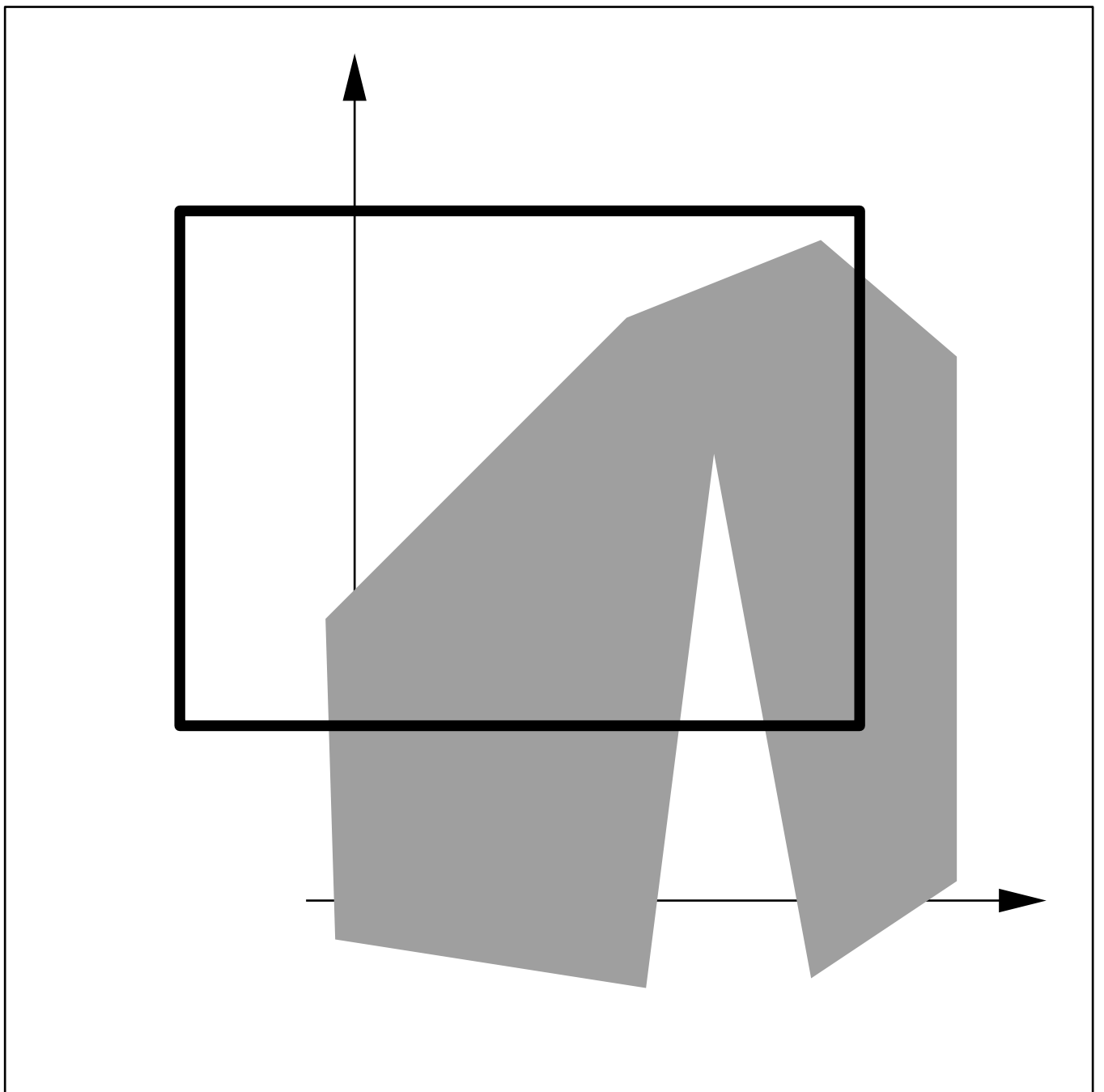
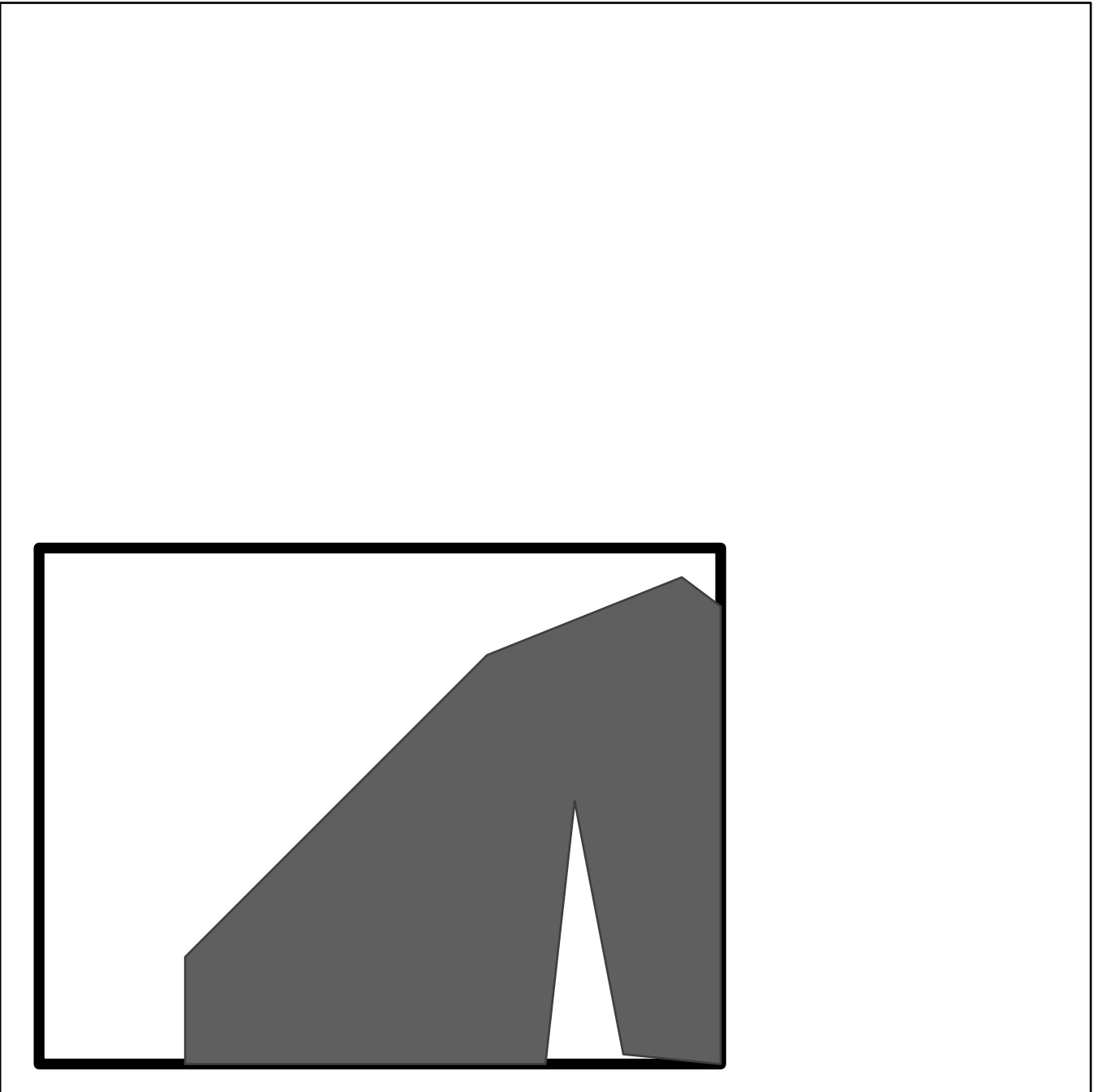


2D Viewing



2D Viewing



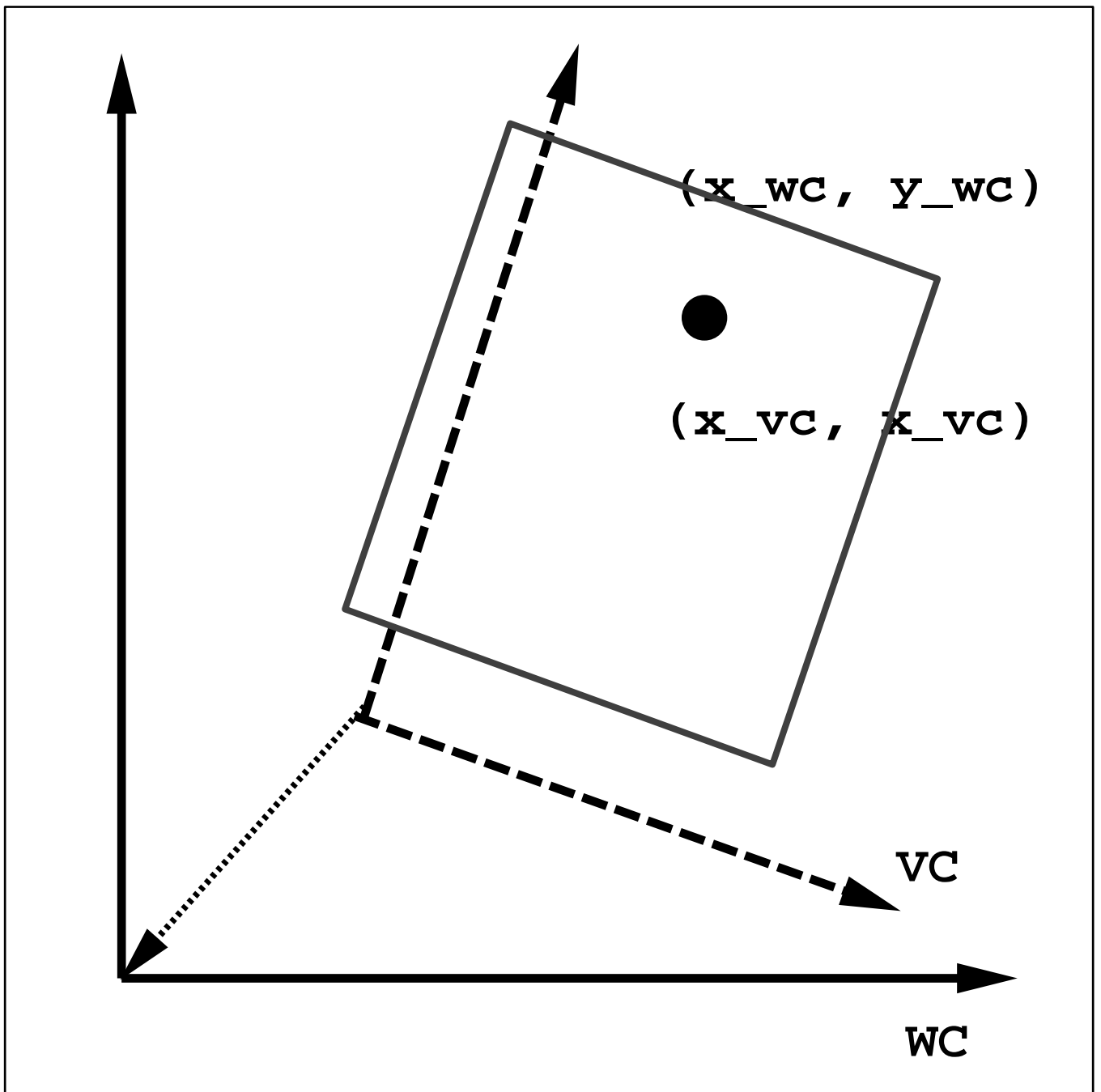
2D Viewing

- How do we specify a **VIEW** of a scene
 - (1) the part of a scene (world) to display (“window”)
 - (2) the place to display that part on the screen (“viewport”)
- Typically, the scene (world) is defined in and convenient coordinate system — “world coordinate system”
- The viewport is generally specified in $([0, 1], [0, 1])$ — “normalized device coordinate system”
- These coordinates are mapped to integer pixel coordinates — “device coordinate system”

2D-Viewing Pipeline

- MC (model coordinate system), models \Rightarrow
- WC (world coordinate system), world \Rightarrow
- VC (view coordinate system) \Rightarrow
- NDC (normalized device coordinate system) \Rightarrow
- DC (device coordinate system)

From WC to VC



From WC to VC

- From model coordinates to world coordinates

$$Obj_{wc} = M_{mc,wc} Obj_{mc}$$

- From world coordinates to viewing coordinates

$$M_{wc,vc} = R(\theta) \star T(\delta x, \delta y)$$

$$Obj_{vc} = M_{wc,vc} Obj_{wc}$$

Window and Viewport

- How do we specify a window?

$x_{w,min}$,

$x_{w,max}$,

$y_{w,min}$,

$y_{w,max}$

- How do we specify a viewport

$x_{v,min}$,

$x_{v,max}$,

$y_{v,min}$,

$y_{v,max}$

- Oftentimes, we want to keep the same relative placement of (x_w, y_w) in the window to

(x_v, y_v) in the viewport
then we must have

$$\frac{x_v - x_{v,min}}{x_{v,max} - x_{v,min}} = \frac{x_w - x_{w,min}}{x_{w,max} - x_{w,min}}$$

$$\frac{y_v - y_{v,min}}{y_{v,max} - y_{v,min}} = \frac{y_w - y_{w,min}}{y_{w,max} - y_{w,min}}$$

- We shall solve for (x_v, y_v)

$$x_v = x_{v,min} + (x_w - x_{w,min})s_x$$

$$y_v = y_{v,min} + (y_w - y_{w,min})s_y$$

where scaling factors are

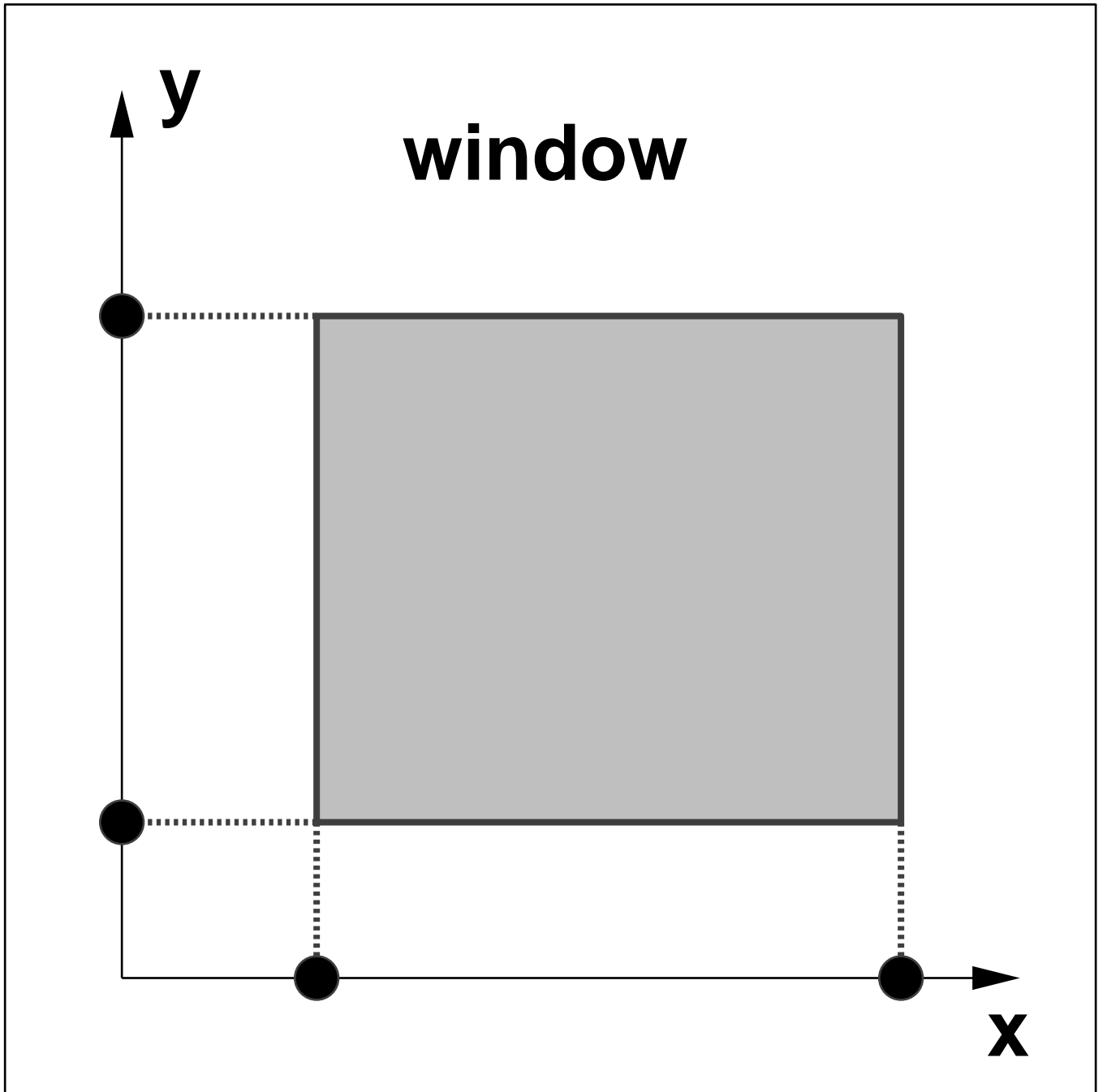
$$s_x = \frac{x_{v,max} - x_{v,min}}{x_{w,max} - x_{w,min}}$$

$$s_y = \frac{y_{v,max} - y_{v,min}}{y_{w,max} - y_{w,min}}$$

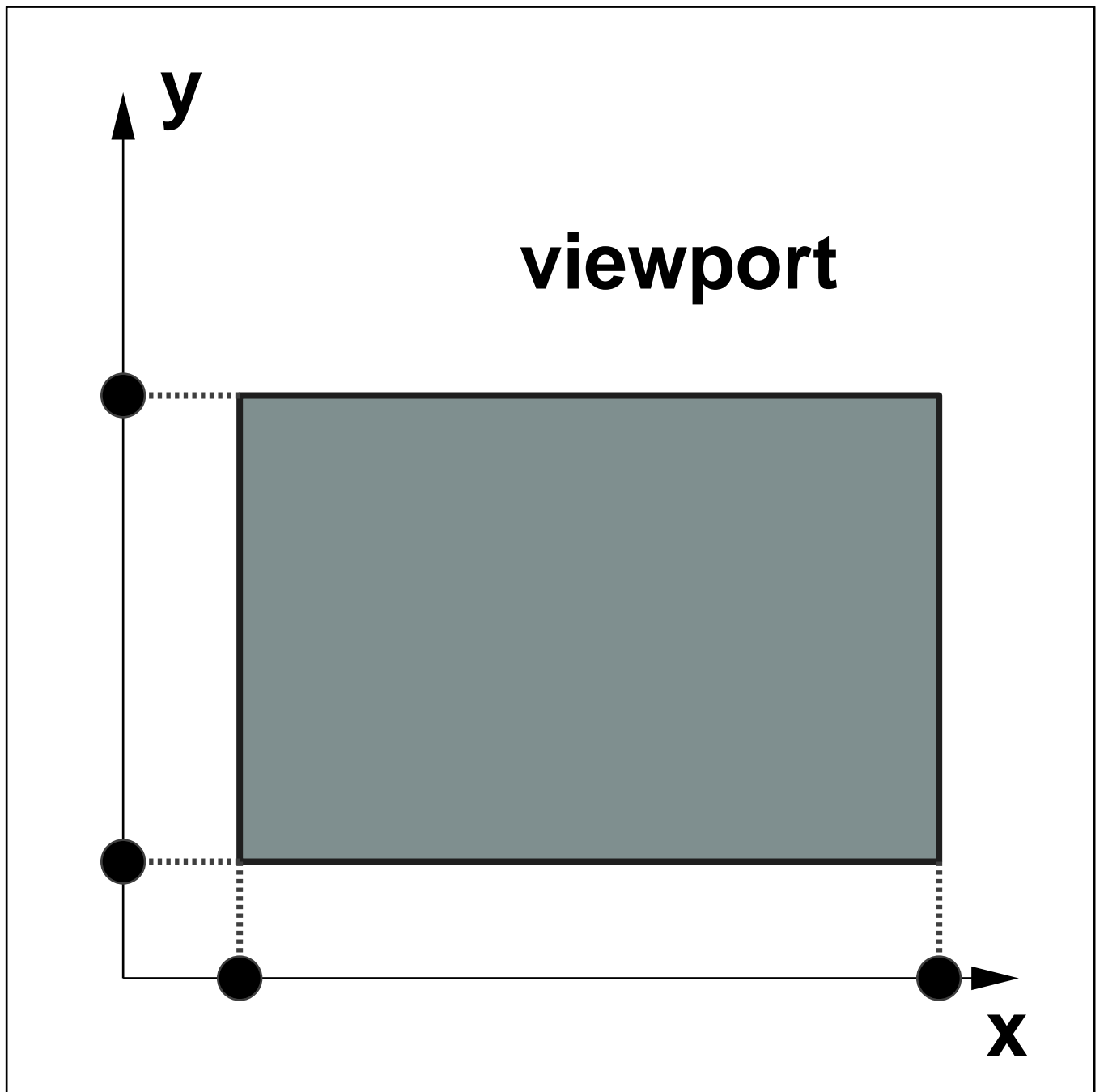
- To maintain relative proportions of objects in this window-to-viewport transformation, we need

$$s_x = s_y$$

Window



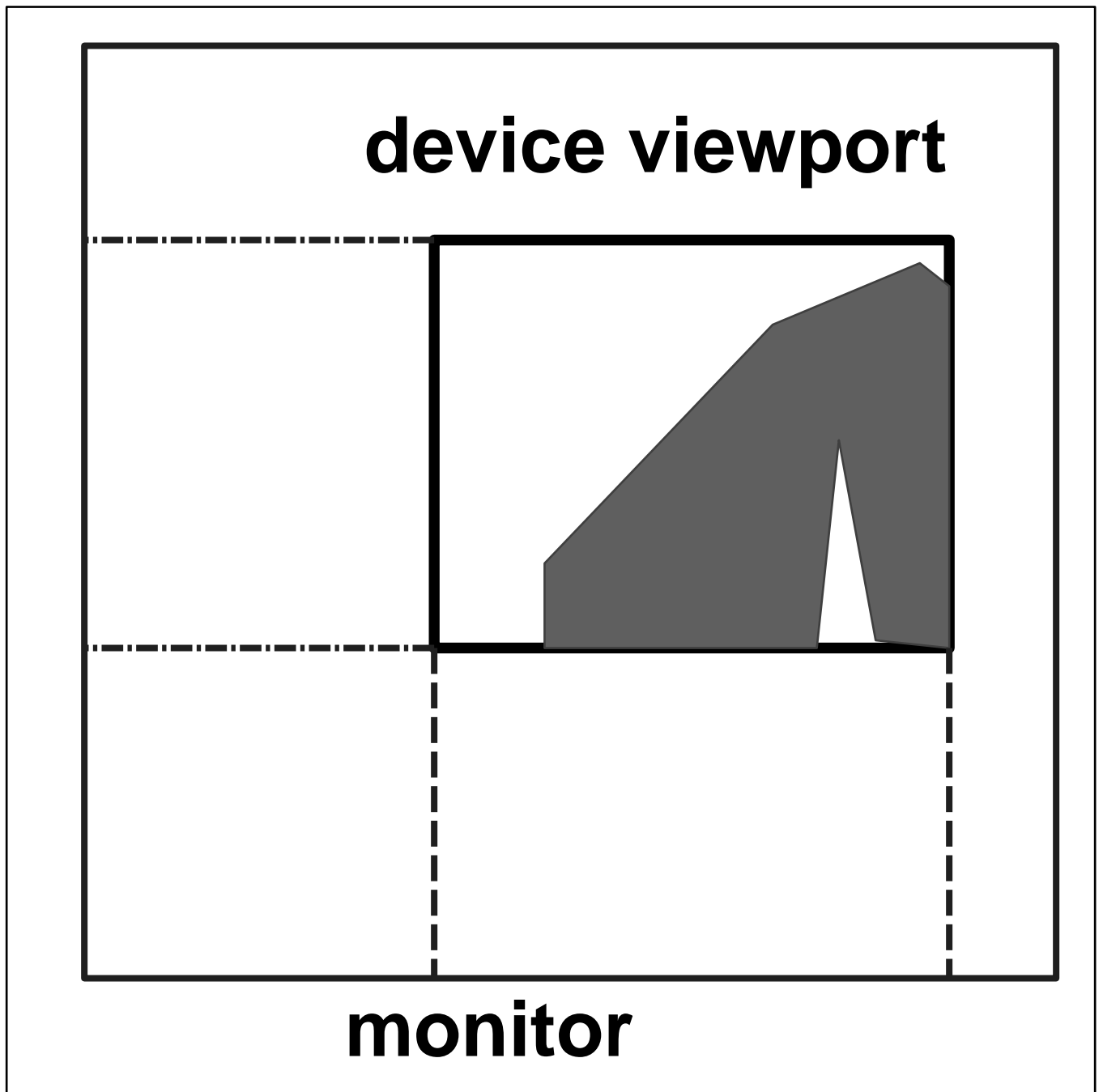
Viewport



Device Coordinate System

- From normalized device coordinate system to device coordinate system
- We need another window-to-viewport transformation
- Why ?
separate transformations from device-dependent requirements
graphics packages become “device-independent”
different devices can be used by providing appropriate device drivers

Device Viewport



Clipping

- Which part(s) of an object should be on or off the screen ?