

Computer Graphics

CSC 418/2504

Patricio Simari
October 19, 2011

Figures courtesy of Peter Shirley,
“Fundamentals of Computer Graphics”, 2nd Ed.

A note about homogeneous coordinates and vectors...

Homogeneous Coordinates: Points vs Vectors

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Point

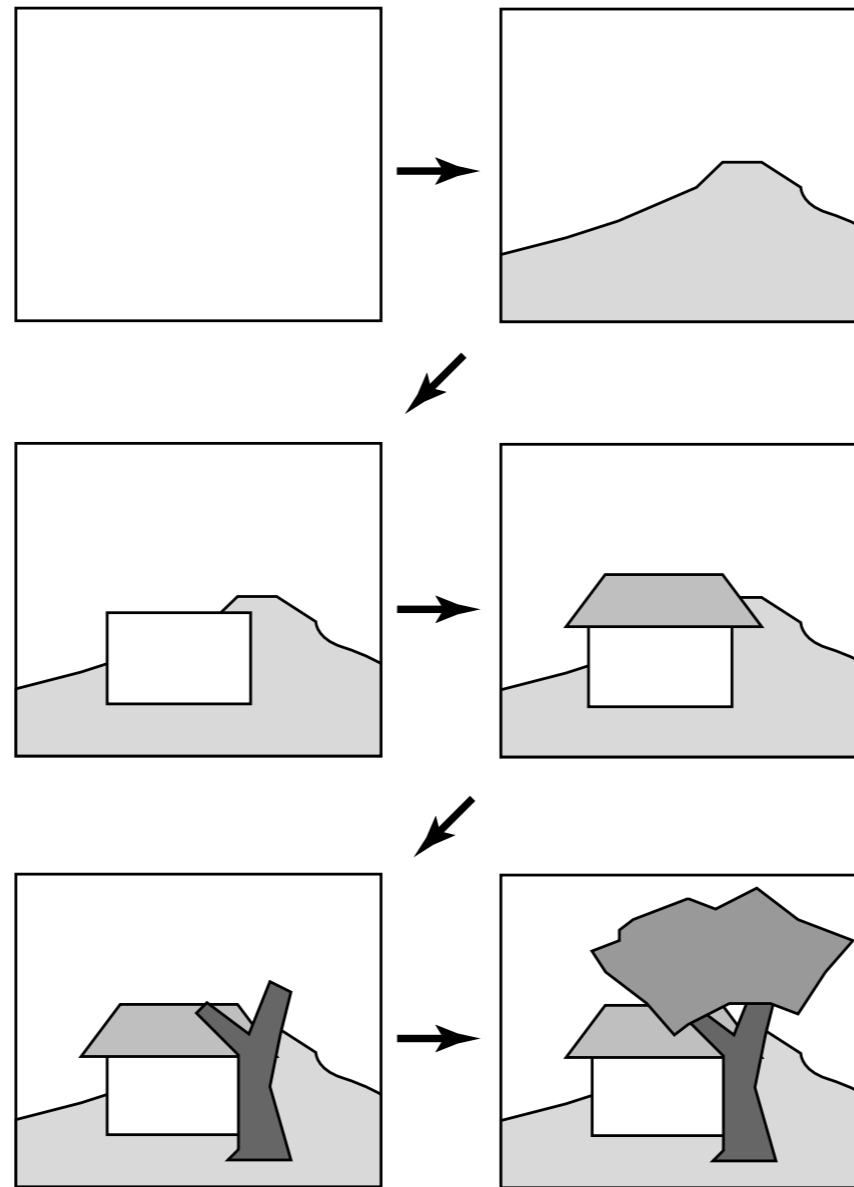
$$\begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

Vector

- Vectors don't translate
- Point + Vector = Point
- Vector + Vector = Vector

Painter's Algorithm, BSP, and Clipping

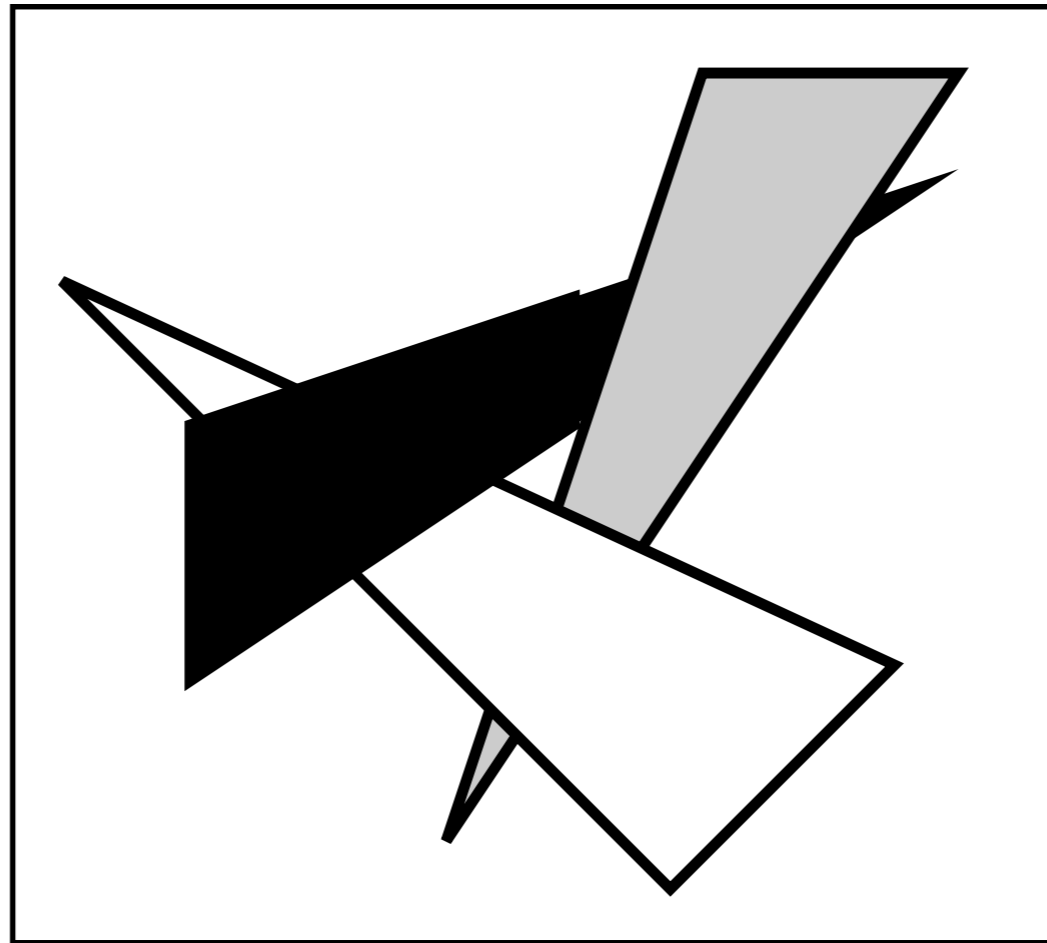
Painter's Algorithm



Painter's Algorithm

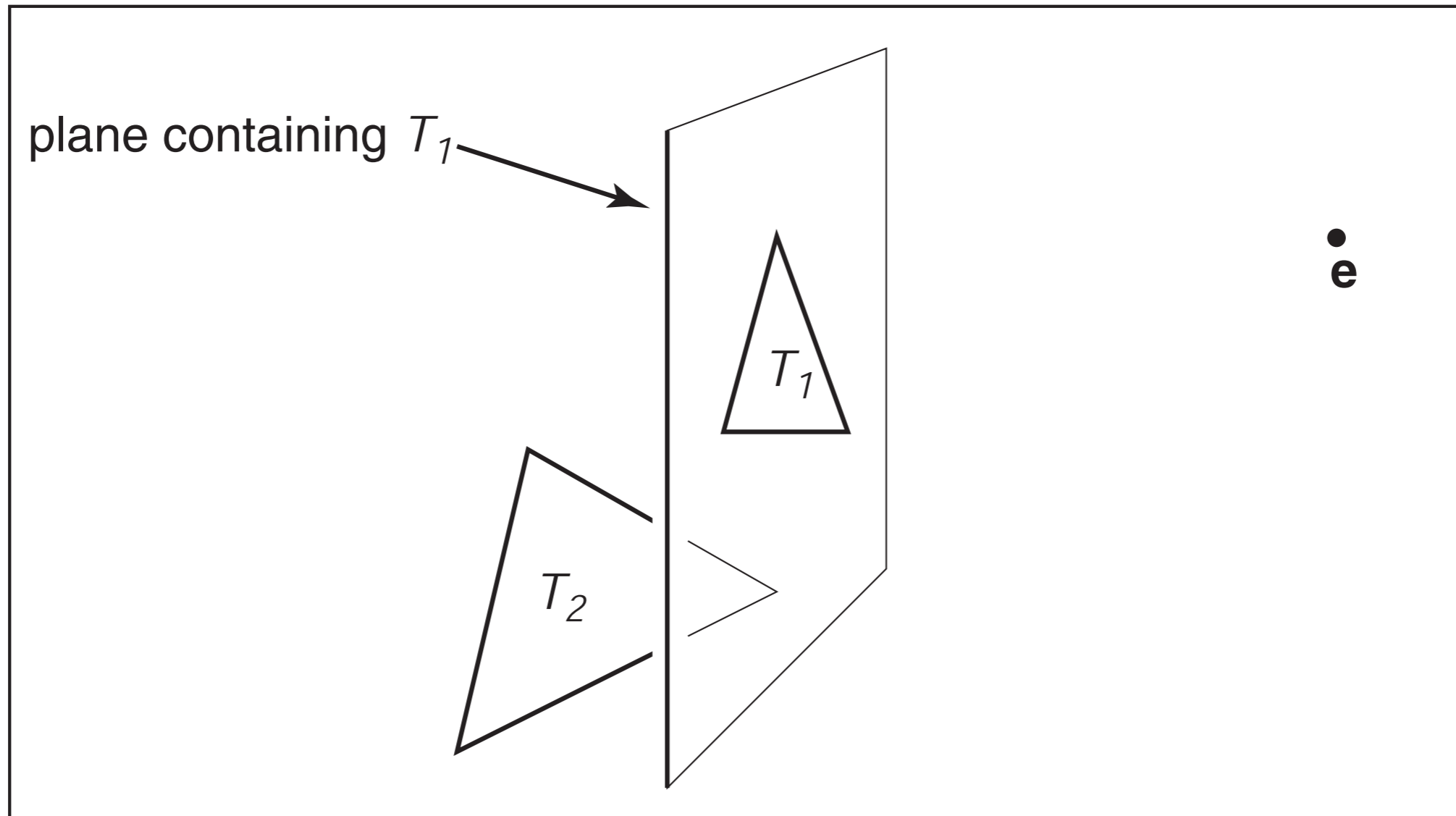
sort objects back to front relative to viewpoint
for each object **do**
 draw object on screen

Limitation



Cycle

Binary Space Partitioning



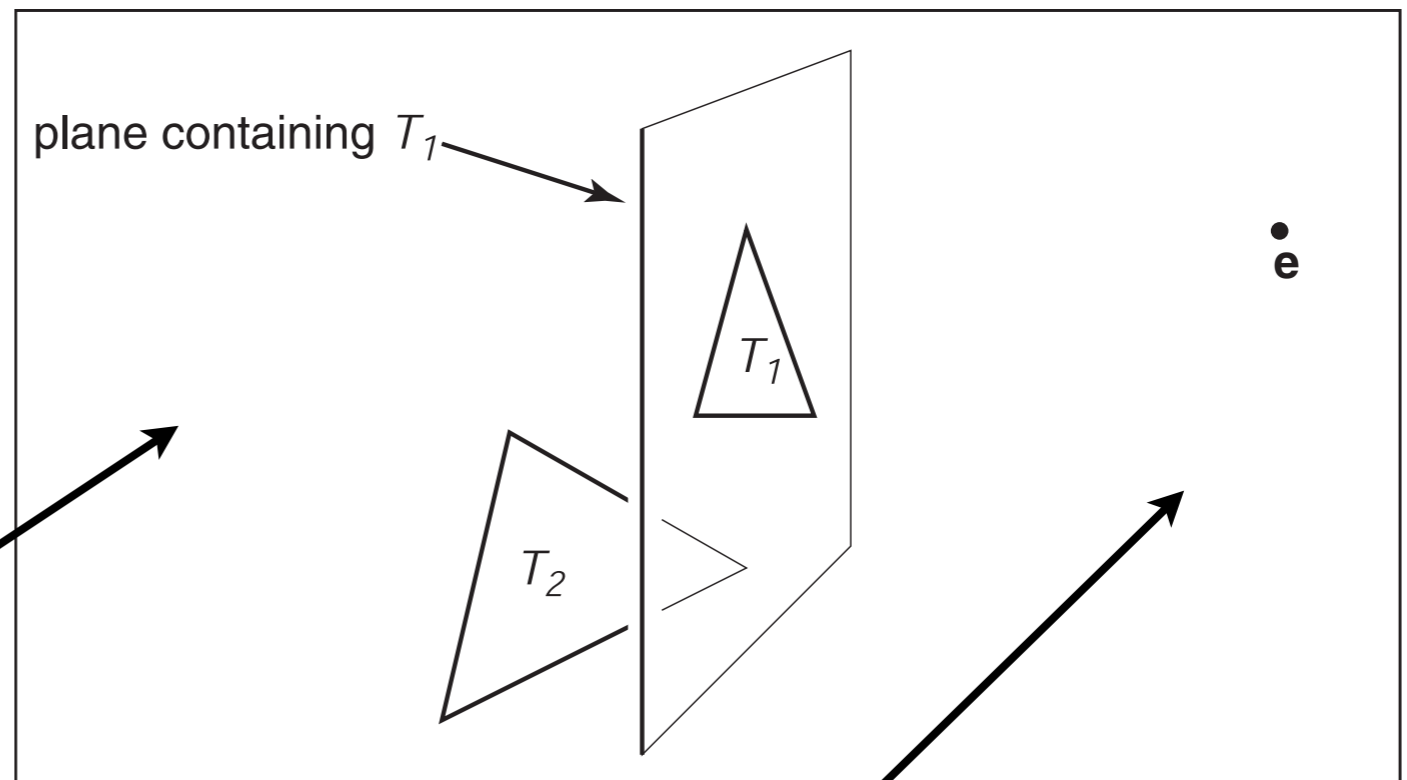
Binary Space Partitioning

Implicit function

$$T_1: f_1(\mathbf{p}) = 0$$

$$f_1(\mathbf{p}) > 0$$

$$f_1(\mathbf{p}) < 0$$



Binary Space Partitioning

if $f_1(\mathbf{e}) < 0$ then

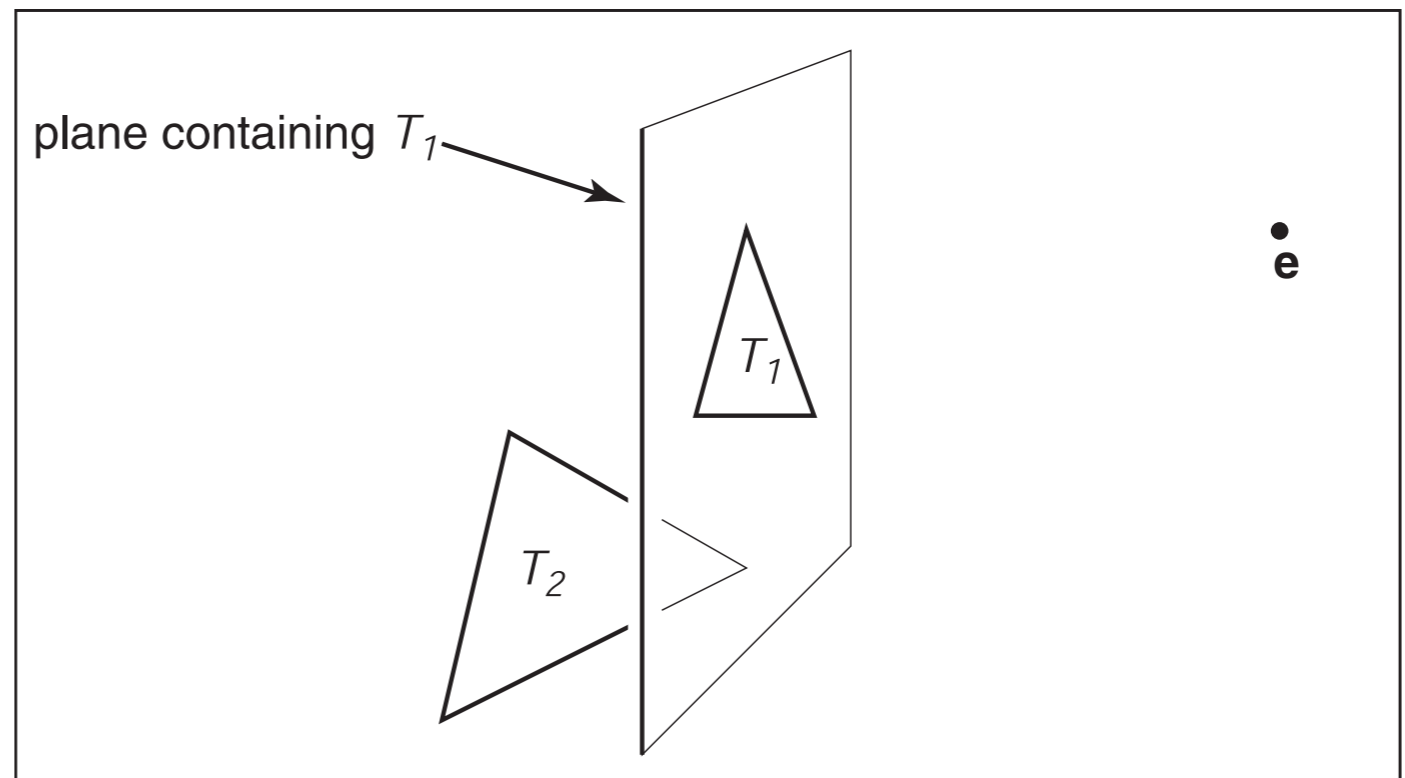
draw T_1

draw T_2

else

draw T_2

draw T_1



BSP Tree

if tree is empty

return

if $f(\mathbf{e}) > 0$ **then**

draw left sub-tree

rasterize T

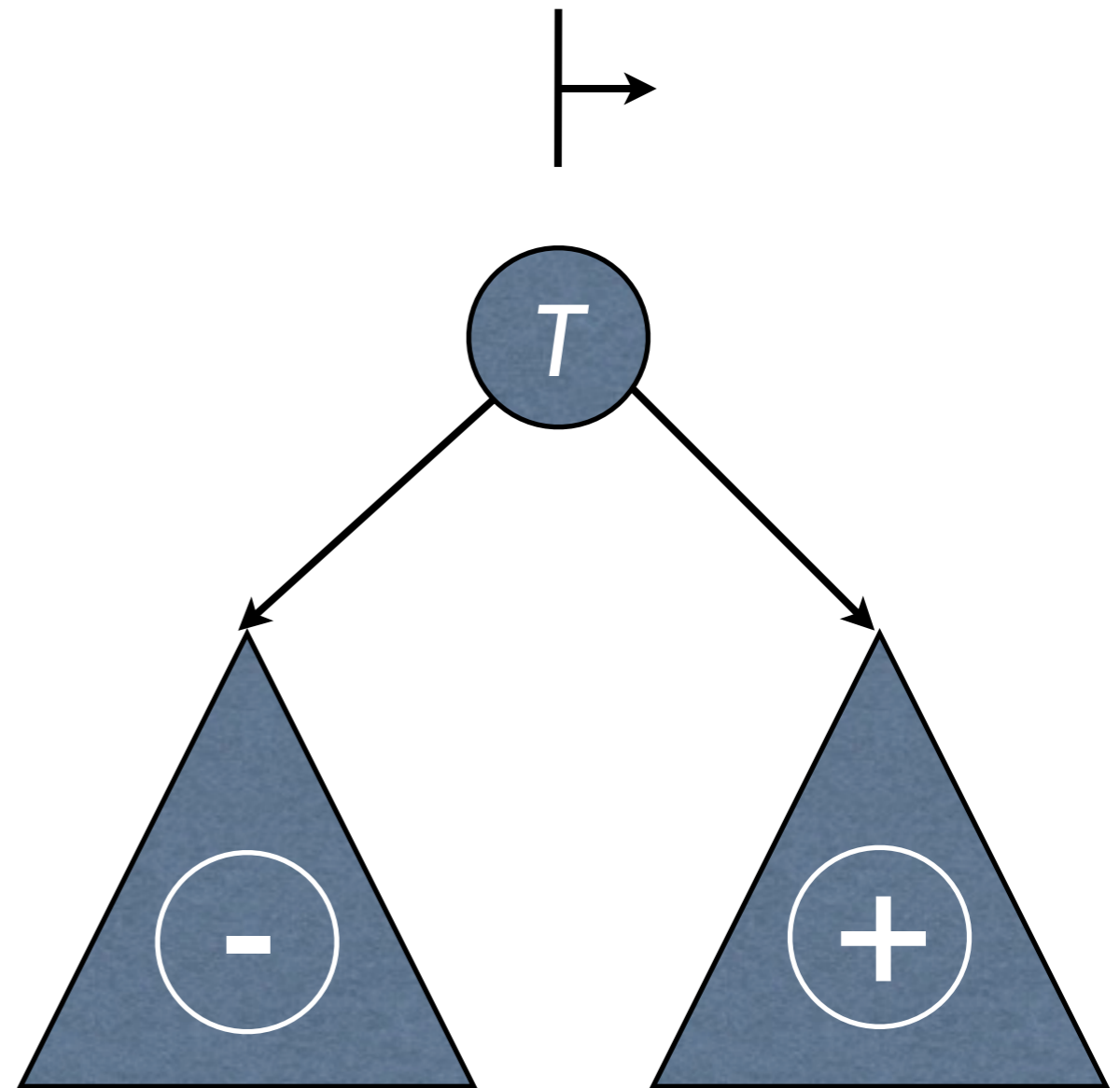
draw right sub-tree

else

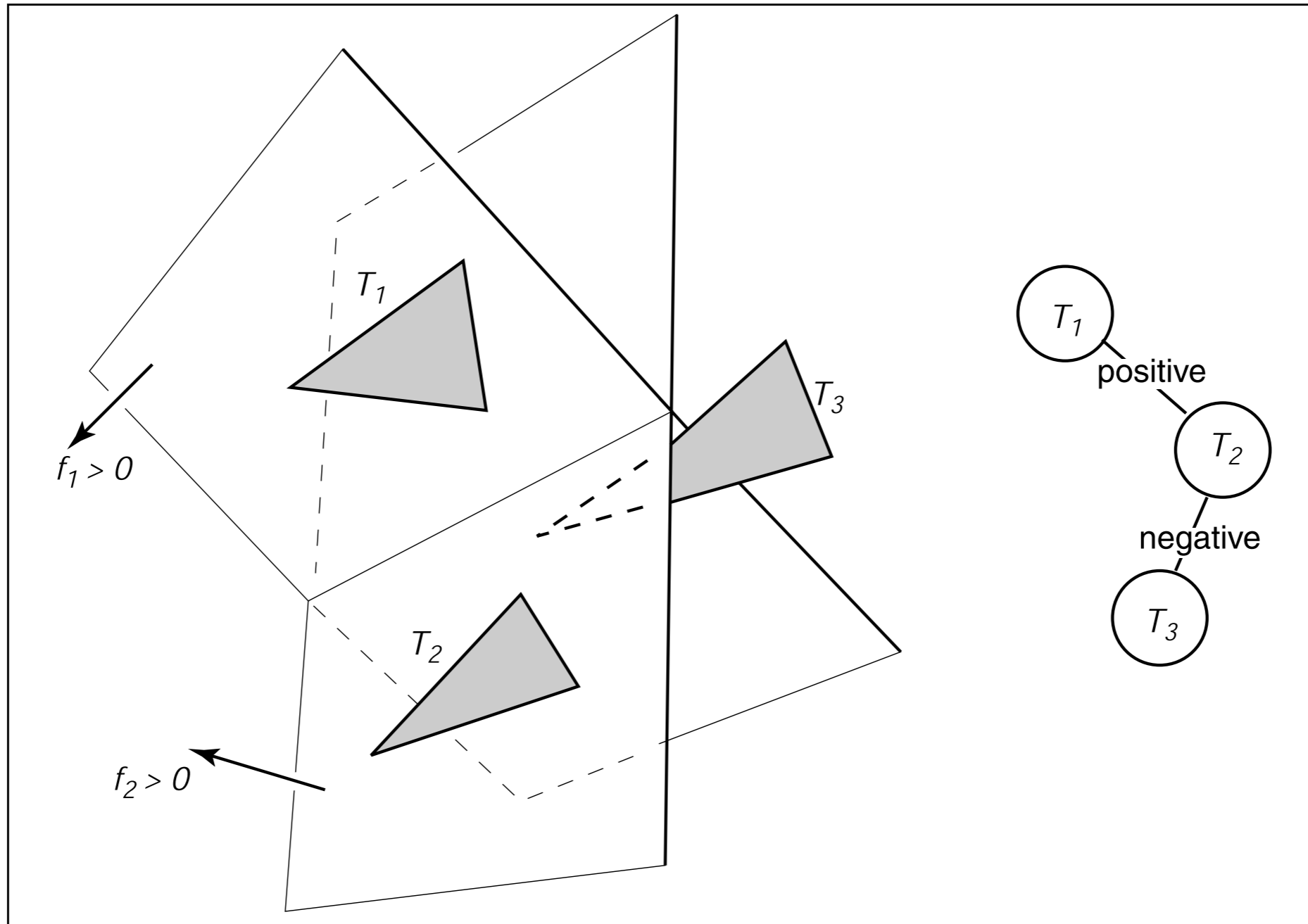
draw right sub-tree

rasterize T

draw left sub-tree



BSP Tree



BSP Tree

```
root = node( $T_1$ )  
for  $i = 2 \dots N$  do  
    tree-root.add( $T_i$ )
```

BSP Tree

algorithm add(root, T)

// Add triangle $T = (\mathbf{a}, \mathbf{b}, \mathbf{c})$ to tree rooted at root

if root is null **then**

 root = node(T)

else **if** root. $f(\mathbf{T.a}) < 0$ and root. $f(\mathbf{T.b}) < 0$ and root. $f(\mathbf{T.c}) < 0$ **then**

 add(root.negative, T)

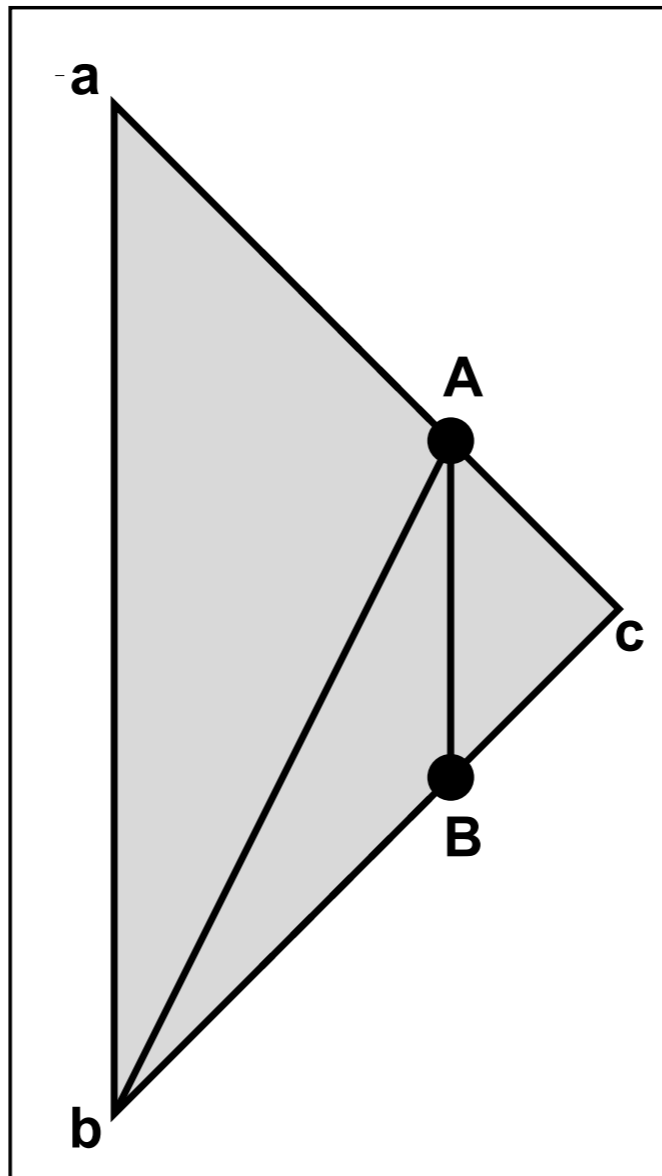
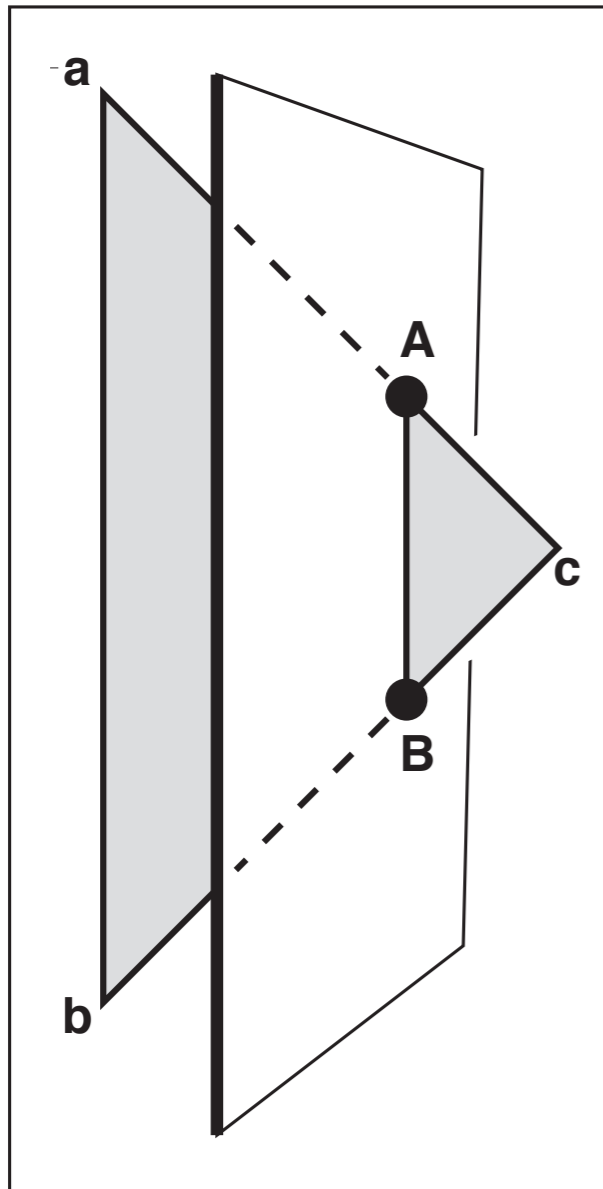
else **if** root. $f(\mathbf{T.a}) > 0$ and root. $f(\mathbf{T.b}) > 0$ and root. $f(\mathbf{T.c}) > 0$ **then**

 add(root.positive, T)

else

so far, we're assuming this doesn't happen

Cutting Triangles



Order matters!

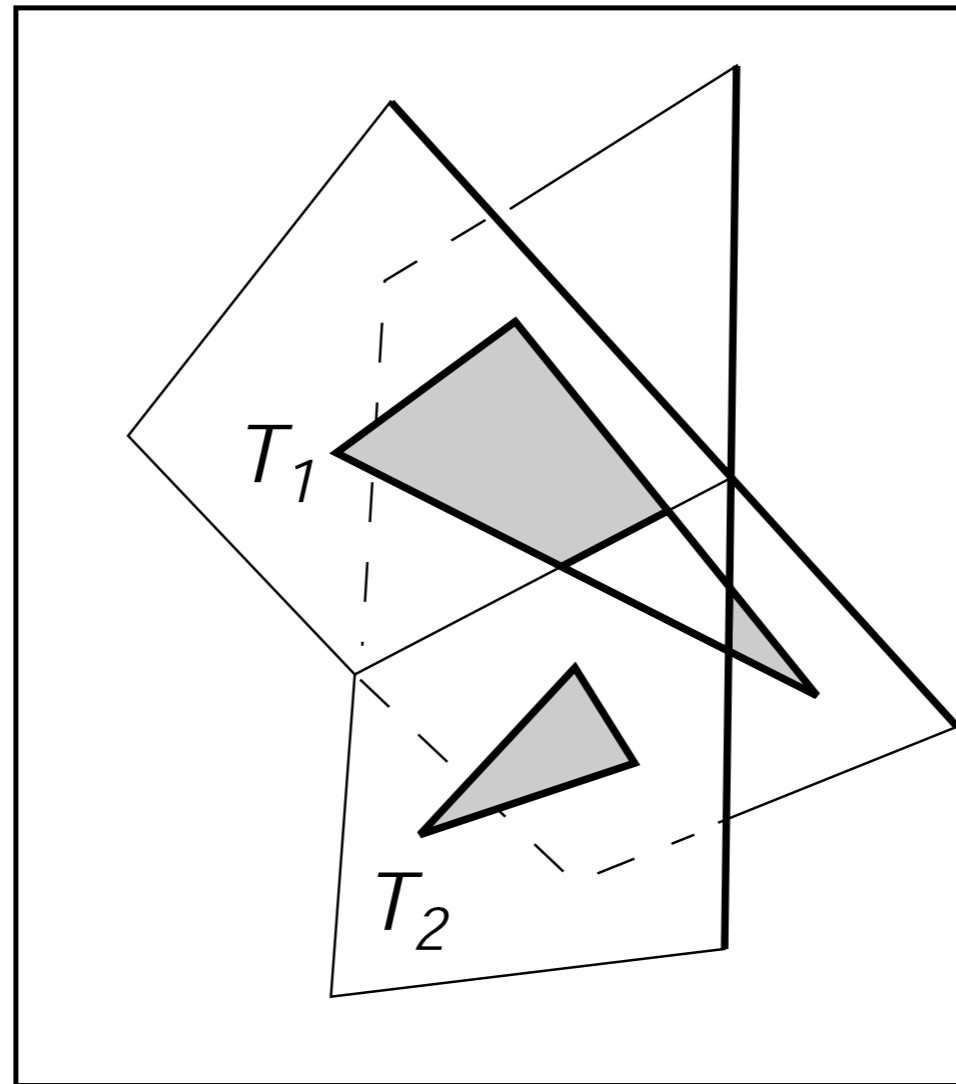
$$T_1 = (a, b, A)$$

$$T_2 = (b, B, A)$$

$$T_3 = (A, B, c)$$

Optimizing Trees

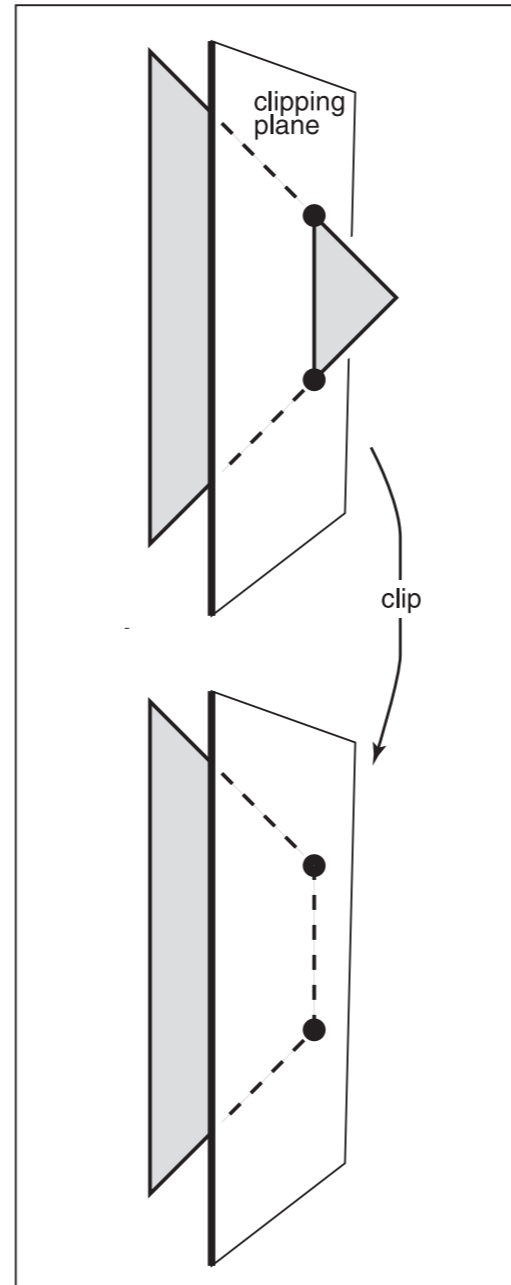
Which root:
 T_1 or T_2
?

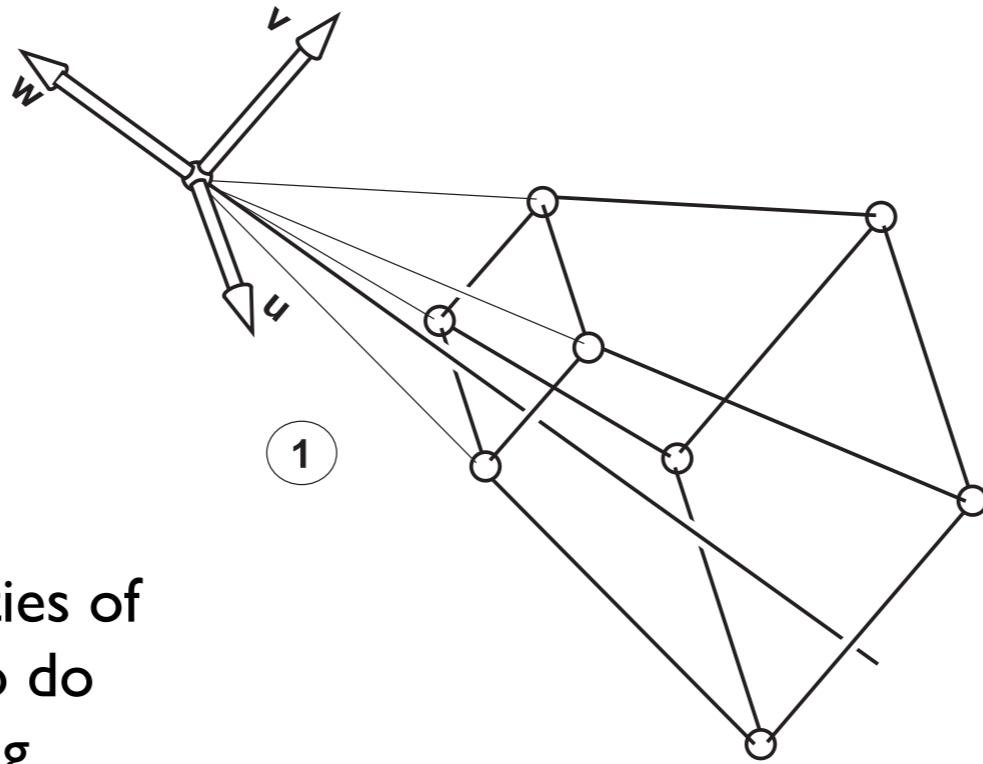


Choosing T_1
2 nodes

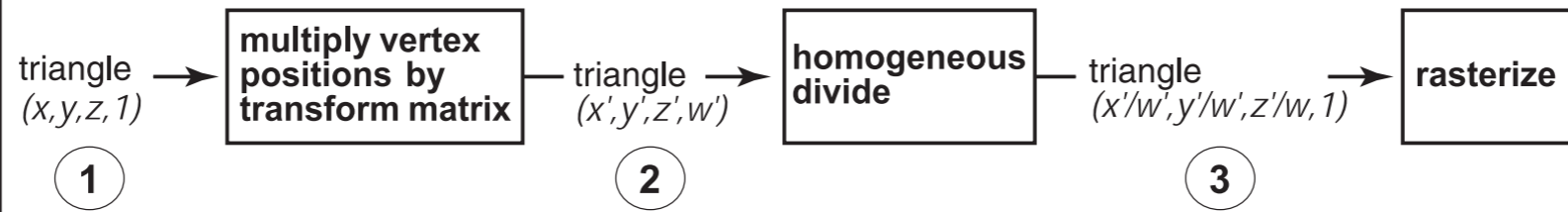
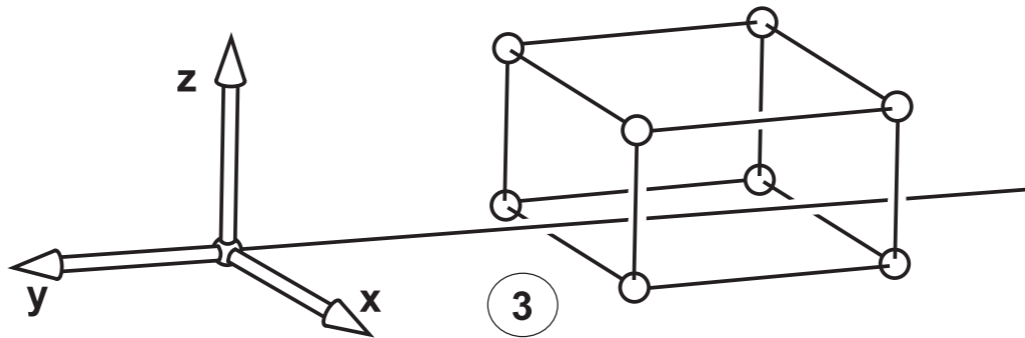
Choosing T_2
3 nodes
(splitting T_1)

Clipping





3 possibilities of where to do clipping



Clipping

```
for each of six planes do  
  if triangle entirely outside plane then  
    break (triangle is not visible)  
  else if triangle straddles plane then  
    clip triangle  
  if quadrilateral is left then  
    break into two triangles
```

Clipping after
homogeneous
divide...

$$z' = n + f - (fn/z)$$

discontinuity at zero

