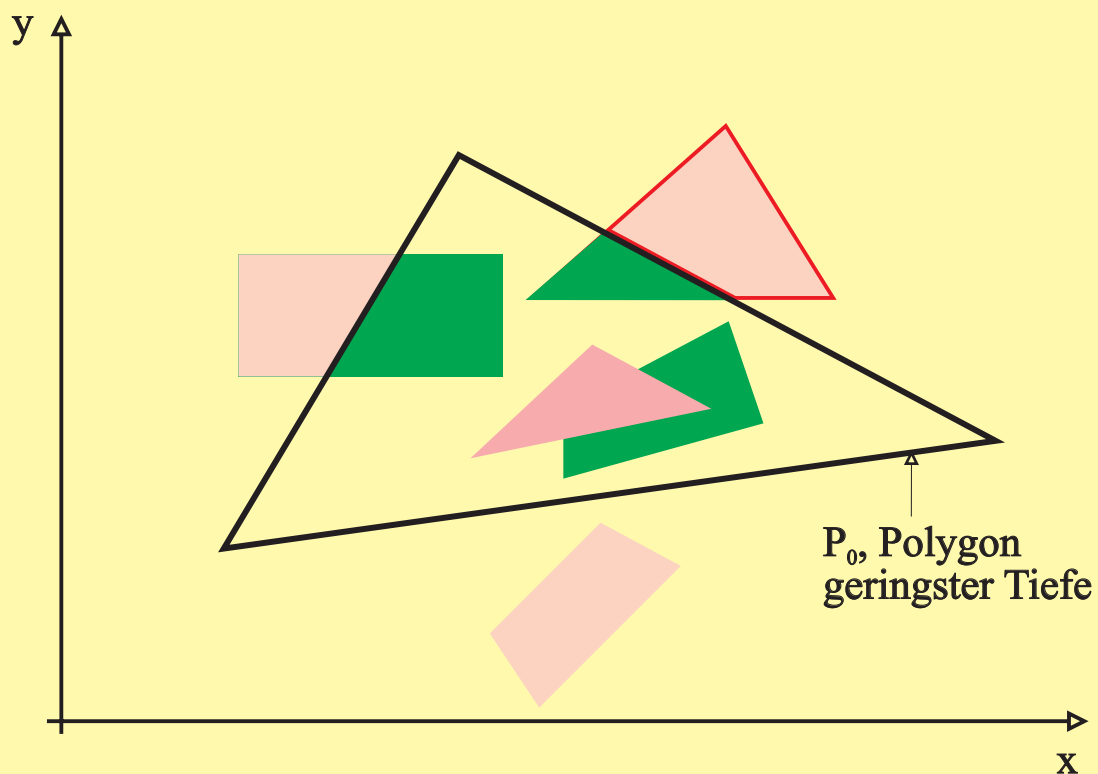
Die weitere Fensterunterteilung ist abubrechen, wenn **eine** der folgenden Bedingungen erfüllt ist:

1. Alle Polygone liegen vollständig außerhalb.
2. Die Liste enthält nur *ein* Polygon vom Typ *vollständig innerhalb*, *schneidend* oder *einfassend*.
3. Ein einfassendes Polygon überdeckt innerhalb des Fensters alle anderen Polygone.
4. Das Fenster hat die Größe von einem Pixel.

Bereichsunterteilungs-Algorithmen, Teil 4

Weiler-Atherton-Algorithmus

Methode: Minimierung der Anzahl der Fensterunterteilungen durch Teilung an den Polygonkanten.



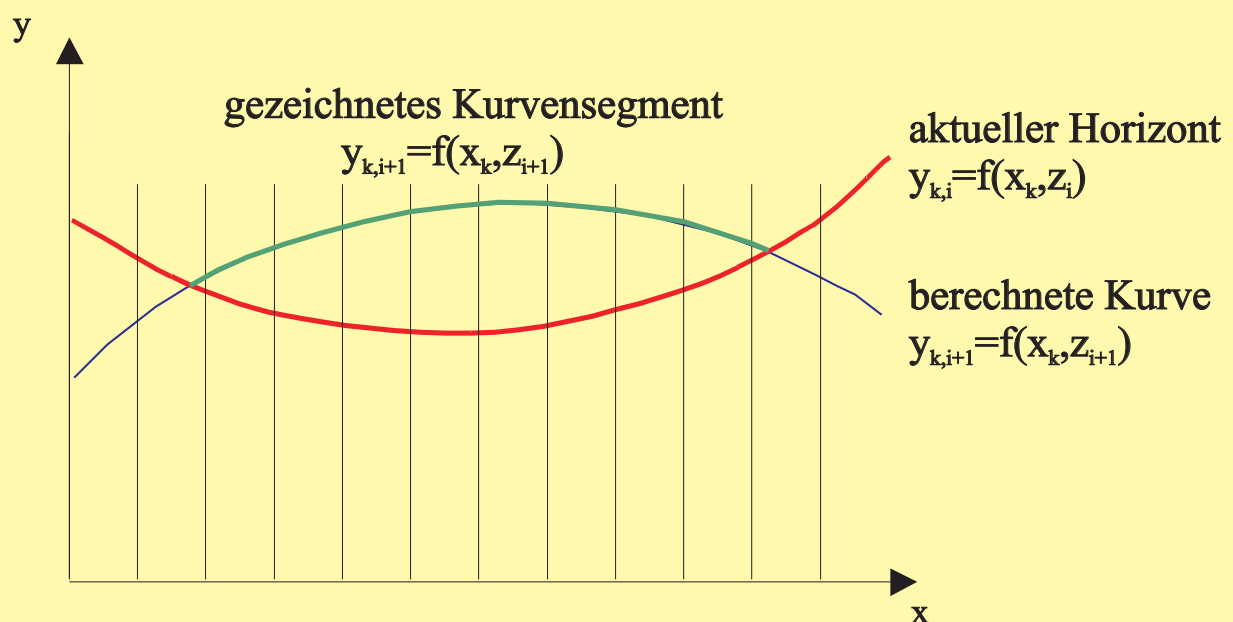
Bereichsunterteilungs-Algorithmen, Teil 5

Weiler-Atherton-Algorithmus

Algorithmus

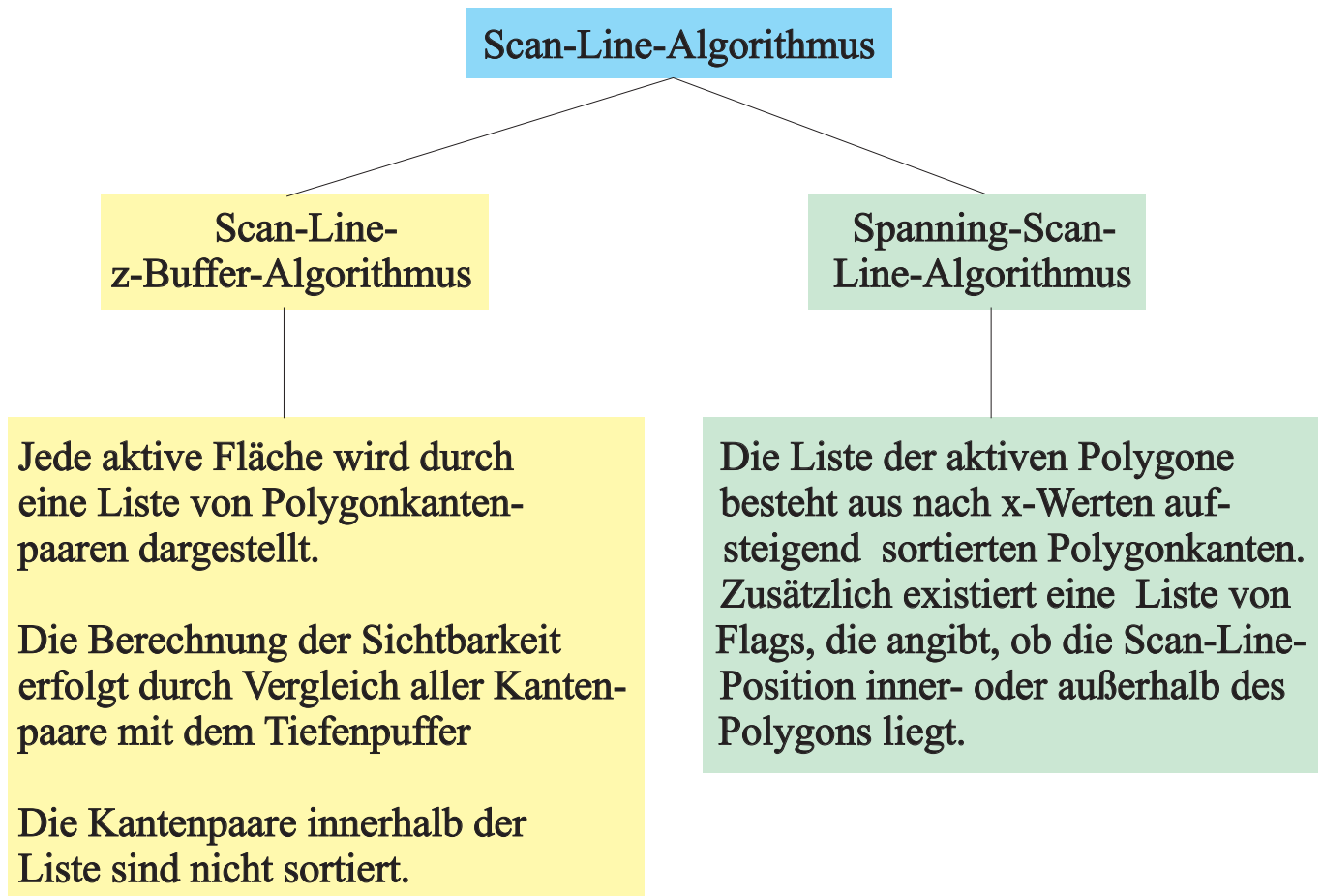
1. Aufbau der Prioritätenliste, geordnet nach zunehmendem Abstand vom Betrachter.
2. Unterteilung des polygonalen Fensters an den projizierten Kanten des Polygons P_0 mit der geringsten Tiefe in einen inneren und einen äußeren Bereich. Die geschnittenen Flächen werden entlang der Begrenzungskanten des Polygons P_0 geteilt.
3. Überprüfung der Bedingungen (2) bis (4) des Warnock-Algorithmus für beide Bereiche.
4. Ist für einen Bereich keine Bedingung erfüllt, so wird rekursiv weitergearbeitet.

Floating-Horizon-Methode

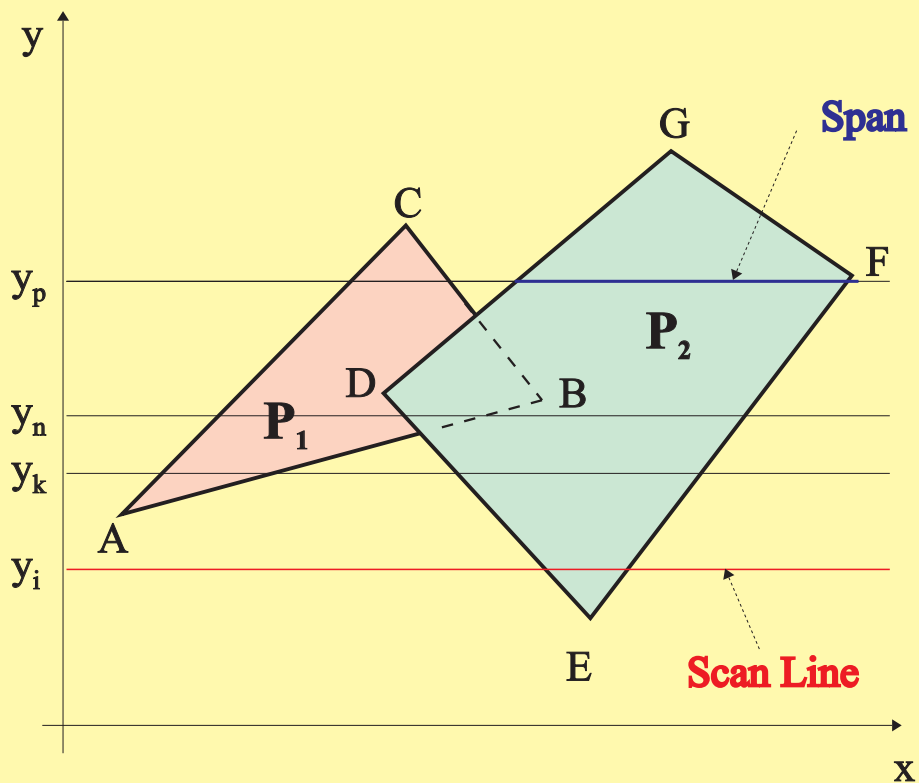


Falls $y_{k,i+1} > y_{k,i}$ ist, so wird $y_{k,i+1} = f(x_k, z_{i+1})$ gezeichnet.

Wenn $\Delta x > 1$ ist, muß zur Bestimmung des Schnittpunktes von $y_{k,i} = f(x_k, z_i)$ und $y_{k,i+1} = f(x_k, z_{i+1})$ interpoliert werden.



Spanning Scan Line-Algorithmus



Eintrag in die ET (Edge Table)

$x_{\min}$	$y_{\max}$	$\Delta x = 1/m$	ID
------------	------------	------------------	----

Eintrag in die PT (Polygon Table)

ID	Koeff. der Ebenengl.	Farb-information	Innen-Außen-Flag
----	----------------------	------------------	------------------