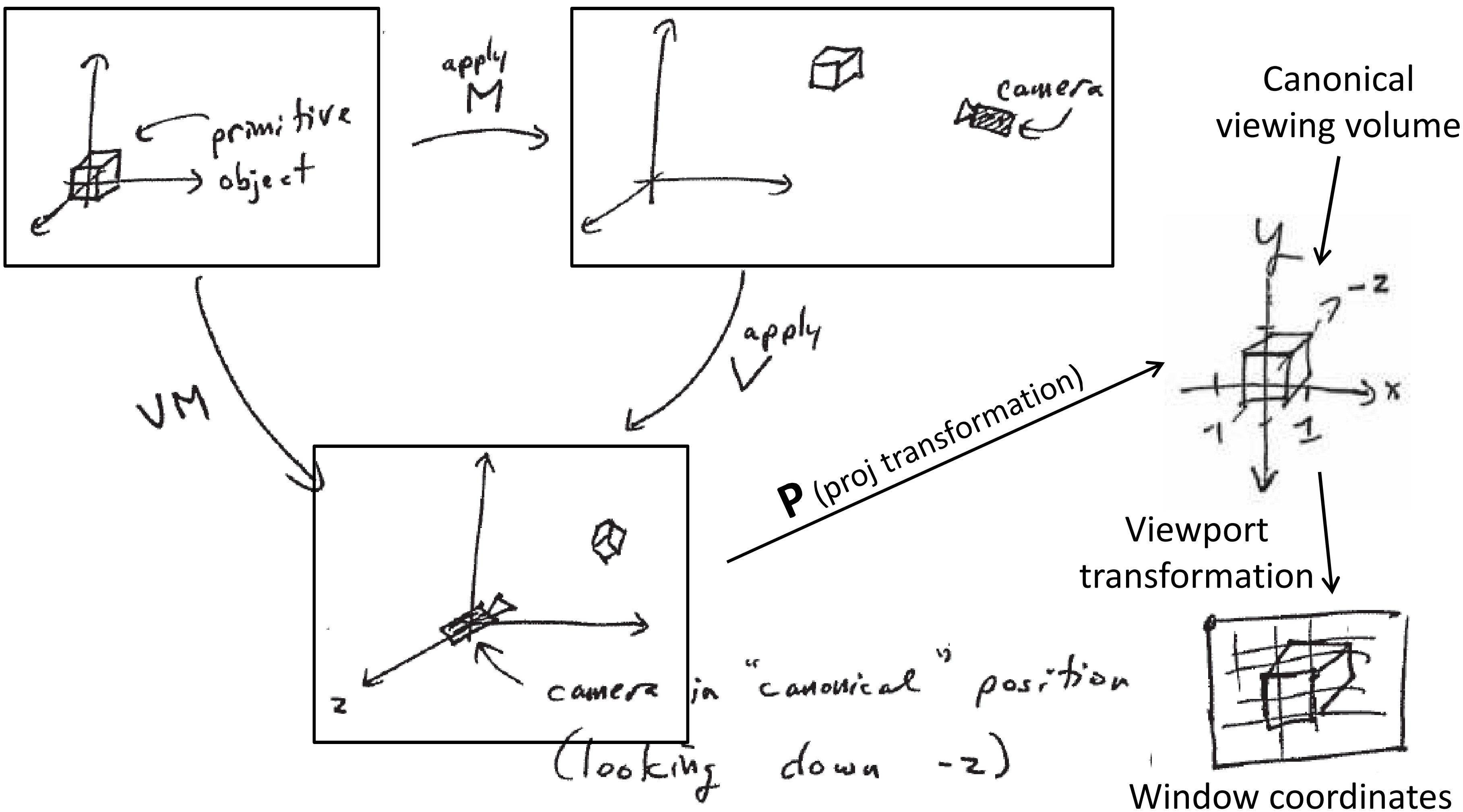
CS 428: Fall 2009

Introduction to Computer Graphics

Viewing and projective transformations

Modeling and viewing transformations



Modeling and viewing transformations

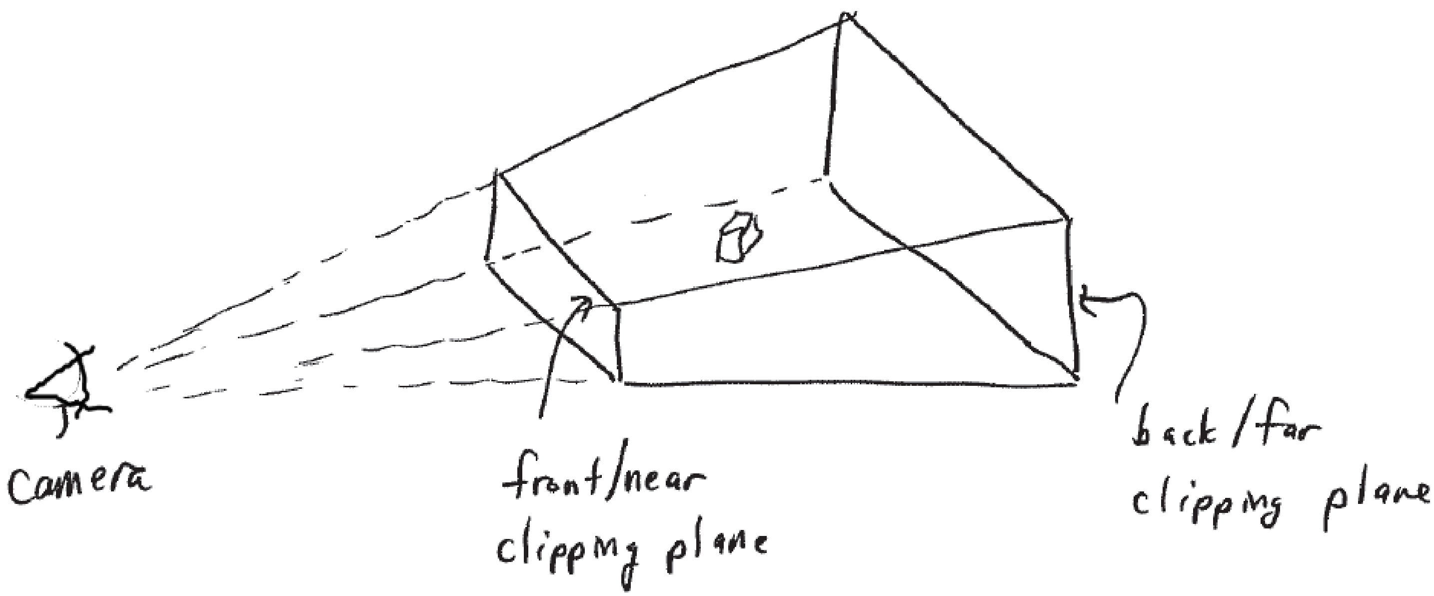
- OpenGL order

```
glMatrixMode(GL_MODELVIEW)
glLoadIdentity()
glMultMatrix(V)
glMultMatrix(M)
draw()          <------------- Transformation is VM
```

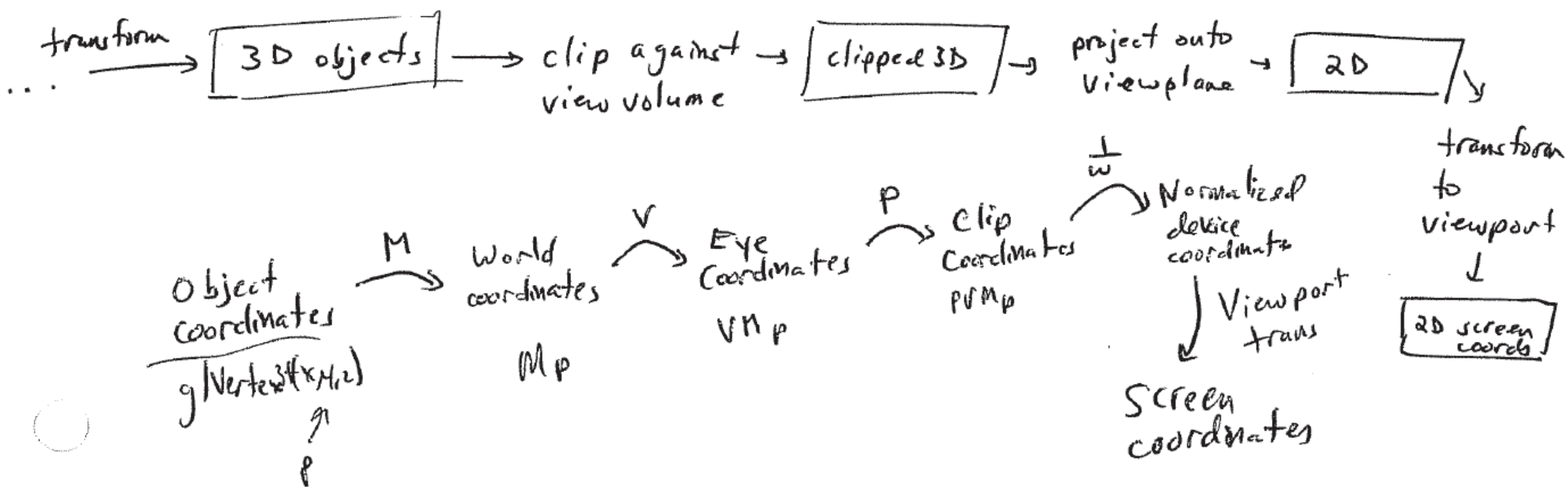
  - In OpenGL, these transformations are place on the **modelview** matrix stack
  - The **projection** matrix stack is only for storing the projection matrix resulting from `glOrtho()`, `glFrustum()`, or `gluProjection()`

3D viewing

- The eye has a view cone
  - Approximated in CG by a "rectangular cone" = square **frustum**
  - Good for rectangular viewing window

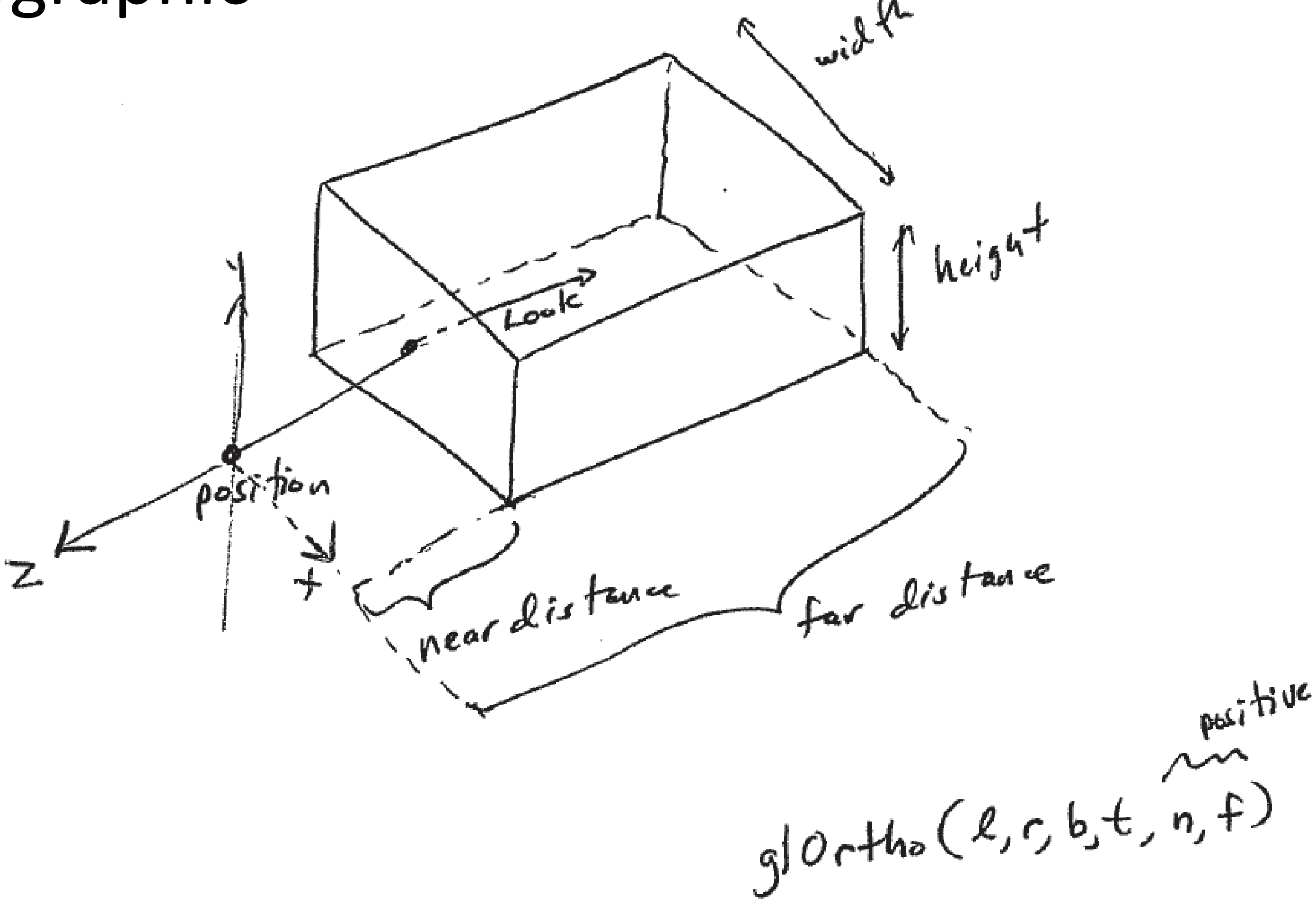


3D viewing process



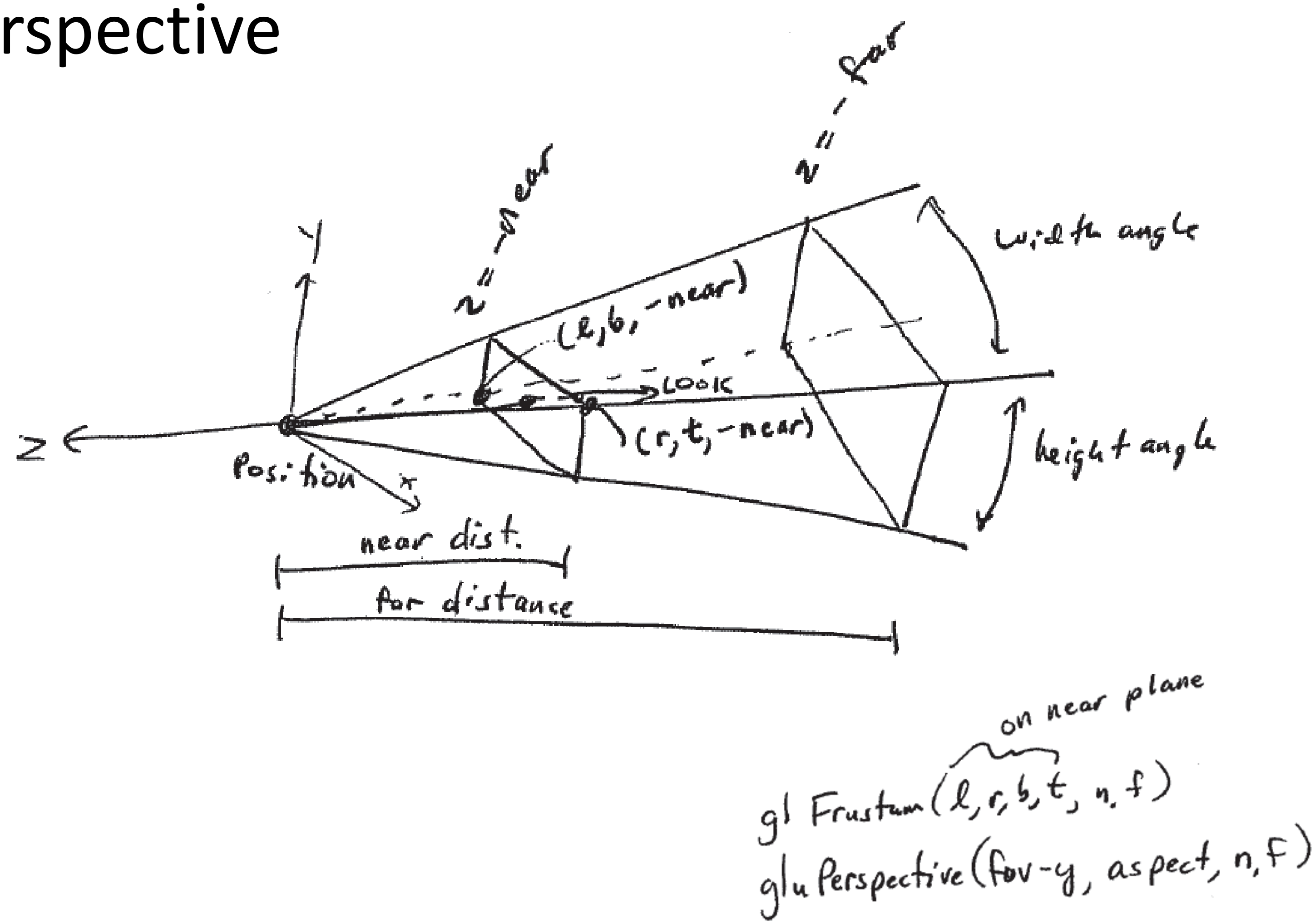
Truncated view volumes

- Orthographic



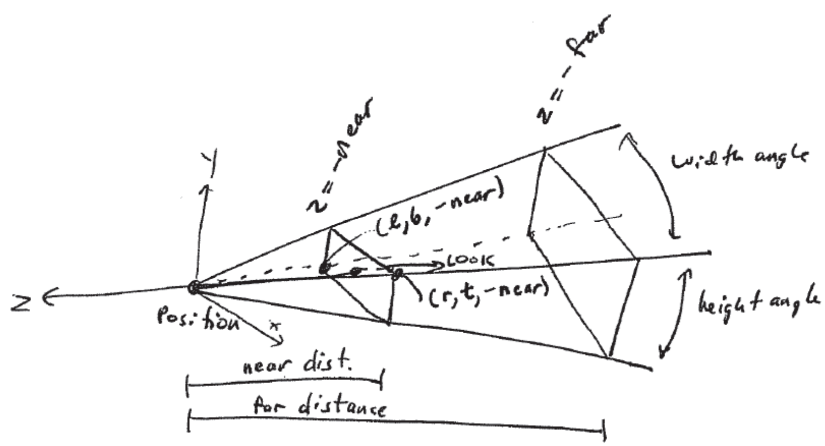
Truncated view volumes

- Perspective



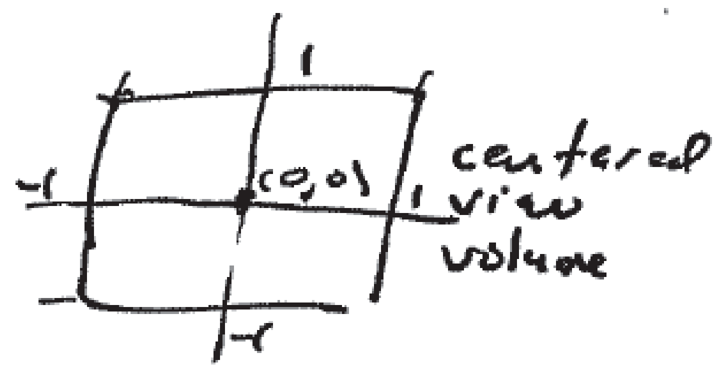
Where's the film?

- A rectangle with known aspect ratio on the infinite film plane
- "Where" doesn't matter, as long as the film plane is parallel to *far* and *near*
- Will be scaled to viewport coordinates



OpenGL projection matrices

- How is this implemented in OpenGL?
- The following matrices assume
  - Camera center at origin
  - Looking down negative z-axis
  - y-axis is "up"
  - *near*, *far* > 0
  - *right* = -*left* = 1
  - *top* = -*bottom* =1



- See **`glOrtho()`** etc. manpages for general case

OpenGL projection matrices

- Orthographic projection

$$P_{ortho} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \frac{2}{n-f} & \frac{n+f}{n-f} \\ & & & 1 \end{bmatrix}$$

$$P_{ortho} \cdot \begin{bmatrix} x \\ y \\ -n \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -1 \\ 1 \end{bmatrix} \qquad P_{ortho} \cdot \begin{bmatrix} x \\ y \\ -f \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \\ 1 \end{bmatrix}$$

maps ortho view volume to $x, y, z \in [-1, 1]$ (cube @ origin)

where near $\rightarrow z = -1$

far $\rightarrow z = +1$

normalized device coordinates

OpenGL projection matrices

- Perspective projection $P_{persp} = \begin{bmatrix} \frac{n}{r} & & & \\ & \frac{n}{t} & & \\ & & \frac{n+f}{n-f} & \frac{2nf}{n-f} \\ & & -1 & \end{bmatrix}$

$$P_{persp} \cdot \begin{bmatrix} x \\ y \\ -n \\ 1 \end{bmatrix} = \begin{bmatrix} (\frac{n}{r})x \\ (\frac{n}{t})y \\ -n \\ n \end{bmatrix} \xrightarrow{\div w} \begin{bmatrix} x/r \\ y/t \\ -1 \\ 1 \end{bmatrix}$$

$$P_{persp} \cdot \begin{bmatrix} x \\ y \\ -f \\ 1 \end{bmatrix} = \begin{bmatrix} (\frac{n}{r})x \\ (\frac{n}{t})y \\ f \\ f \end{bmatrix} \xrightarrow{\div w} \begin{bmatrix} \frac{n}{f} \cdot \frac{x}{r} \\ \frac{n}{f} \cdot \frac{y}{t} \\ 1 \\ 1 \end{bmatrix}$$

maps frustum to cube @ origin $[-1,1]^3$

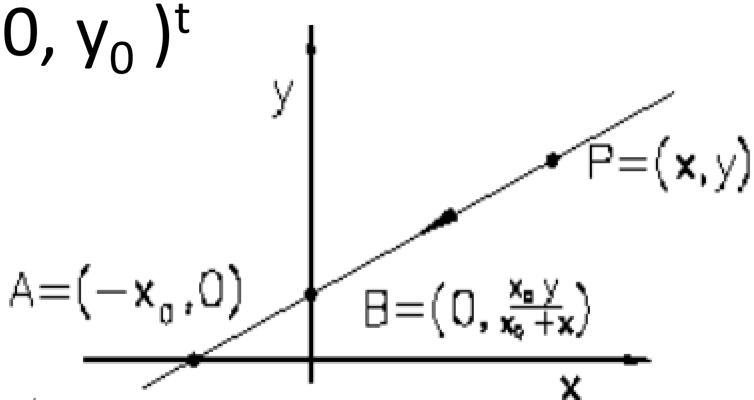
$$P_{persp} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{n}{r} x \\ \frac{n}{t} y \\ \cdots \\ -z \end{bmatrix} \xrightarrow{\div w} \begin{bmatrix} -\frac{n}{r} \frac{x}{z} \\ -\frac{n}{t} \frac{y}{z} \\ \cdots \\ 1 \end{bmatrix} \leftarrow$$

x, y divided by z → foreshortening

Perspective projections

- Perspective projections are not affine transformations
  - Relative lengths are no longer invariant
  - Distant objects (of same size) are made smaller than near ones (= foreshortening)
  - Given a projected point $P=(x, y)^t$ and eye position $A=(-x_0, 0)^t$, then by the theorem of intersecting lines the image B is $(0, y_0)^t$

$$\frac{y_0}{y} = \frac{x_0}{x + x_0}$$



Perspective projections

- In general, the mapping is

$$y_0 = y \cdot \frac{x_0}{x_0 + x} \qquad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ \dfrac{y \cdot x_0}{x_0 + x} \end{pmatrix}$$

- Resulting in the homogeneous 3 × 3-Matrix
  - 2D-Geometry!

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & x_0 & 0 \\ 1 & 0 & x_0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ \dfrac{1}{x_0} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ x_0 \cdot y \\ x + x_0 \end{bmatrix} = \begin{bmatrix} 0 \\ \dfrac{x_0}{x + x_0} \cdot y \\ 1 \end{bmatrix}$$

Perspective projections

- The projection is composed of two transformations
  - The perspective transformation
  - The subsequent parallel projection

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{x_0} & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{x_0} & 0 & 1 \end{bmatrix}$$

Perspective projection **Parallel-projection** **Perspective transformation**

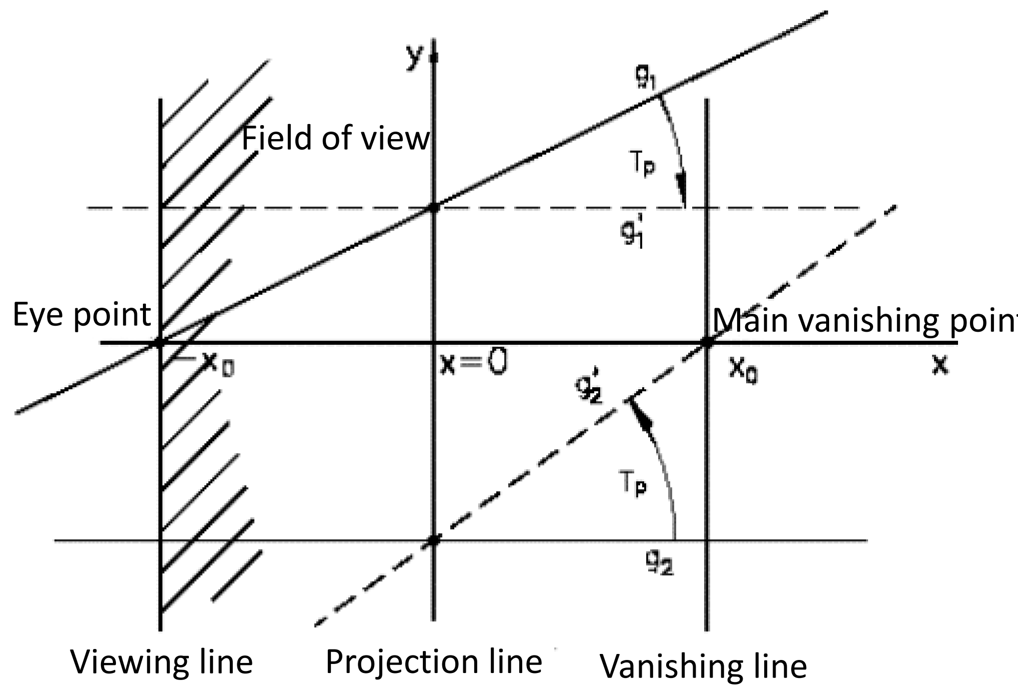
Perspective transformations

- Properties of perspective transformations of the form

$$T_p \cdot \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{x_0} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{x}{x_0} + w \end{bmatrix}$$

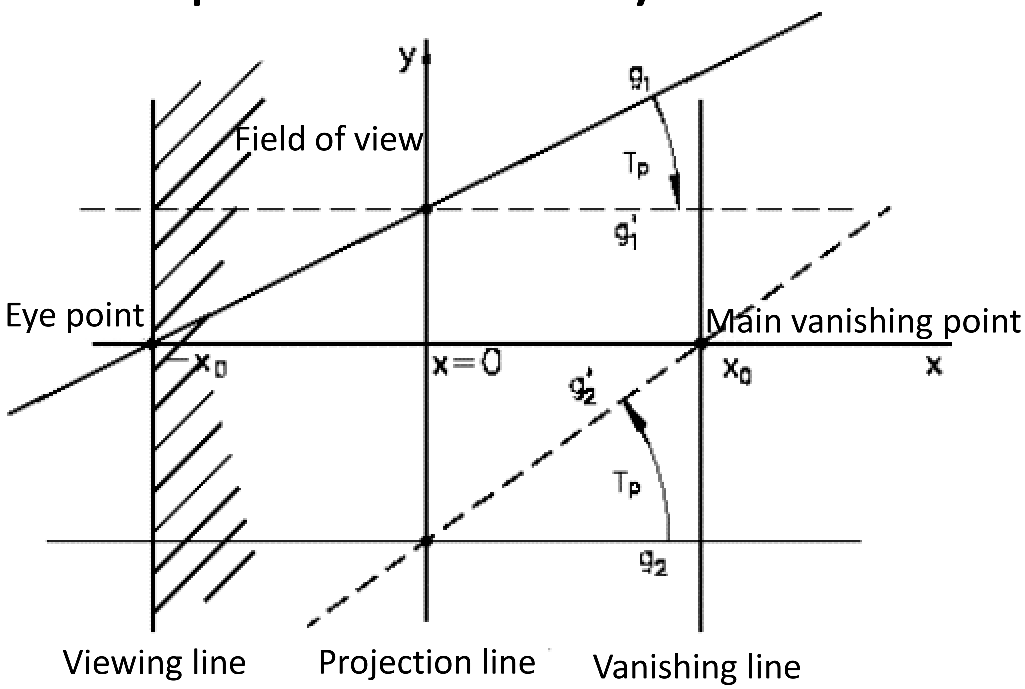
Field of view and viewing line

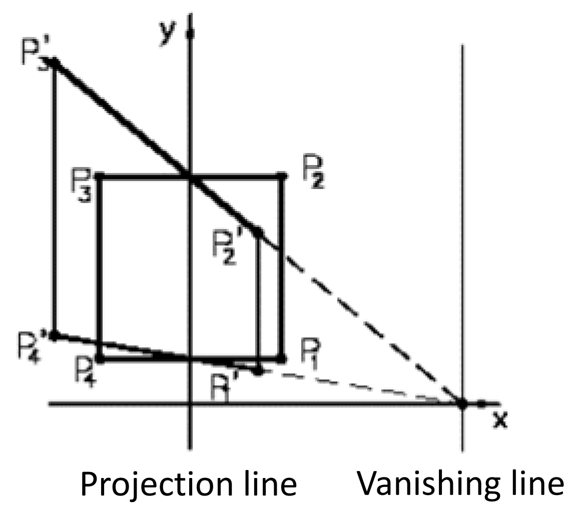
- All points on the affine Line $x = -x_0$ are mapped to infinite points
- Only points on one side of this line are transformed
- These points are in the field of view, and the line $x = -x_0$ is the viewing line



Fixed points and lines

- Points on the projection line x=0 (= y-axis) together with the infinite point [0, 1, 0] are fixed points of this transformation (the y-axis is invariant)
- Lines parallel to the y-axis remain parallel





Fixed points and lines

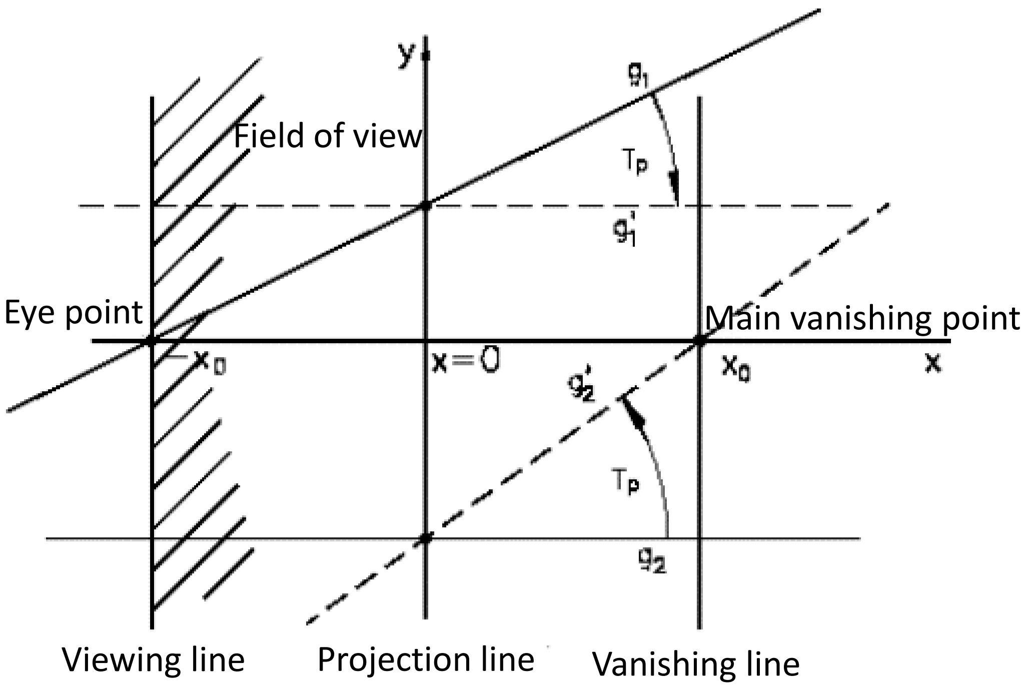
- Points on the projection line x=0 (= y-axis) together with the infinite point [0, 1, 0] are fixed points of this transformation (the y-axis is invariant)
- Lines parallel to the y-axis remain parallel

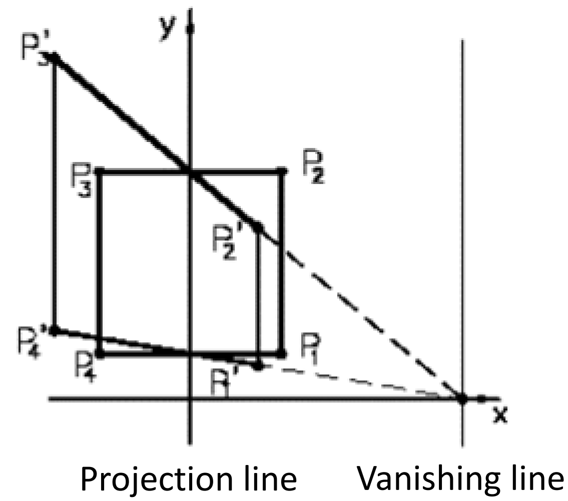
$$T_p \cdot \begin{bmatrix} x_{obj} \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{x_0} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{obj} \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x_{obj} \\ y \\ \frac{x_{obj}}{x_0} + 1 \end{bmatrix} = \begin{bmatrix} x_{obj} \\ y \\ \frac{x_{obj} + x_0}{x_0} \end{bmatrix} = \begin{bmatrix} \frac{x_{obj} \cdot x_0}{x_{obj} + x_0} \\ \frac{y \cdot x_0}{x_{obj} + x_0} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{image} \\ y \cdot a_{factor} \\ 1 \end{bmatrix}$$

- Lines are transformed to lines

Parallel lines

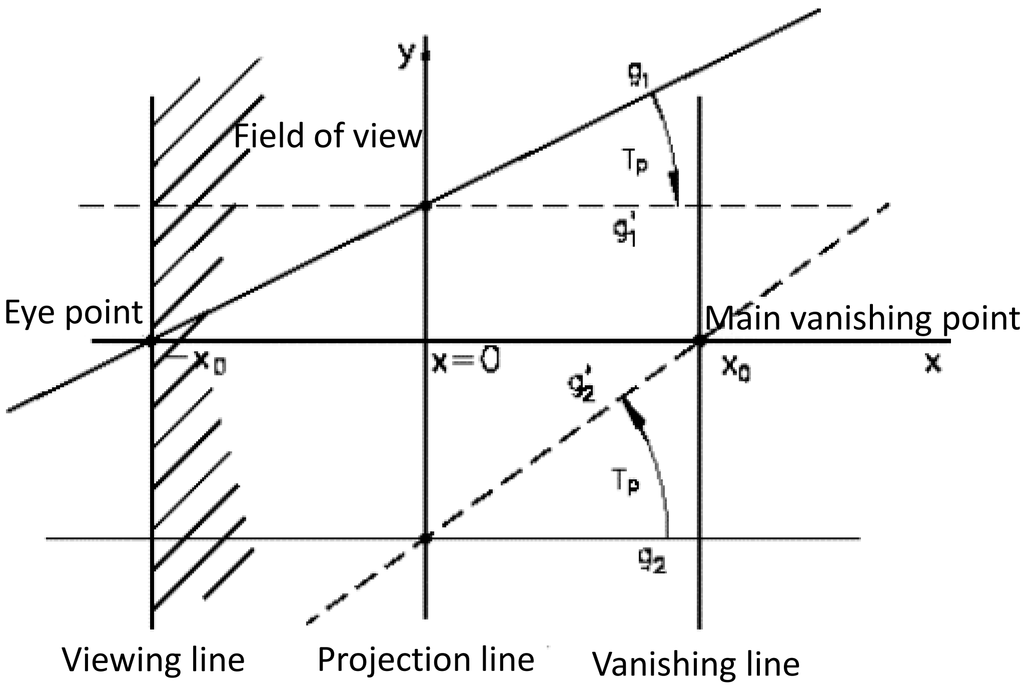
- The affine (eye) point $[-x_0, 0, 1]^t$ is transformed to the infinite point $[-x_0, 0, 0]^t = [-1, 0, 0]^t$
- The points on the affine y-axis are invariant

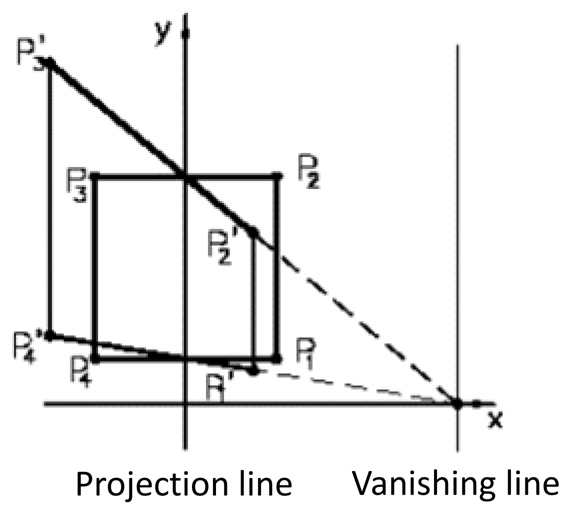




Parallel lines

- Lines are mapped to lines
  - A line through eye point $[-x_0, 0, 1]^t$, which intersects the y-axis at $[0, y_0, 1]^t$ is mapped to a line parallel to the x-axis passing through $[0, y_0, 1]^t$



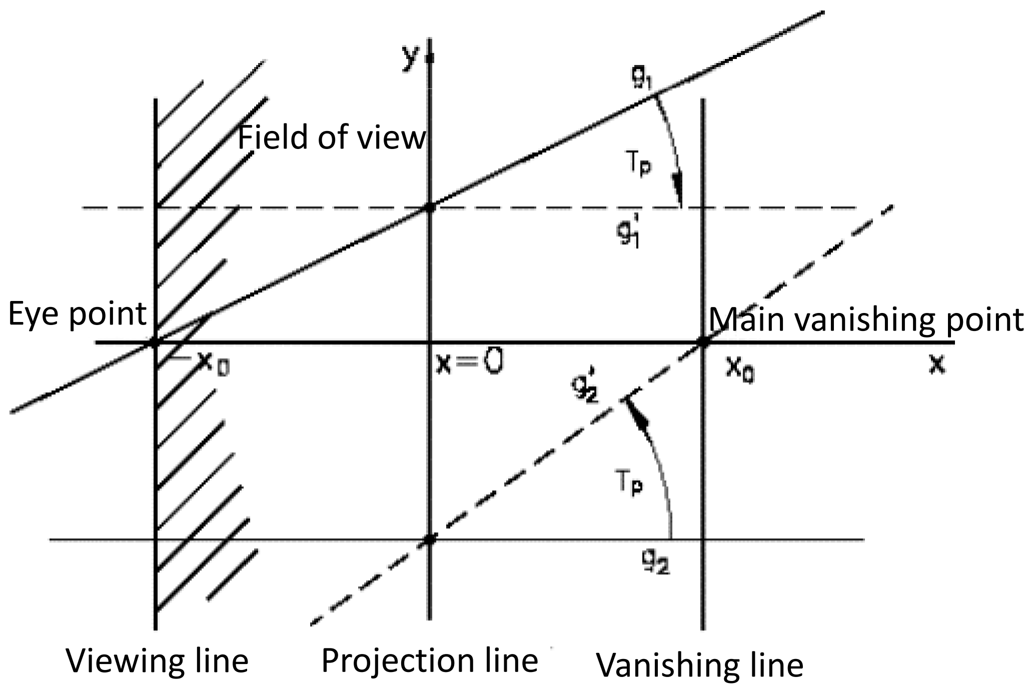


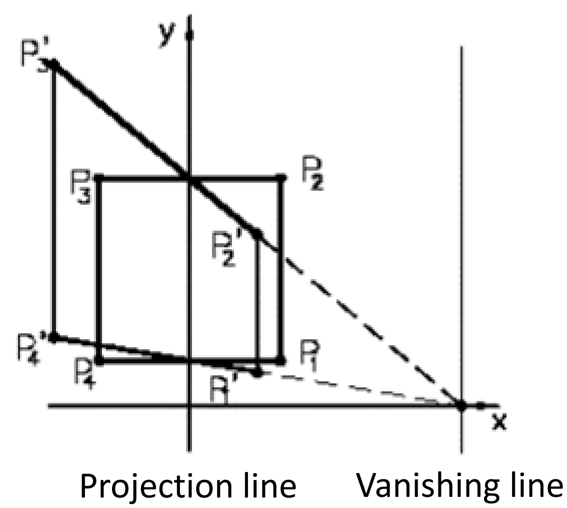
Vanishing line

- The point $[x, y, 0]^t$ is mapped to $[x_0, x_0 \cdot y/x, 1]^t$
  - Note that $[x, y, 0]^t$ is a **direction** and $[x_0, x_0 \cdot y/x, 1]^t$ is a **point**
- The mappings of all lines parallel to the affine line with direction $[x, y, 0]^t$ contain the point $[x_0, x_0 y/x, 1]^t$, meaning they all intersect in this point
- The union of the mappings of all lines with direction $[x, y, 0]^t$ lie on the line $x=x_0$ (= vanishing line)

Vanishing line

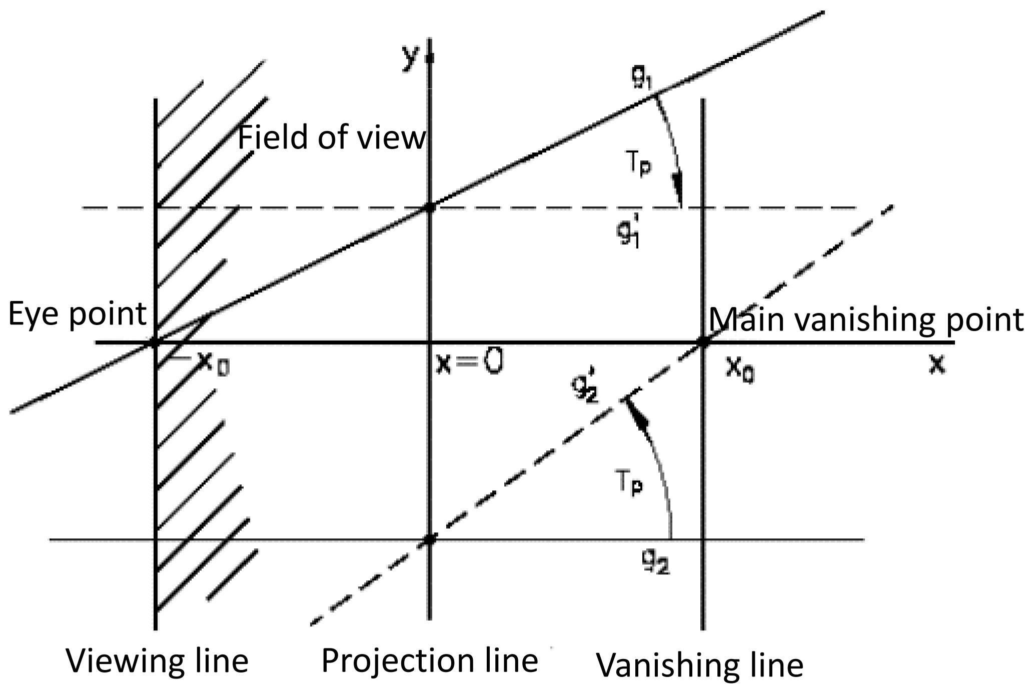
- The point $[x, y, 0]^t$ is mapped to $[x_0, x_0 \cdot y/x, 1]^t$
  - Note that $[x, y, 0]^t$ is a **direction** and $[x_0, x_0 \cdot y/x, 1]^t$ is a **point**

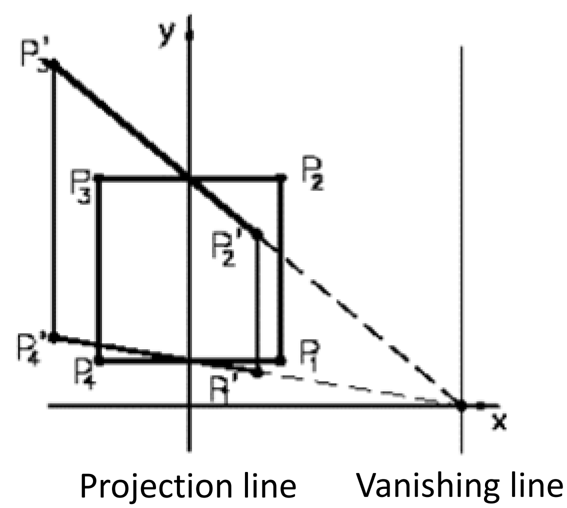




Vanishing point

- The point $[x, y, 0]^t$ is mapped to $[x_0, x_0 \cdot y/x, 1]^t$
  - Note that $[x, y, 0]^t$ is a **direction** and $[x_0, x_0 \cdot y/x, 1]^t$ is a **point**
  - **All lines with direction $[x, 0, 0]^t$ are mapped to $[x_0, 0, 0]^t$**



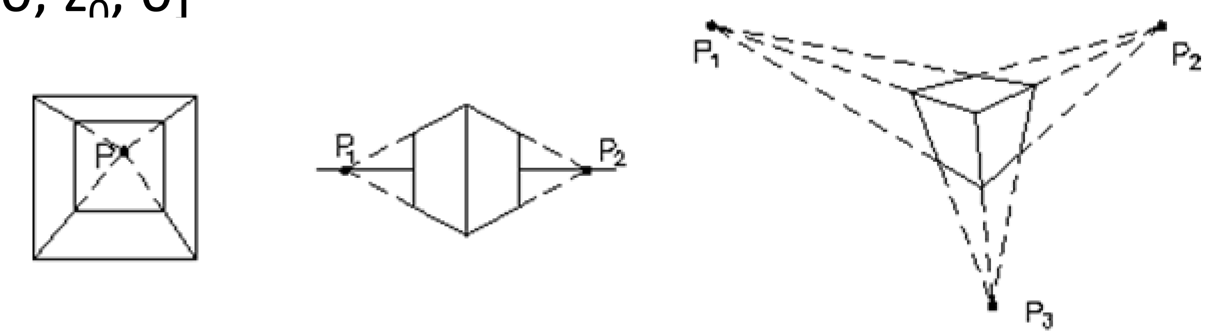


One, two and three vanishing point perspectives

- General perspective transformation

$$T_p \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{x_0} & \frac{1}{y_0} & \frac{1}{z_0} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \frac{x}{x_0} + \frac{x}{y_0} + \frac{x}{z_0} + w \end{bmatrix}$$

  - The directions of lines parallel to the coordinate axes are mapped to the vanishing points $[x_0, 0, 0, 0]^t$, $[0, y_0, 0, 0]^t$, $[0, 0, z_0, 0]^t$



In OpenGL code

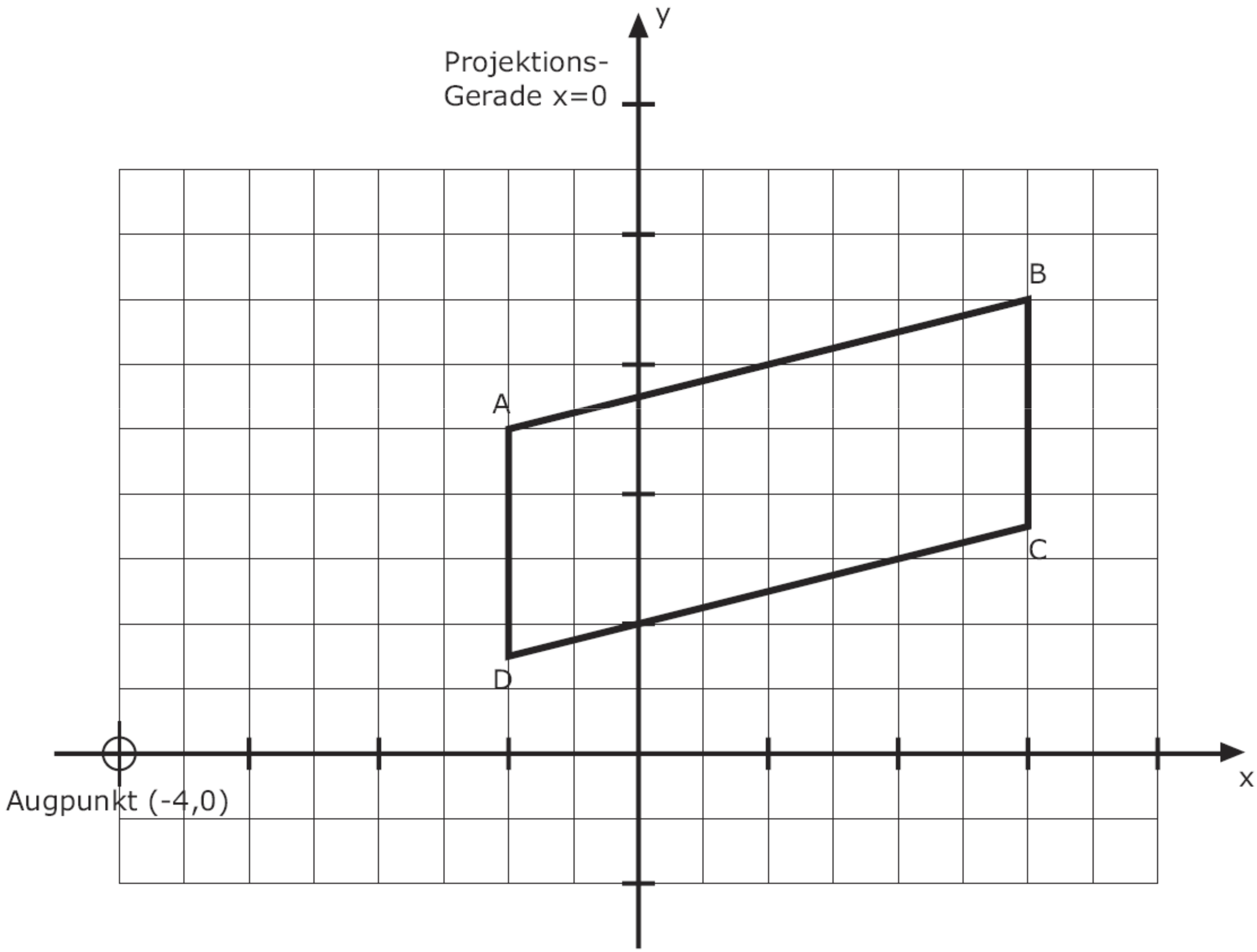
- For example, in **`reshape(...)`**

```
glViewport(0, 0, width, height)
glMatrixMode(GL_PROJECTION)
glLoadIdentity()
gluPerspective(...)      ← Perspective transformation
glMatrixMode(GL_MODELVIEW)
```

- In **`render()`**

```
gluLookAt(...)           ← Viewing transformation
glTranslatef(...)  ┐
glRotatef(...)     ┘     ← Modeling transformations
draw_scene()
```

Geometric construction

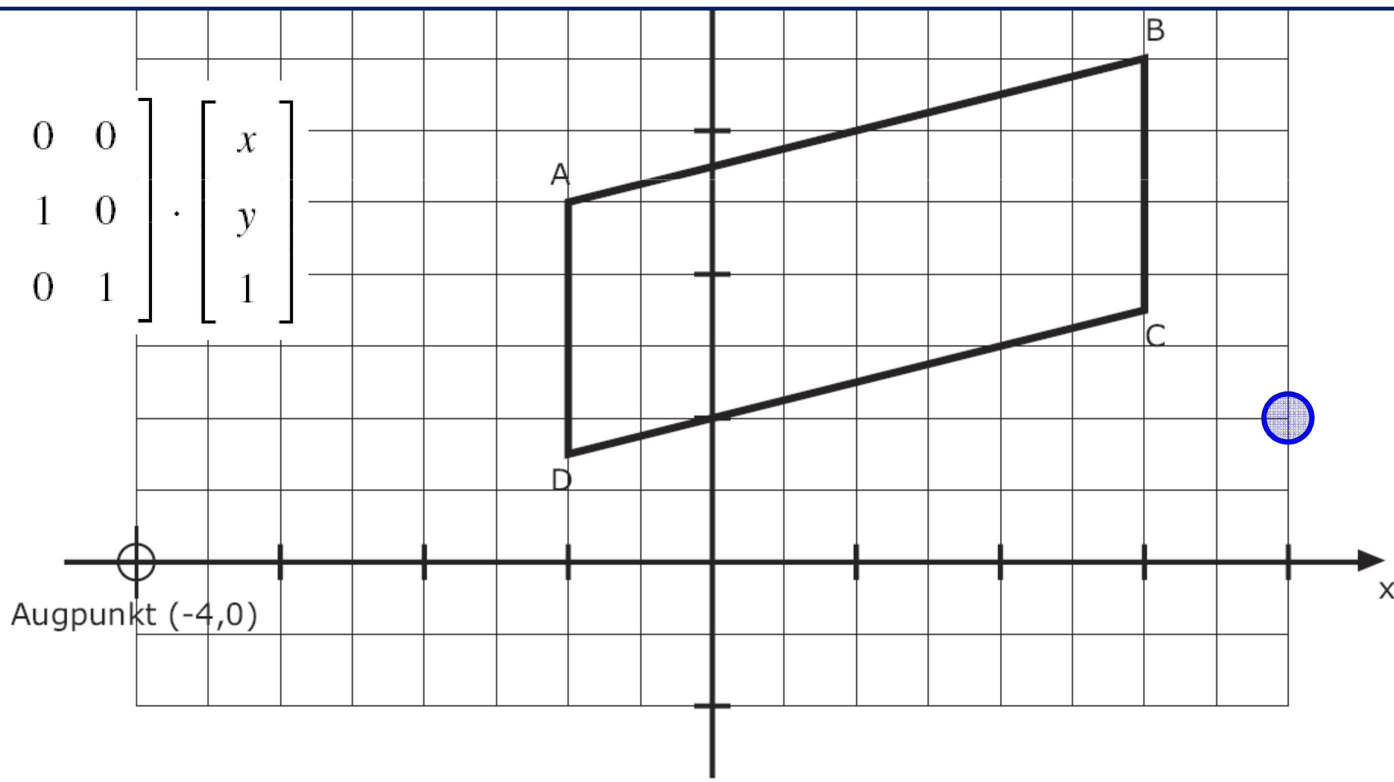


Geometric construction

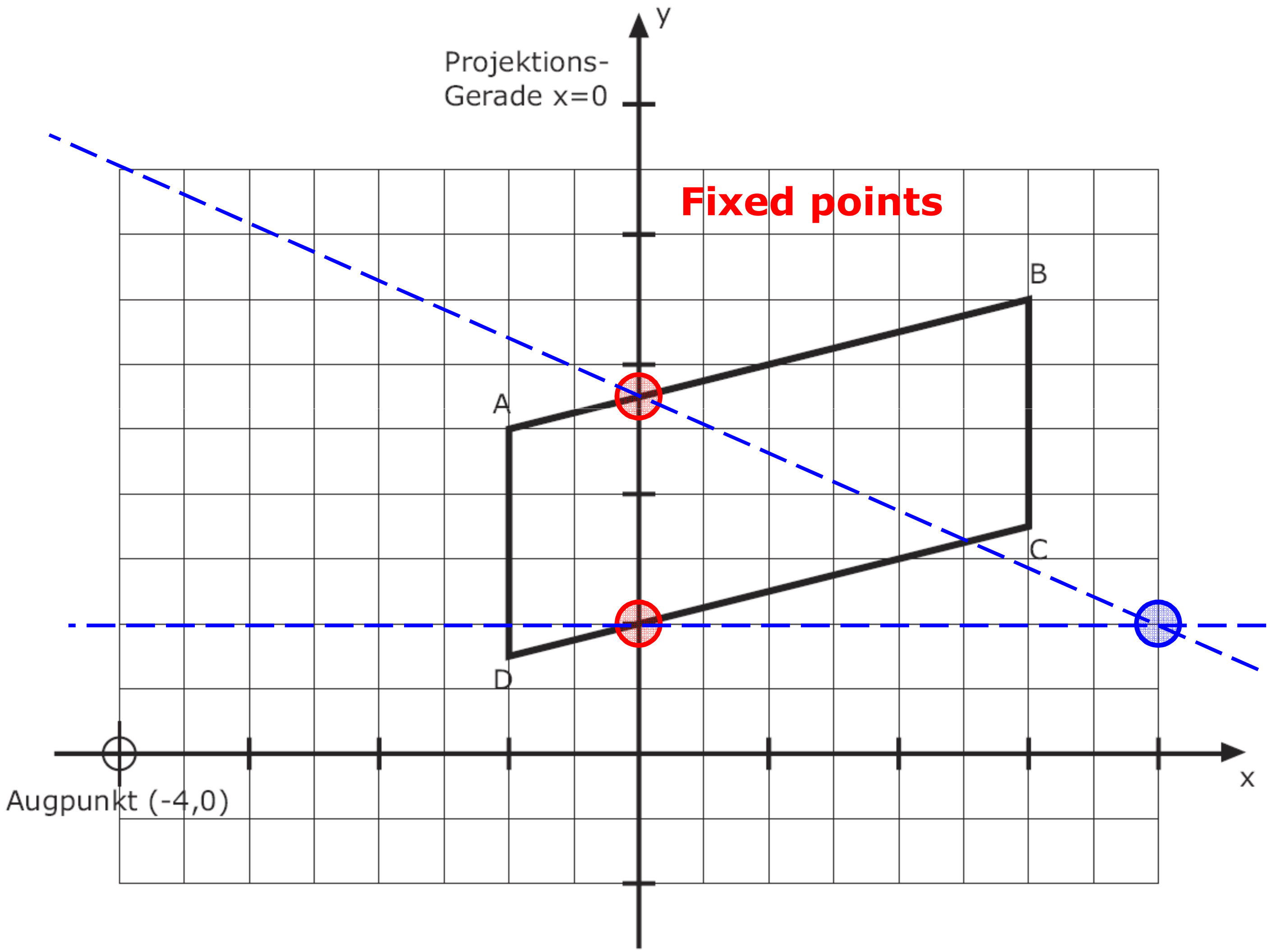
This transformation maps points [x,y,0] to [x,y,x/4]

The vanishing point of parallel lines AB and DC can be computed from the direction [4,1,0] as [4,1,1], which is the 2D point [4,1]

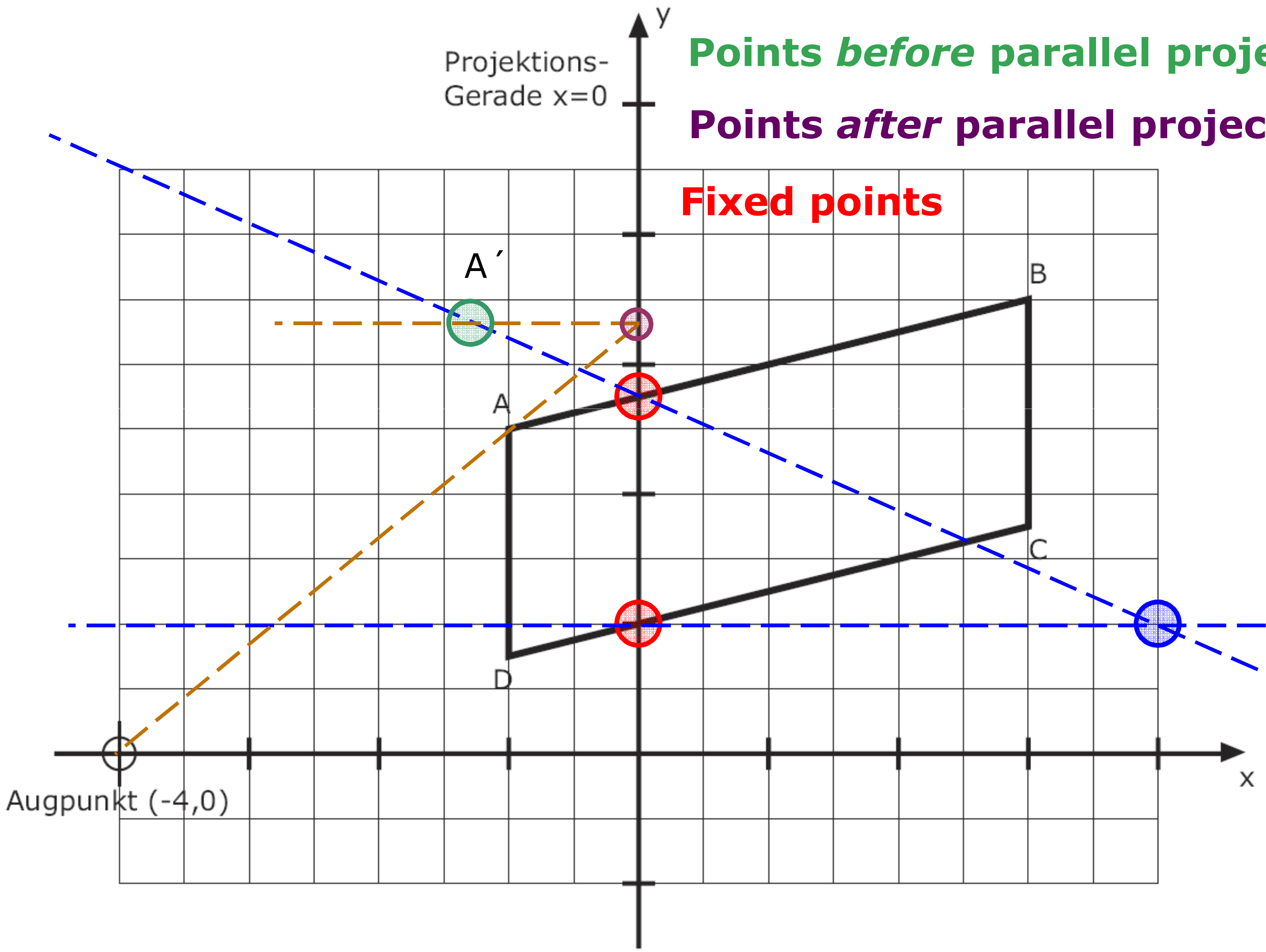
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{x_0} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



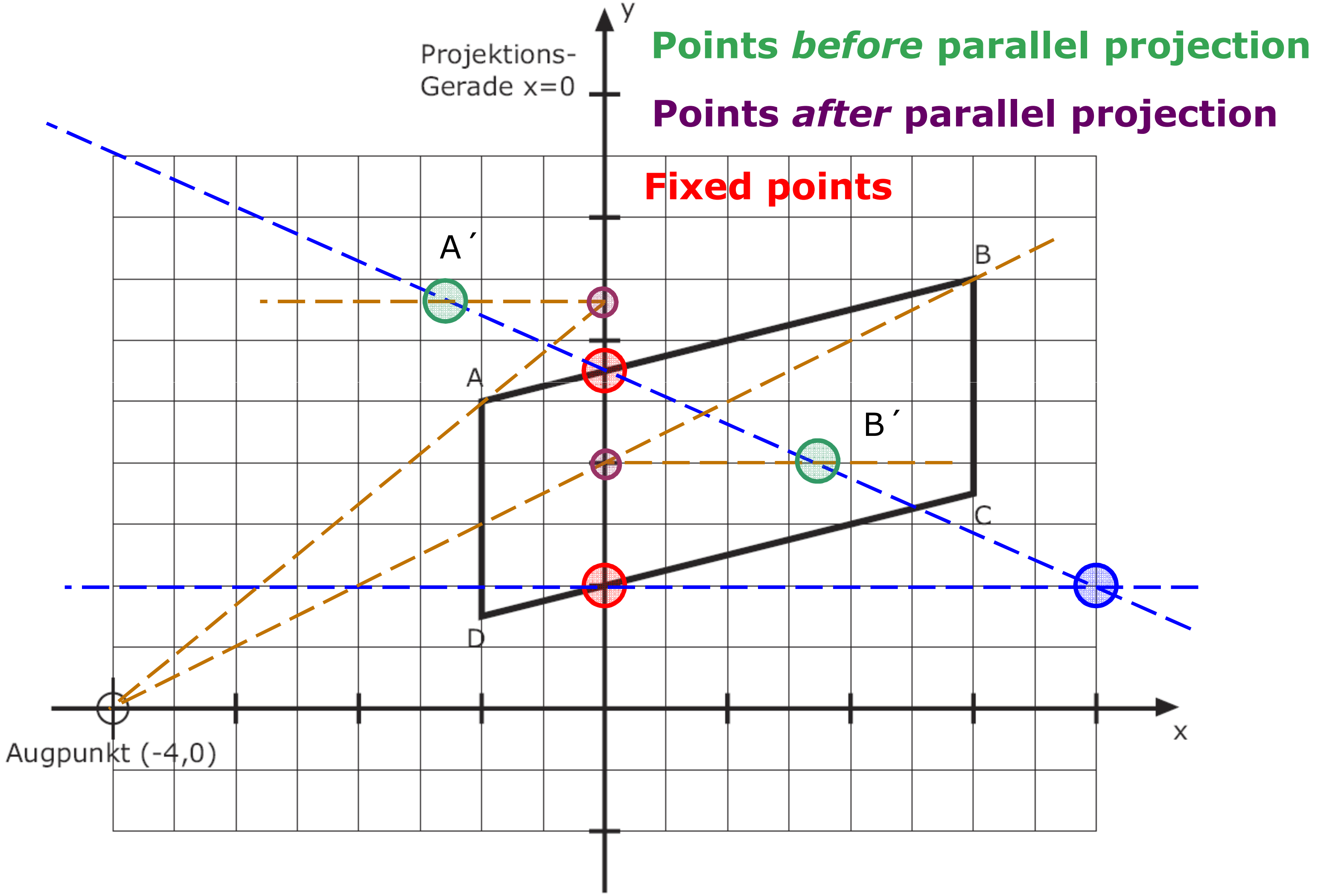
Geometric construction



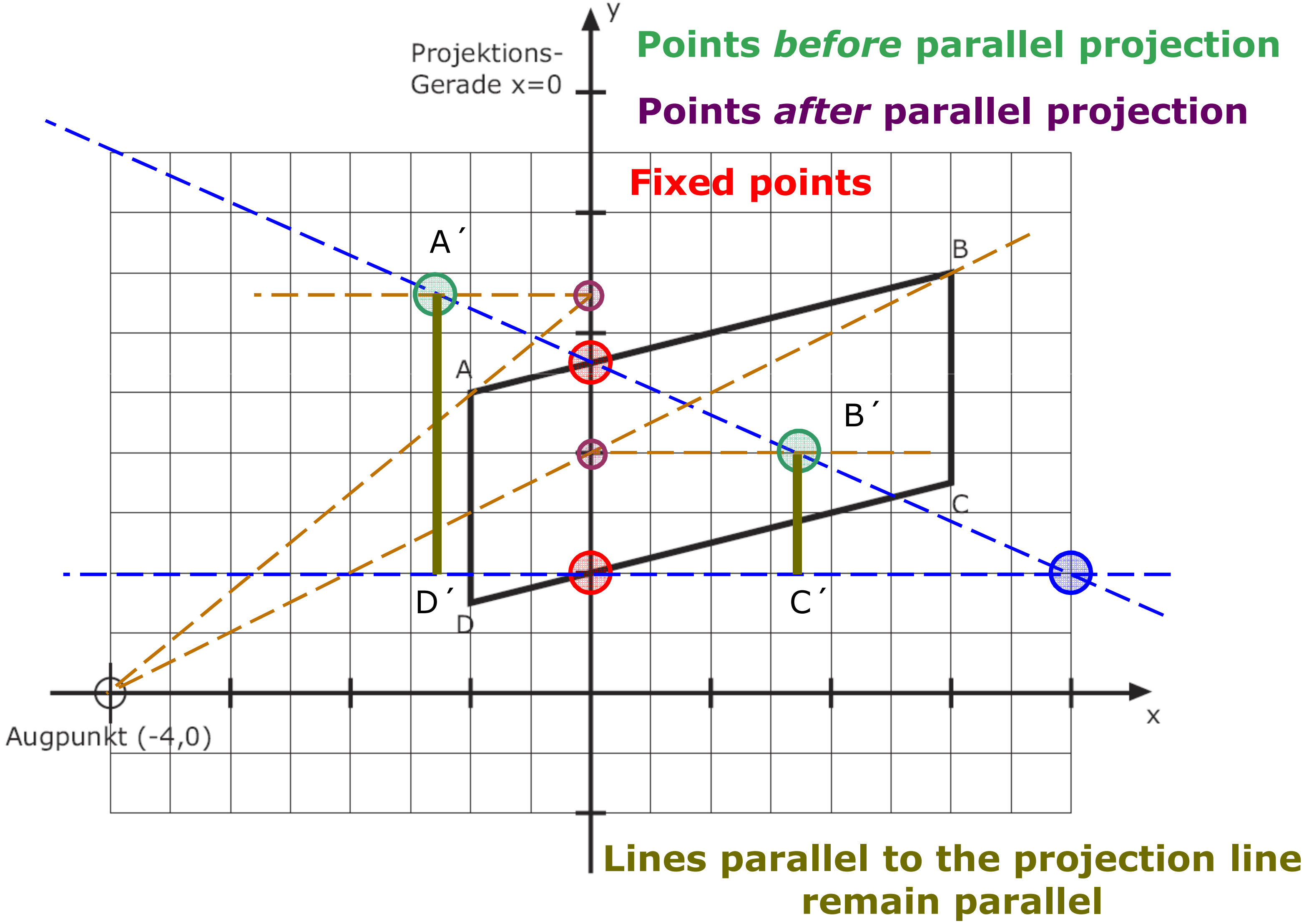
Geometric construction



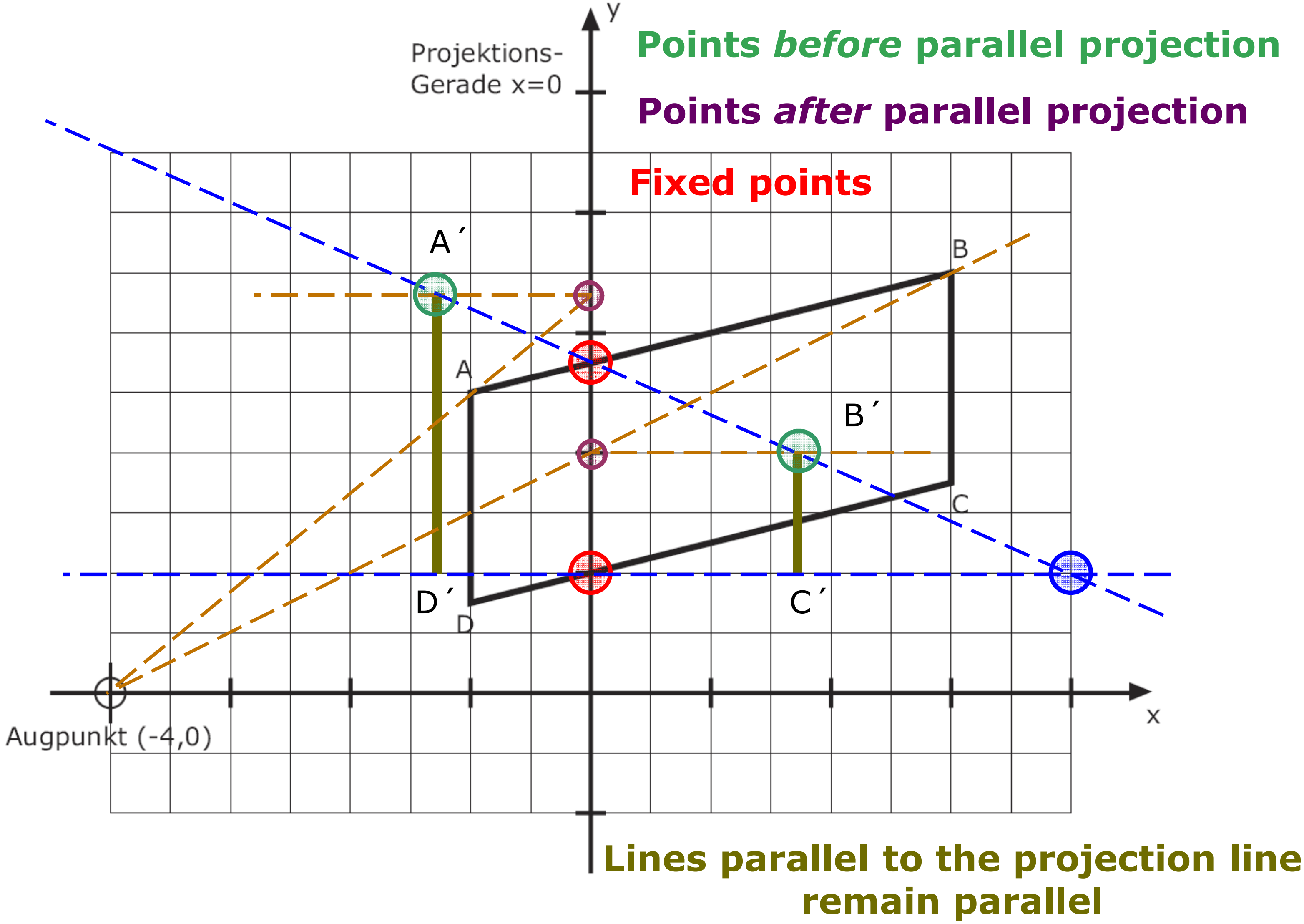
Geometric construction



Geometric construction



Geometric construction



Geometric construction