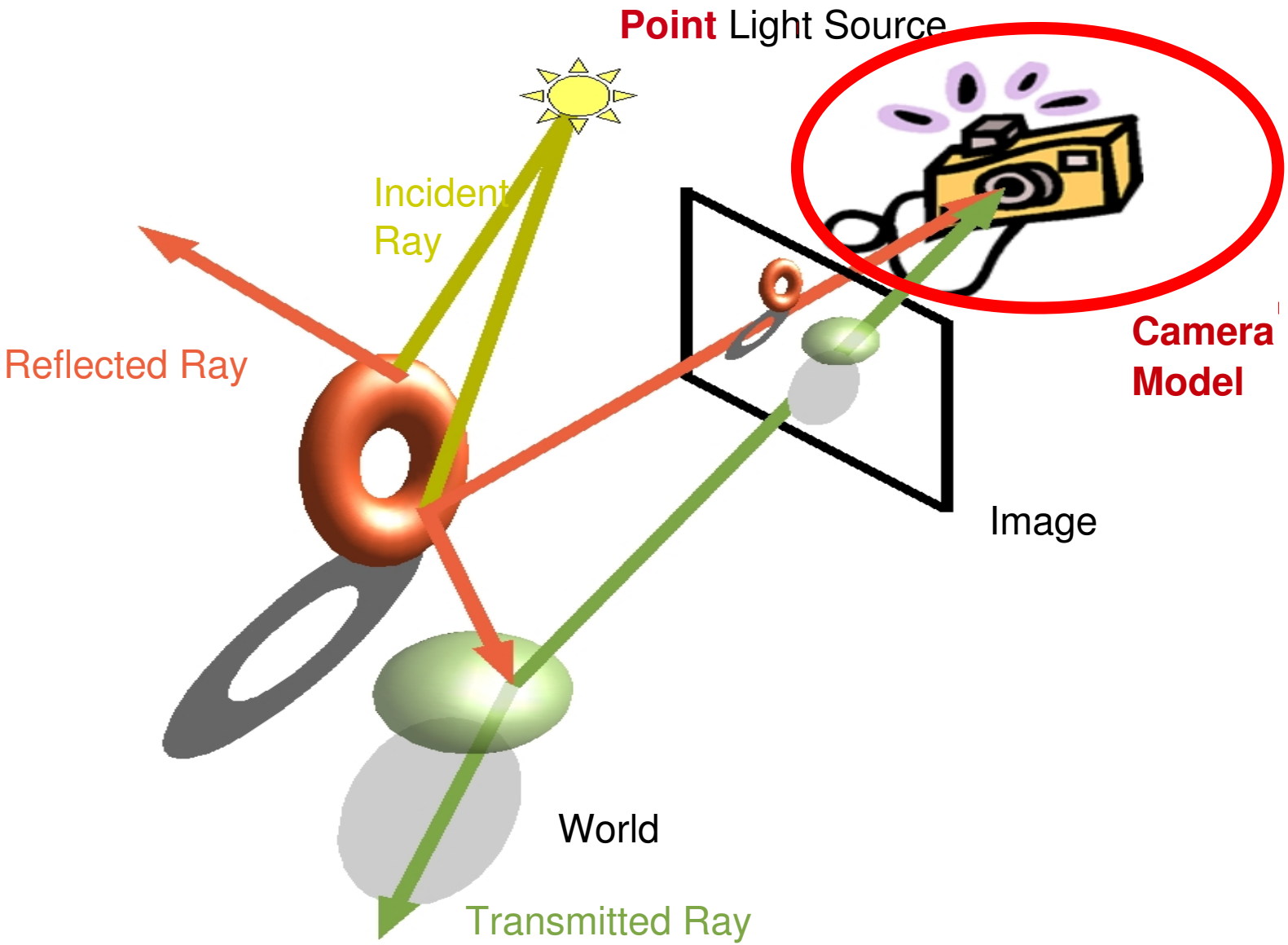
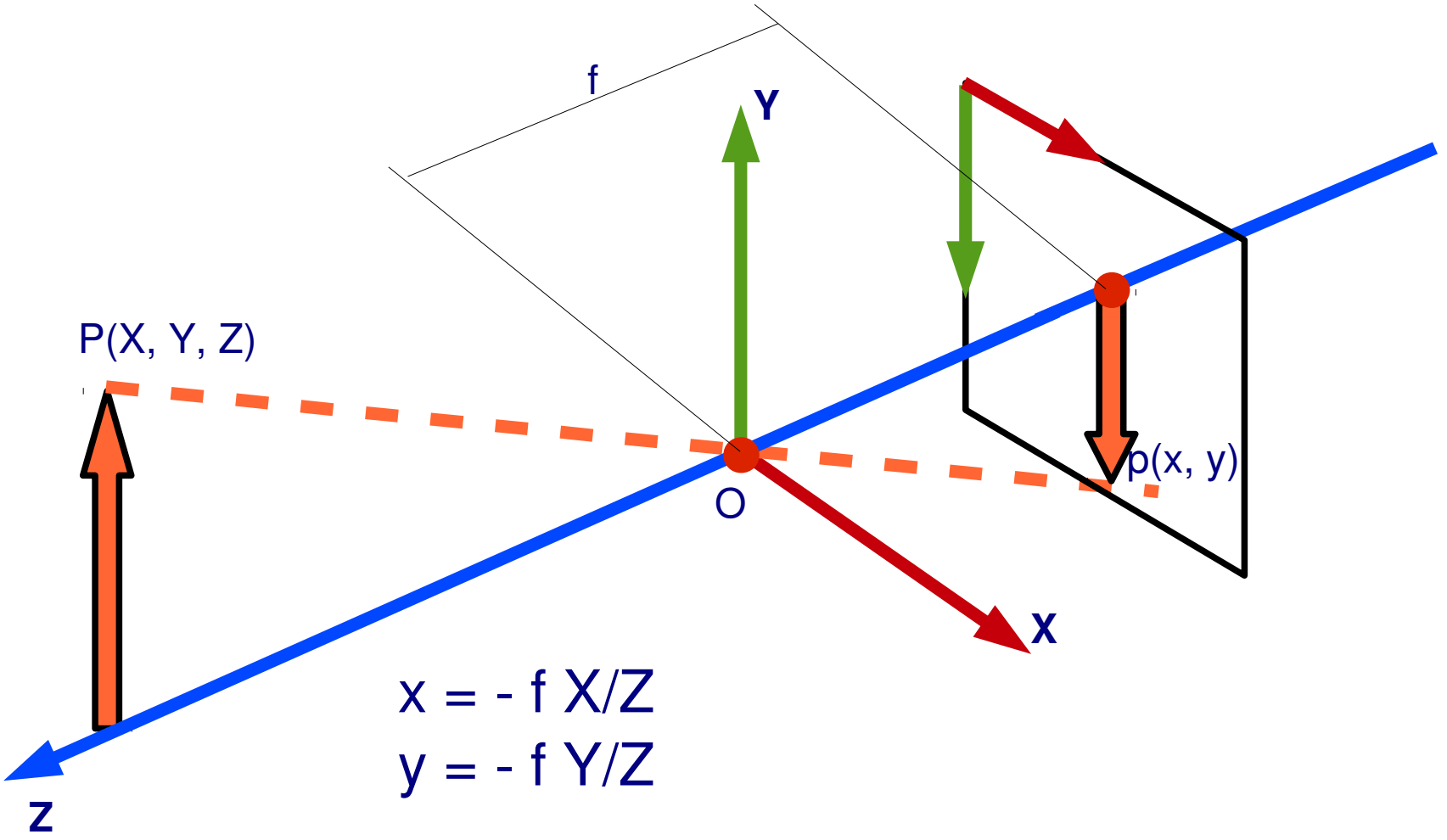
CS475/CS675 Computer Graphics

Clipping

Image Formation

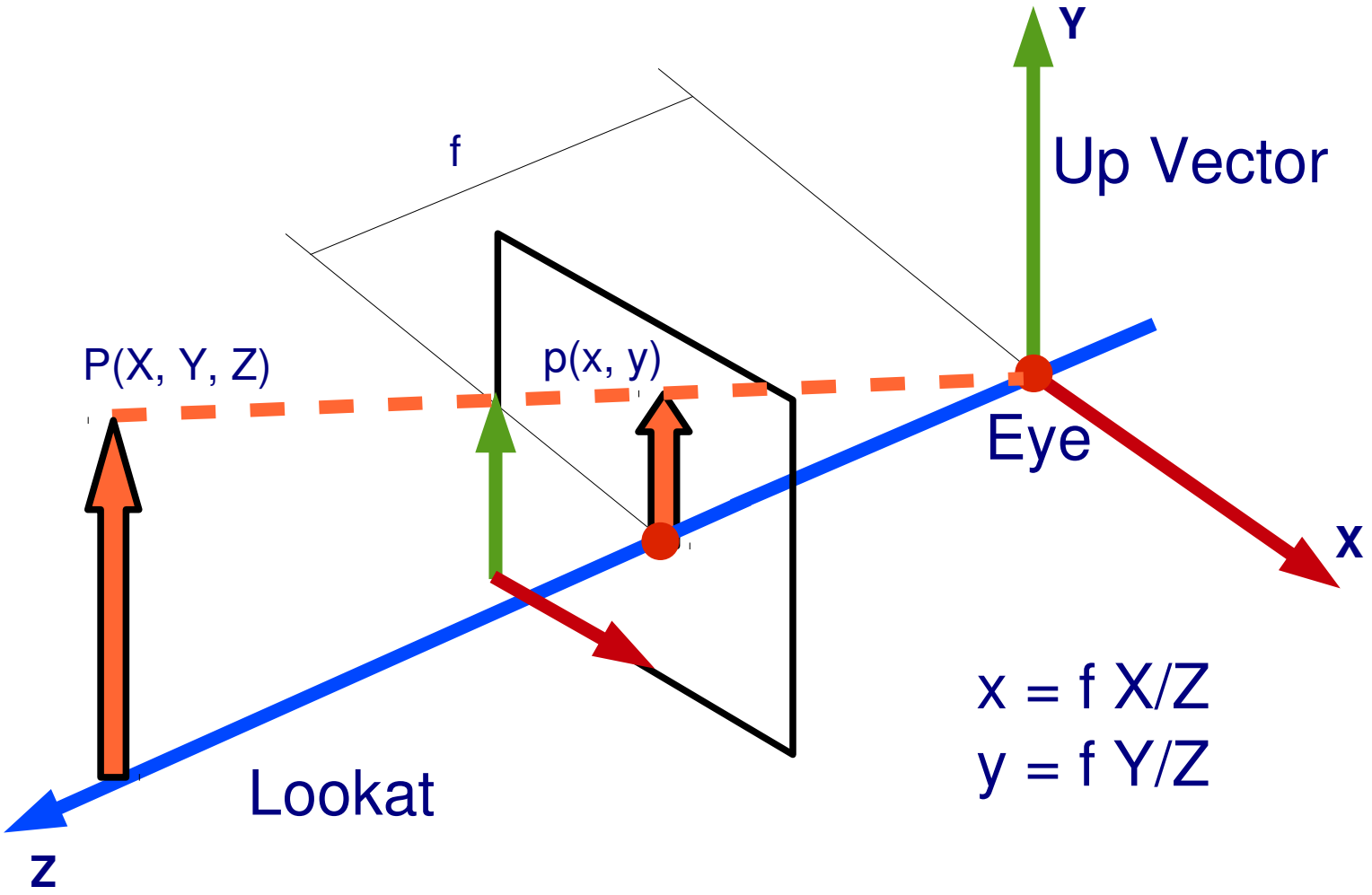


Camera Model



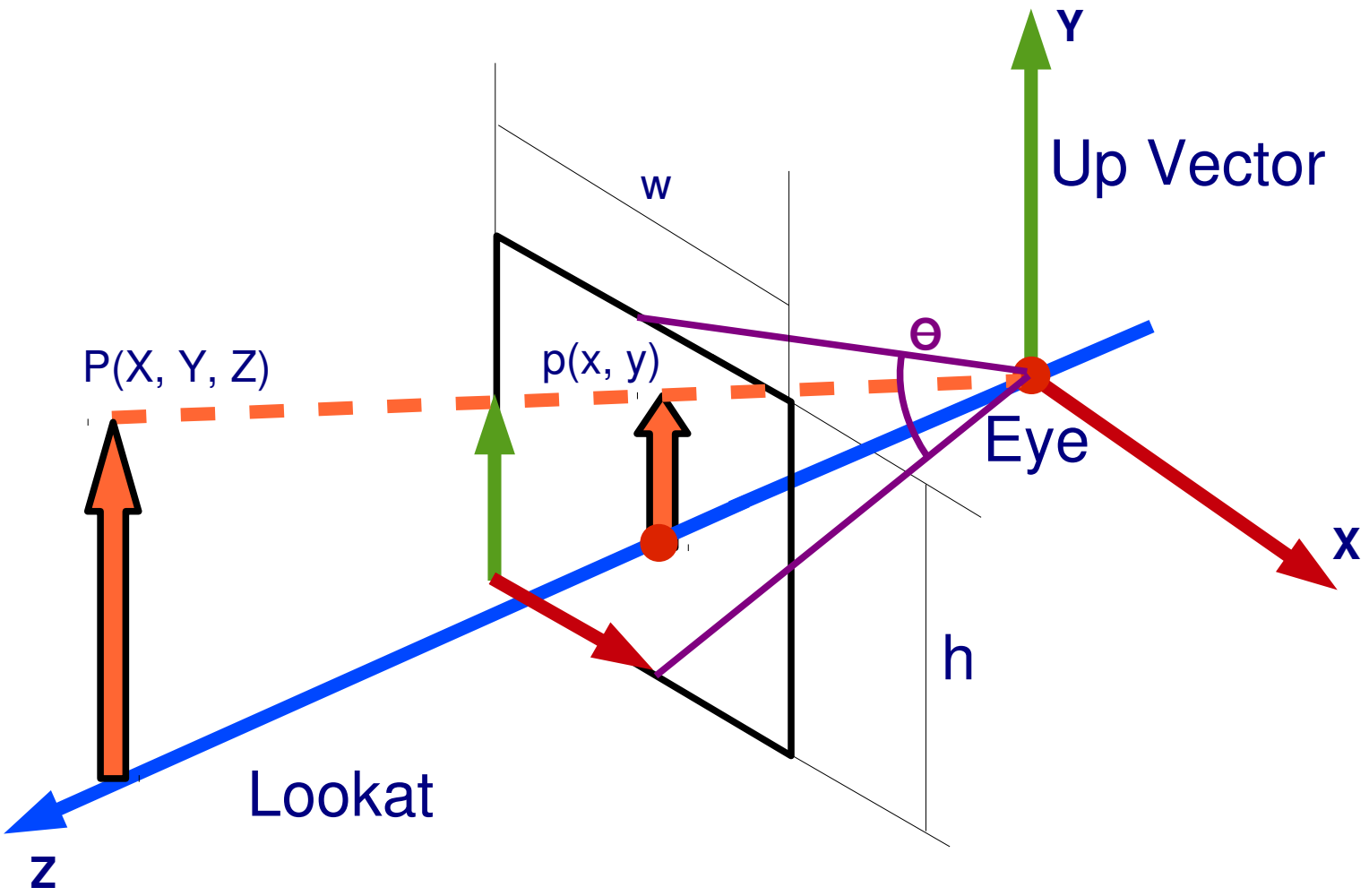
Pinhole Camera

Synthetic Camera Model



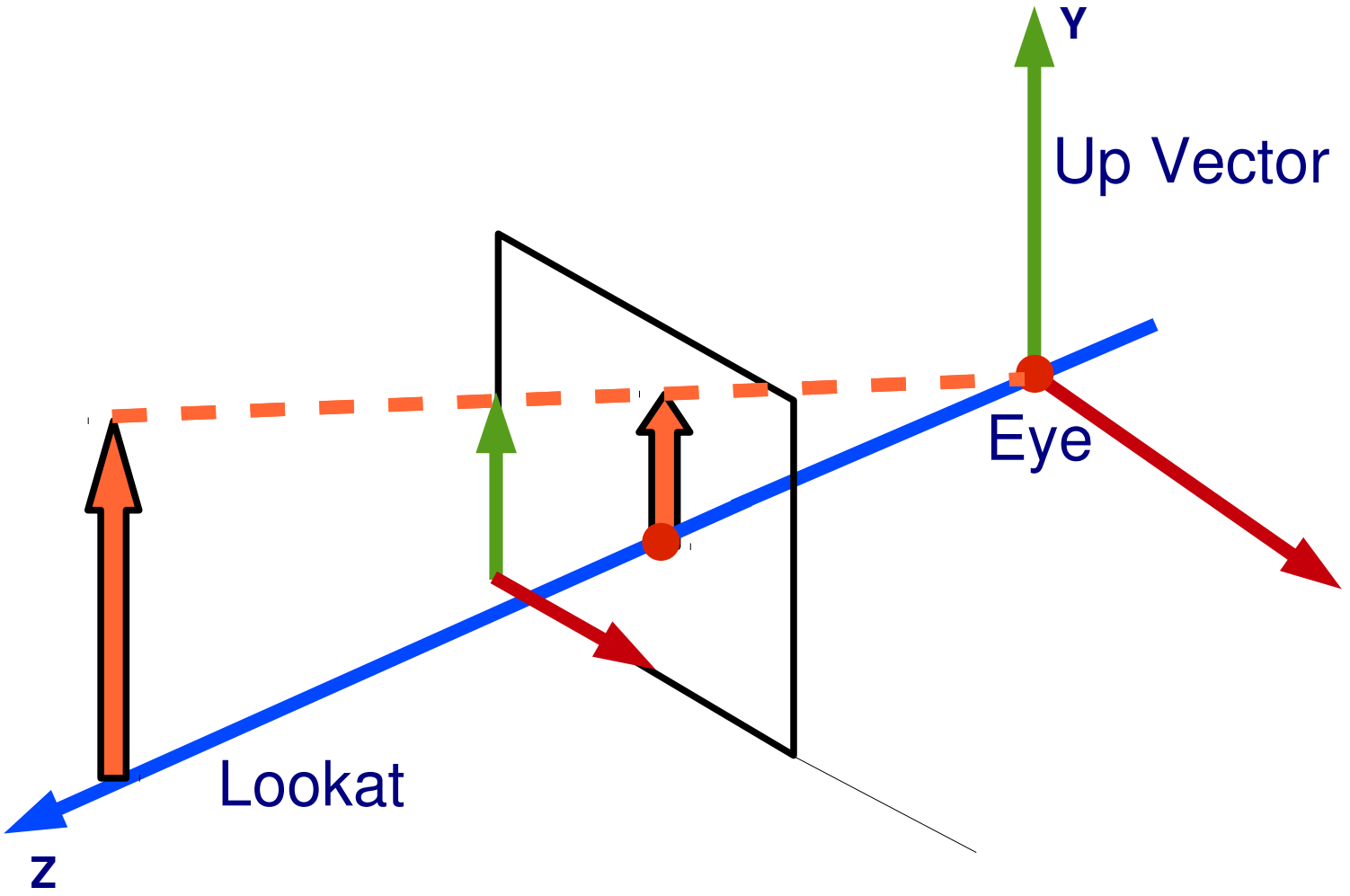
Is the Eye, Lookat and Up Vector is enough to define the camera?

Synthetic Camera Model



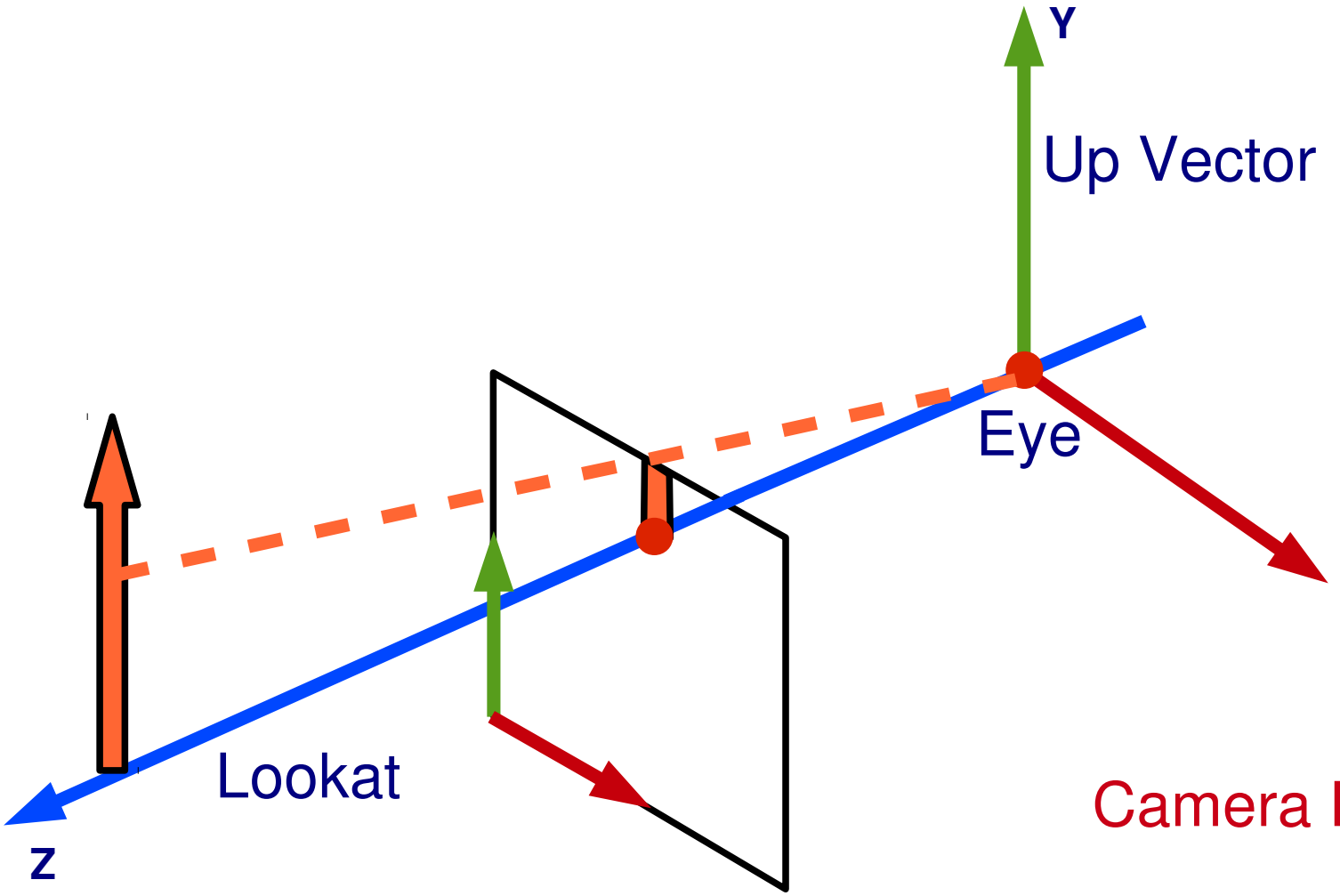
The *field of view (θ)* is also needed alongwith the window aspect ratio (w/h).

Synthetic Camera Model



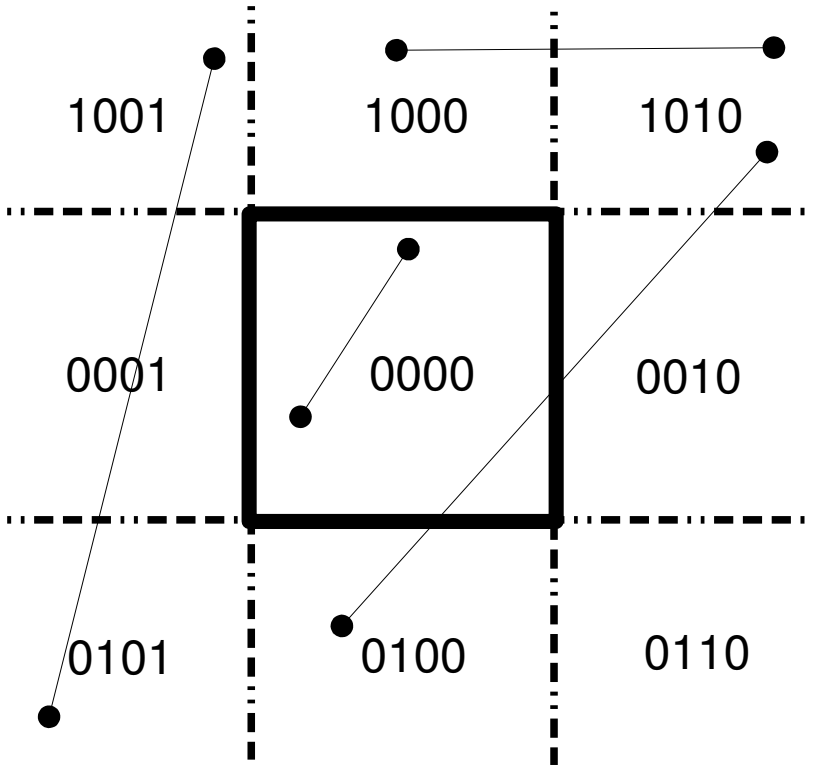
What if the window is shifted?

Synthetic Camera Model



If the window is shifted the the scene gets *clipped* at the window edges.

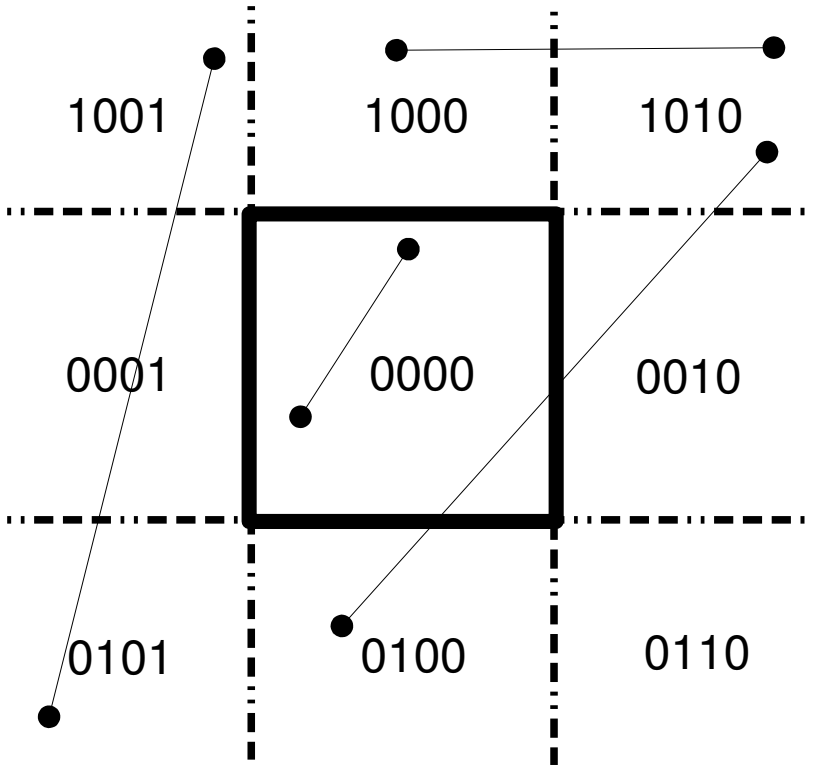
Line Clipping



Cohen – Sutherland Algorithm

- Divide the plane into 9 regions.
- Each region has its own 4 bit outcode.
- Compute the outcodes OC_0 and OC_1 for the vertices of the line segment.
- Trivially accept if $OC_0 \vee OC_1 = 0$ (TA)
- Trivially reject if $OC_0 \wedge OC_1 = 1$ (TR)
- If cannot TA/TR, subdivide line into two segments at a clip edge and TA/TR one or both segments.
- Repeat until entire line has been processed.

Line Clipping



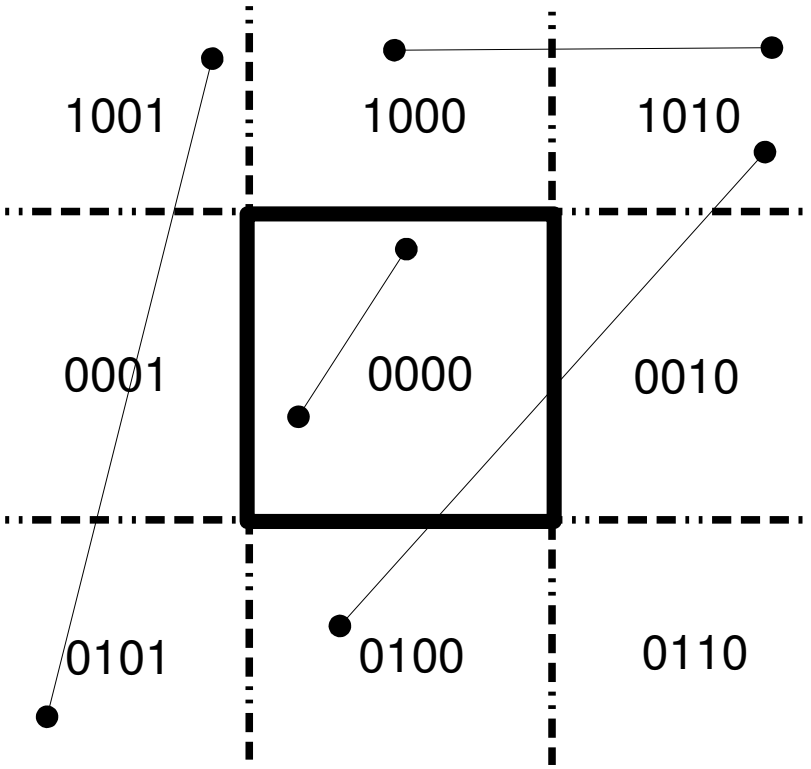
Cohen – Sutherland Algorithm

clipline(x_0, y_0,x_1, y_1)
{
ComputeOutcode(x_0, y_0, OC_0);
ComputeOutcode(x_1, y_1, OC_1);
repeat
Check for TA and TR. If either happens then *done*.

Choose a vertex that is outside the clip rectangle.

If (vertex lies over TOP edge)
then
$x = x_0 + 1/slope * (y_{max} - y_0)$
$y = y_{max}$
....

Line Clipping



Cohen – Sutherland Algorithm

else if (vertex lies below BOTTOM edge)
then

$x = x_0 + 1/slope * (y_{min} - y_0)$

$y = y_{min}$

else if (vertex lies to right of RIGHT edge)
then

$y = y_0 + slope * (x_{max} - x_0)$

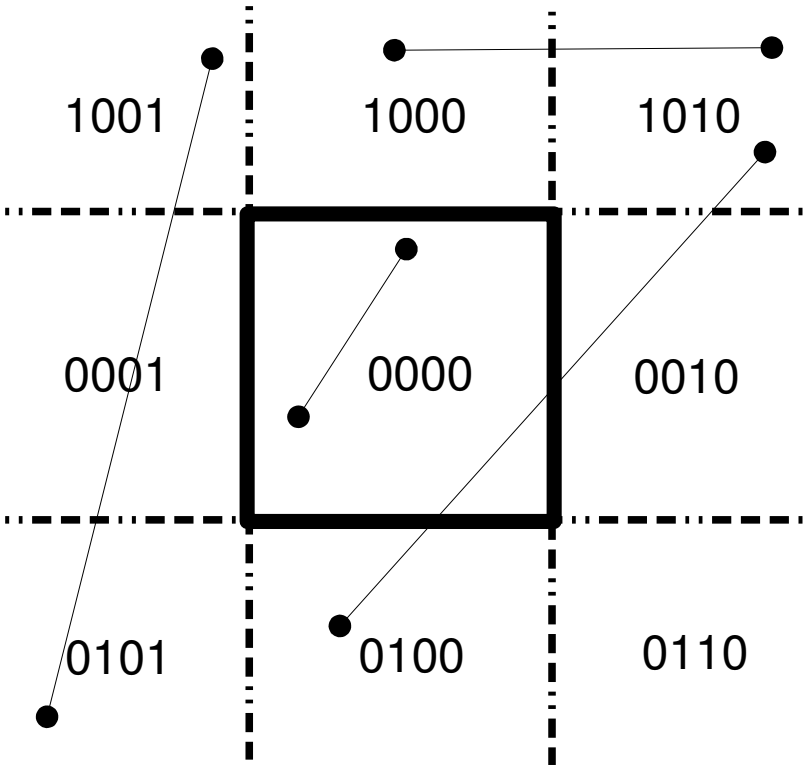
$x = x_{max}$

else if (vertex lies to left of LEFT edge)
then

$y = y_0 + slope * (x_{min} - x_0)$

$x = x_{min}$

Line Clipping



Cohen – Sutherland Algorithm

else if (vertex lies below BOTTOM edge)
then

$x = x_0 + 1/slope * (y_{min} - y_0)$

$y = y_{min}$

else if (vertex lies to right of RIGHT edge)
then

$y = y_0 + slope * (x_{max} - x_0)$

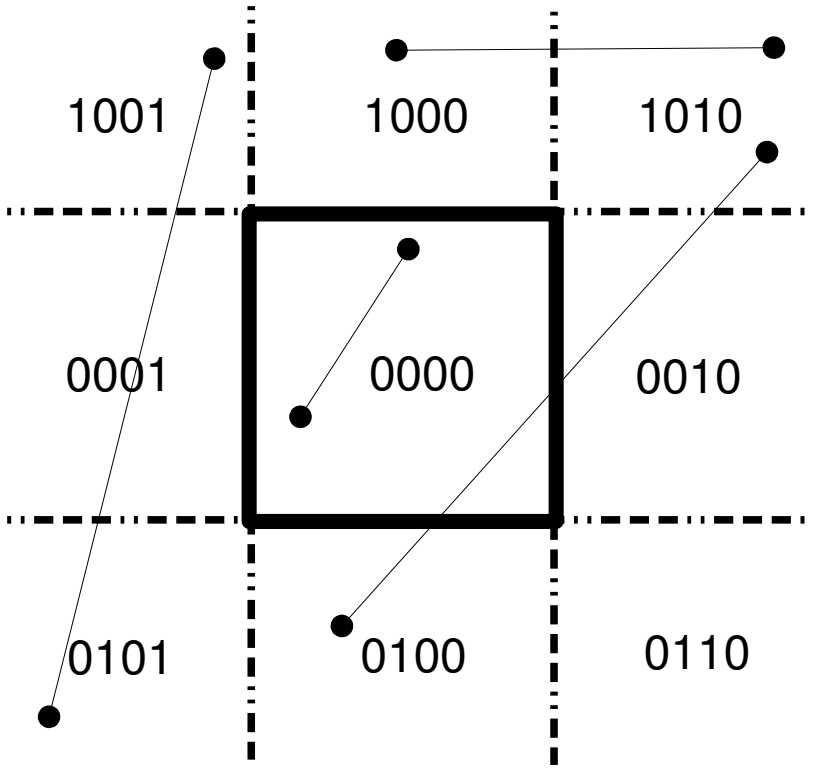
$x = x_{max}$

else if (vertex lies to left of LEFT edge)
then

$y = y_0 + slope * (x_{min} - x_0)$

$x = x_{min}$

Line Clipping



Cohen – Sutherland Algorithm

if (x_0, y_0) was the outer point
then
$x_0 = x, y_0 = y$
ComputeOutcode(x_0, y_0, OC_0)
else
$x_1 = x, y_1 = y$
ComputeOutcode(x_1, y_1, OC_1)

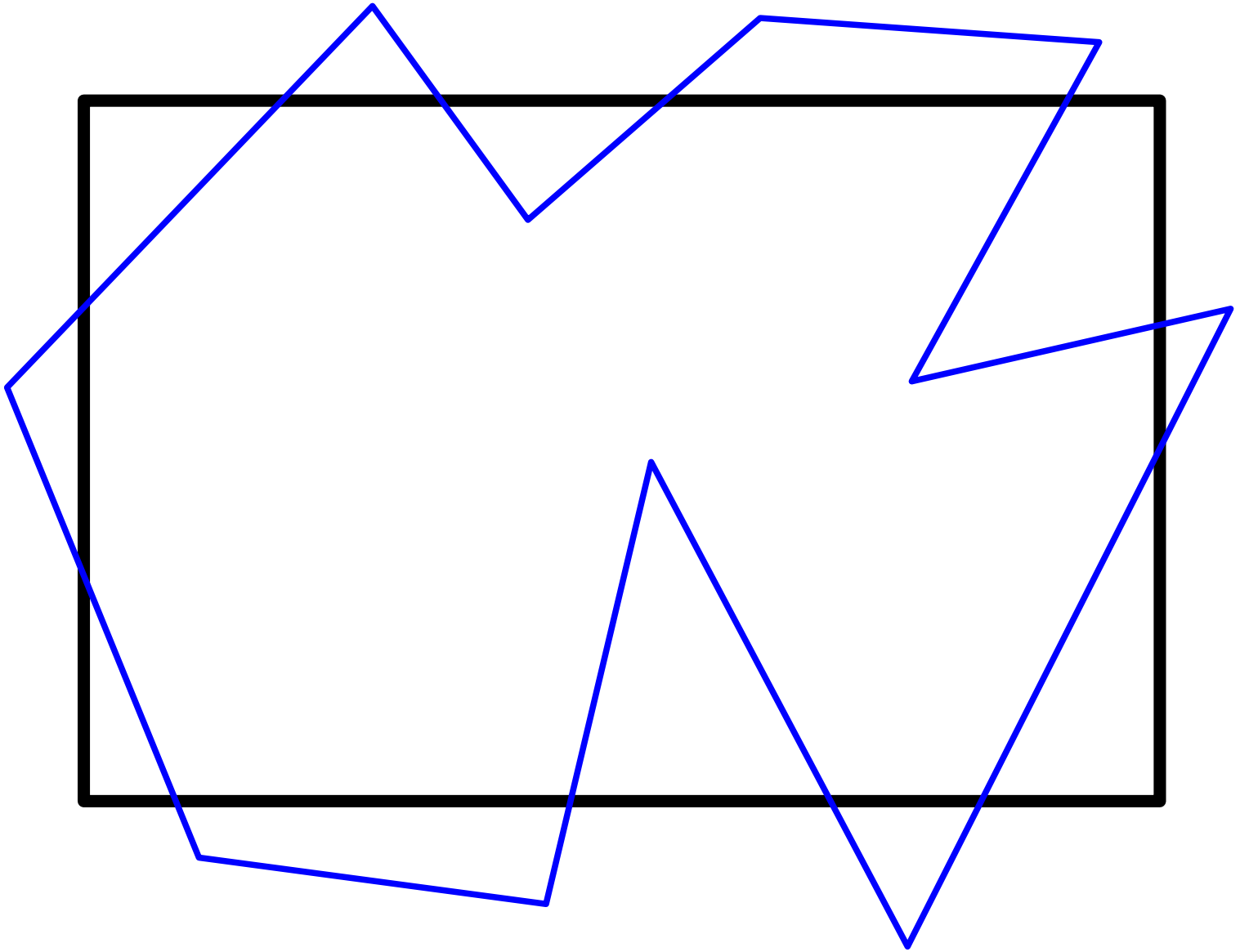
until (*done*)

}

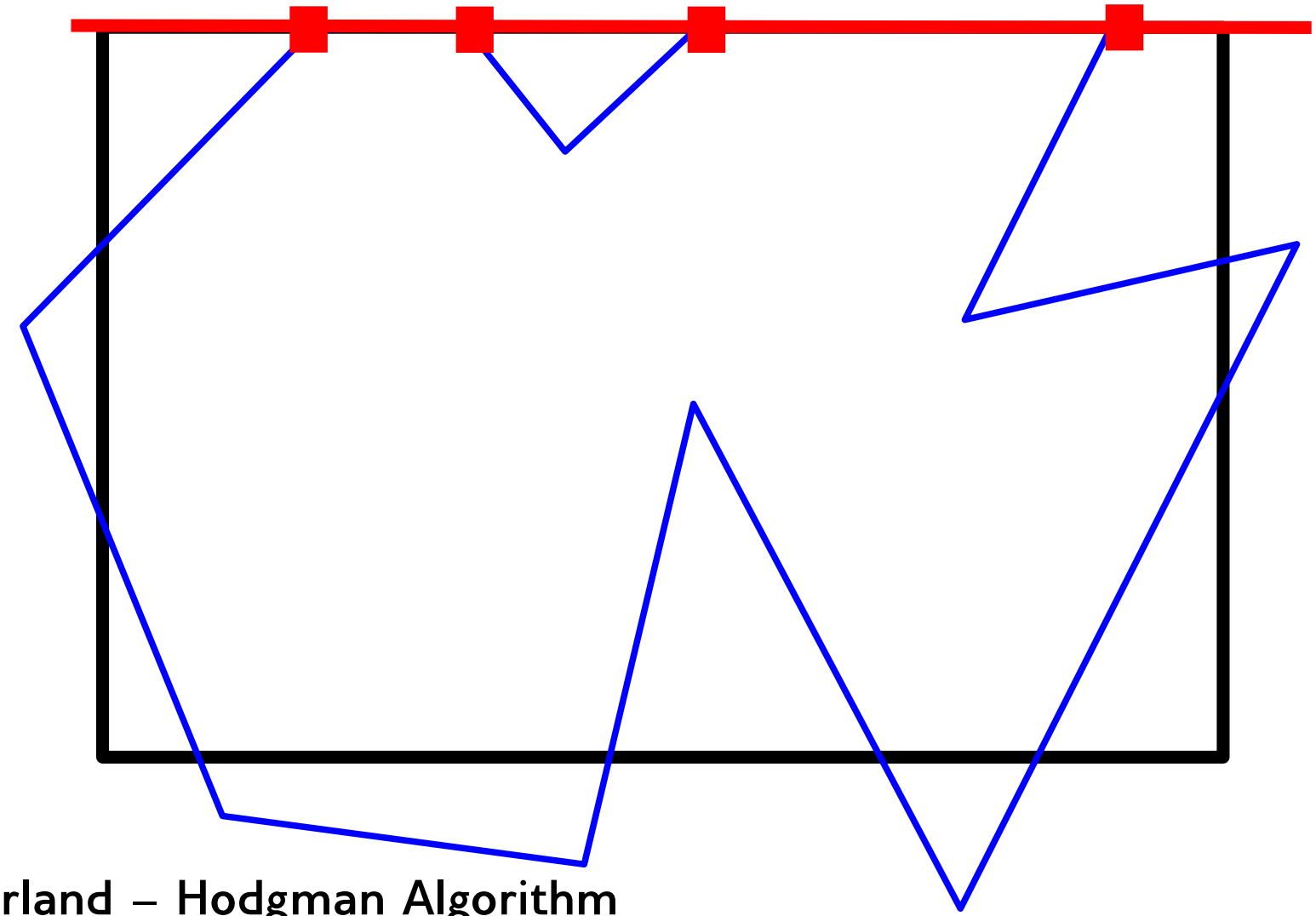
Issues in
- Clipping
- Scan Conversion and Clipping

Read notes on **Cyrus-Beck** Parametric Line Clipping Algorithm.

Polygon Clipping

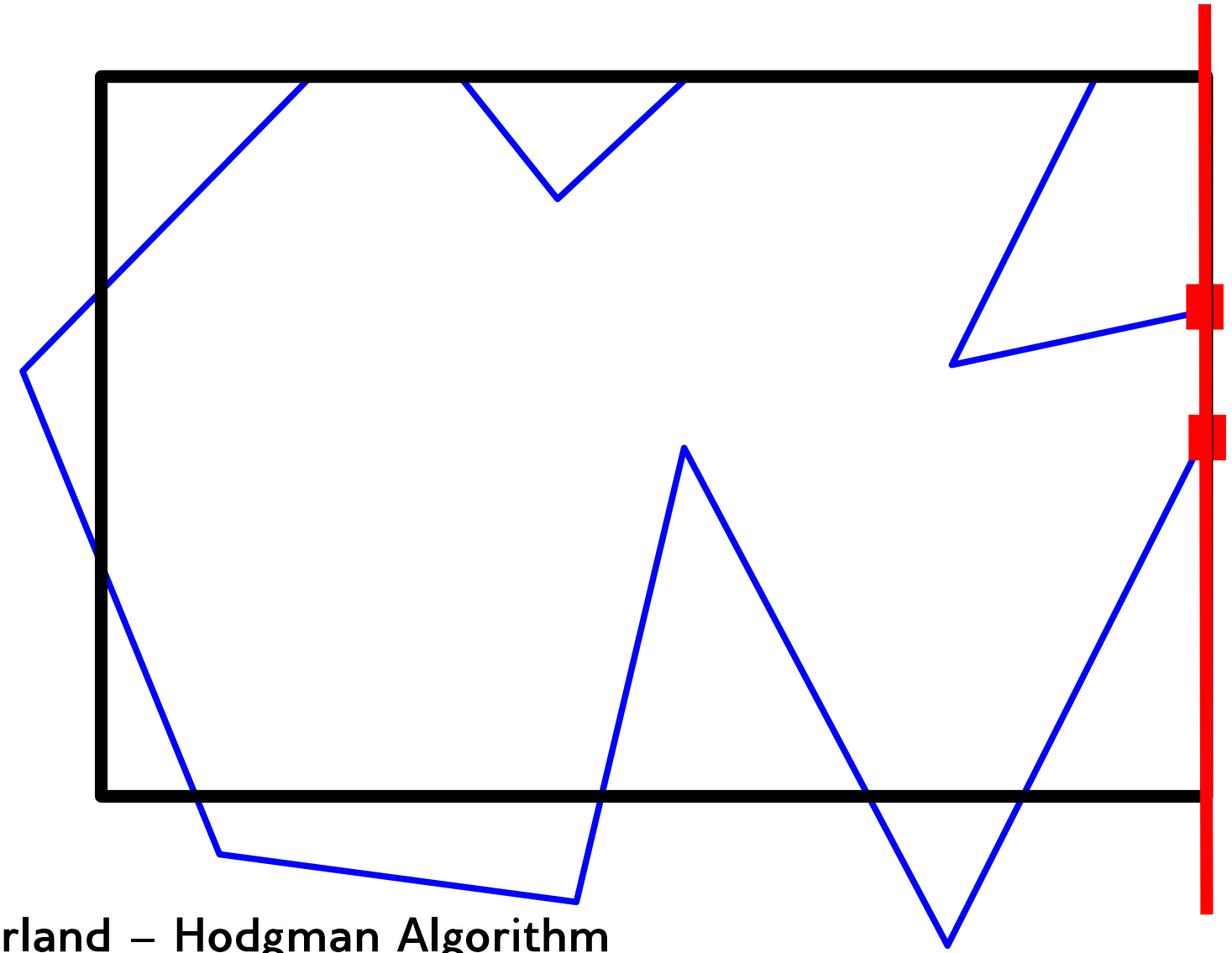


Polygon Clipping



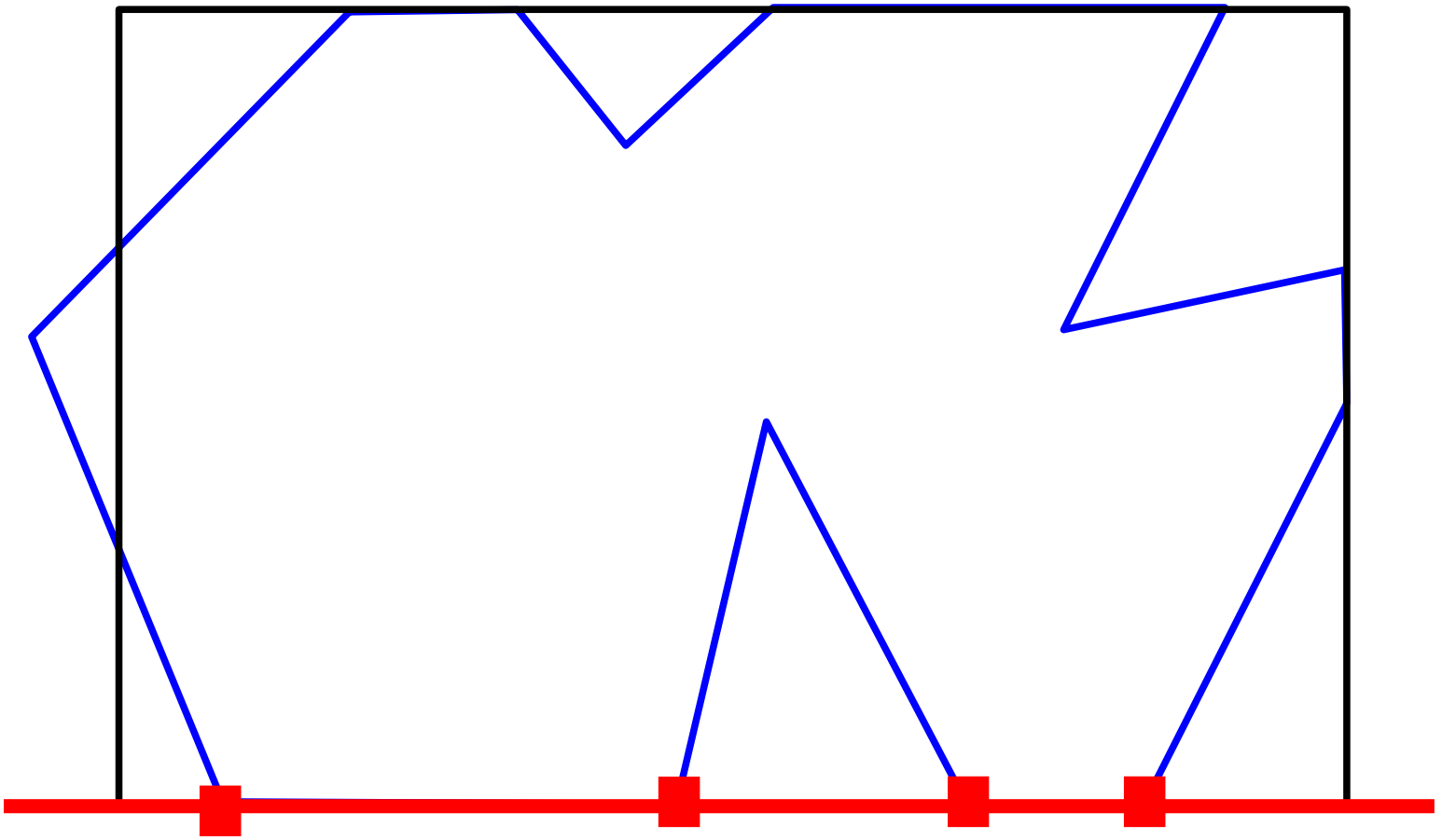
Sutherland – Hodgman Algorithm

Polygon Clipping



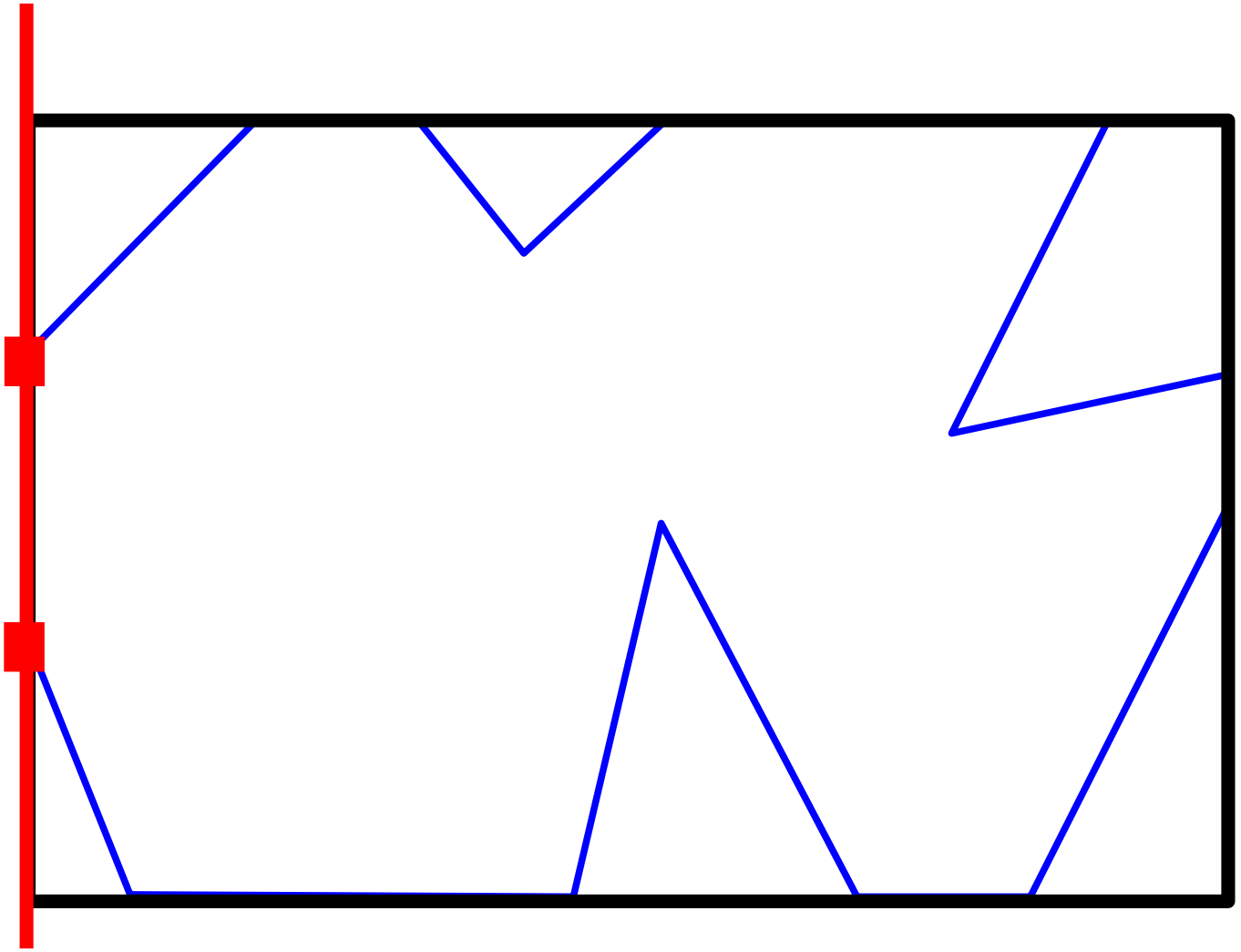
Sutherland – Hodgman Algorithm

Polygon Clipping



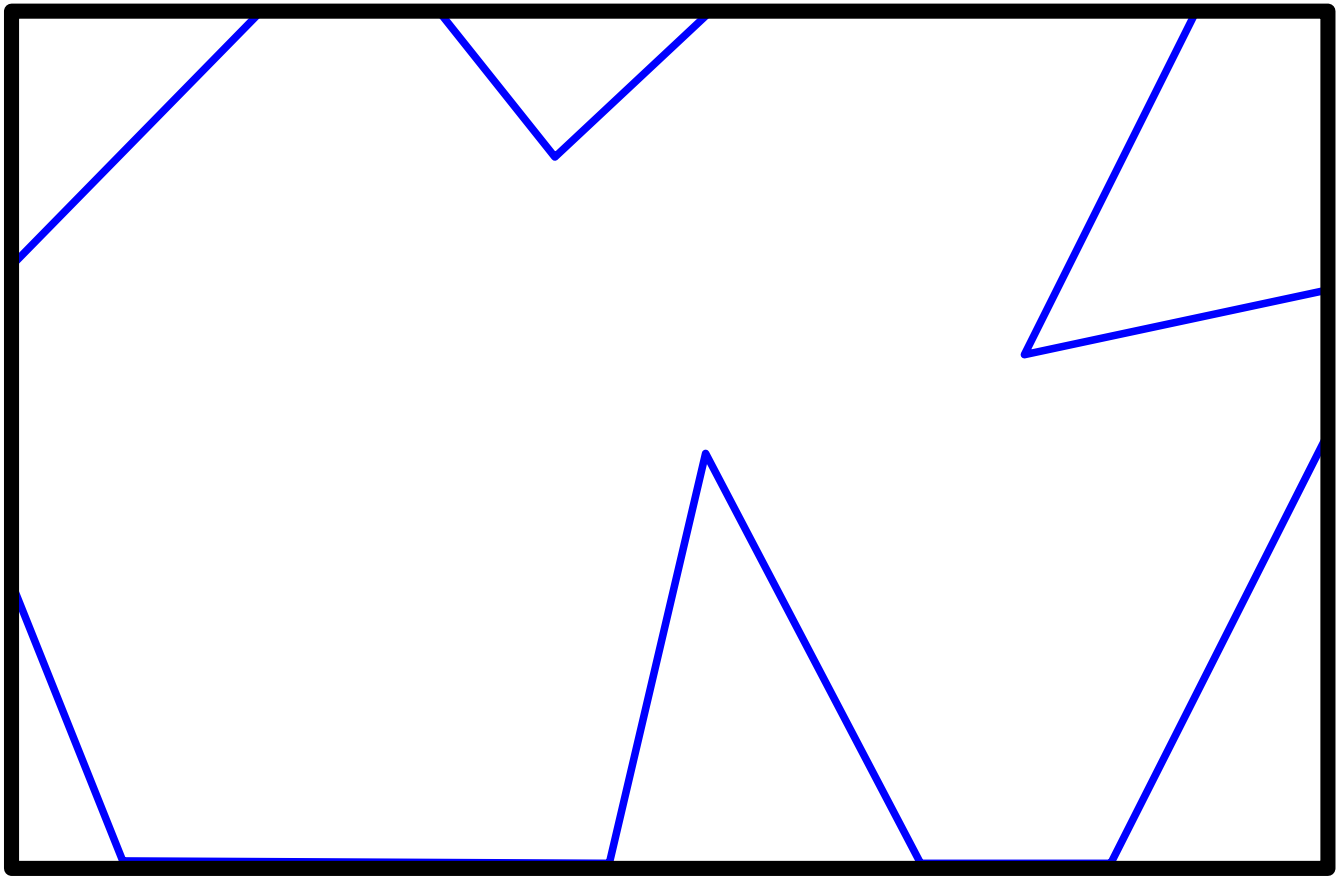
Sutherland – Hodgman Algorithm

Polygon Clipping



Sutherland – Hodgman Algorithm

Polygon Clipping



Sutherland – Hodgman Algorithm