

## Polygon Clipping in 2D

- Clipping ist symmetrisch
- Polygon Clipping ist komplex
  - Sogar bei konvex Polygonen
- Polygon Clipping  $\neq$  Clipping der Liniensegmente!

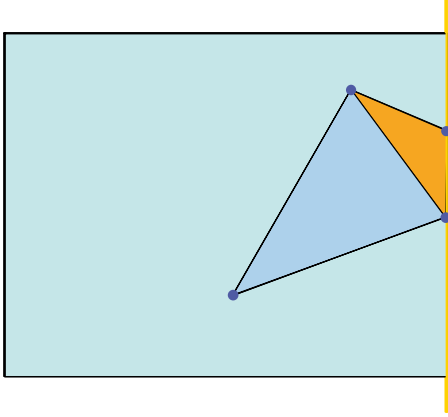
falsch!

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 35

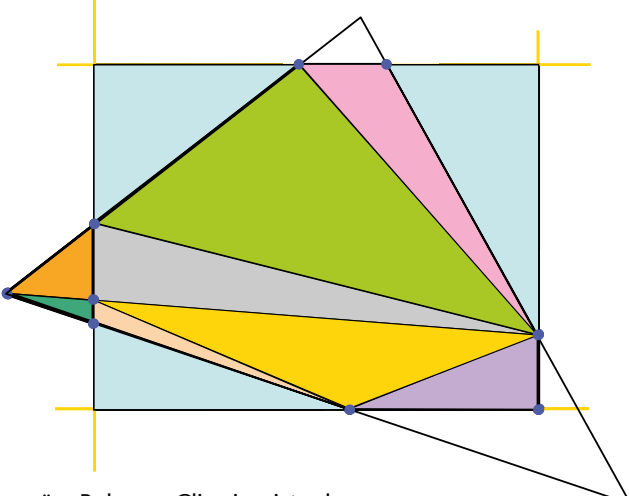
- Polygon-Clipping kann, insbesondere bei nicht-konvexen Polygonen, unangenehm werden
- Es können sogar mehrere Polygone entstehen

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 36

## Naiver Clipping-Algorithmus



G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 37



Fazit: das naive Polygon-Clipping ist schon für Dreiecke zu ineffizient (erzeugt viel zu viel Output)

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 38

## Sutherland-Hodgman

- Clipping eines Polygonzugs gegen ein konvexes Clip-Polygon (z.B. Viewport); der Polygonzug darf konkav sein

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 39

## Vorgehen

- Eingabe =
  - Liste der Eckpunkte in der richtigen Reihenfolge (gegen den Uhrzeigersinn)
  - Menge von Clip-Kanten, die ein konvexes Clip-Window definieren
- Betrachte eine (beliebige) Clip-Kante:
  - Nach dem Schnitt mit dieser Clip-Kante wird eine neue Liste von Eckpunkten erzeugt
  - Dieses Ergebnis ist wieder ein geschlossener Polygonzug
  - Alle Punkte des neuen Polygons befinden sich auf der "Innenseite" (der "richtigen" Seite) dieser Clip-Kante (Schleifeninvariante)
- Das wird mit allen Clip-Kanten wiederholt
  - (Im Prinzip ist die Reihenfolge der Clip-Kanten egal)

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 40

Der Polygonzug wird der Reihe nach an den Clip-Kanten geschnitten

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 41

### 4 Fälle

- Annahme: der Punkt A wurde bereits behandelt

	<p>Beide Punkte drinnen: → Output B</p>		<p>Linie „zeigt“ nach außen: → Output S</p>
	<p>Linie „zeigt“ nach innen: → Output S, B</p>		<p>Beide Punkte draußen: → kein Output</p>

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 42

### Beispiel

Input: A B C D

Output: S<sub>1</sub> A B C S<sub>2</sub>

Man beginnt mit der Kante [letzter Punkt – erster Punkt], hier also D-A.  
(D wird am Ende der Schleife betrachtet)

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 43

Input: S<sub>1</sub> A B C S<sub>2</sub>

Output: S<sub>3</sub> S<sub>1</sub> A S<sub>4</sub> S<sub>5</sub> C S<sub>6</sub>

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 44

- Kleines Problem des Sutherland-Hodgeman-Algos: falls das ursprüngliche Polygon in mehrere Teile zerfällt beim Clipping, dann entsteht eine unschöne Polygonkante am Rand des Windows
- Beispiel:

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 45

Eingabe:  
 $P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8$

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 47

Etwas komplexeres Beispiel

Eingabe:  
 $P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8$

Ausgabe:  
 $I_1 P_2 P_3 P_4 P_5 I_2 I_3 P_7 I_4$

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 48

Eingabe:  
 $I_1 P_2 P_3 P_4 P_5 I_2 I_3 P_7 I_4$

Ausgabe:  
 $I_1 I_5 I_6 P_4 P_5 I_2 I_3 P_7 I_4$

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 49

Eingabe:  
 $l_1 l_5 l_6 P_4 P_5 l_2 l_3 P_7 l_4$

Ausgabe:  
 $l_7 l_8 l_6 P_4 l_9 l_{10} l_3 P_7 l_4$

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 50

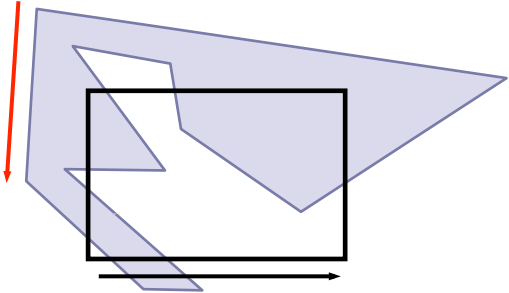
Ausgabe:  
 $l_7 l_8 l_6 P_4 l_9 l_{10} l_3 P_7 l_4$

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 51



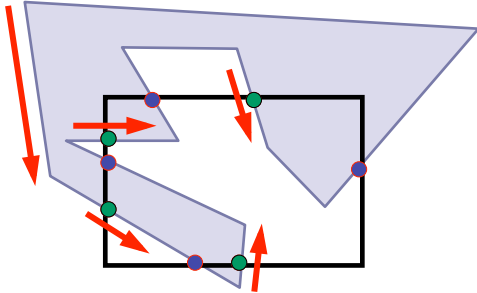
## Der Algorithmus von Weiler-Atherton

- Strategie: "Wandere" auf dem Polygonzug oder dem Window-Rand
- Konvention (wie immer): Polygone sind CCW orientiert



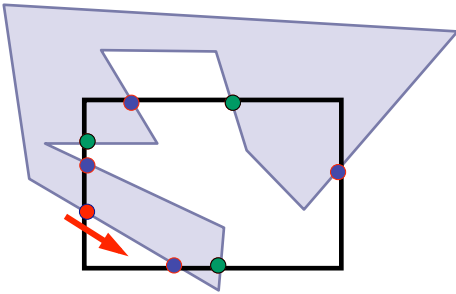
G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 56

- Berechne alle Schnittpunkte
- Markiere die Punkte, an denen das Polygon in das Clipping-Window eintritt (hier grün)



G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 57

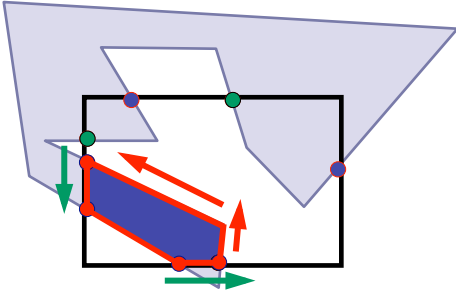
- Solange noch ein unbearbeiteter Eintrittsschnittpunkt vorhanden ist wird das Polygon weiter umlaufen
- Die Ausgabe ist (wie bei Sutherland-Hodgman) ein oder mehrere Listen von Punkten (Eckpunkte des Polygons, Schnittpunkte, und/oder Eckpunkte des Windows)
  - Unterschied: die Eingabe wird nur einmal abgearbeitet



G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 58

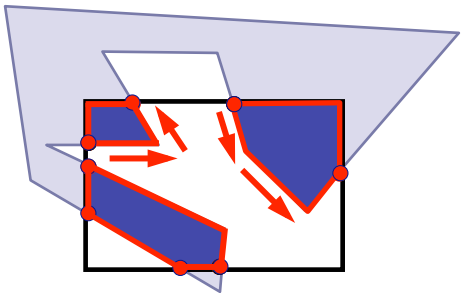
### Umlauf-Regeln

- Trifft man beim Umlauf auf einen Schnittpunkt, dann:
  - Füge Schnittpunkt zur Ausgabe hinzu
  - Falls Schnittpunkt = "Out-to-in": folge dem Polygonzug (ccw)
  - Falls Schnittpunkt = "In-to-out": folge dem Window-Rand (ccw)



G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 59

- Solange noch ein unbearbeiteter Eintrittsschnittpunkt vorhanden ist wird das Polygon weiter umlaufen

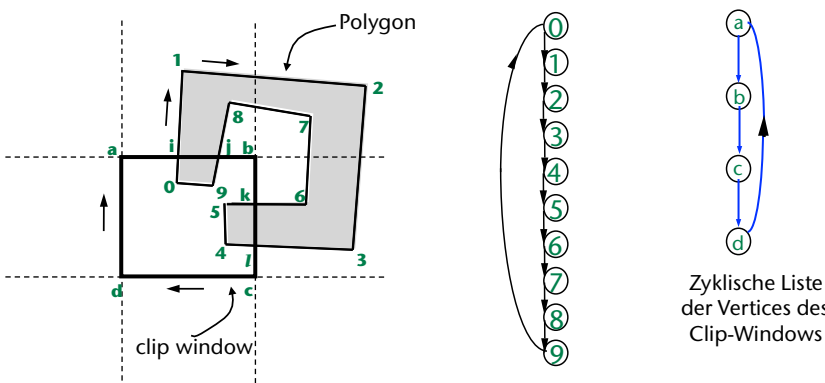


- Der Weiler-Atherton-Algorithmus erzeugt echt separate Polygonzüge für jedes sichtbare Fragment

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 60

### Implementierung des Algorithmus

- Eingabe (hier ist die Konvention *clockwise ordering*):



Zyklische Liste der Vertices des Polygons

Zyklische Liste der Vertices des Clip-Windows

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 61

Find the intersection points and add them to **both** lists

Clipping 62

Find the intersection points and add them to **both** lists

Clipping 63

Find the intersection points and add them to **both** lists

**Add Vertex k:**

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 64

Find the intersection points and add them to **both** lists

**Add Vertex j:**

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 65

▪ Klassifiziere jeden Schnittpunkt als „*entering*“ oder „*leaving*“

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 66

▪ Erzeuge das geclippte Polygon:

- Starte bei einem "entering"-Vertex
- Falls man auf einen "leaving"-Vertex trifft, dann wechsele auf die Liste des Clip-Polygons (blaue Zeiger)
- Falls man auf einen "entering"-Vertex trifft, dann wechsele auf die Liste des Polygons (schwarze Zeiger)
- Eine Polygonzug ist beendet, wenn der Startpunkt wieder erreicht ist
- Wiederhole, solange noch unbesuchte "entering"-Vertices vorhanden sind

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 67

**Beispiel**

- Polygonzug 1 = l, 4, 5, k
- Polygonzug 2 = j, 9, 0, l

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 68

**Robustheit, Präzision, Entartungen**

- Die üblichen (leidigen) Fragen:
  - Was passiert wenn ein Vertex (beinahe) auf dem Rand des anderen Polygonzuges liegt?

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 69

## Absch(l)ießende Bemerkungen

- Ältere Hardware machte volles Clipping
- Moderne Hardware vermeidet Clipping weitgehend:
  - Nur bzgl. der Ebene  $z=z_0$
- Im Allgemeinen ist es nützlich, Clipping zu kennen, da es viele ähnliche geometrische Probleme und Algorithmen gibt, z.B.:
  - Zur Bestimmung, welche Objekte innerhalb eines "Picking-Frustums" liegen
  - Schnittpunkte zwischen Objekten
  - Berechnung analytischer Schatten

Modeling Transformations

Illumination (Shading)

Viewing Transformation (Perspective / Orthographic)

Clipping

Projection (to Screen Space)

Scan Conversion (Rasterization)

Visibility / Display

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 70



"Oh, lovely — just the hundredth time you've managed to cut everyone's head off."

G. Zachmann Computer-Graphik 1 - WS 09/10 Clipping 73