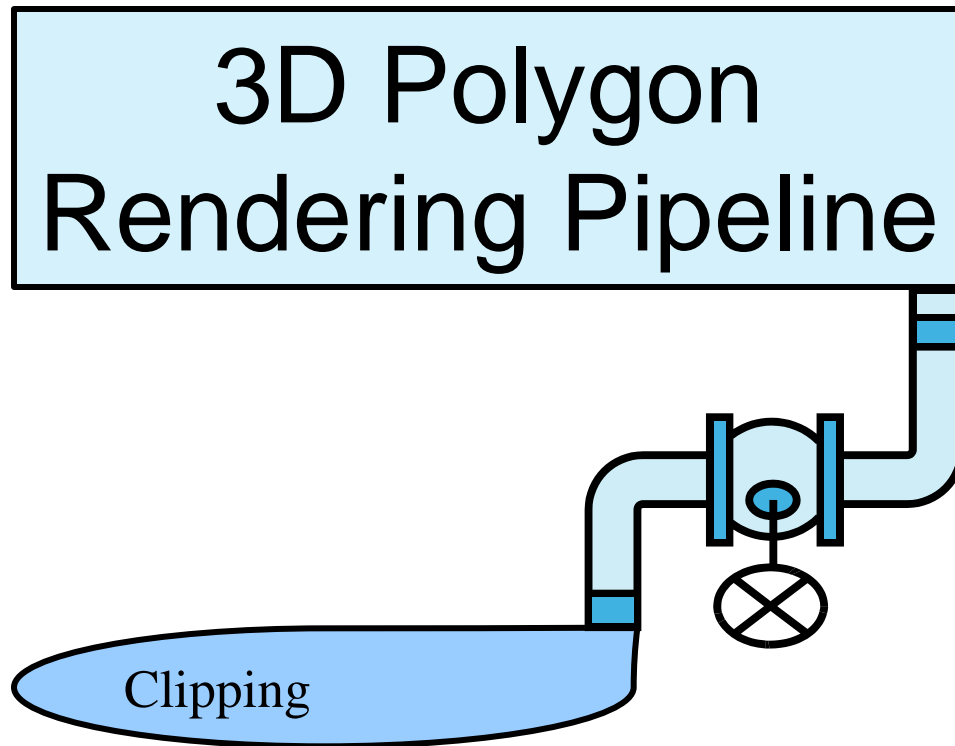
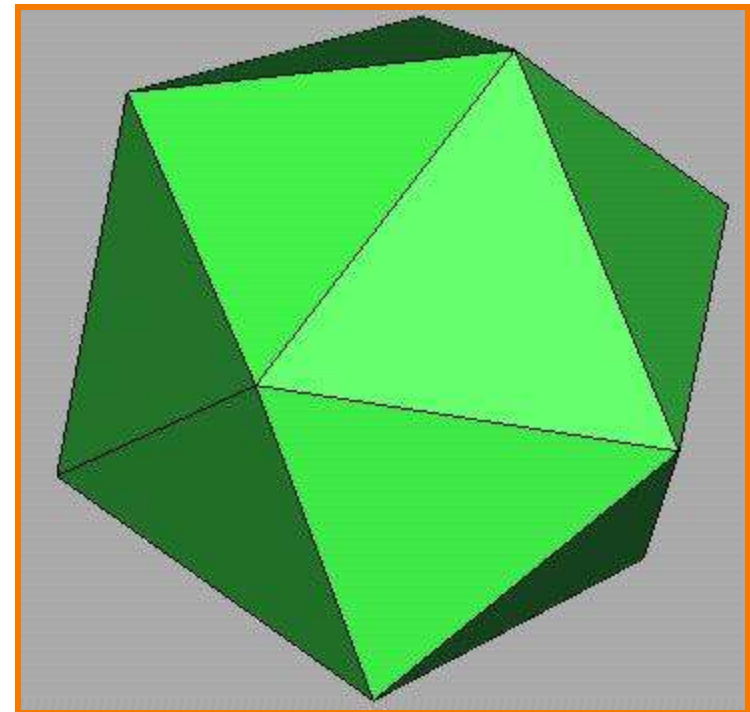
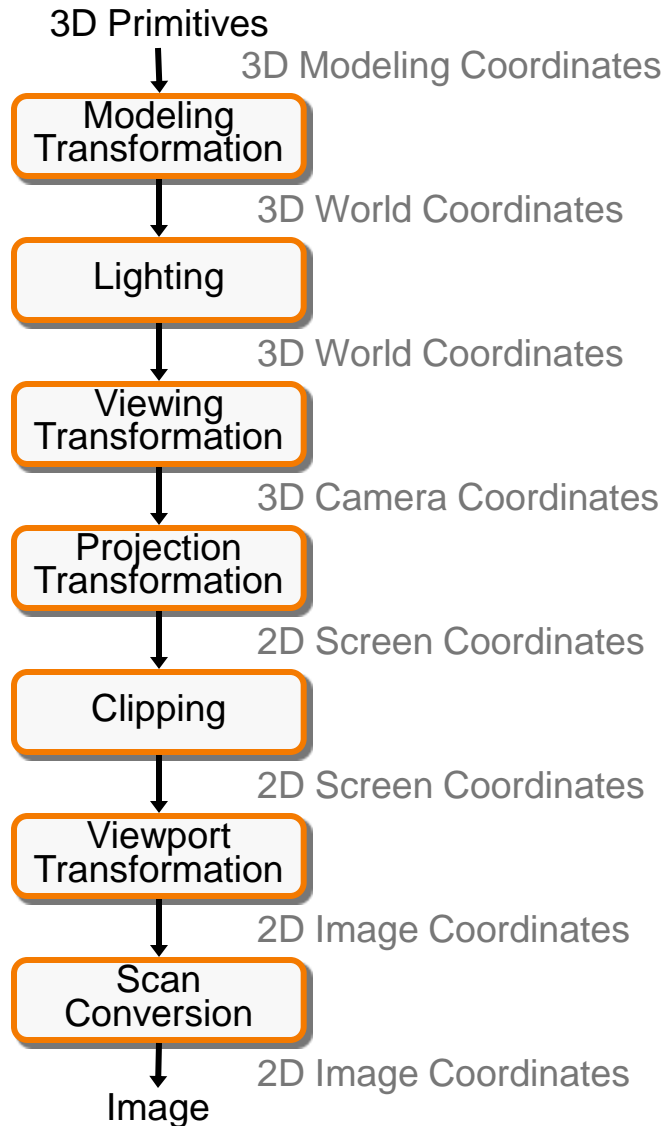


# קורס גרפיקה ממוחשבת

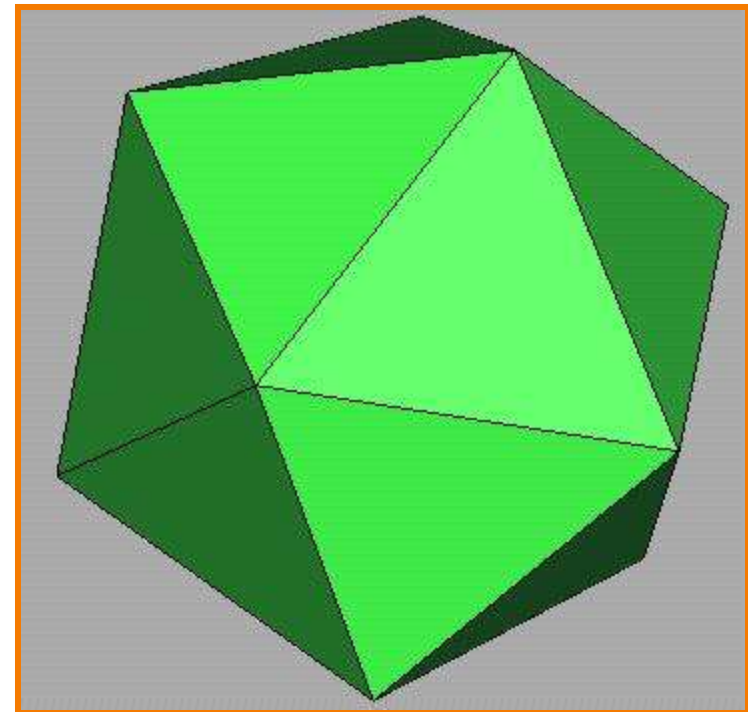
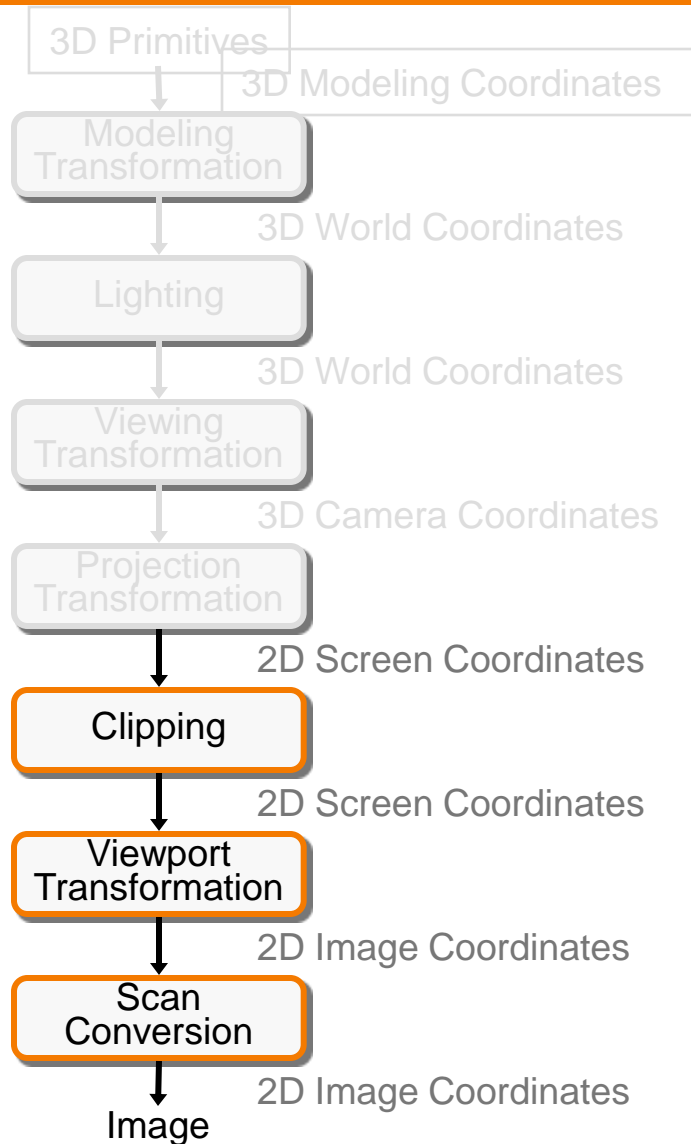


Thomas Funkhouser  
Princeton University  
COS 426, Fall 1999

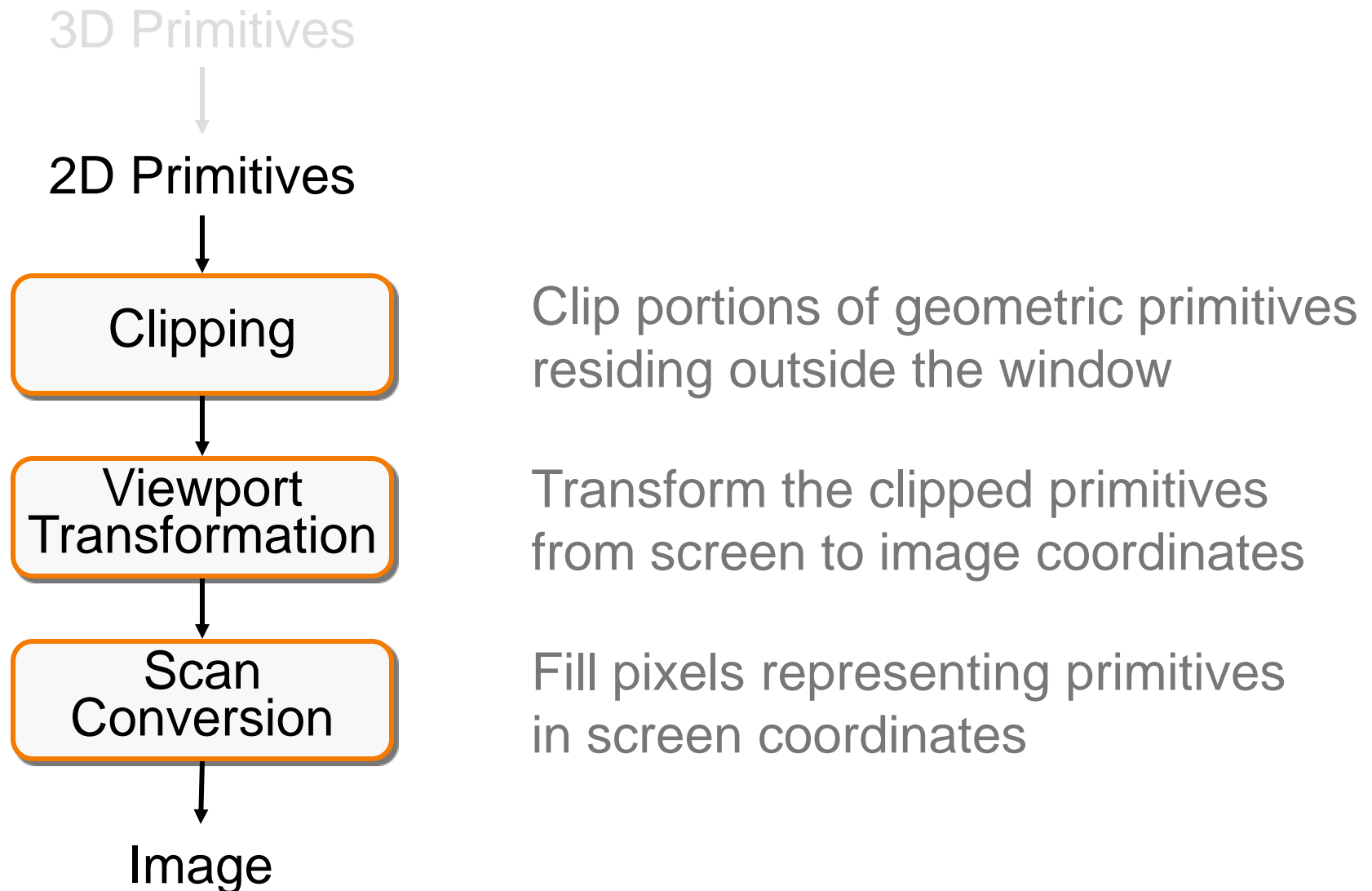
# 3D Rendering Pipeline (for direct illumination)



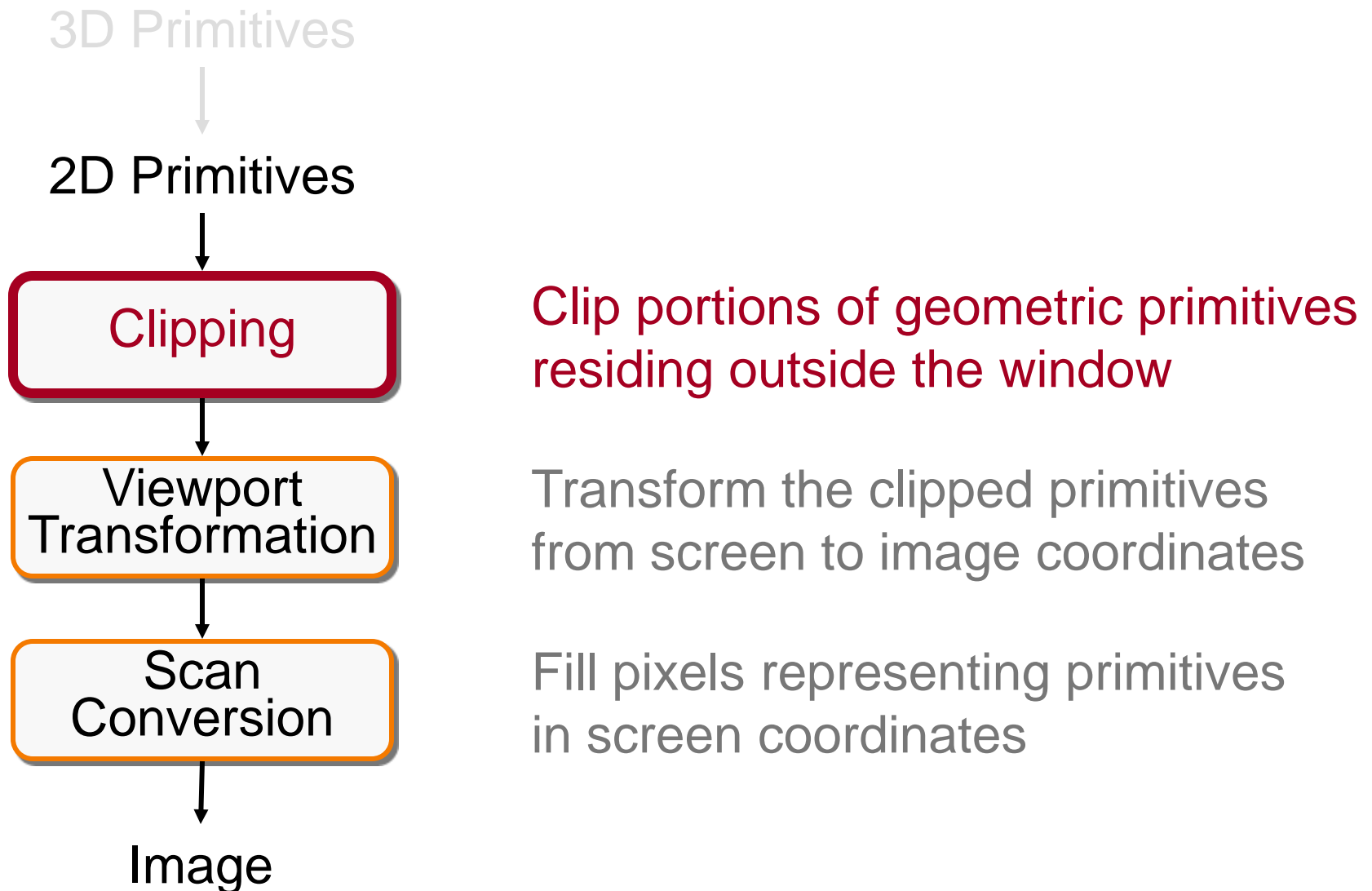
# 3D Rendering Pipeline (for direct illumination)



# 2D Rendering Pipeline



# 2D Rendering Pipeline



# Clipping

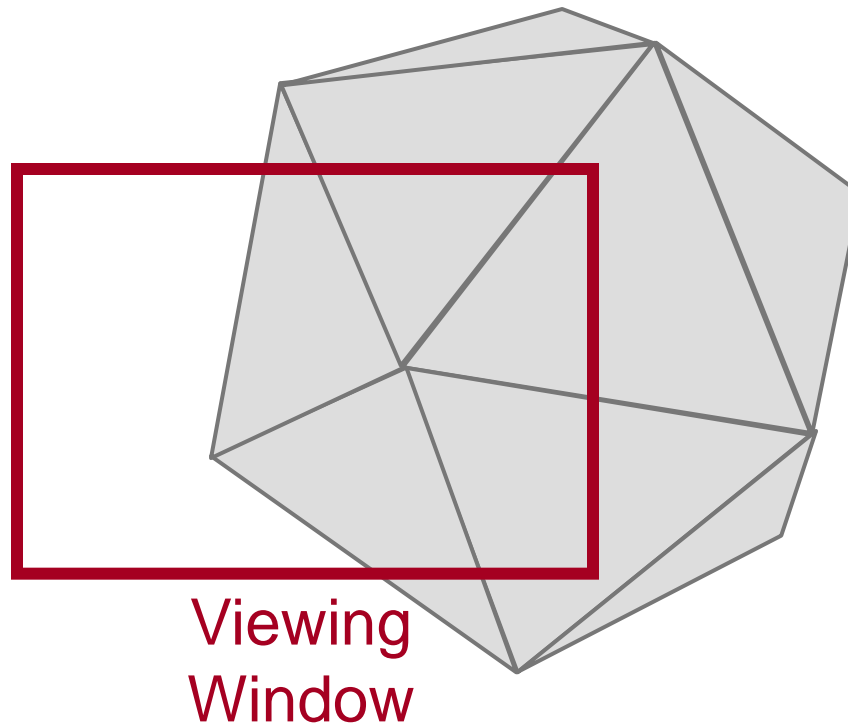
- Avoid drawing parts of primitives outside window
  - Window defines part of scene being viewed
  - Must draw geometric primitives only inside window



Screen Coordinates

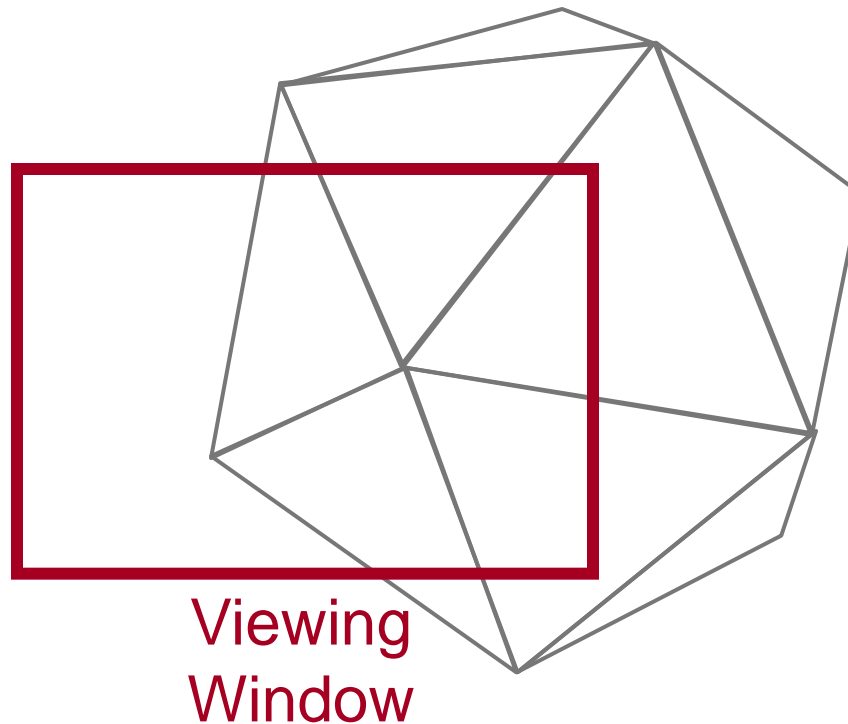
# Clipping

- Avoid drawing parts of primitives outside window
  - Window defines part of scene being viewed
  - Must draw geometric primitives only inside window



# Clipping

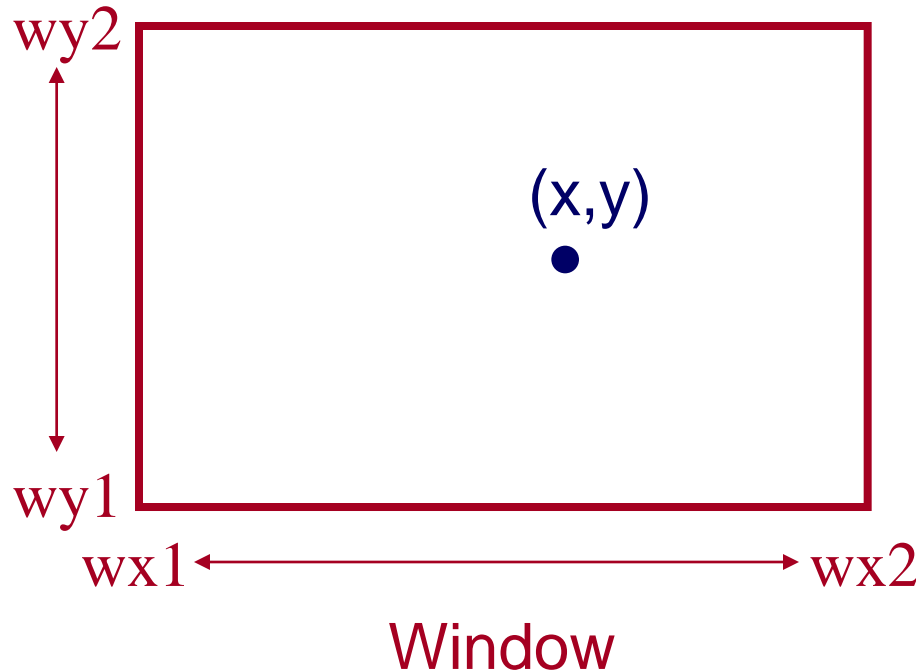
- Avoid drawing parts of primitives outside window
  - Points
  - Lines
  - Polygons
  - Circles
  - etc.





# Point Clipping

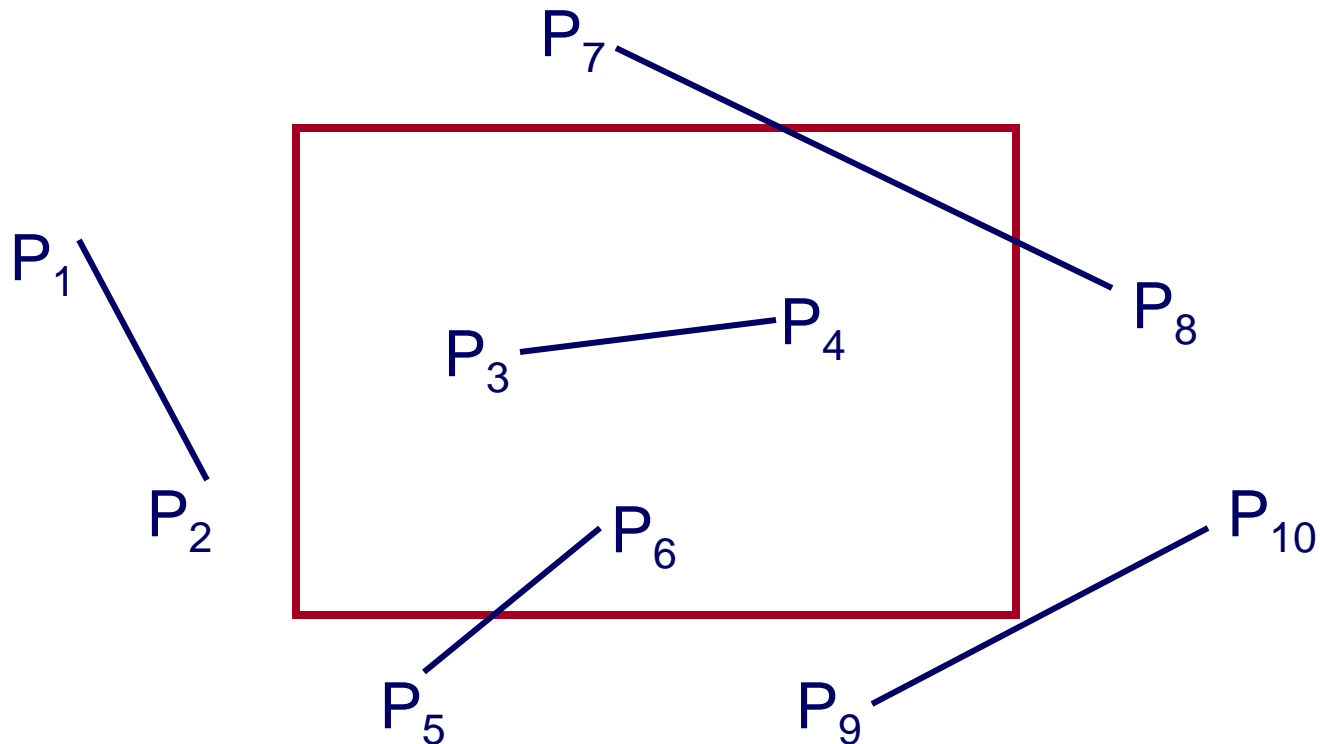
- Is point  $(x,y)$  inside the clip window?



```
inside =  
  (x >= wx1) &&  
  (x <= wx2) &&  
  (y >= wy1) &&  
  (y <= wy2) ;
```

# Line Clipping

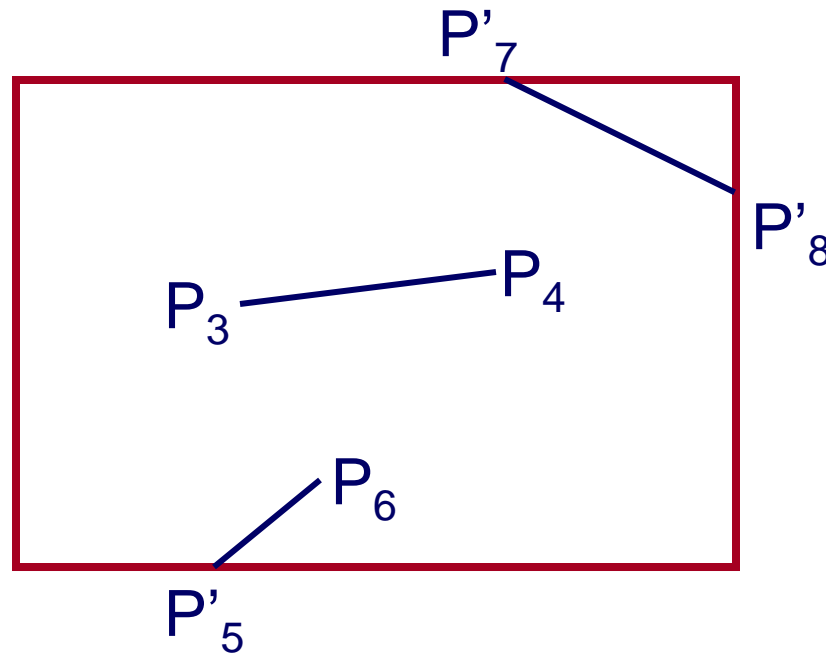
- Find the part of a line inside the clip window



Before Clipping

# Line Clipping

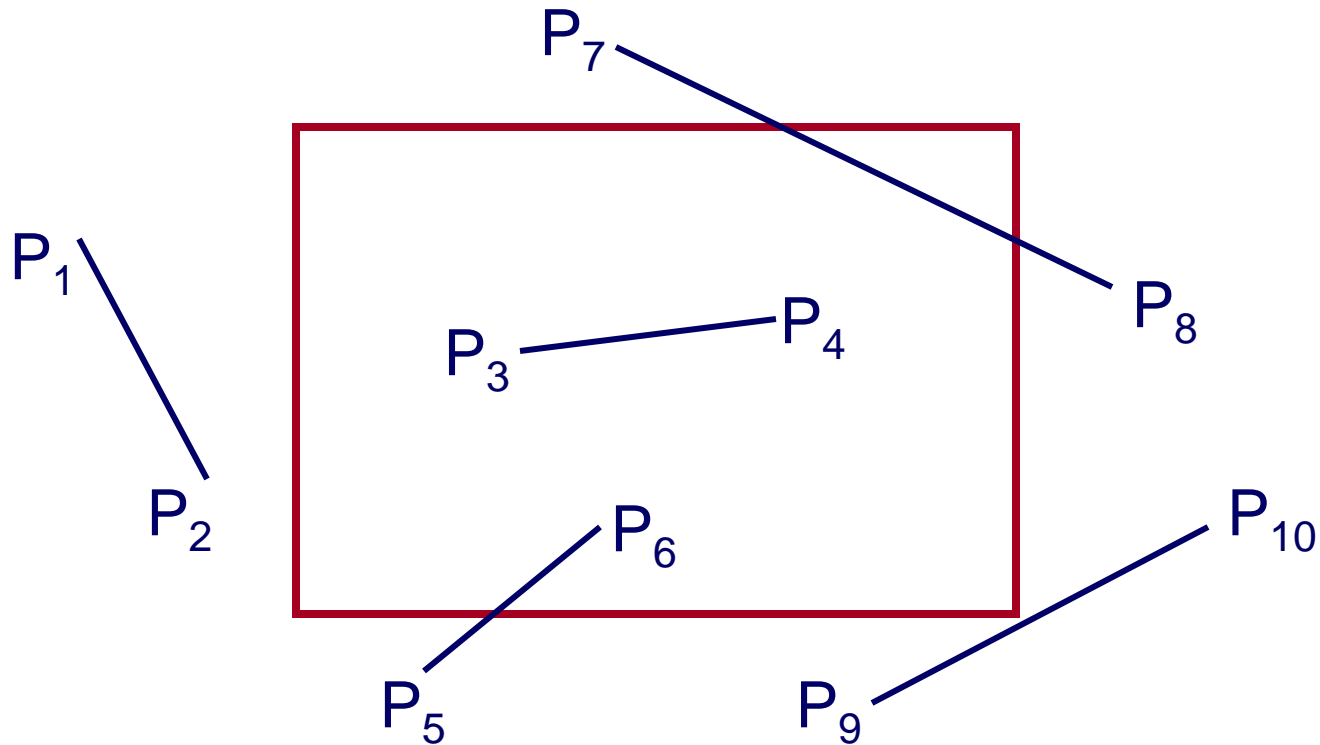
- Find the part of a line inside the clip window



After Clipping

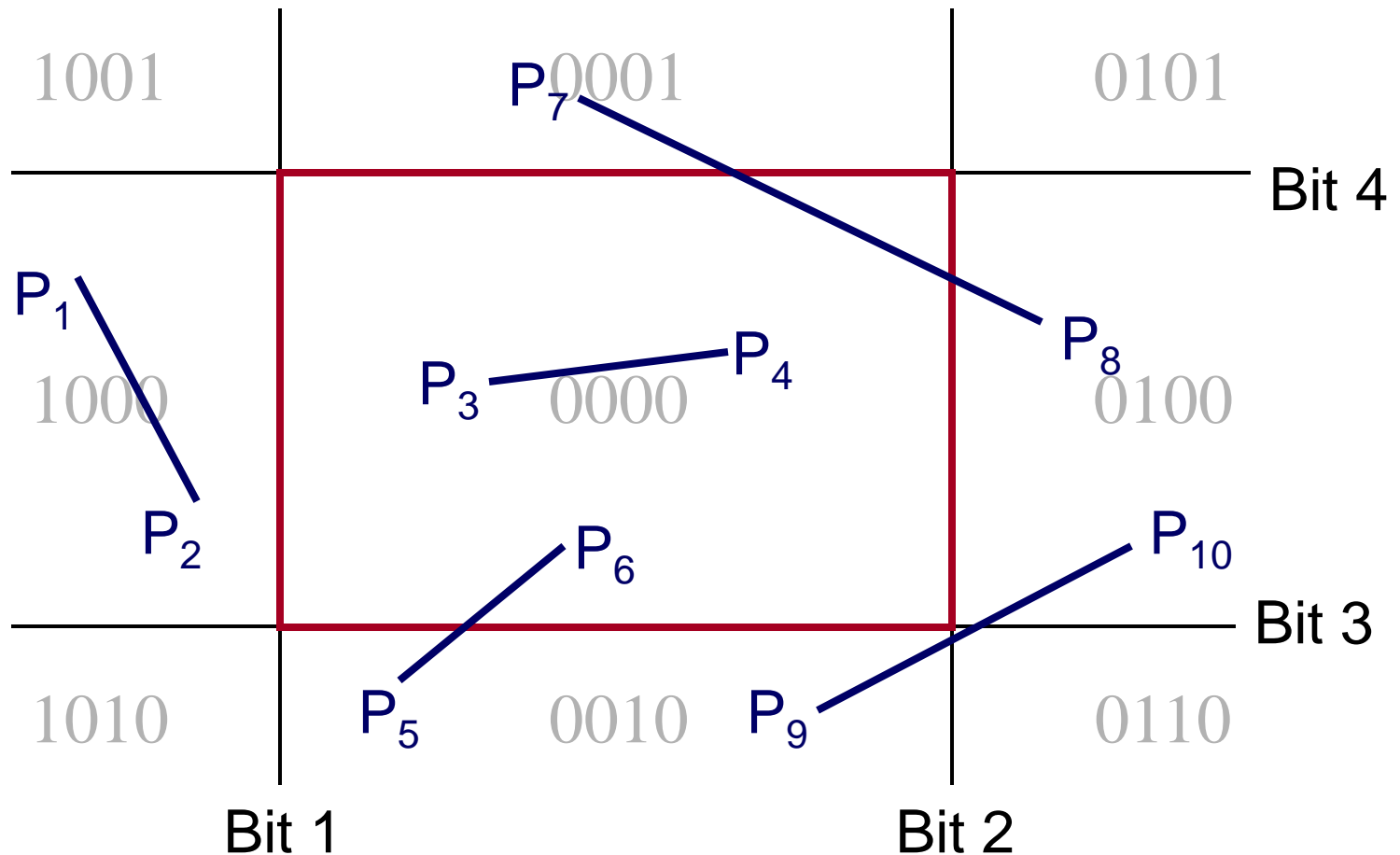
# Cohen Sutherland Line Clipping

- Use simple tests to classify easy cases first



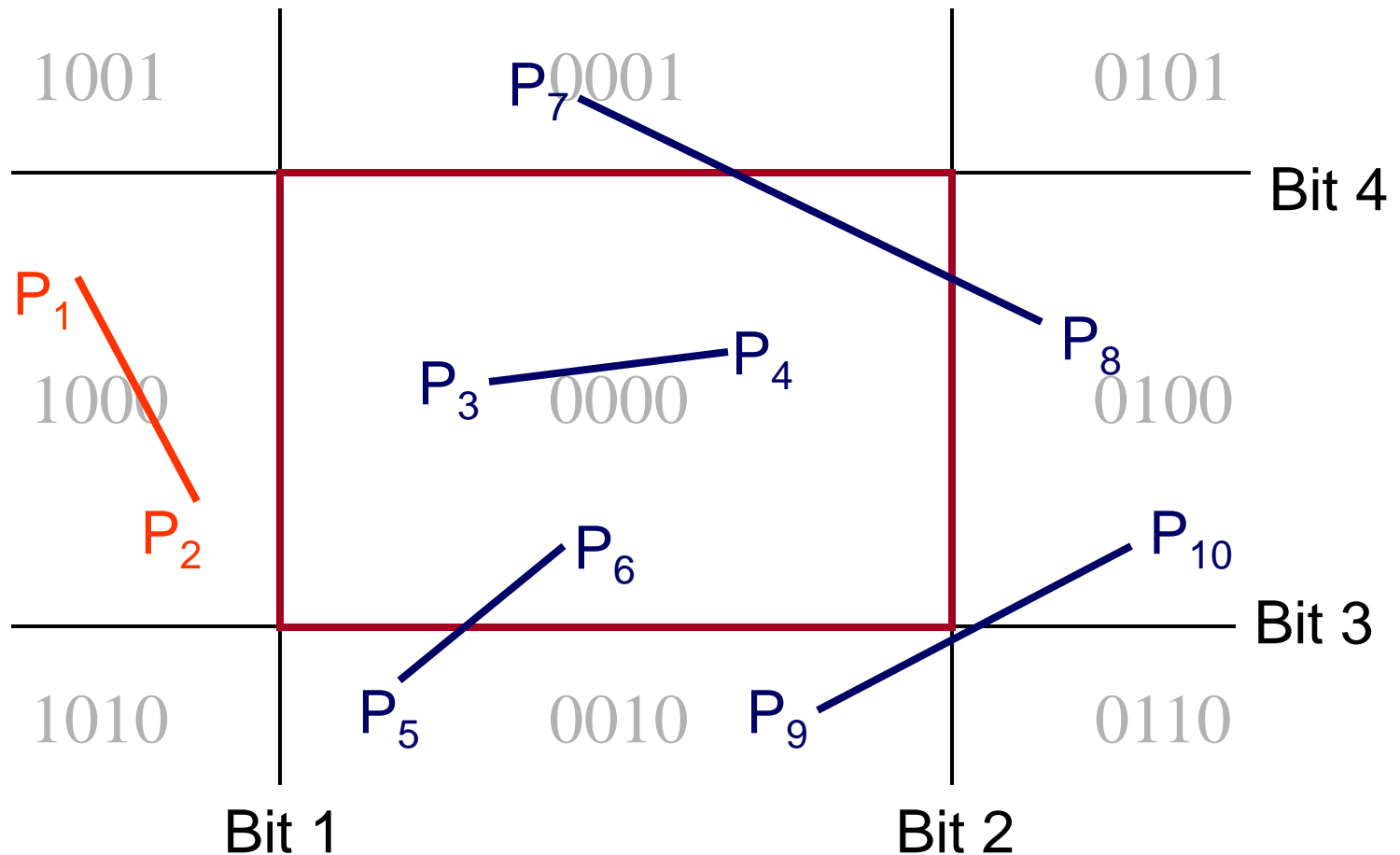
# Cohen Sutherland Line Clipping

- Classify some lines quickly by AND of bit codes representing regions of two endpoints (must be 0)



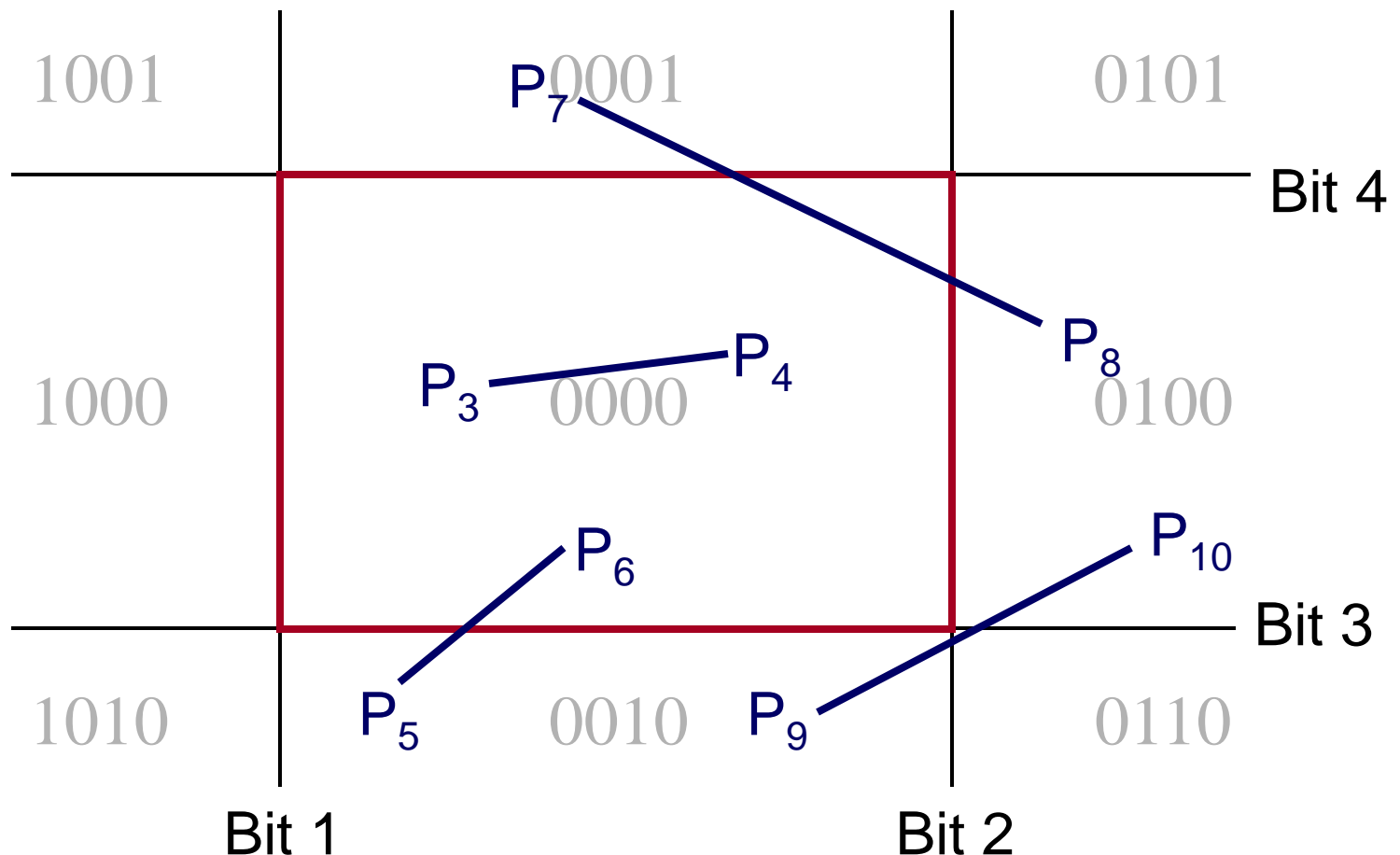
# Cohen Sutherland Line Clipping

- Classify some lines quickly by AND of bit codes representing regions of two endpoints (must be 0)



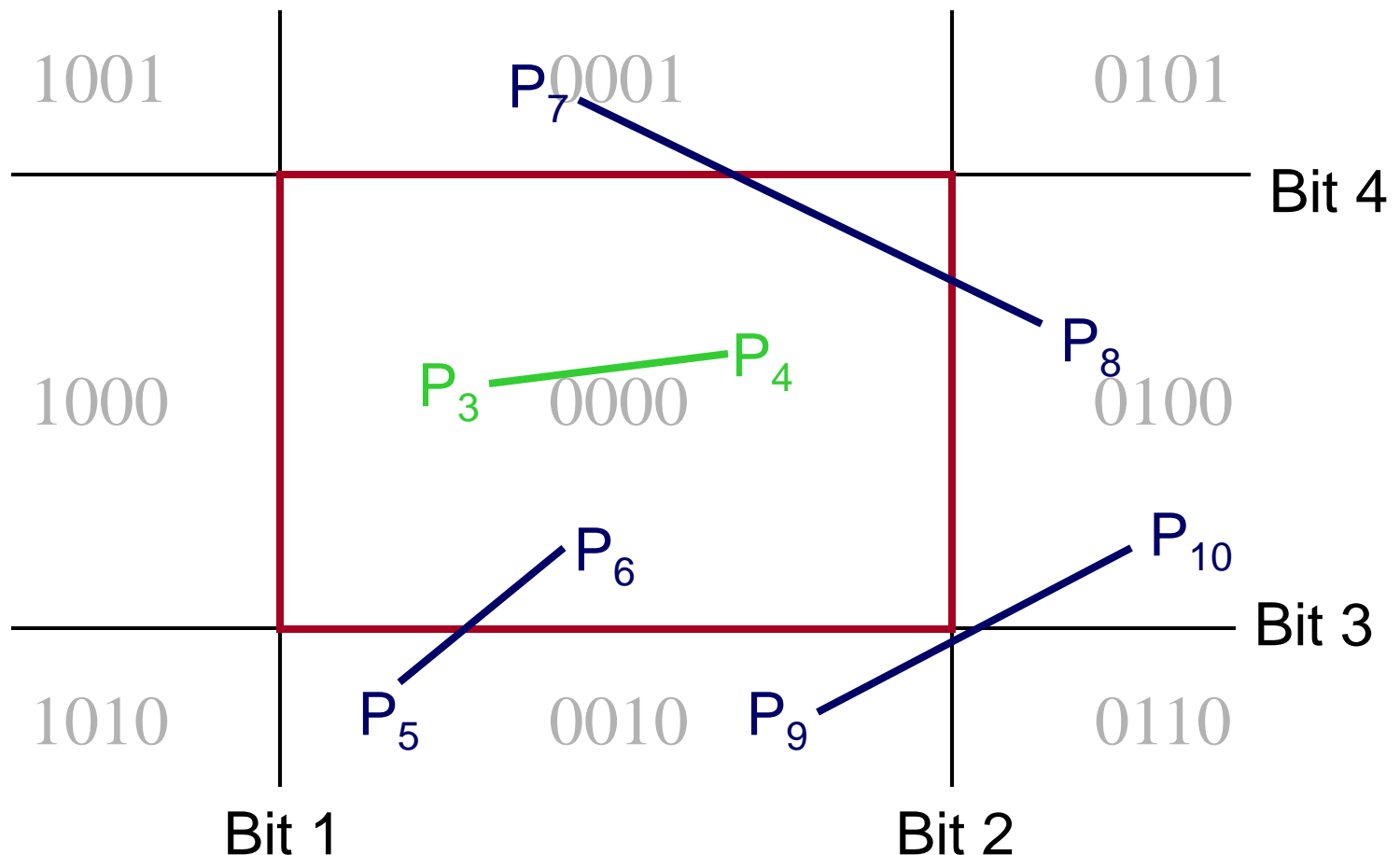
# Cohen Sutherland Line Clipping

- Classify some lines quickly by AND of bit codes representing regions of two endpoints (must be 0)



# Cohen Sutherland Line Clipping

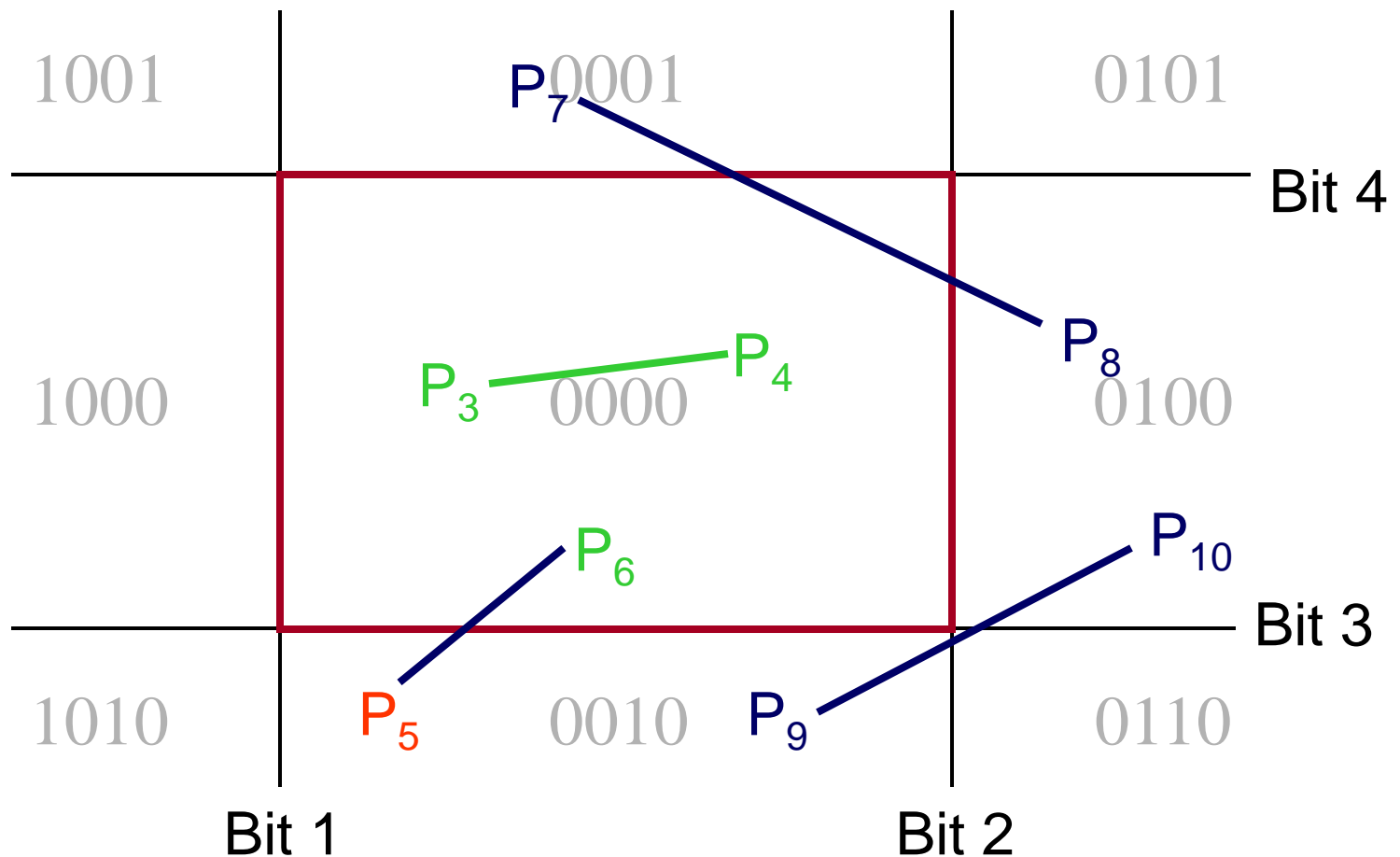
- Classify some lines quickly by AND of bit codes representing regions of two endpoints (must be 0)





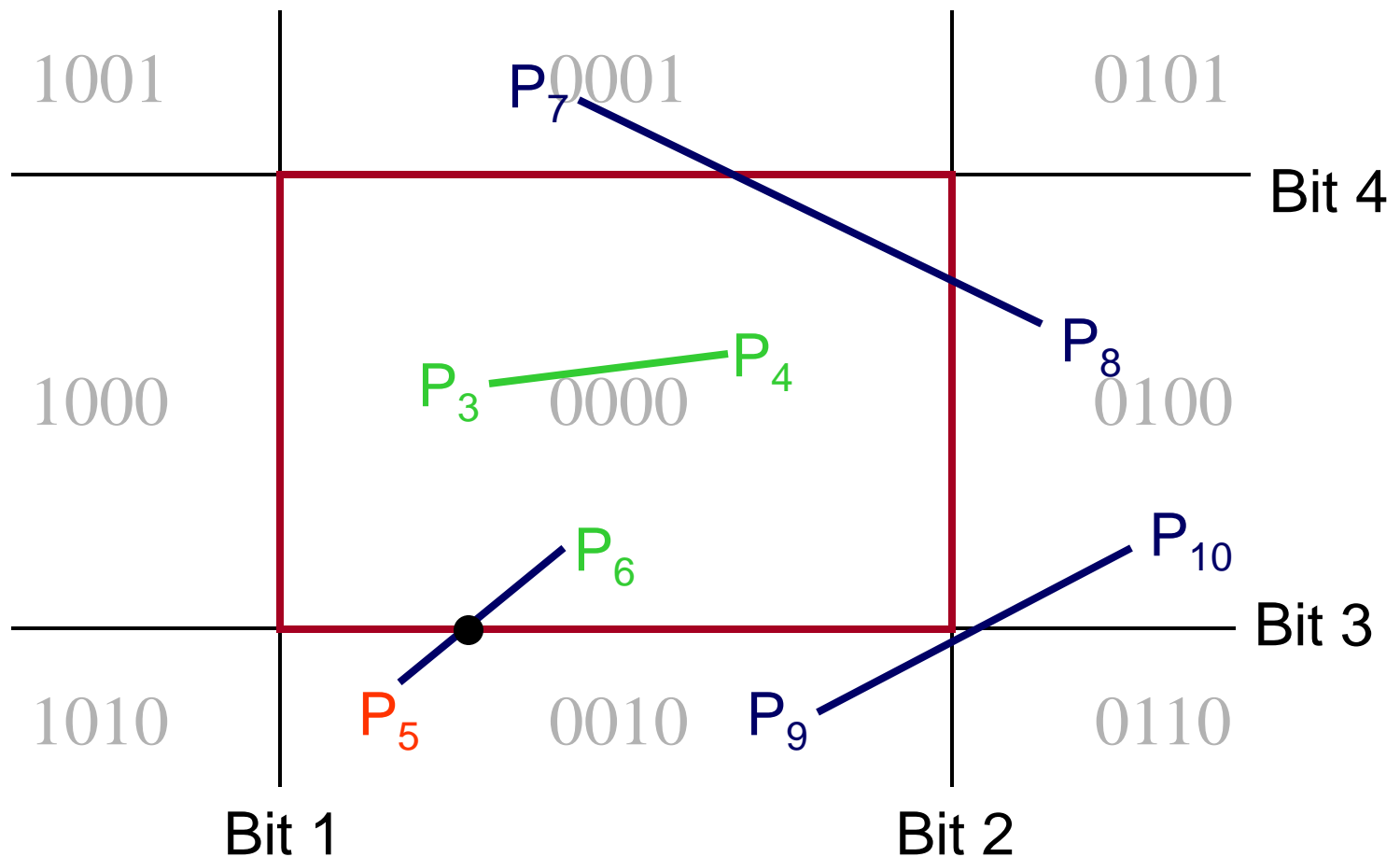
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



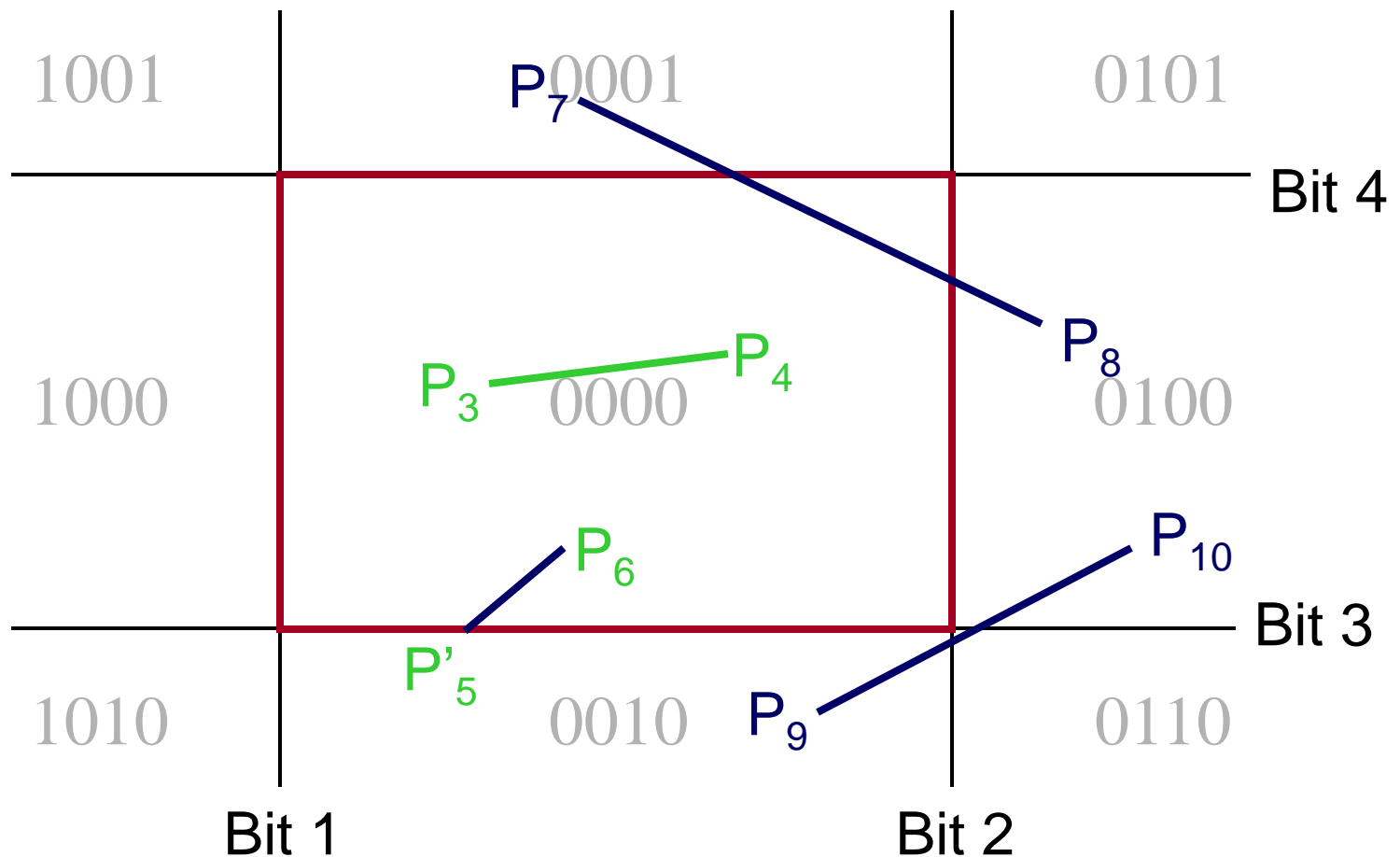
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



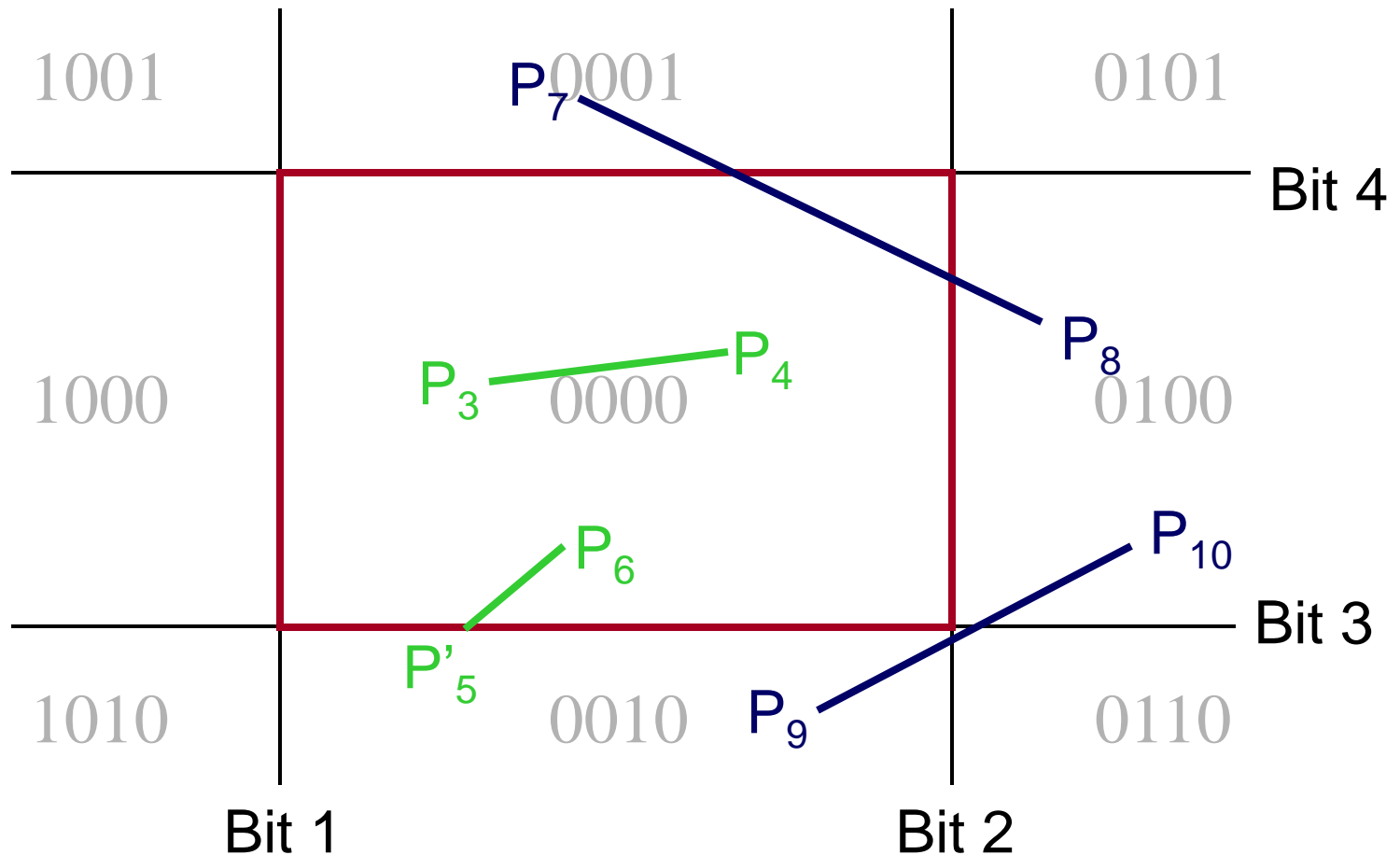
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



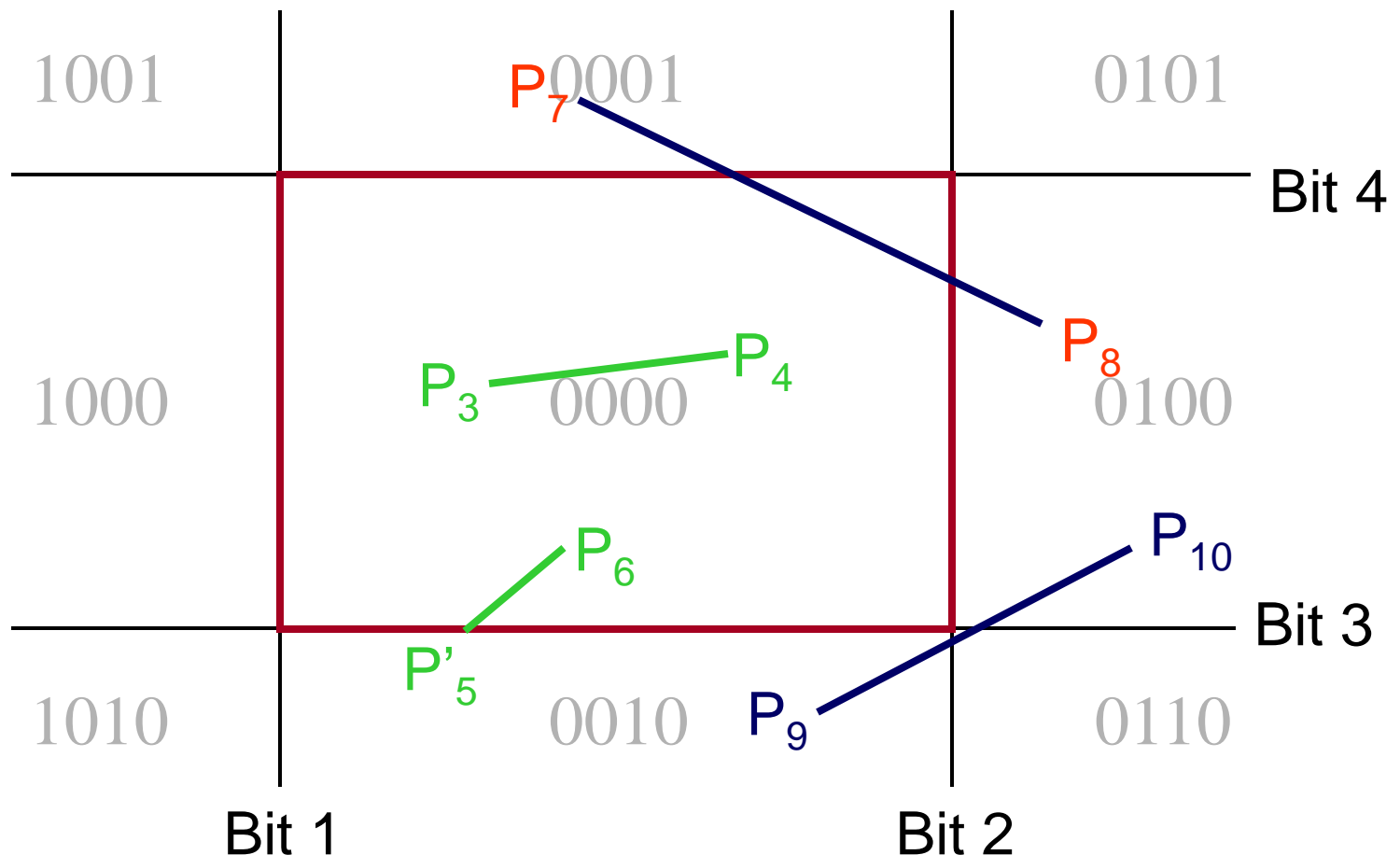
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



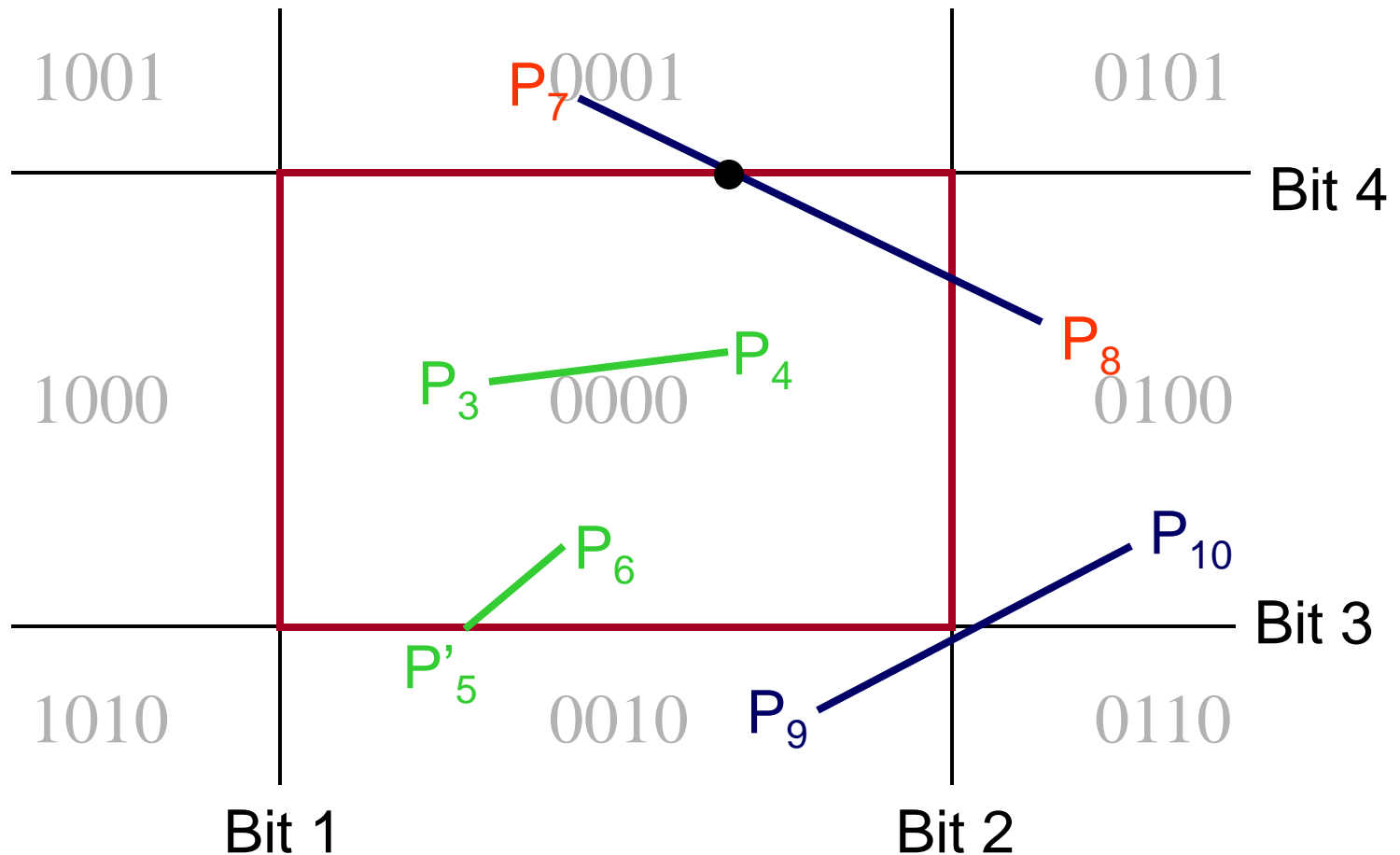
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



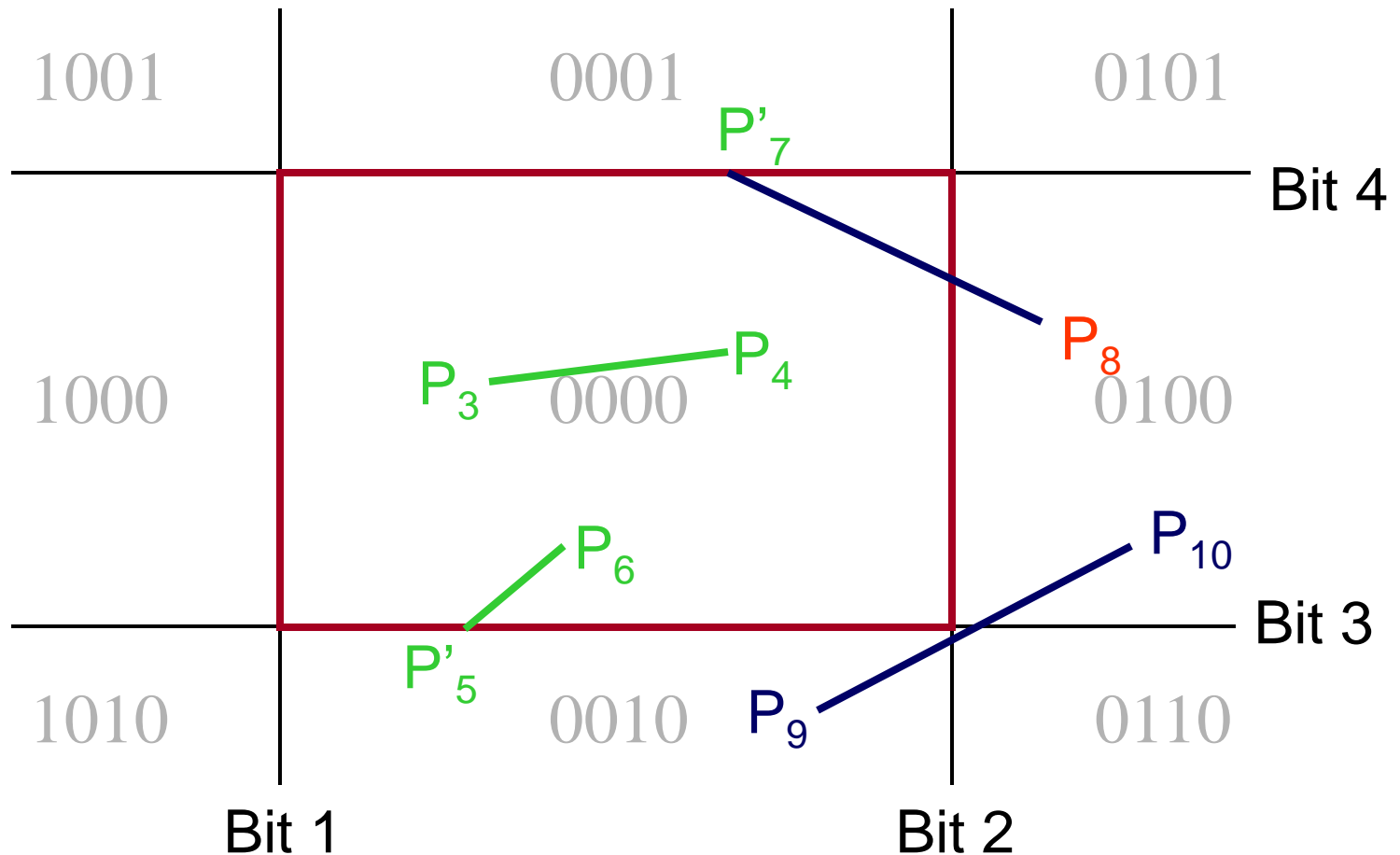
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



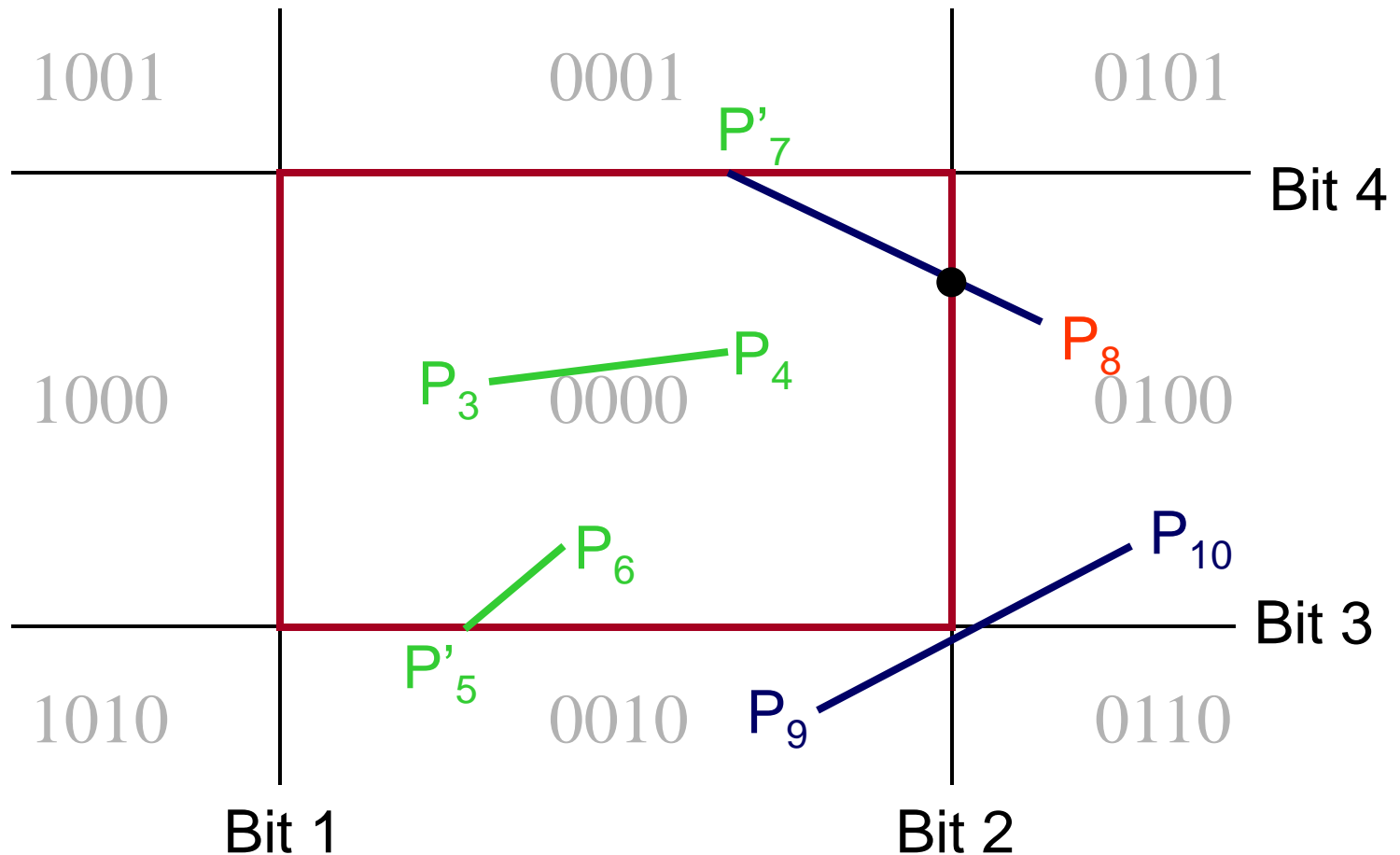
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



# Cohen-Sutherland Line Clipping

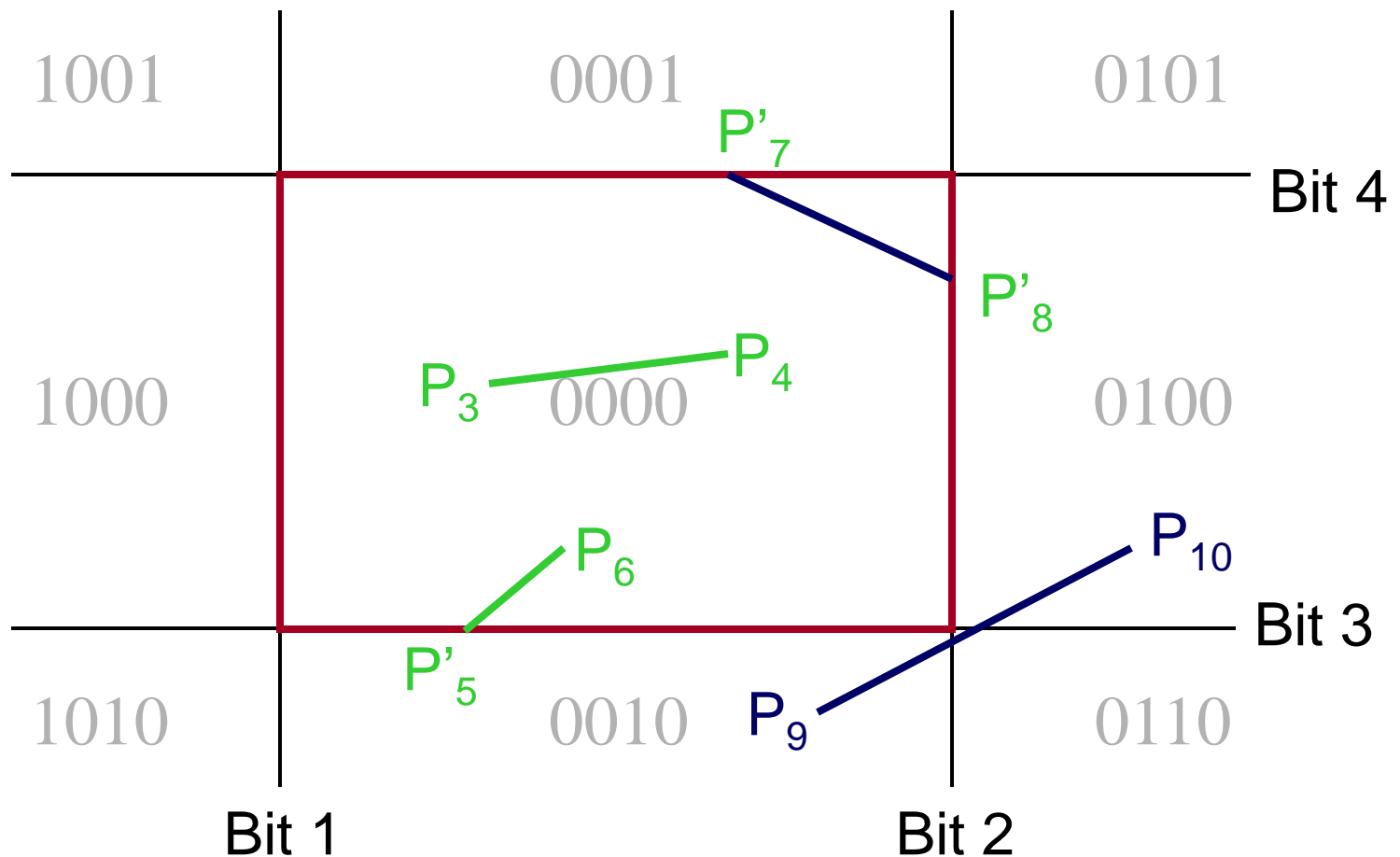
- Compute interesections with window boundary for lines that can't be classified quickly





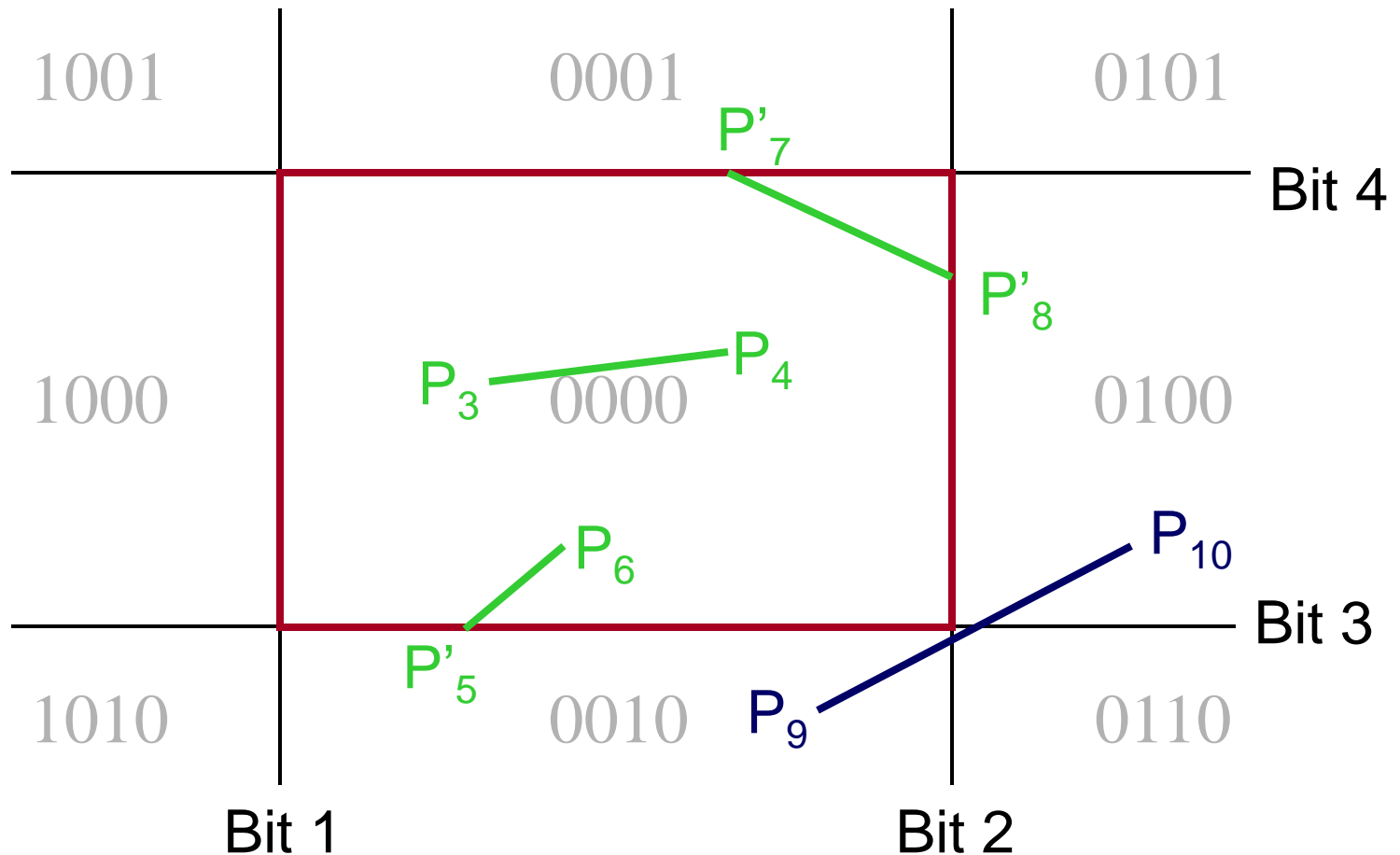
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



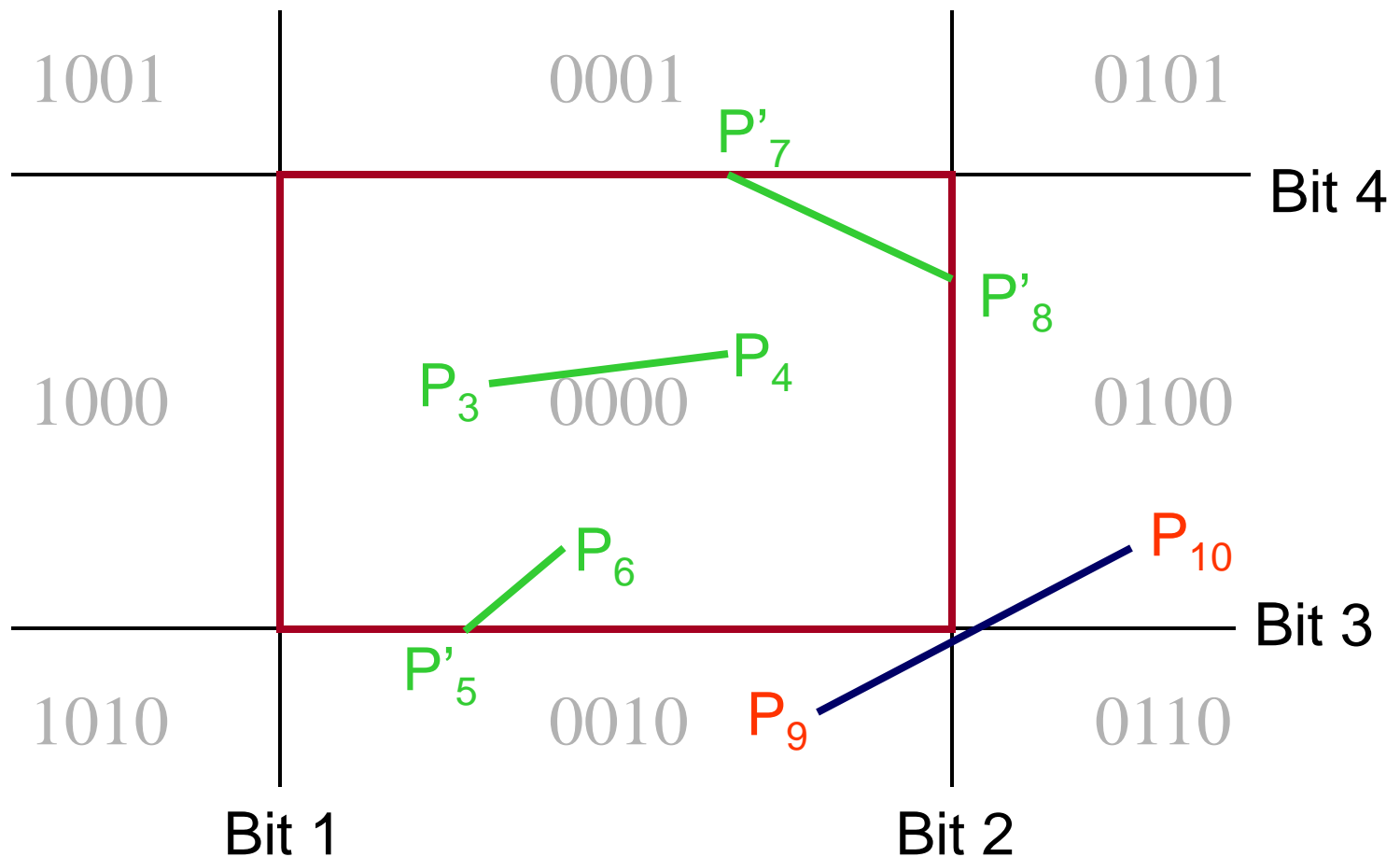
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



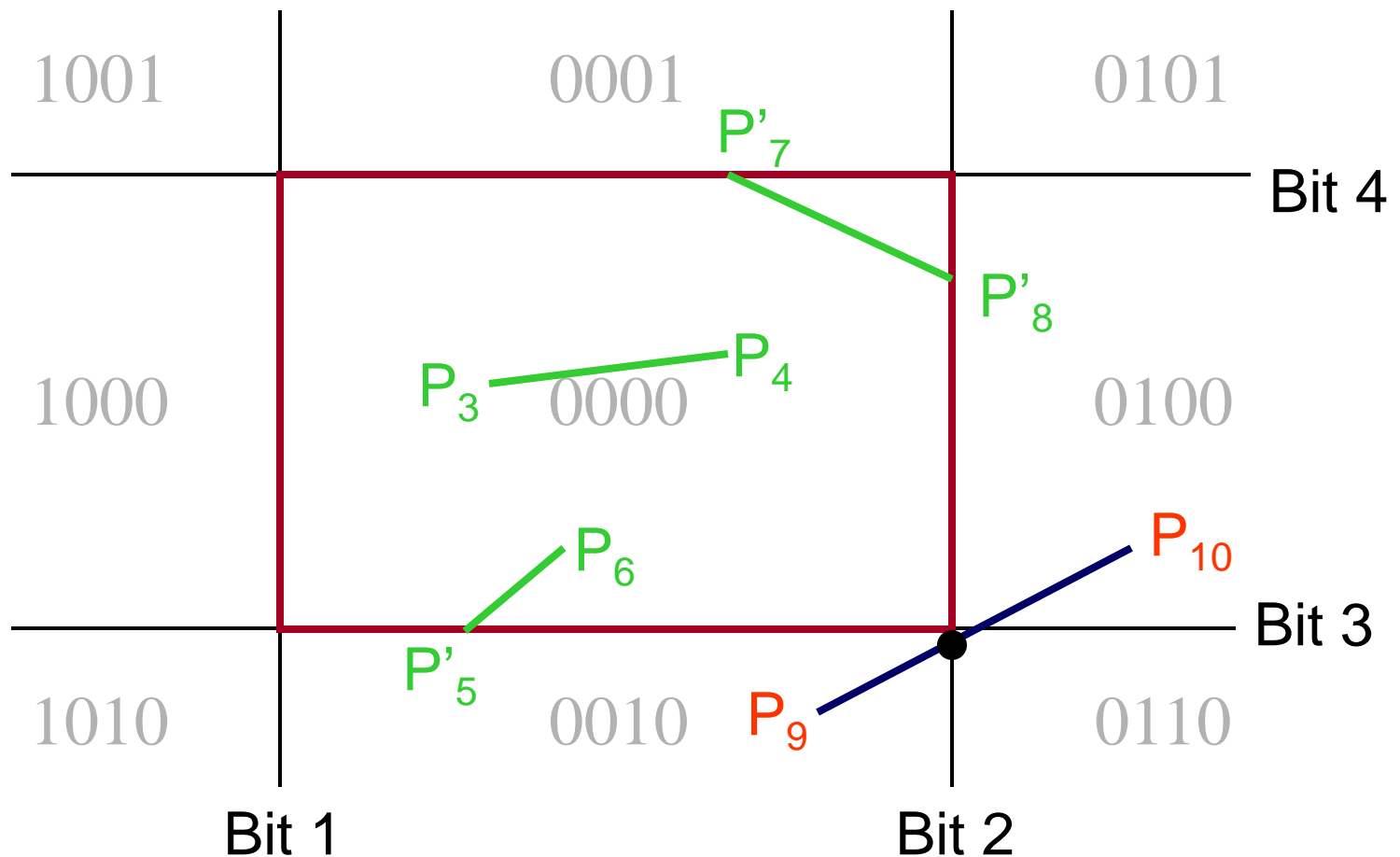
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



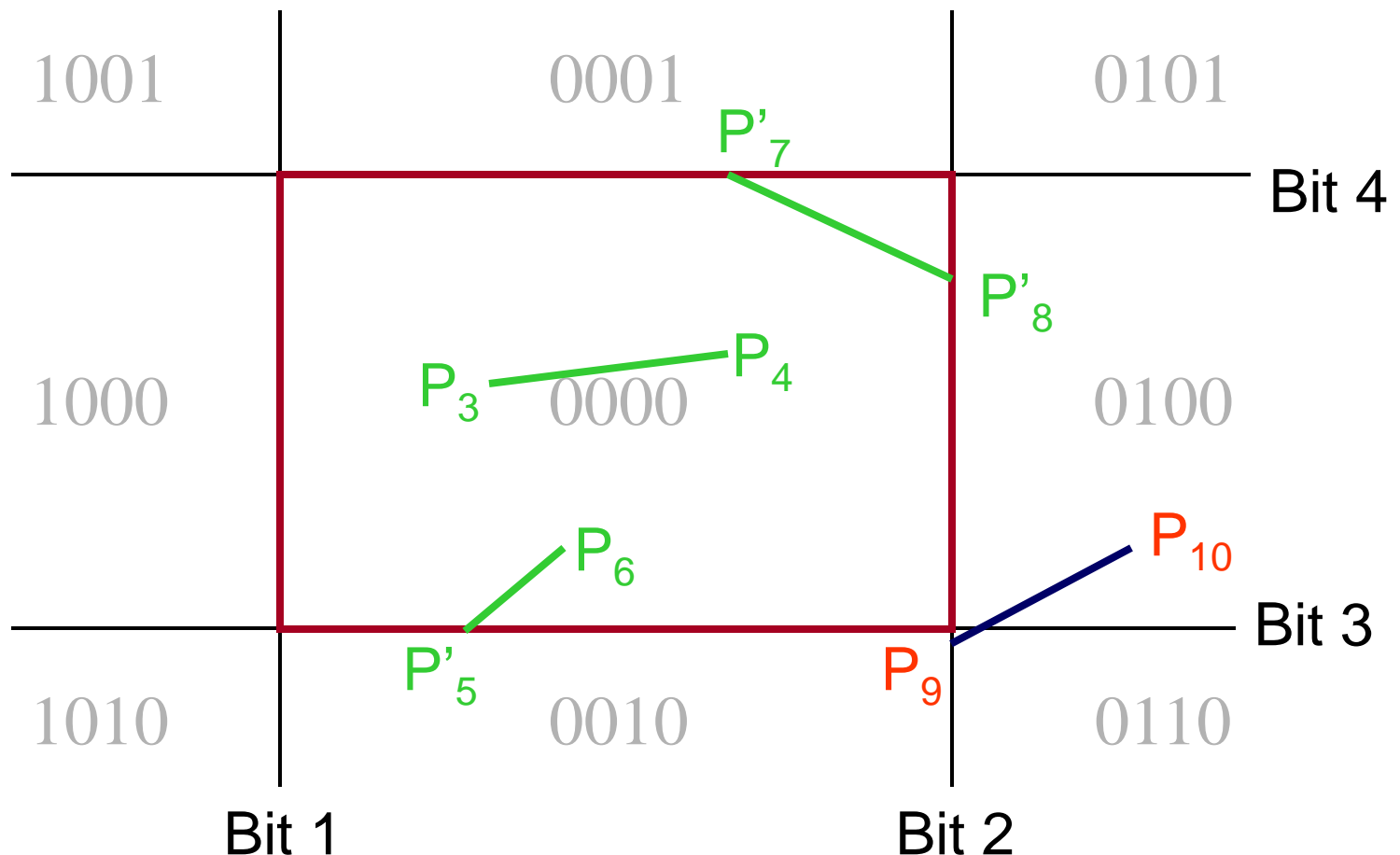
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



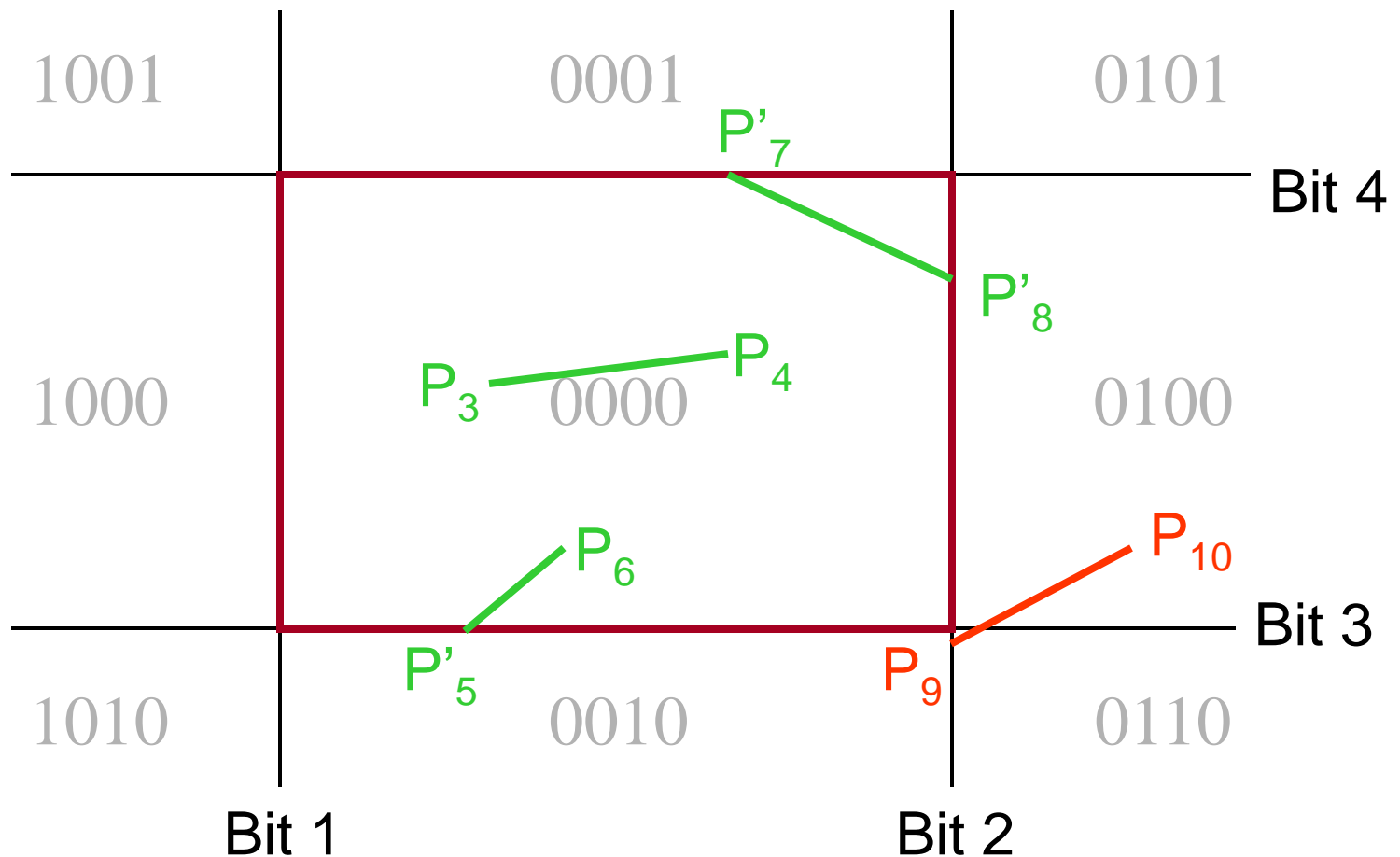
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



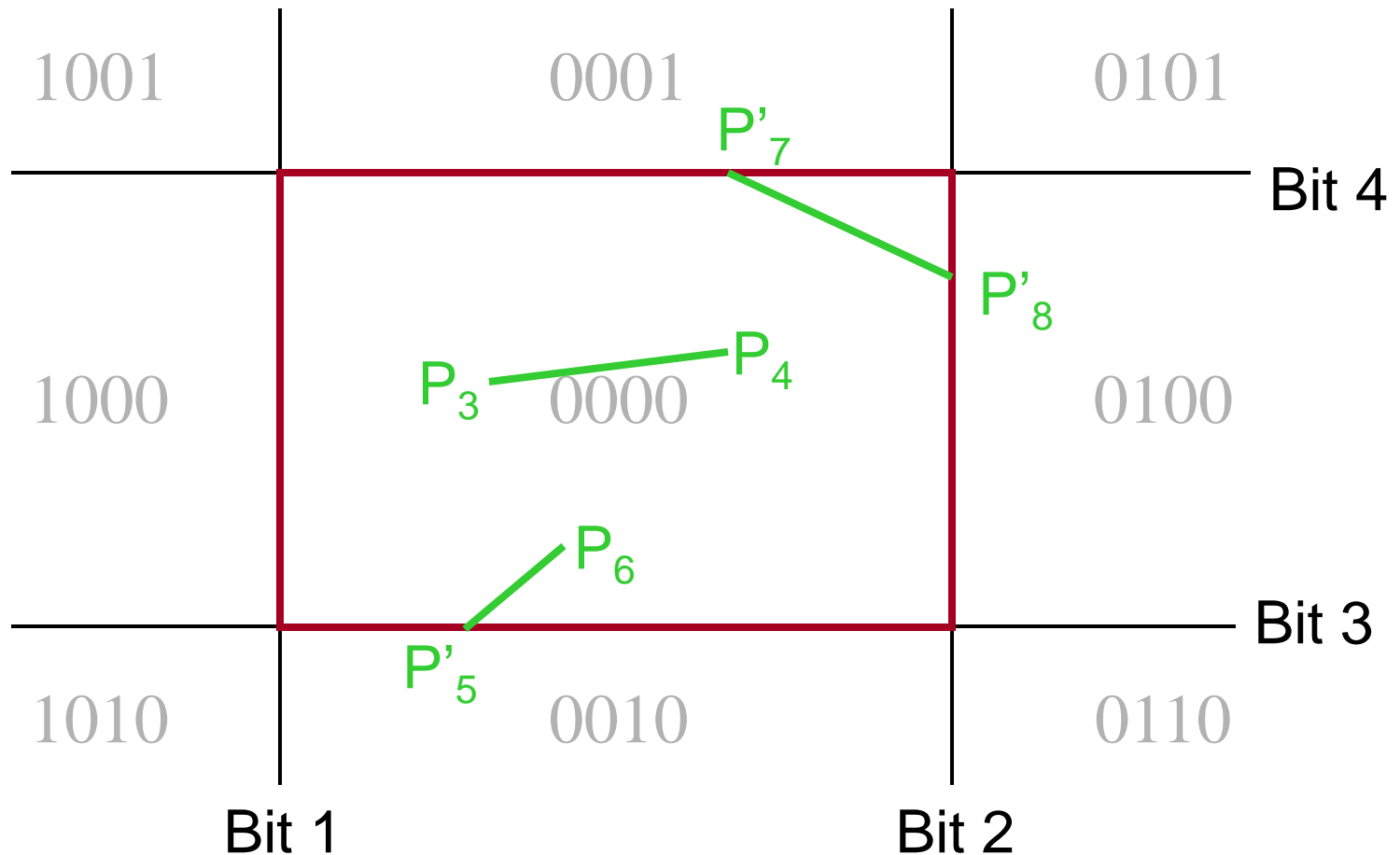
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



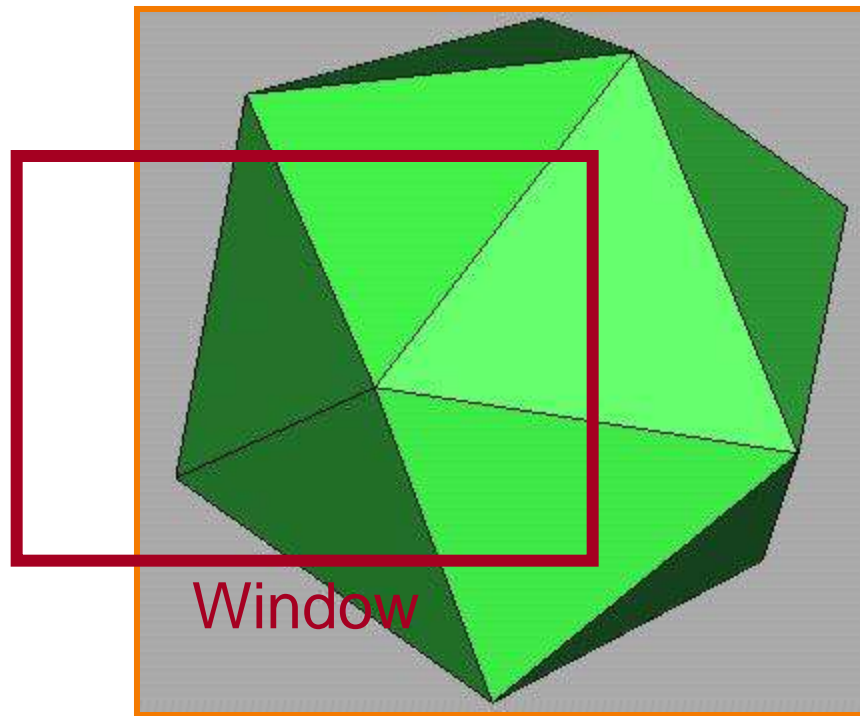
# Cohen-Sutherland Line Clipping

- Compute interesections with window boundary for lines that can't be classified quickly



# Clipping

- Avoid drawing parts of primitives outside window
  - Points
  - Lines
  - Polygons
  - Circles
  - etc.

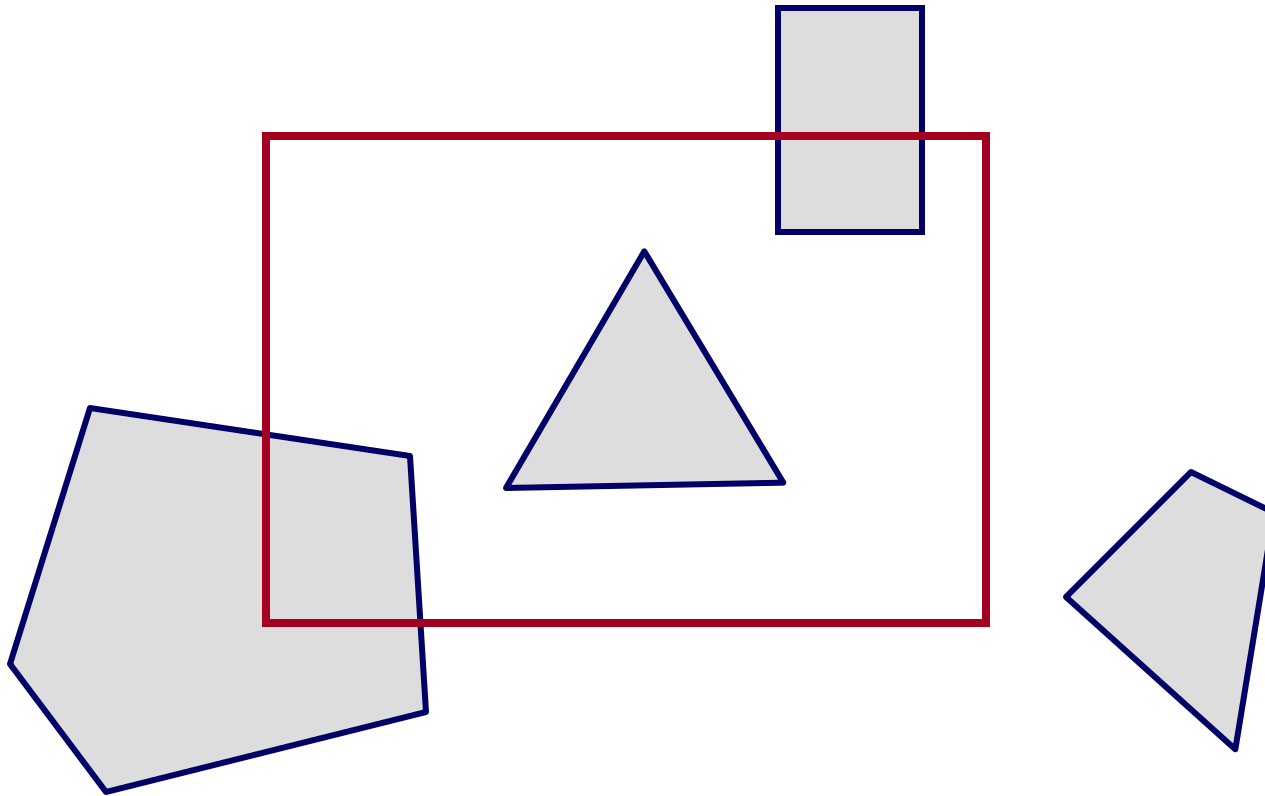


2D Screen Coordinates



# Polygon Clipping

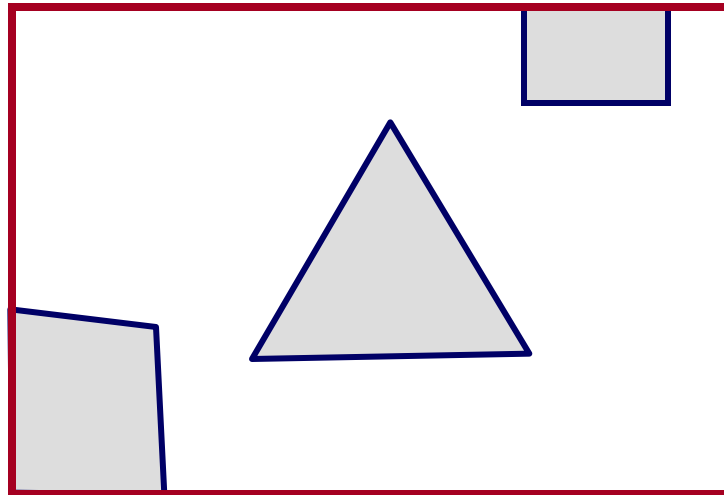
- Find the part of a polygon inside the clip window?



Before Clipping

# Polygon Clipping

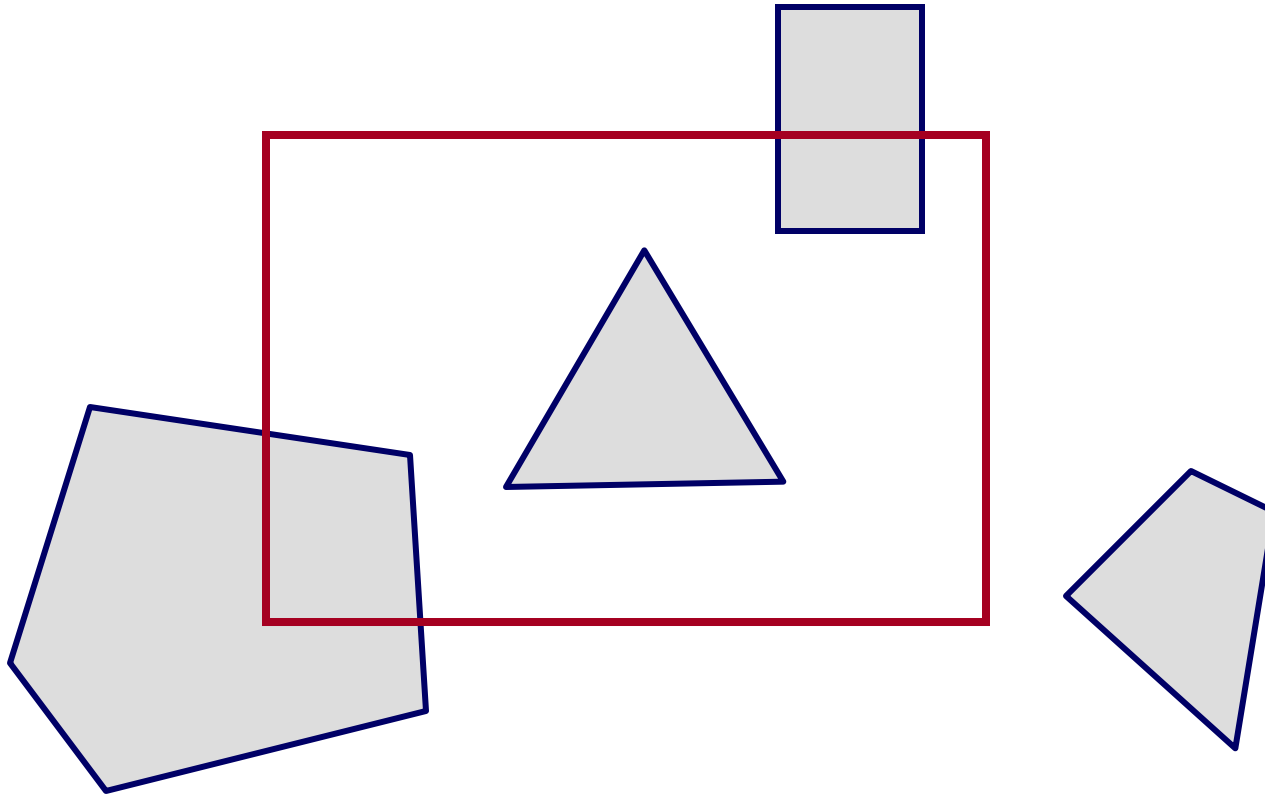
- Find the part of a polygon inside the clip window?



After Clipping

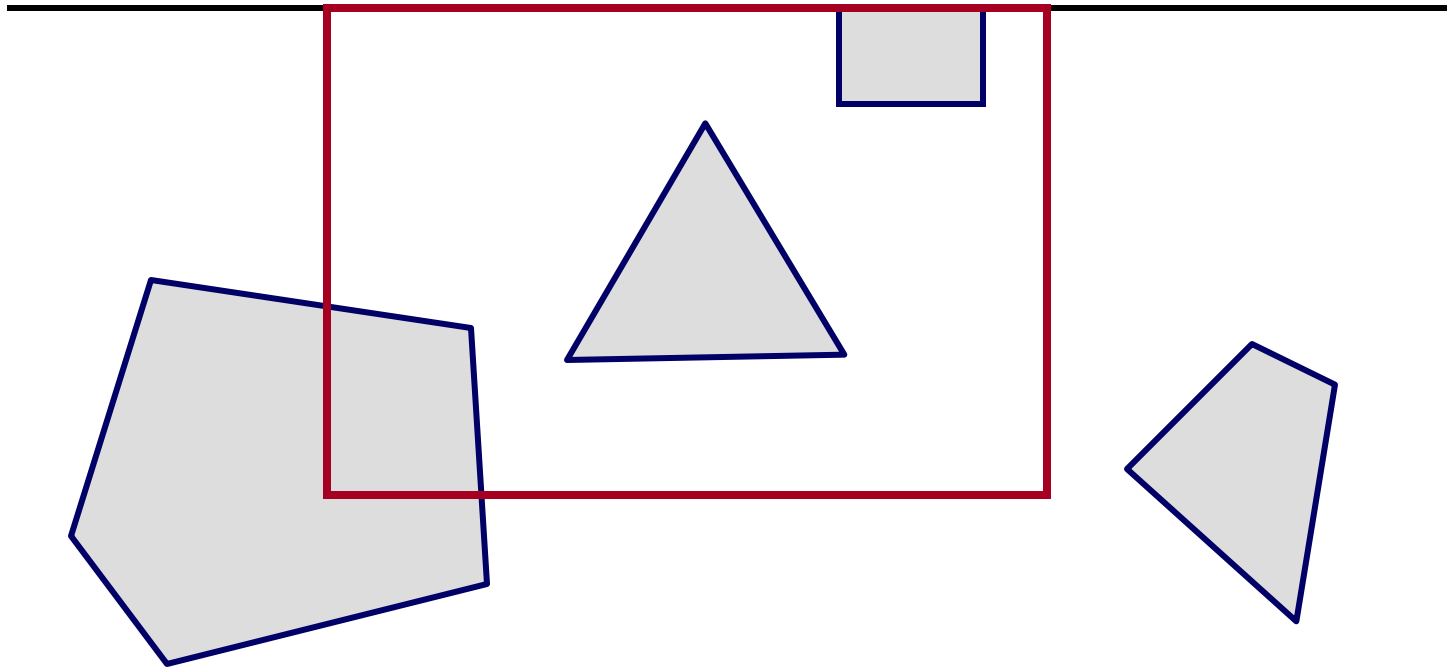
# Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time



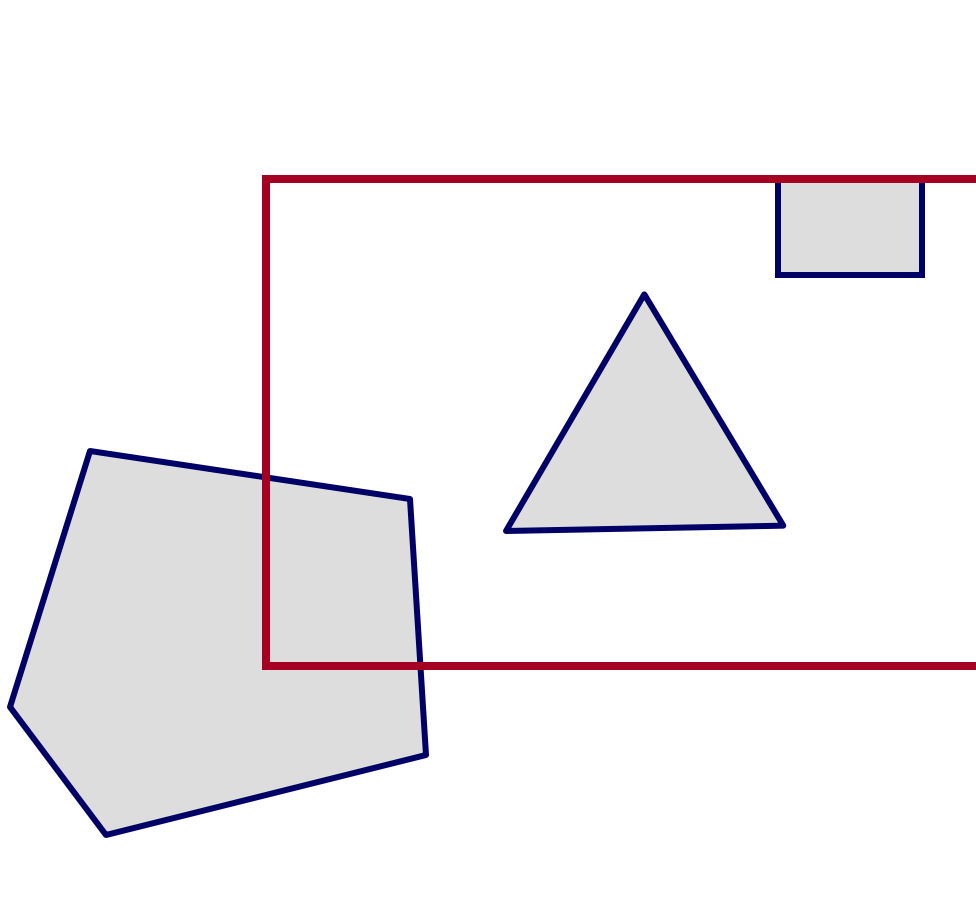
# Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time



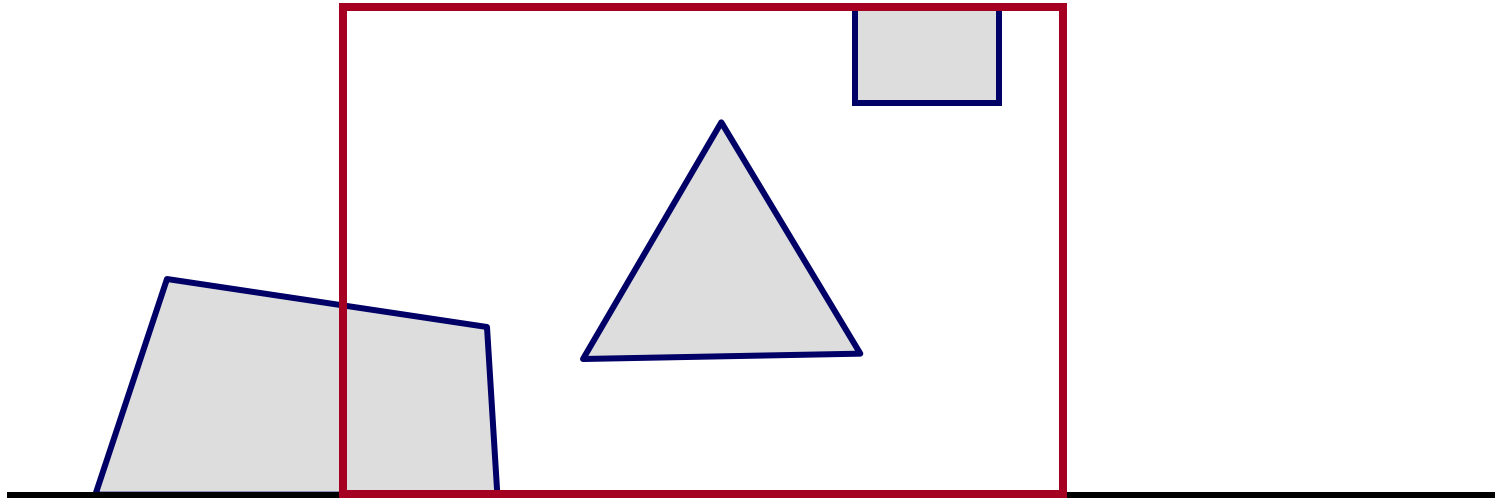
# Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time



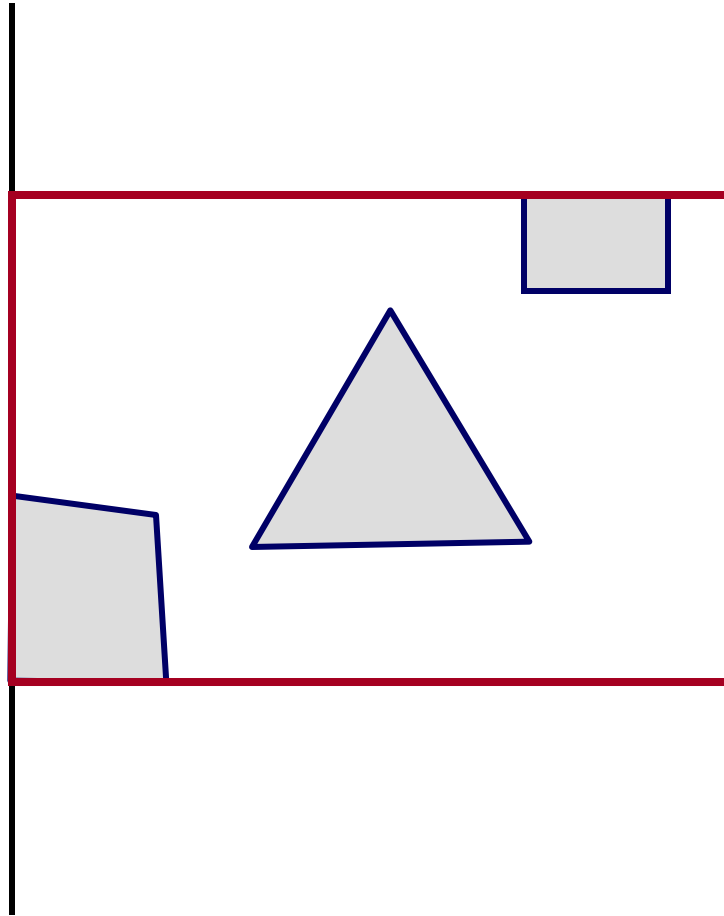
# Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time



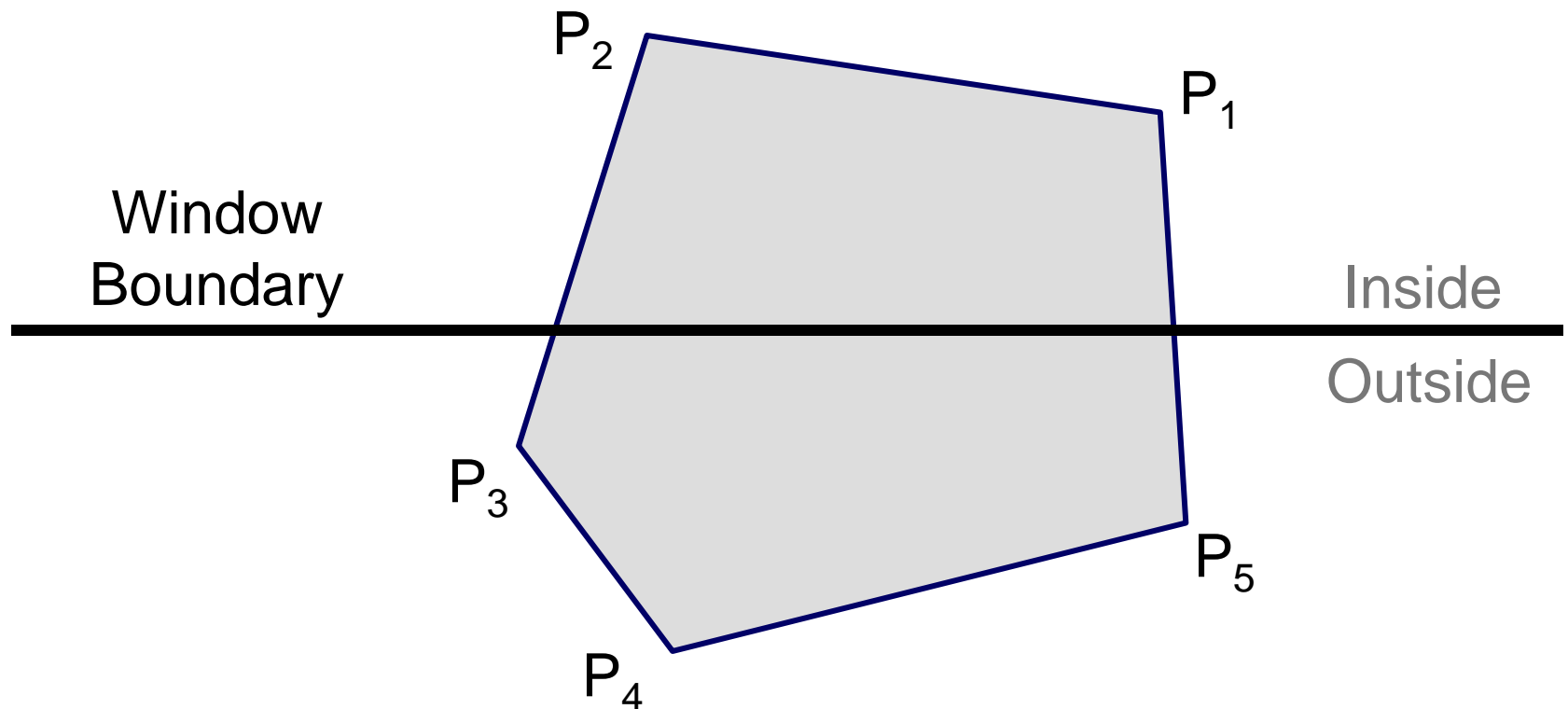
# Sutherland Hodgeman Clipping

- Clip to each window boundary one at a time



# Clipping to a Boundary

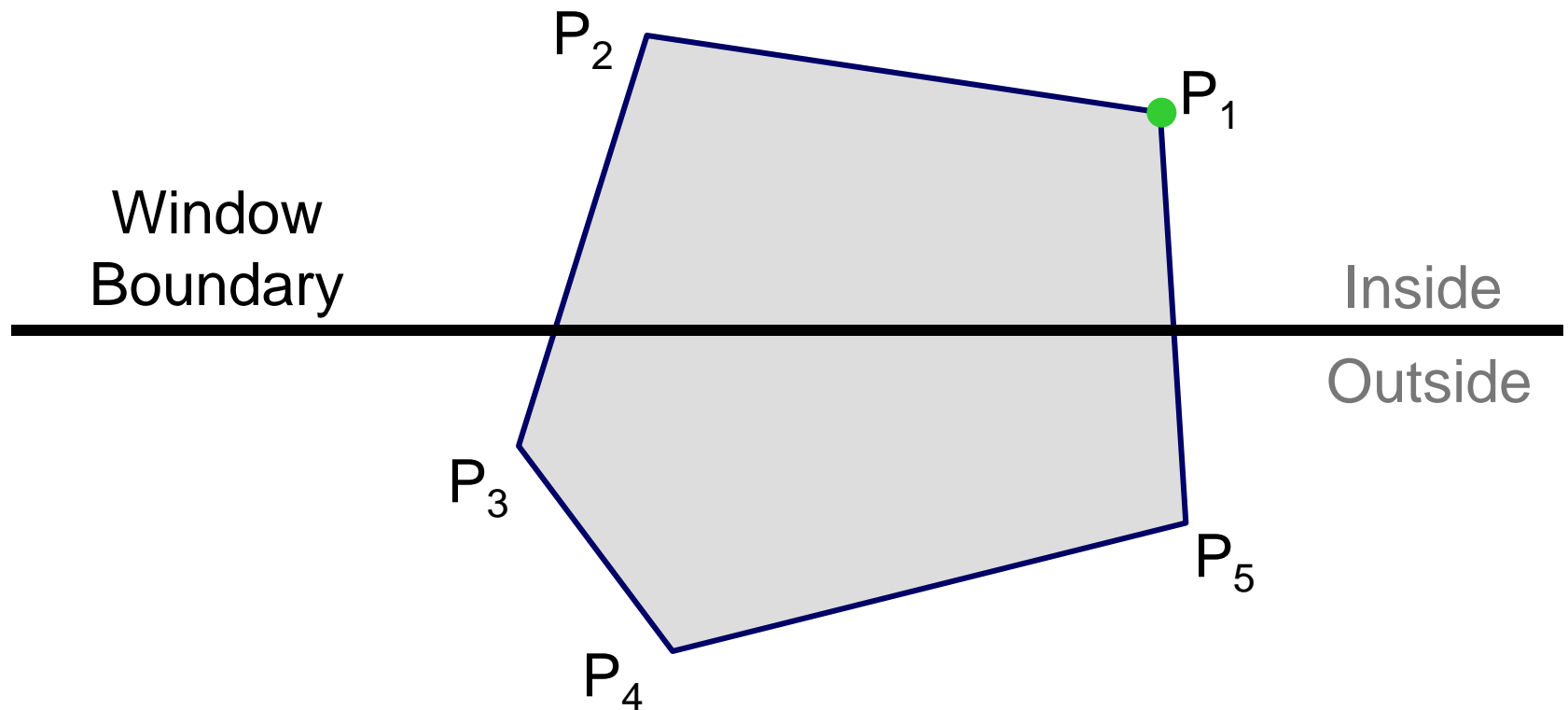
- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary





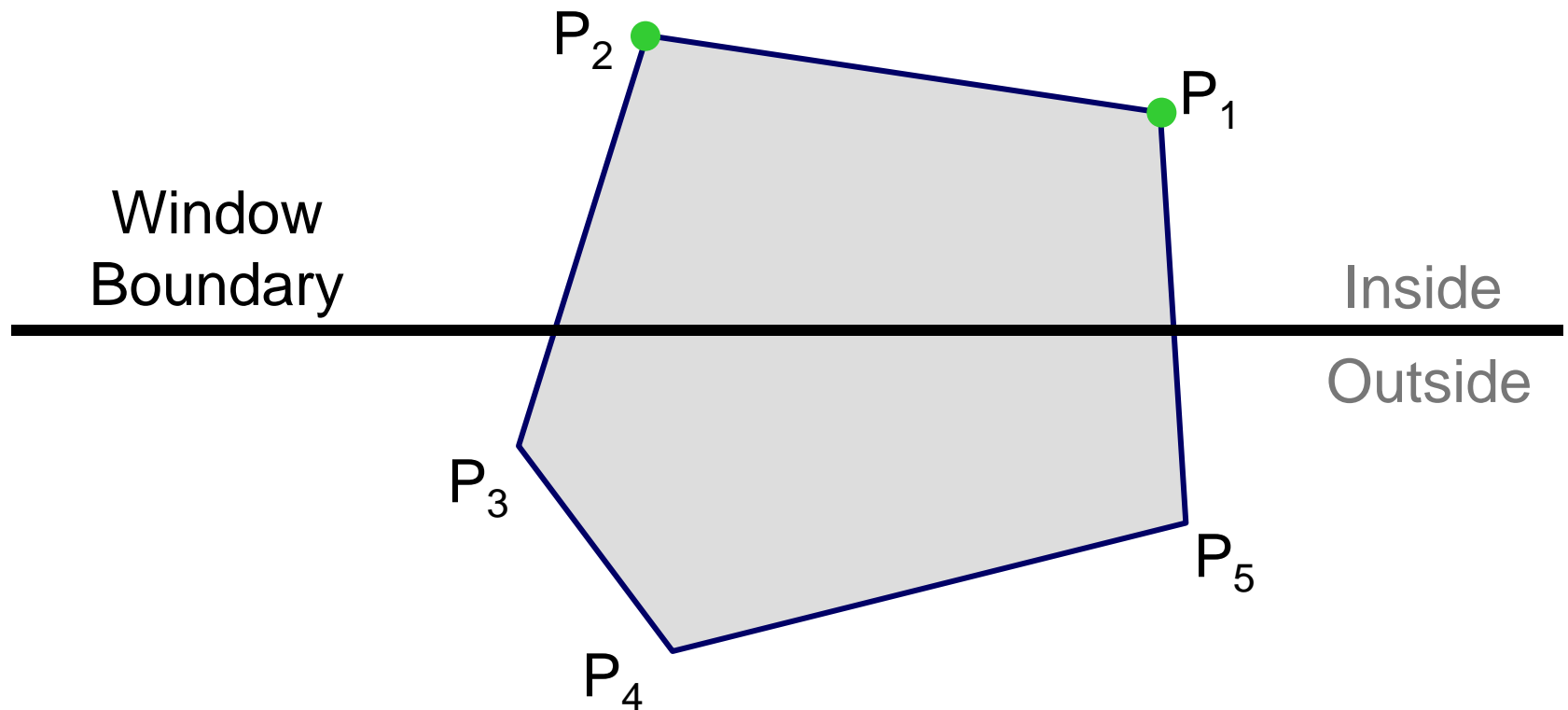
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



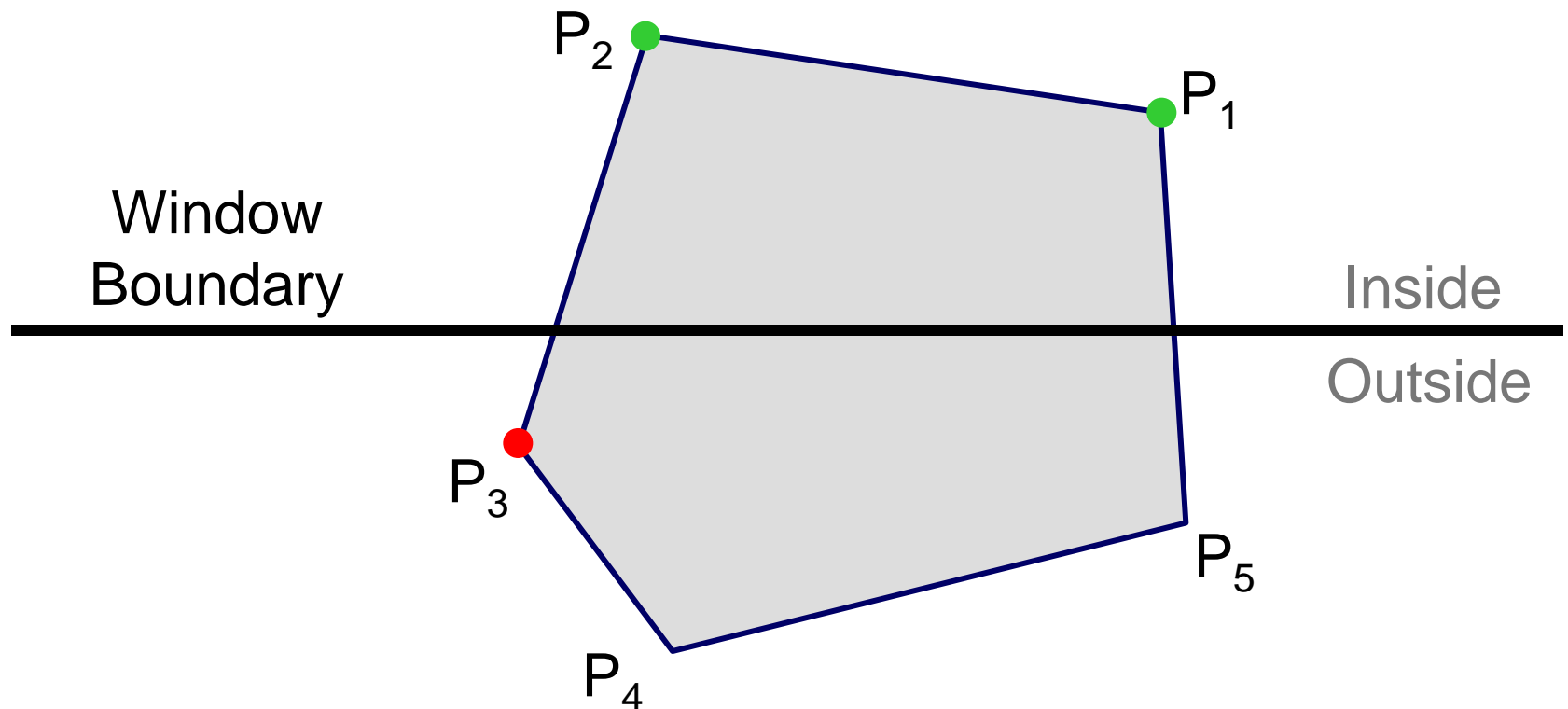
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



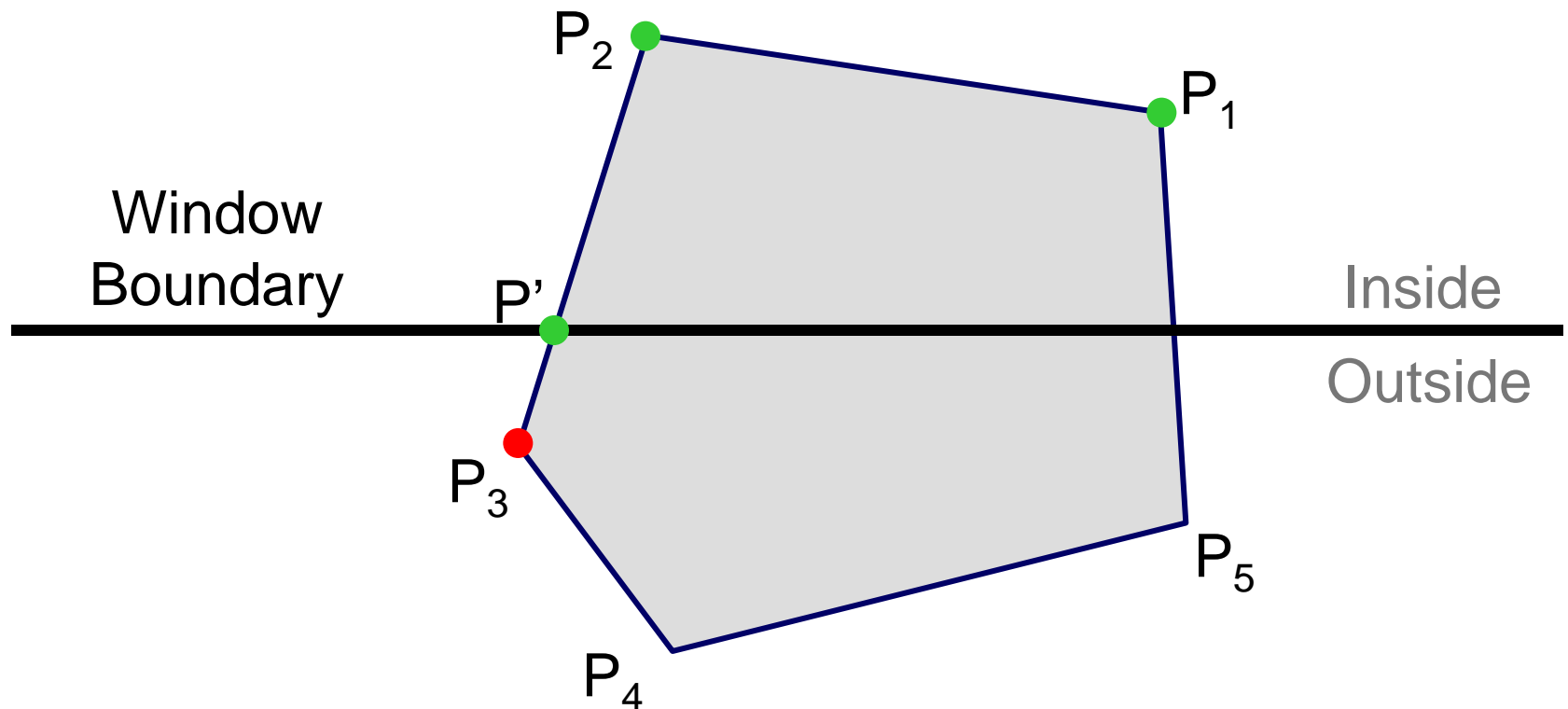
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



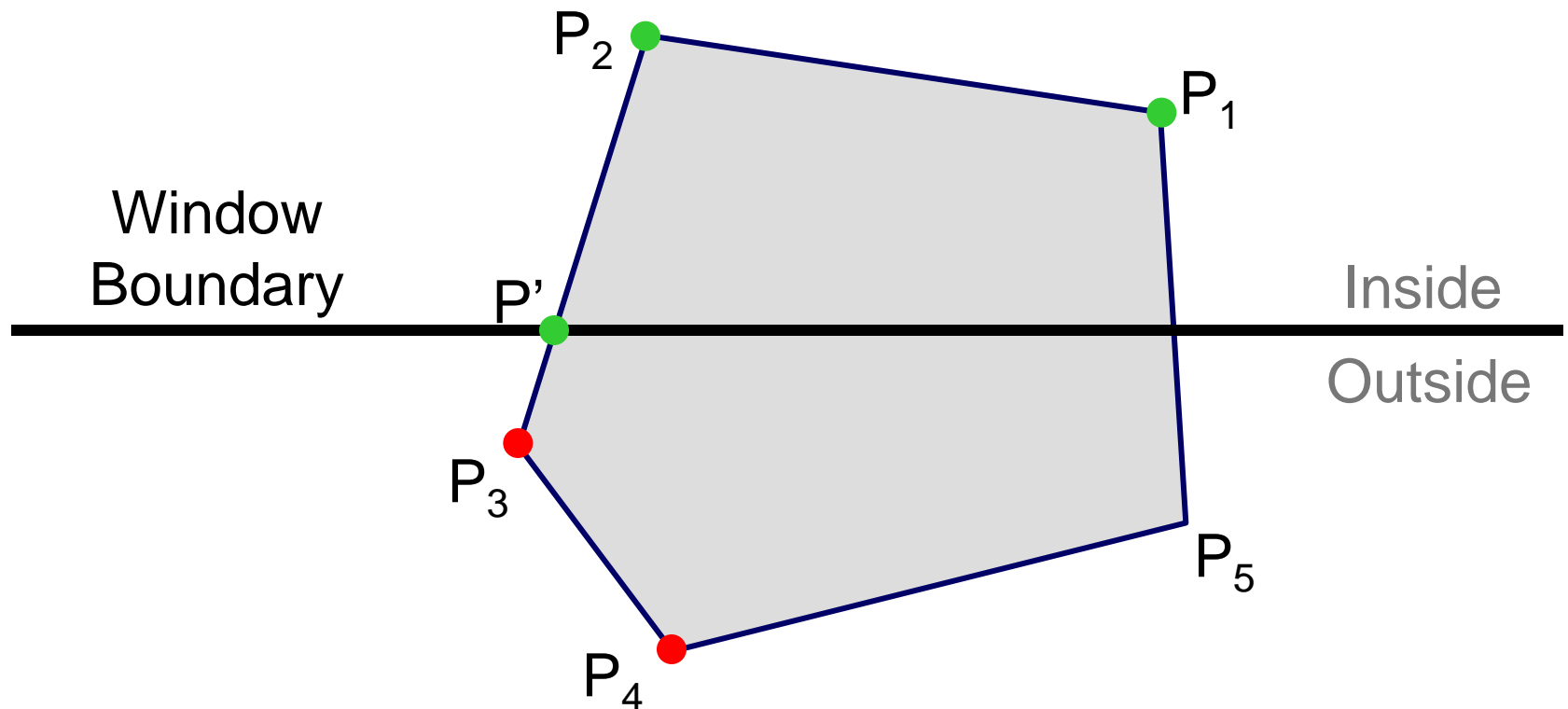
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



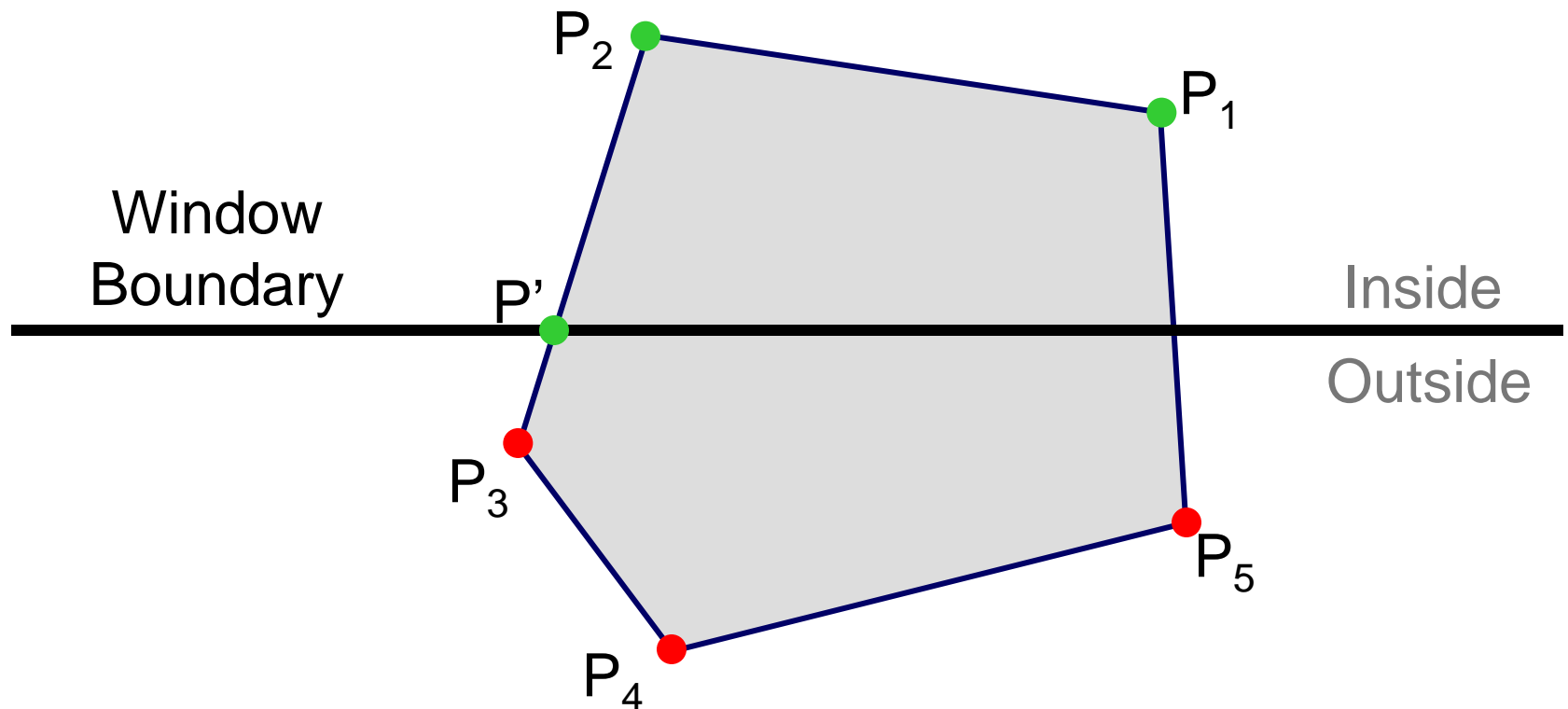
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



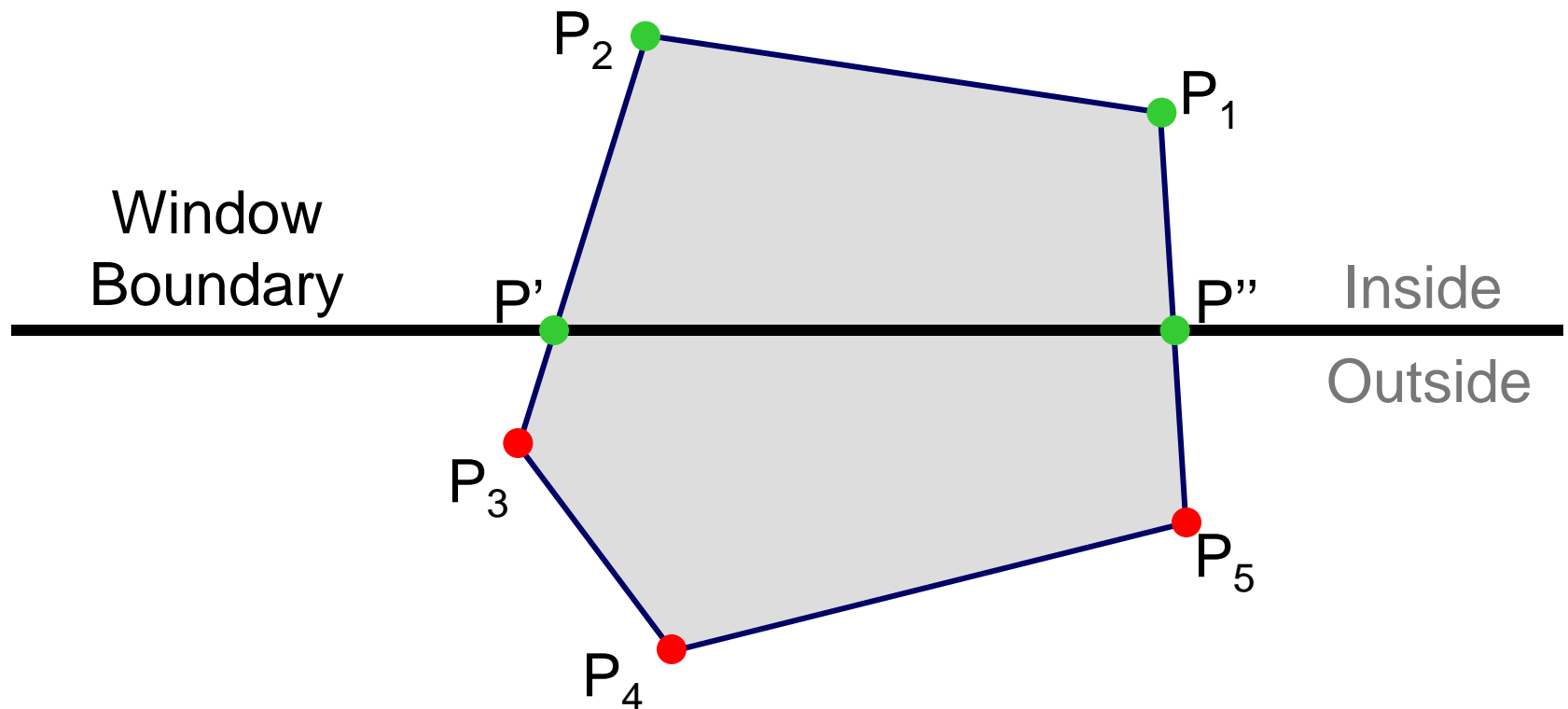
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



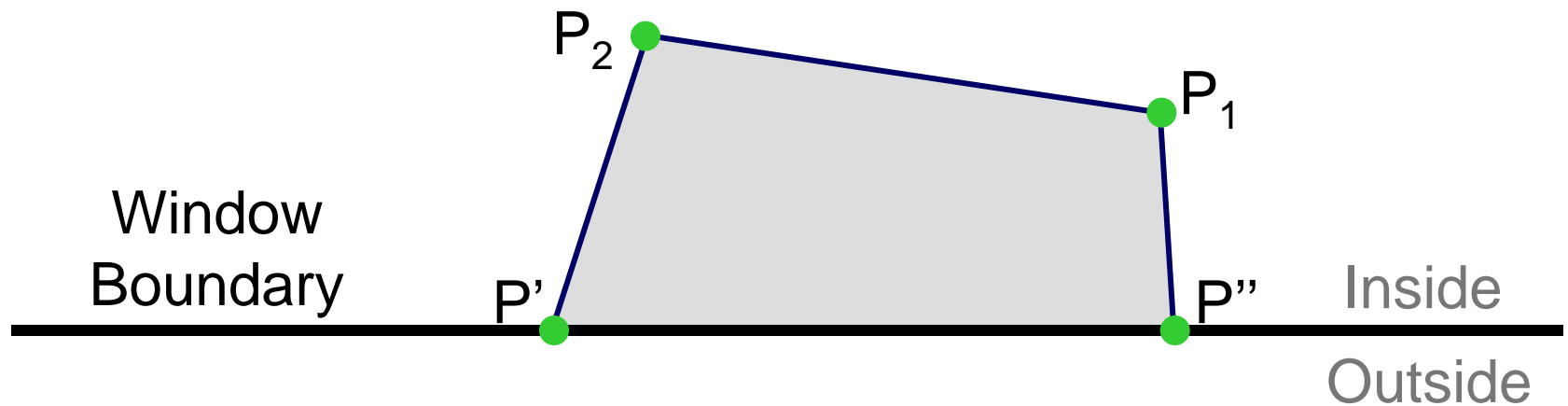
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



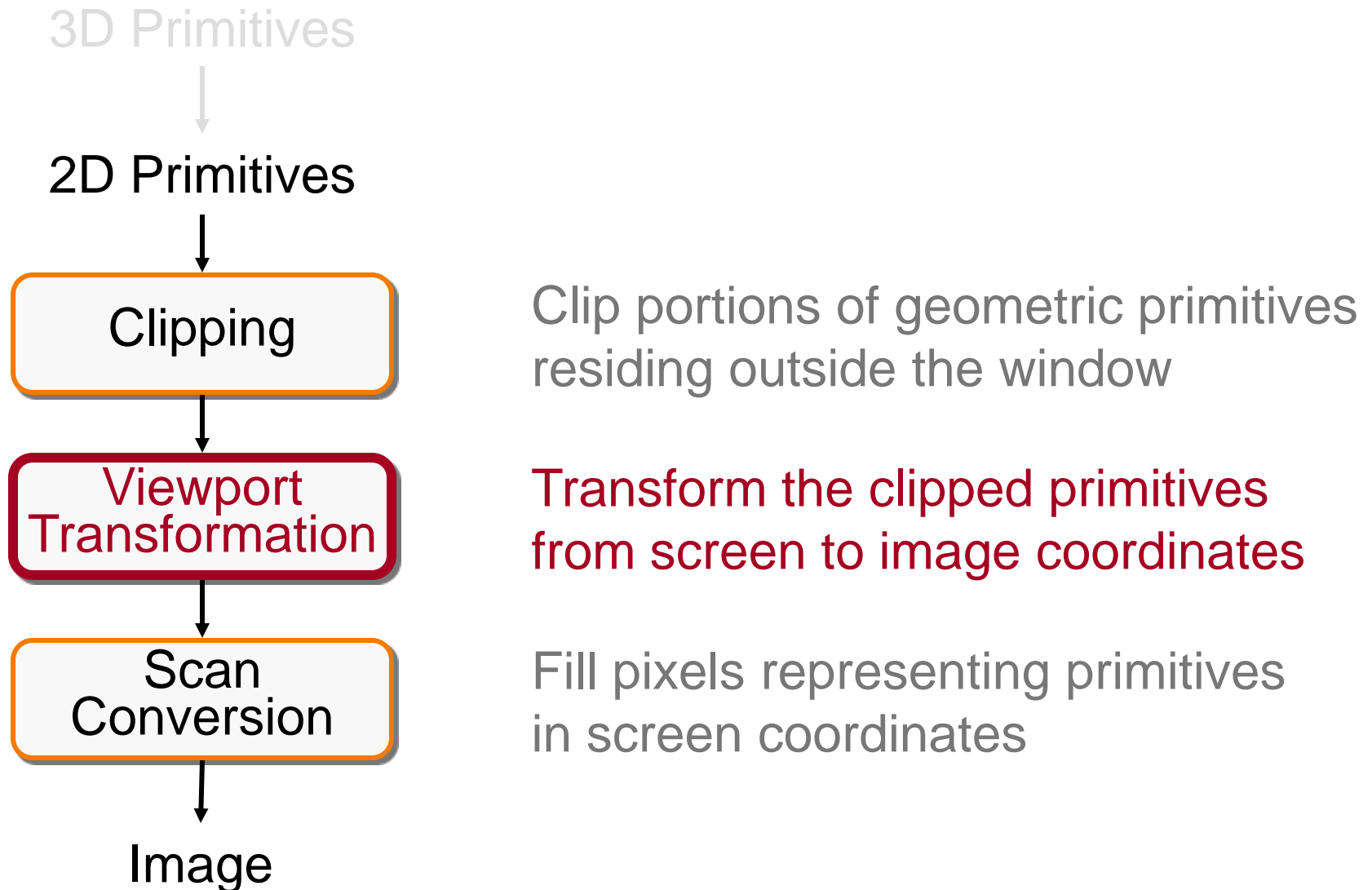
# Clipping to a Boundary

- Do inside test for each point in sequence,  
Insert new points when cross window boundary,  
Remove points outside window boundary



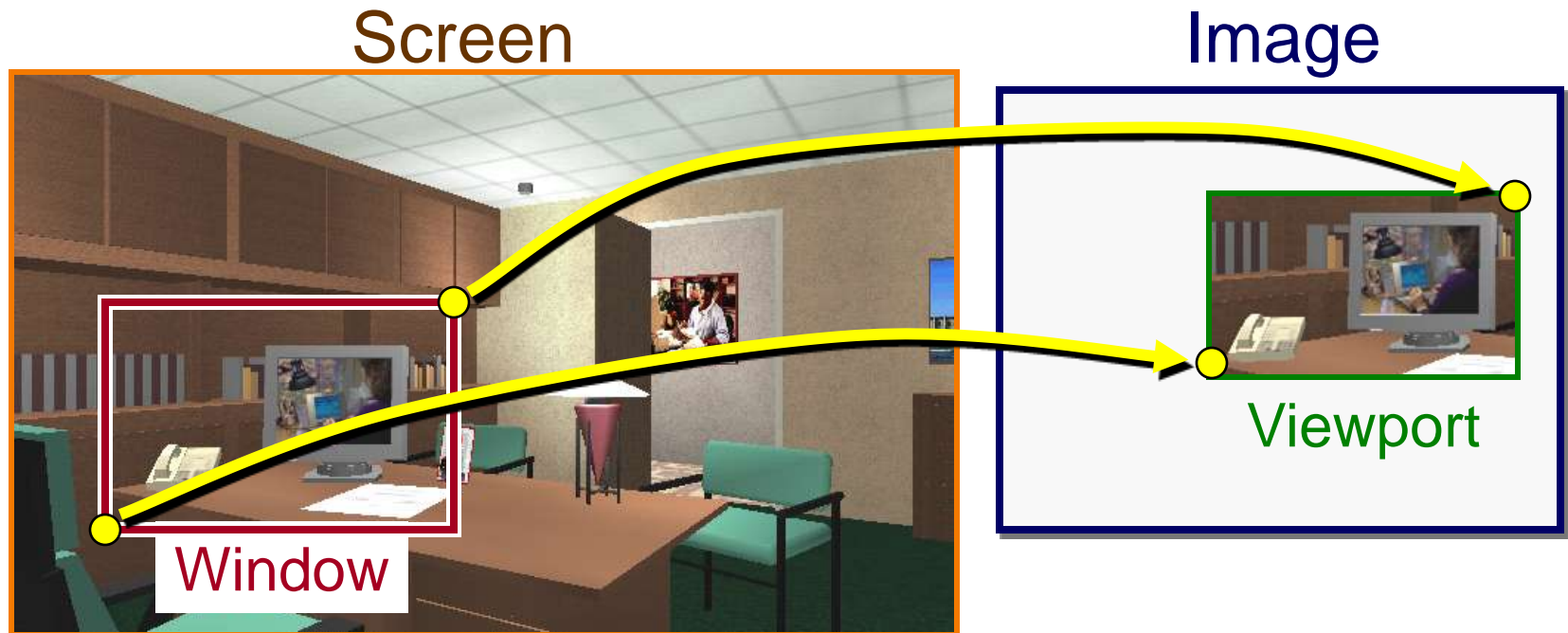


# 2D Rendering Pipeline



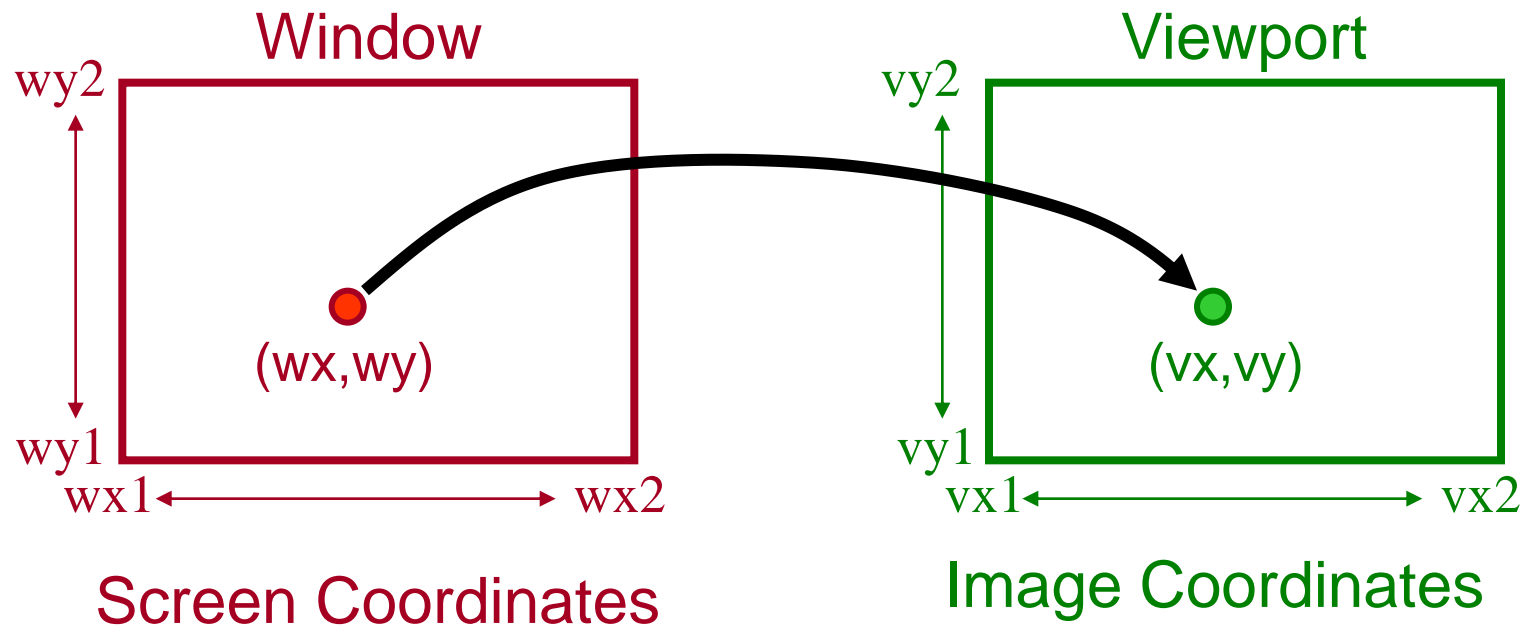
# Viewport Transformation

- Transform 2D geometric primitives from screen coordinate system (normalized device coordinates) to image coordinate system (pixels)



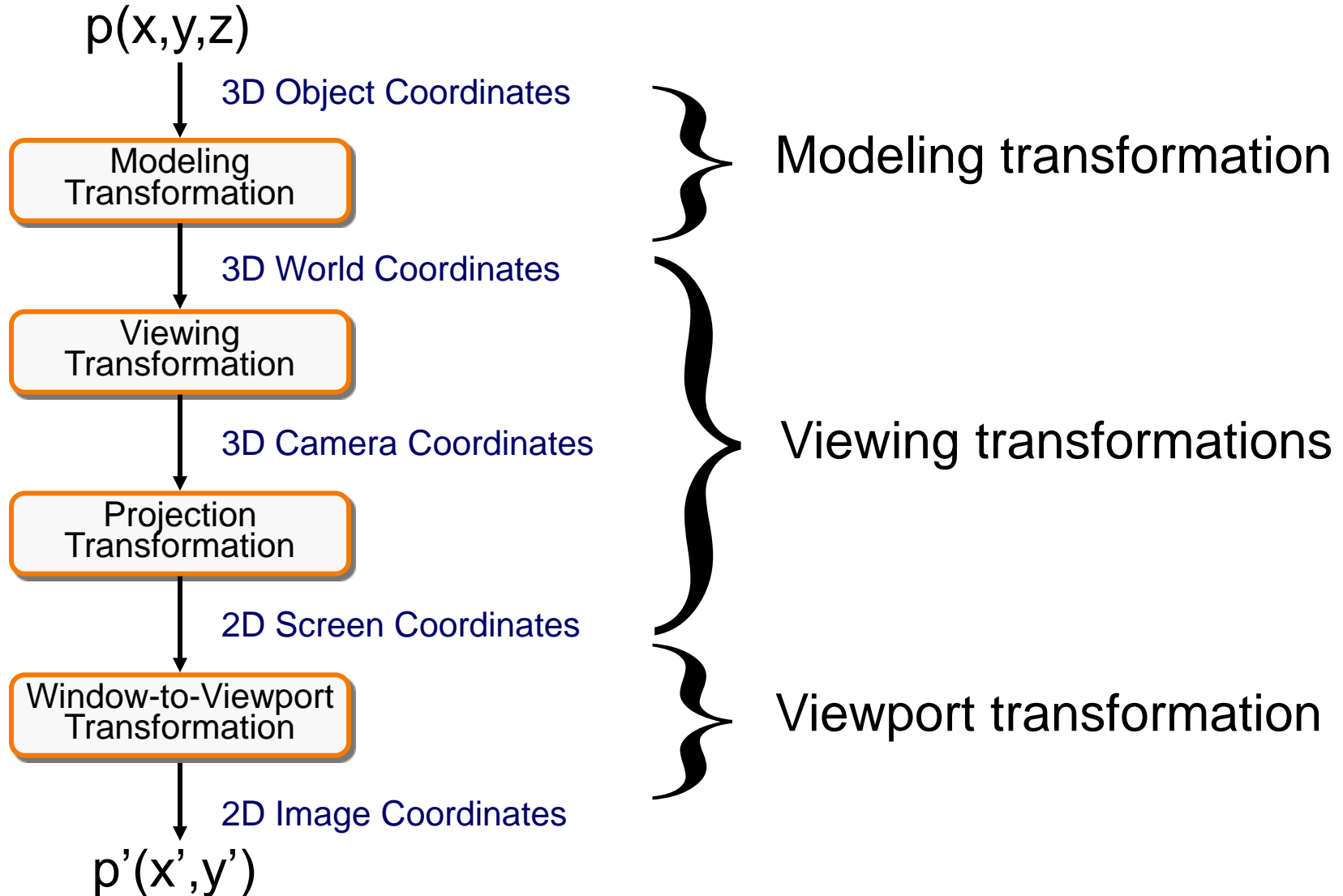
# Viewport Transformation

- Window-to-viewport mapping

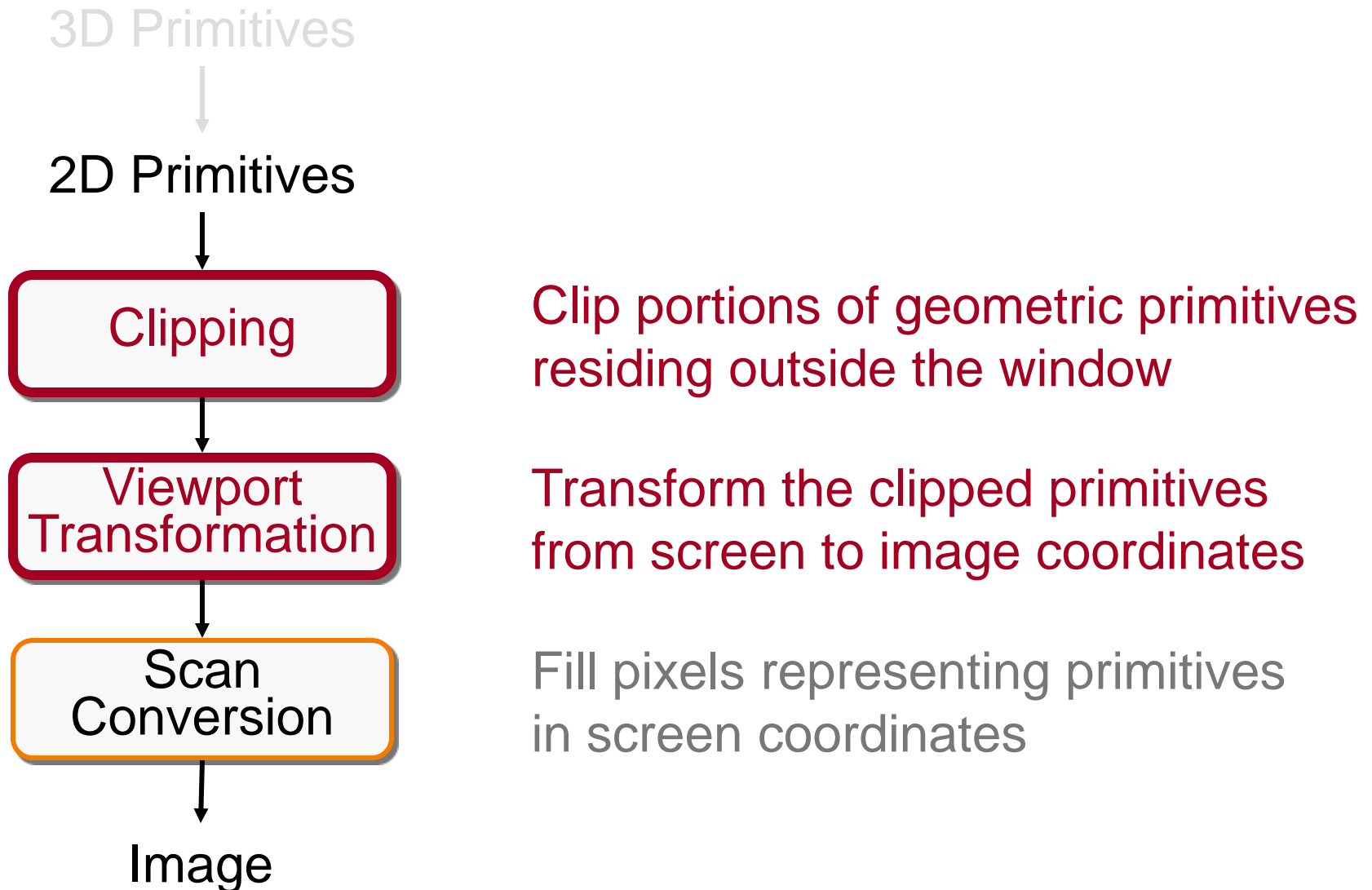


$$\begin{aligned} vx &= vx1 + (wx - wx1) * (vx2 - vx1) / (wx2 - wx1) ; \\ vy &= vy1 + (wy - wy1) * (vy2 - vy1) / (wy2 - wy1) ; \end{aligned}$$

# Summary of Transformations



# Summary



# Summary

