



2D Viewing



Polygon and Text Clipping

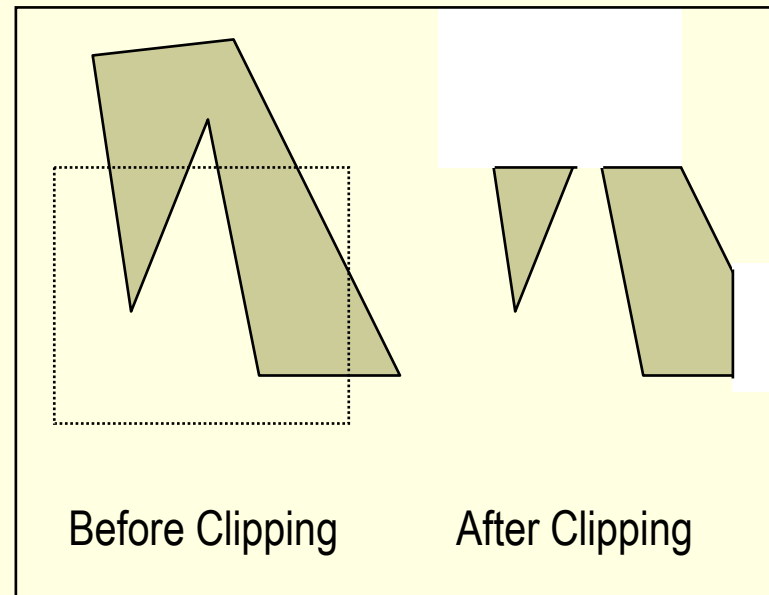
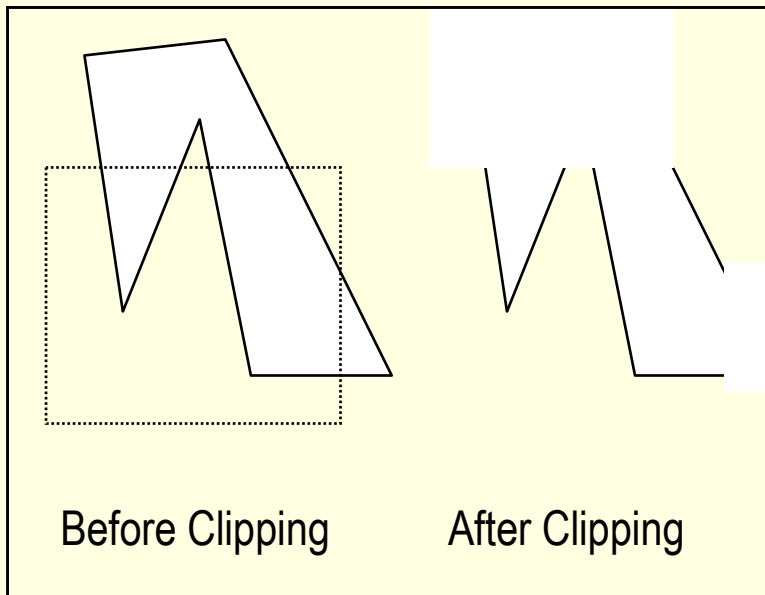
Overview

- Sutherland-Hodgeman Polygon Clipping
- Weiler-Atherton Polygon Clipping
- Text Clipping
- Exterior Clipping

Polygon Clipping

- To clip a polygon fill area, we cannot directly apply a line clipping method to the individual polygon edges.
- Because this approach would not, in general, produce a closed polyline.
- Instead, a line clipper would often produce a disjoint set of lines with no complete information about how we might form a closed boundary around the clipped fill area.

Sutherland-Hodgman Polygon-Clipping



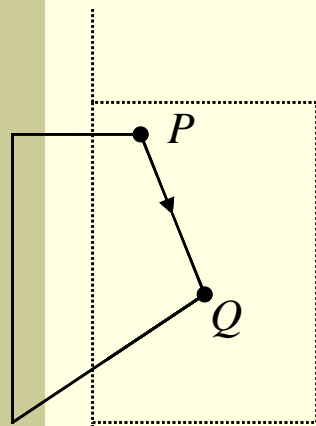
Sutherland-Hodgeman Polygon Clipping Algorithm (A divide-and-conquer strategy)

- Polygons can be clipped against each edge of the window one at a time.
- Clip the output polygon against the next edge.
- Repeat for all edges
- Edge intersections, if any, are easy to find since the X or Y coordinates are already known.
- Note that the number of vertices usually change and will often increase.

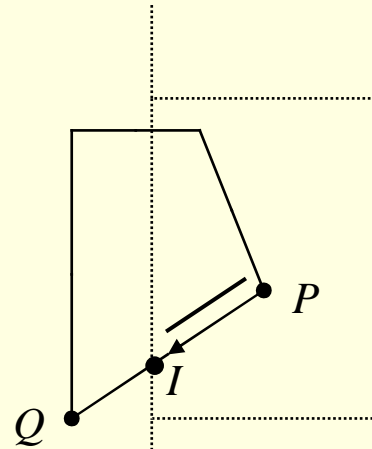
Sutherland-Hodgeman Polygon Clipping Algorithm

- Clip one boundary at a time: left, top, right, bottom.
- Check each adjacent pair of vertices (P,Q), in order to make a new vertex list.
 1. If P and Q are in, add Q.
 2. If P is in and Q is out, add the intersection point with boundary only.
 3. If P and Q are both out, add nothing.
 4. If P is out and Q is in, add intersection point with boundary and also Q.

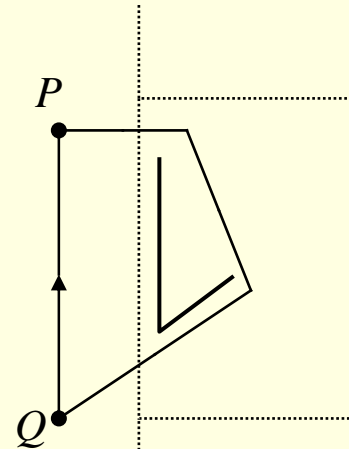
Sutherland-Hodgeman Algorithm(*cont.*)



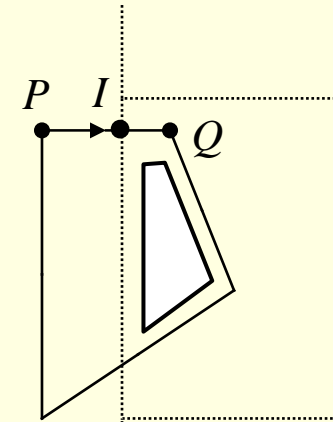
Save Q
(a)



Save I
(b)

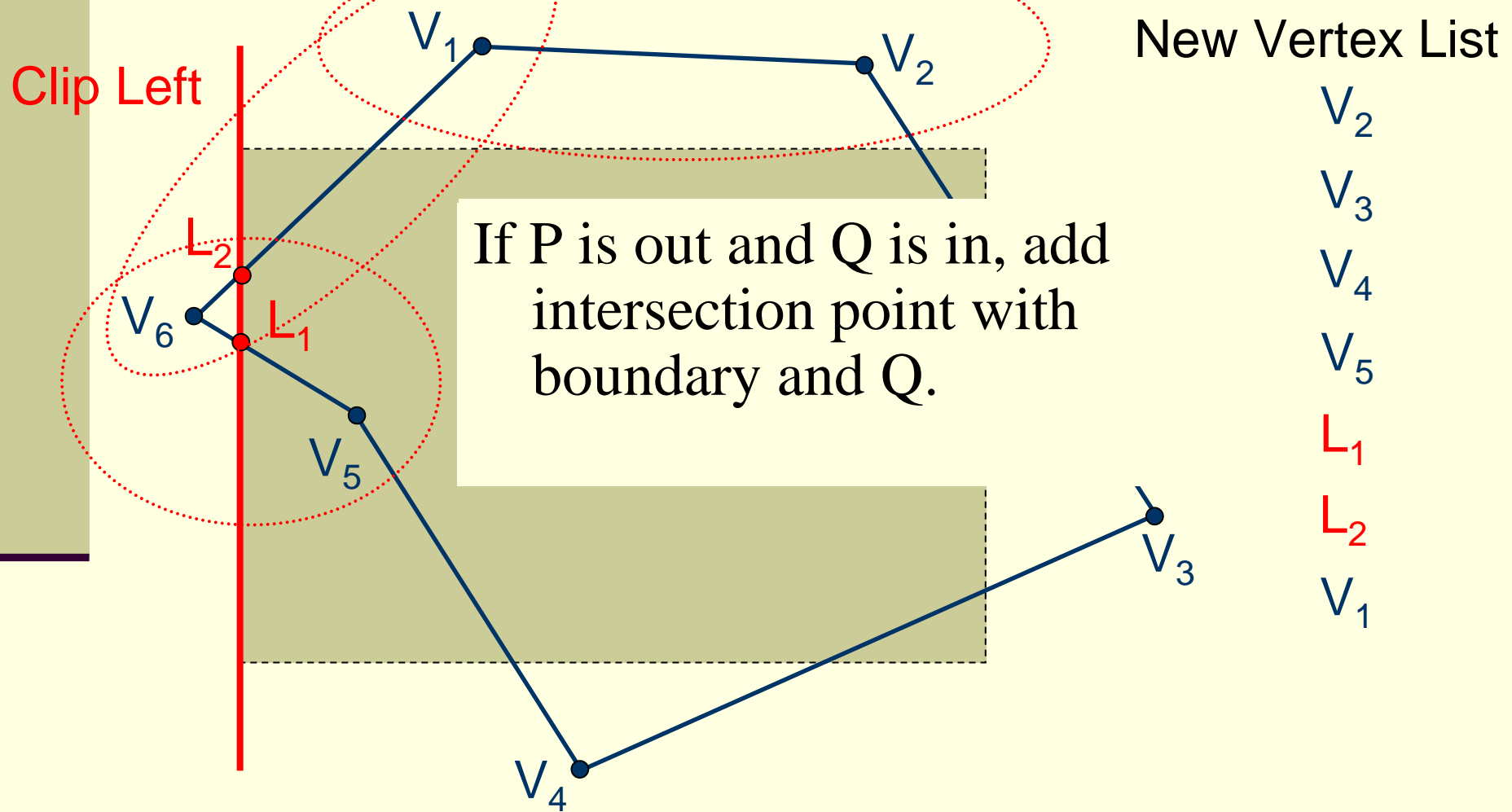


No Points
Saved
(c)

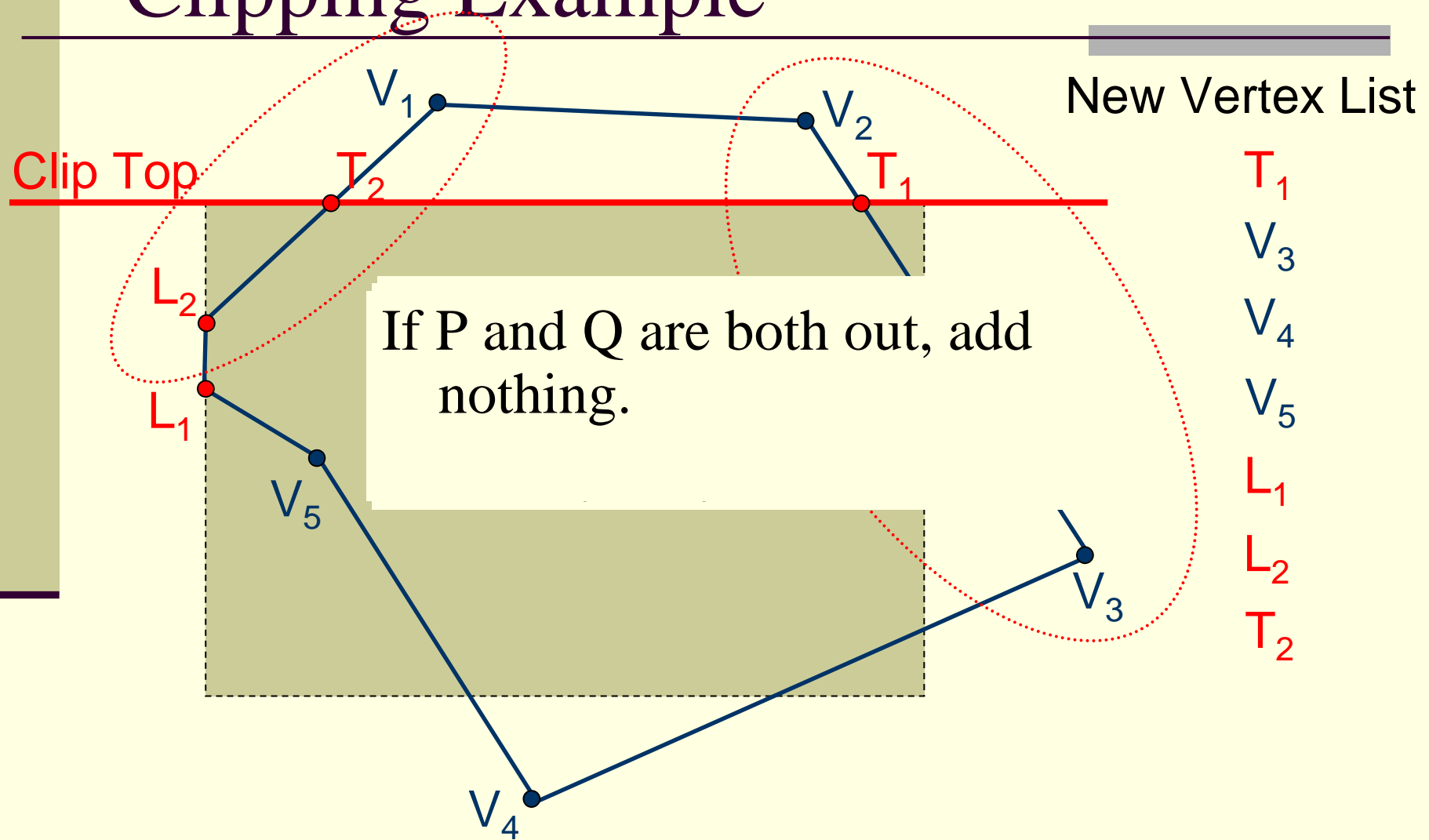


Save I, Q
(d)

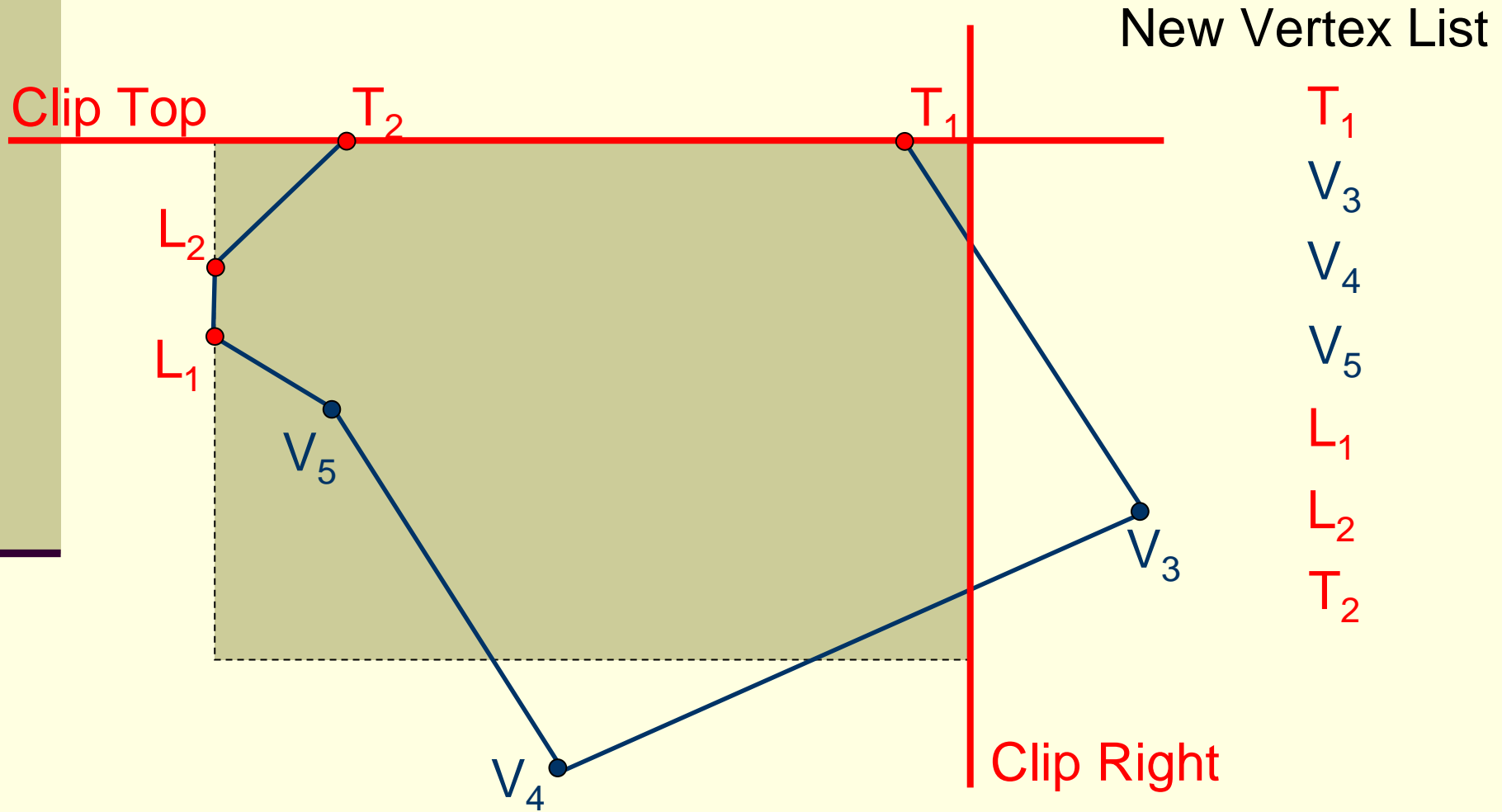
Sutherland-Hodgeman Clipping Example



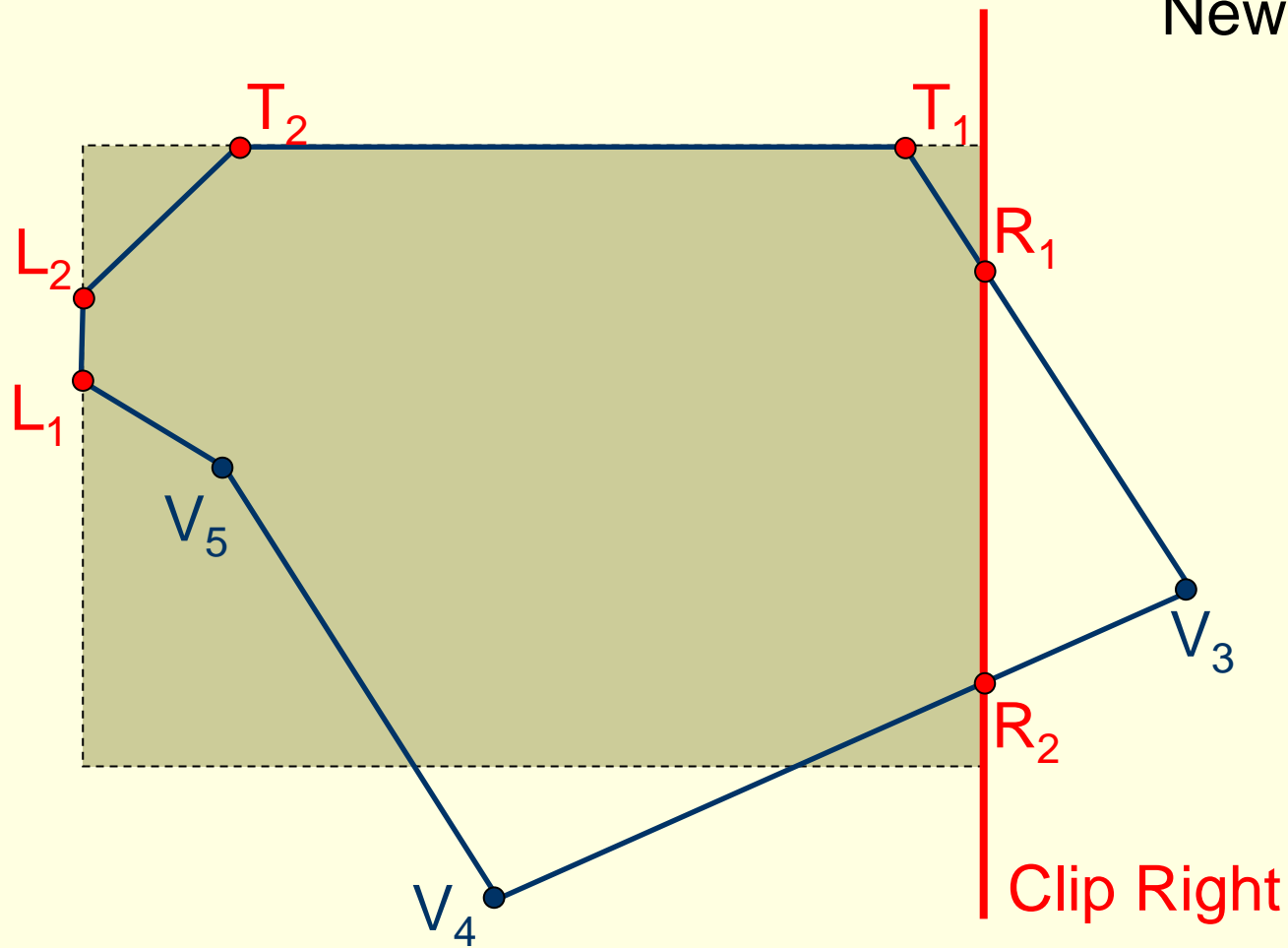
Sutherland-Hodgeman Clipping Example



Sutherland-Hodgeman Clipping Example



Sutherland-Hodgeman Clipping Example



New Vertex List

R_1

R_2

V_4

V_5

L_1

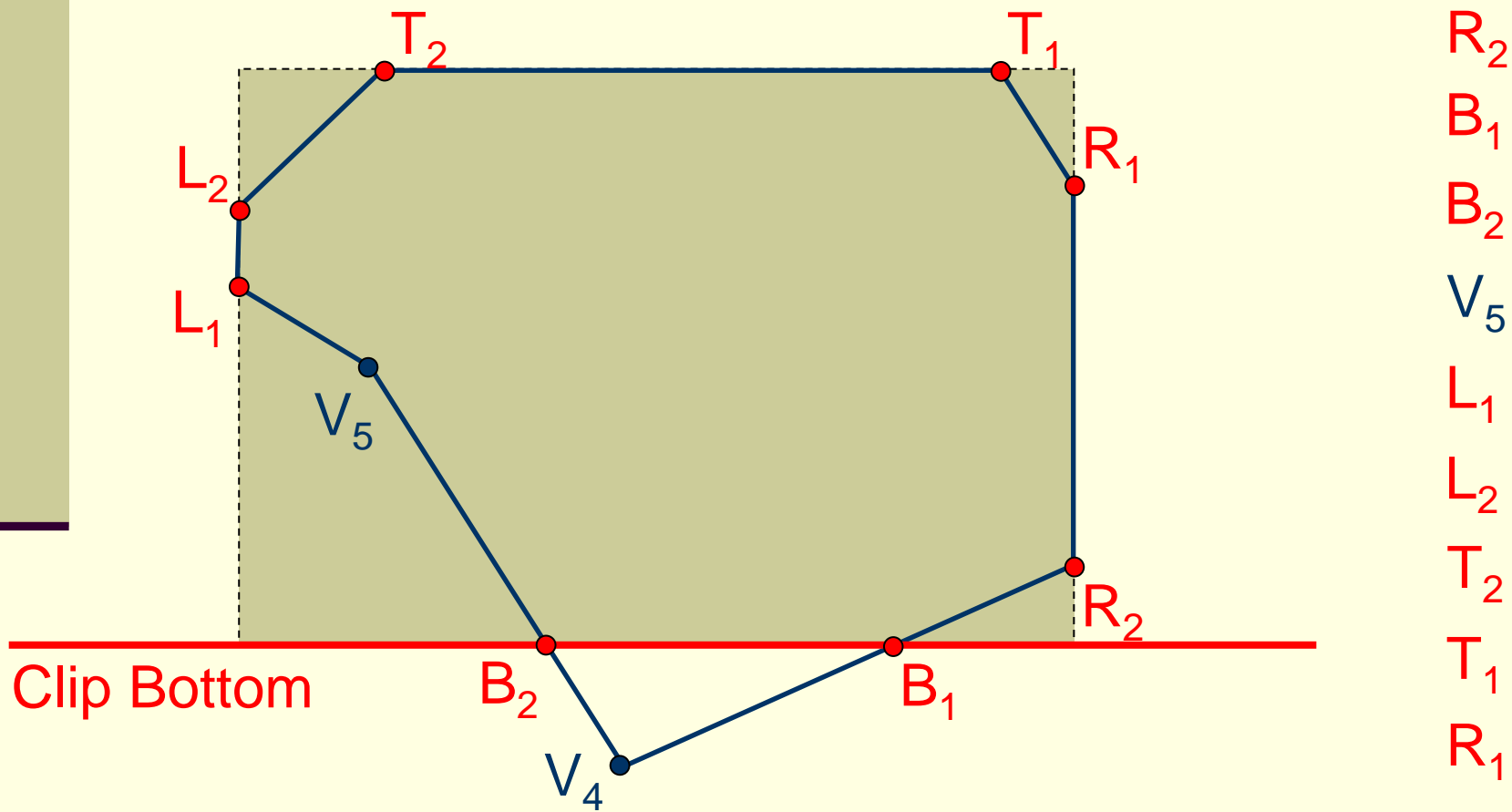
L_2

T_2

T_1

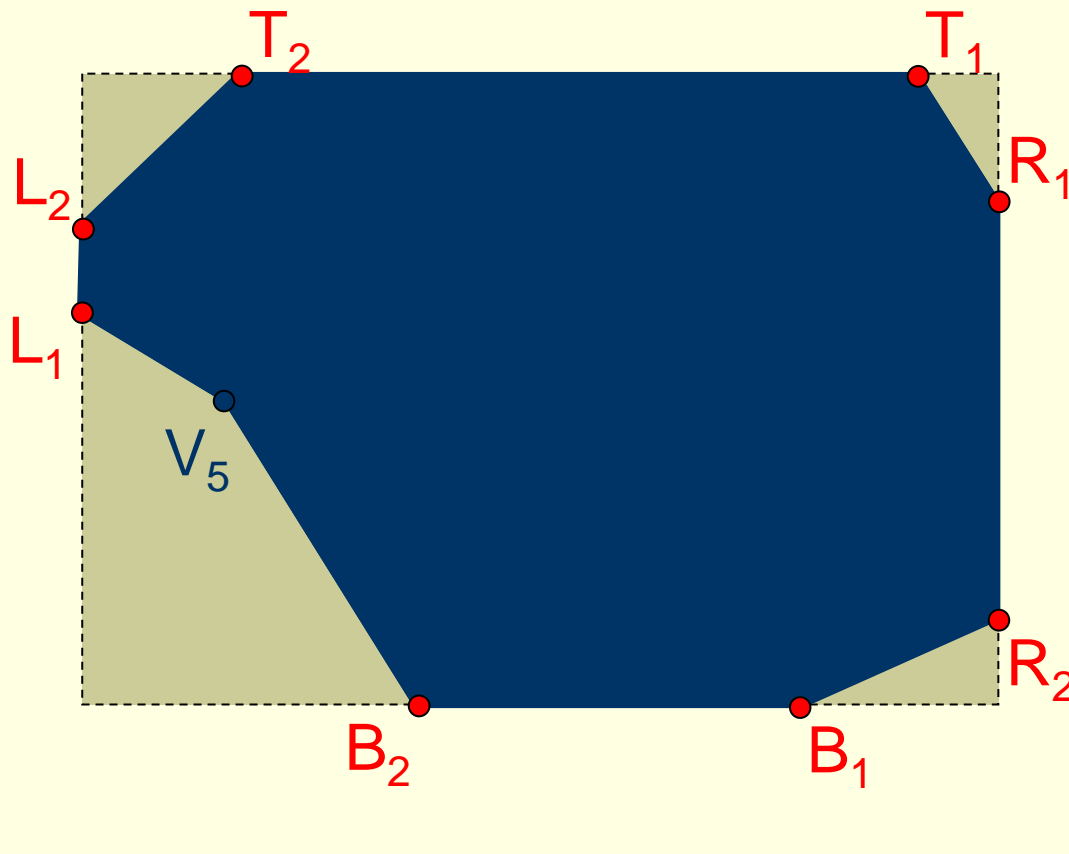
Sutherland-Hodgeman Clipping Example

New Vertex List



Sutherland-Hodgeman Clipping Example

New Vertex List

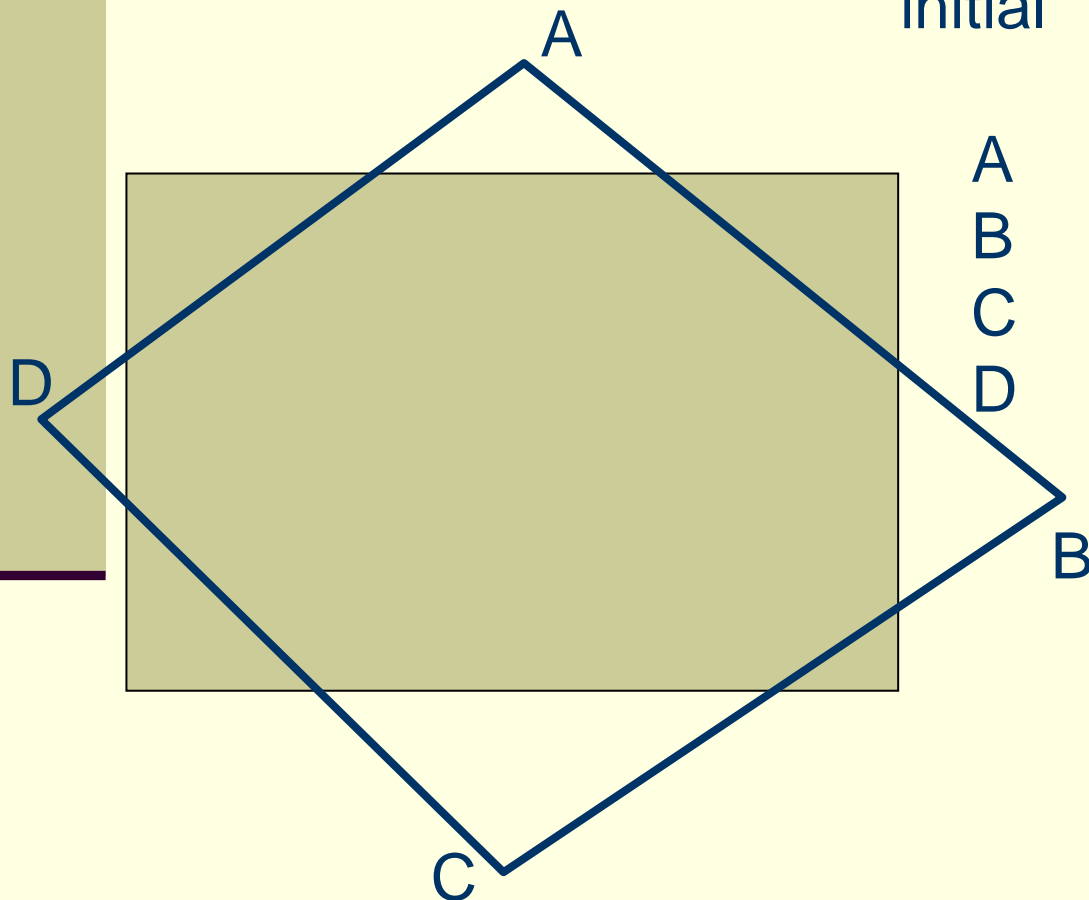


R_2
 B_1
 B_2
 V_5
 L_1
 L_2
 T_2
 T_1
 R_1

Sutherland-Hodgeman Exercise 1

Vertex List

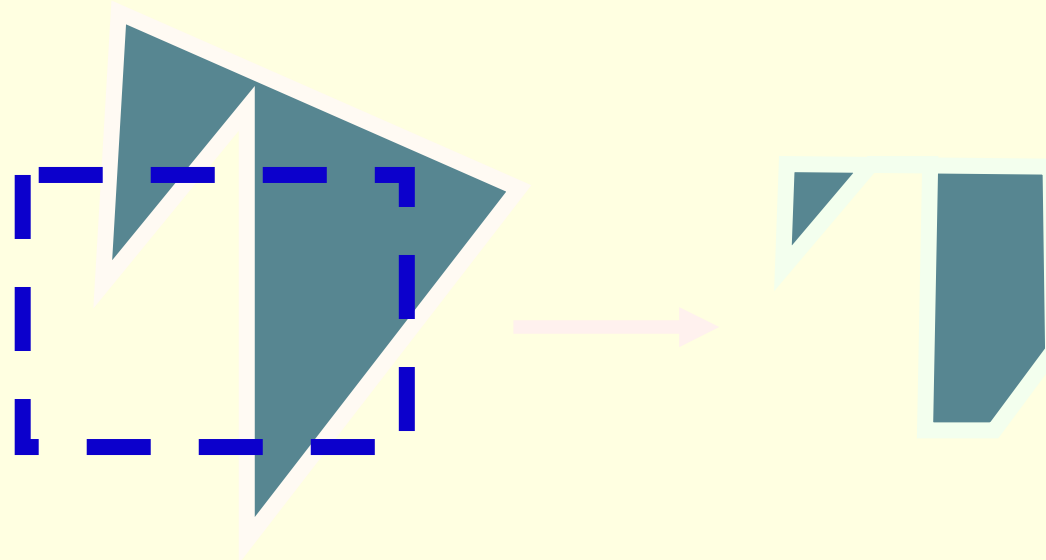
initial	after left	after top	after right	after bottom
---------	---------------	--------------	----------------	-----------------



Sutherland-Hodgeman Polygon Clipping Algorithm

- Convex polygons are correctly clipped by sutherland hodgeman algorithm
- Concave polygons may be displayed with extraneous lines.
- Occurs when clipped polygon have two separate sections.
- Only one output vertex list, the last vertex in the list is always joined to the first vertex

Weiler-Atherton Clipping



- A different clipping algorithm, the Weiler-Atherton algorithm, creates separate polygons

Weiler-Atherton Clipping

- The vertex processing procedures for window boundaries are modified so that concave polygons are displayed.
- Consider the window boundaries along with the polygon edges.
- Which path to follow depends on the polygon processing direction.
- For clockwise processing of polygon vertices, use the following rules:
 - For an outside-to-inside pair of vertices, follow the polygon boundary
 - For an inside-to-outside pair of vertices, follow the window boundary in a clockwise direction

Weiler-Atherton Clipping

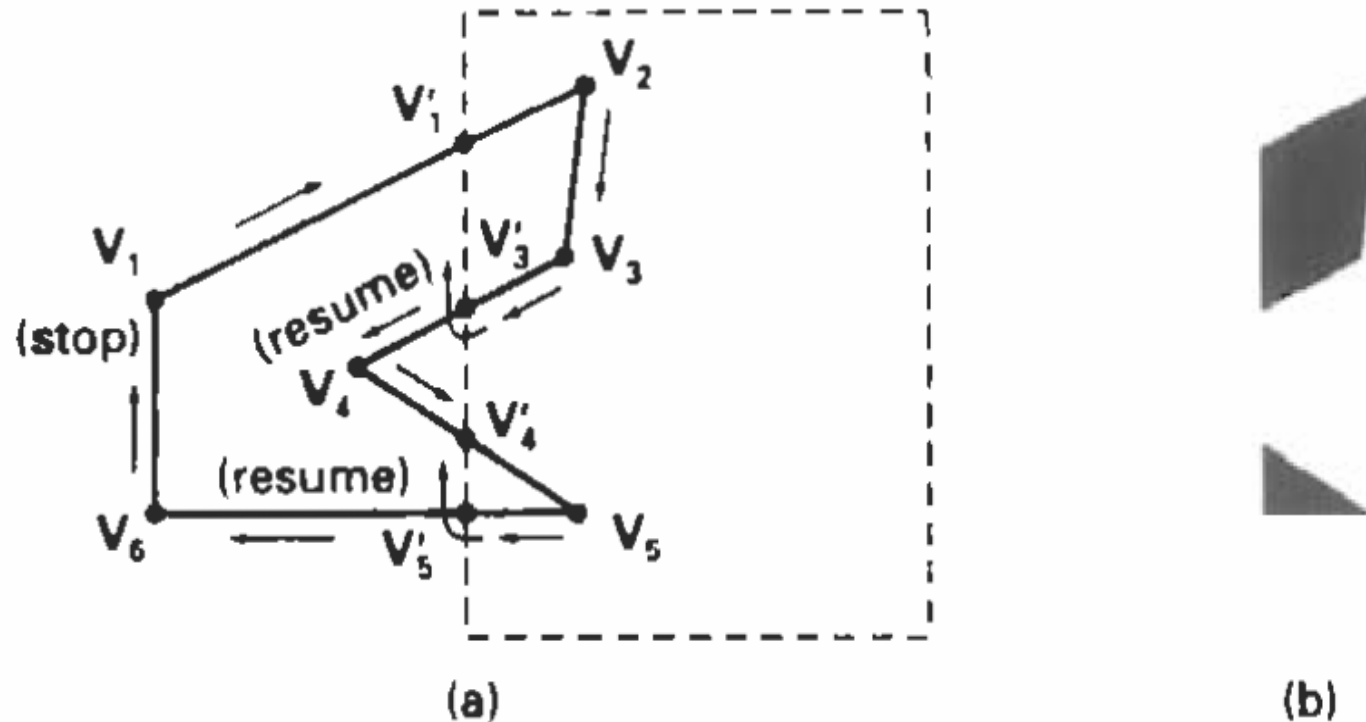


Figure 6-25

Clipping a concave polygon (a) with the Weiler-Atherton algorithm generates the two separate polygon areas in (b).

Weiler-Atherton Clipping

- Polygon clipping using nonrectangular polygon clip windows

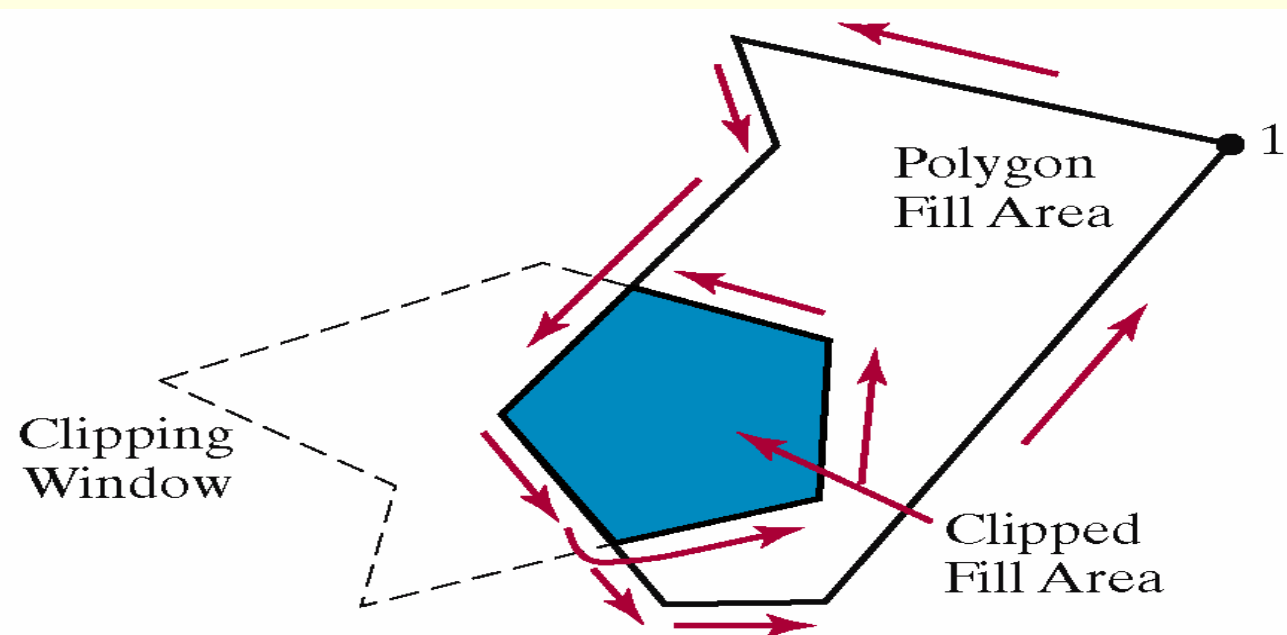
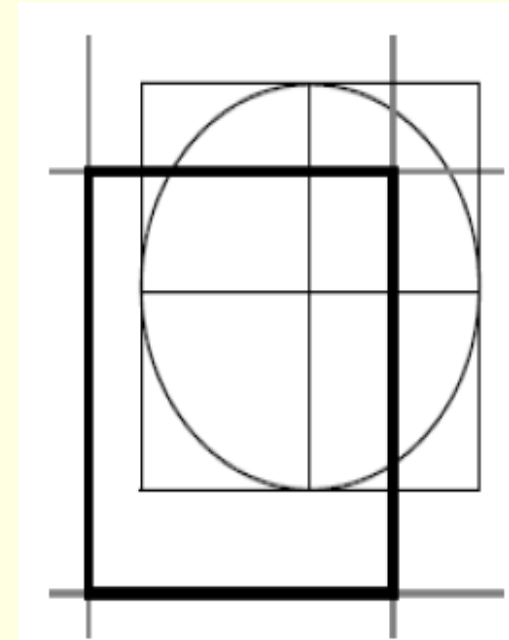


Figure 6-30

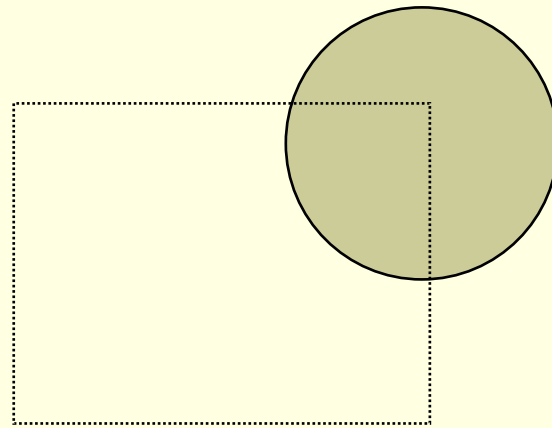
Clipping a polygon fill area against a concave-polygon clipping window using the Weiler-Atherton algorithm.

Curve Clipping

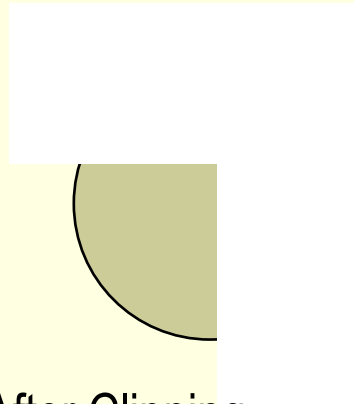
- Curve clipping procedures involve non-linear equations for intersection tests.
- A good strategy is to utilize bounding information, e.g.:
- Check bounding box for trivial accept/reject



Curve Clipping



Before Clipping



After Clipping

Text Clipping

- Several techniques are used for text clipping in a graphics package.
- **All-or-none text clipping**
 - Using boundary box for the entire text
 - If all text inside the clip window , keep it
 - String is discarded if there is any overlap of bounding rectangle with window boundary.

Text Clipping

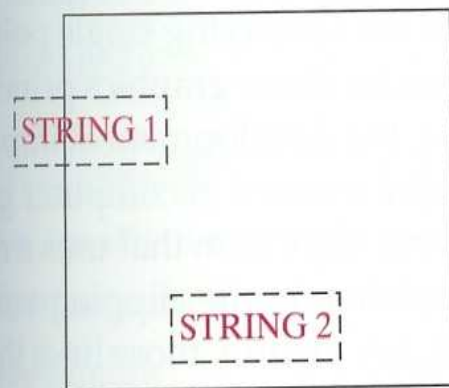
- **All-or-none character clipping**

- Using boundary box for each individual character
- The boundary limits of the individual characters are compared to the window.
- Any character which is outside or overlaps a window boundary is clipped.

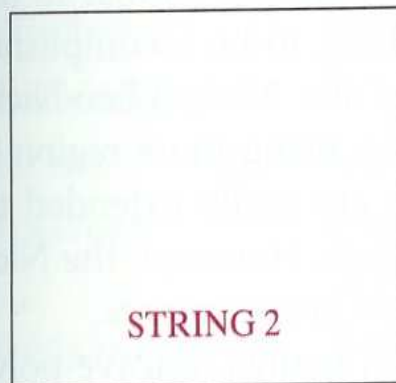
- **Character Component Clipping**

- If individual character overlaps a clip window boundary, clip off the parts of the character that are outside the window.

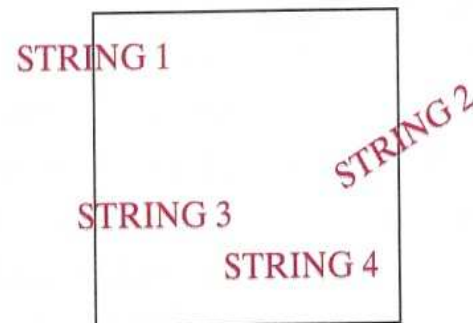
Text Clipping



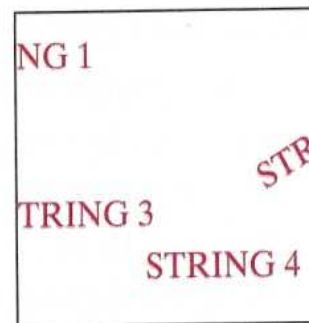
Before Clipping



After Clipping



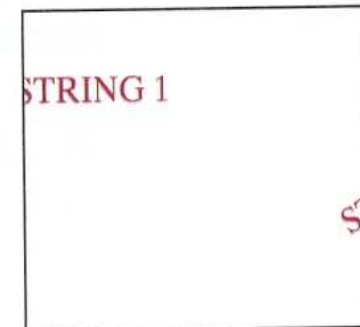
Before Clipping



After Clipping



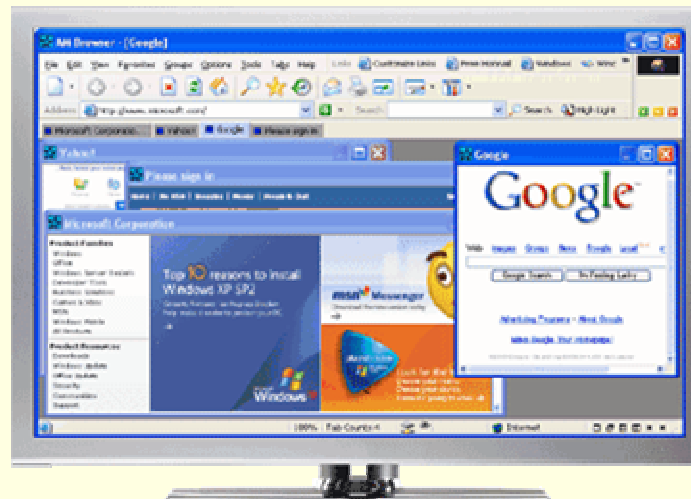
Before Clipping



After Clipping

Exterior Clipping

- The picture to be saved are those that are outside the region.
- Example : Multiple window systems
- Objects within the window are clipped to the interior of the window.
- Other high priority windows overlap these objects , they are clipped to the exterior of the overlapping windows.





■ Thank You

