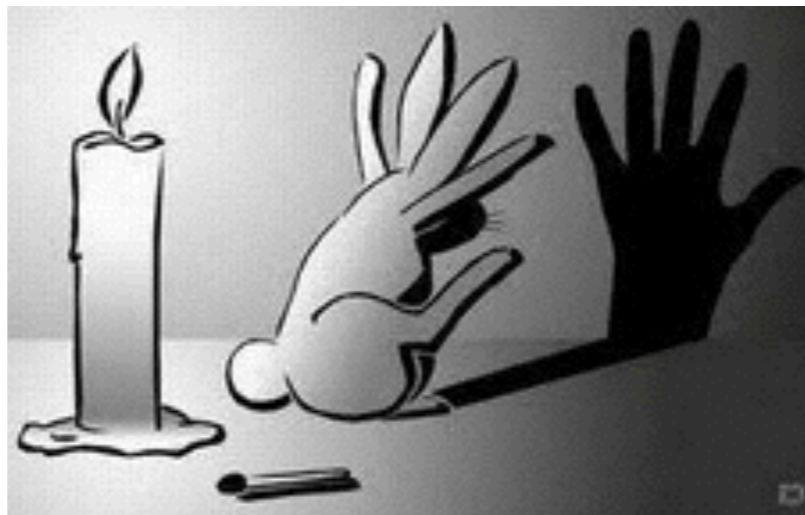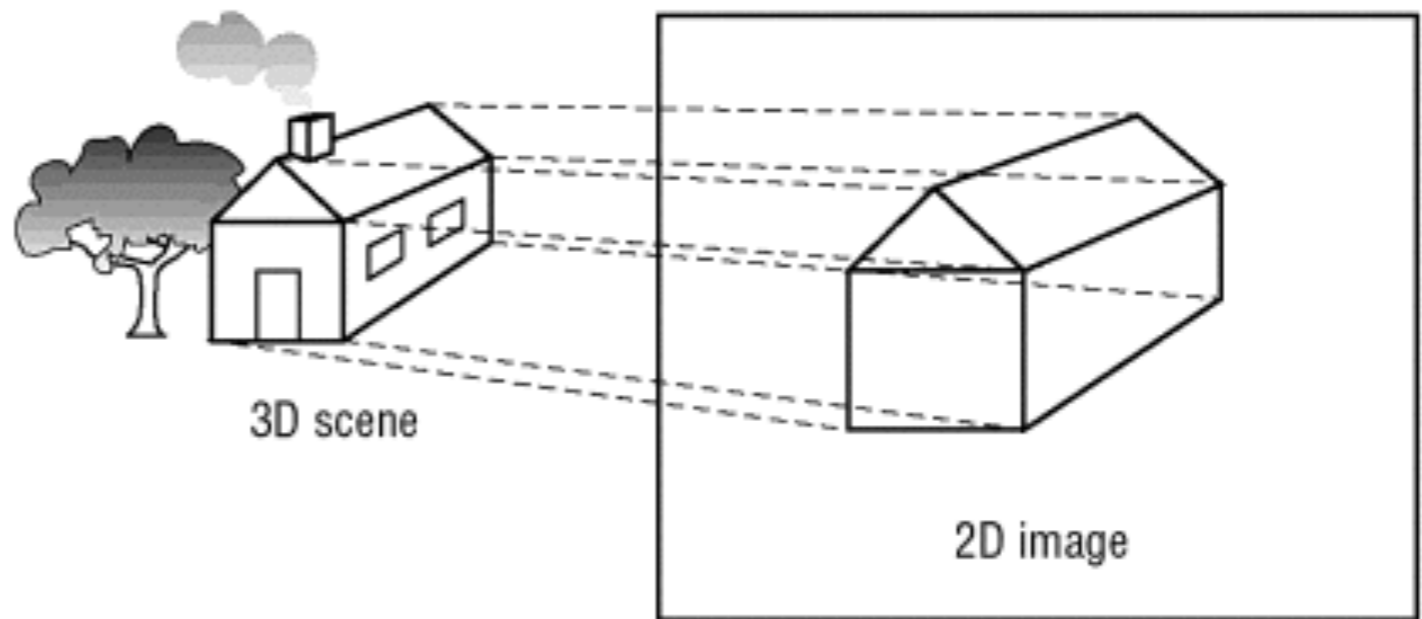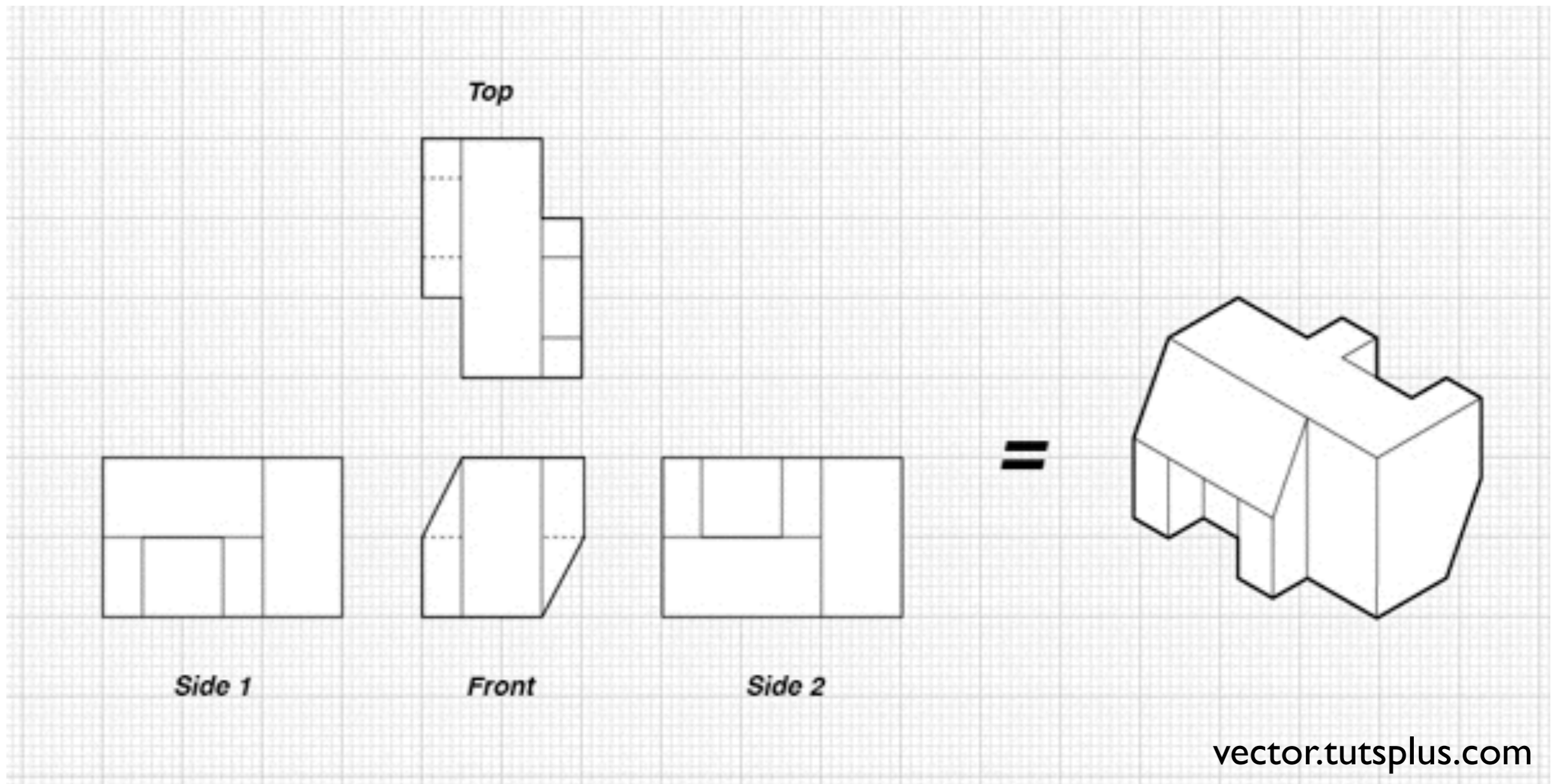# Viewing Transformations

Projection: map 3D scene to 2D image
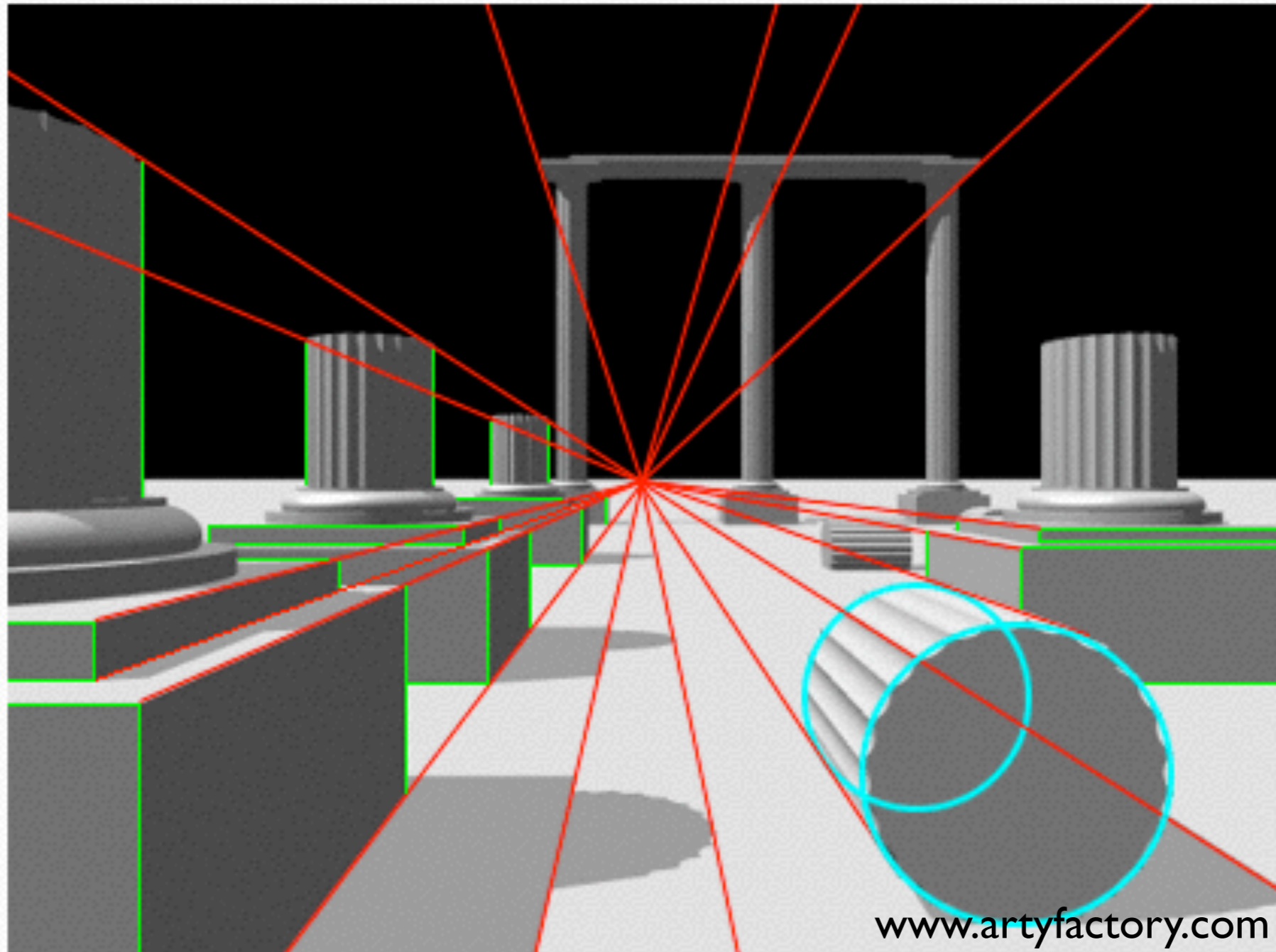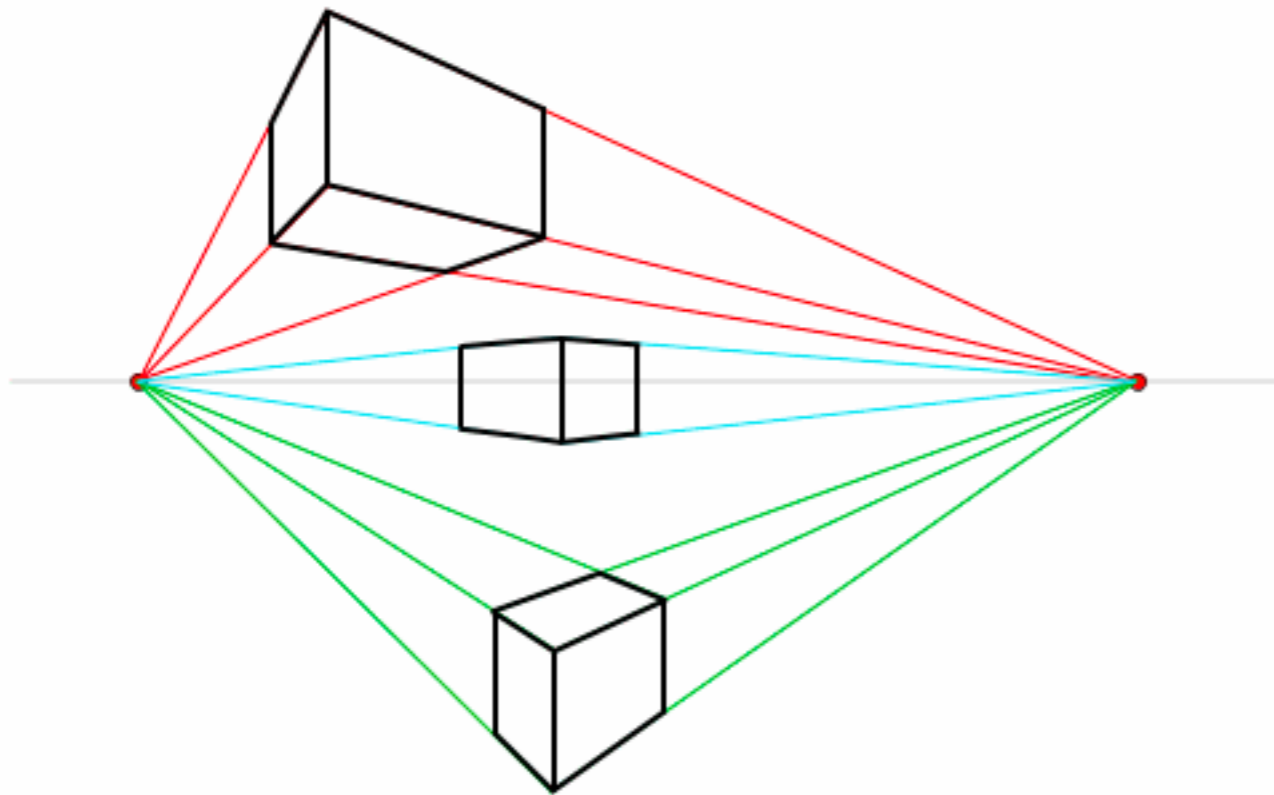

OpenGL Super Bible, 5th Ed.

Orthographic projection

Top

Side 1 Front Side 2
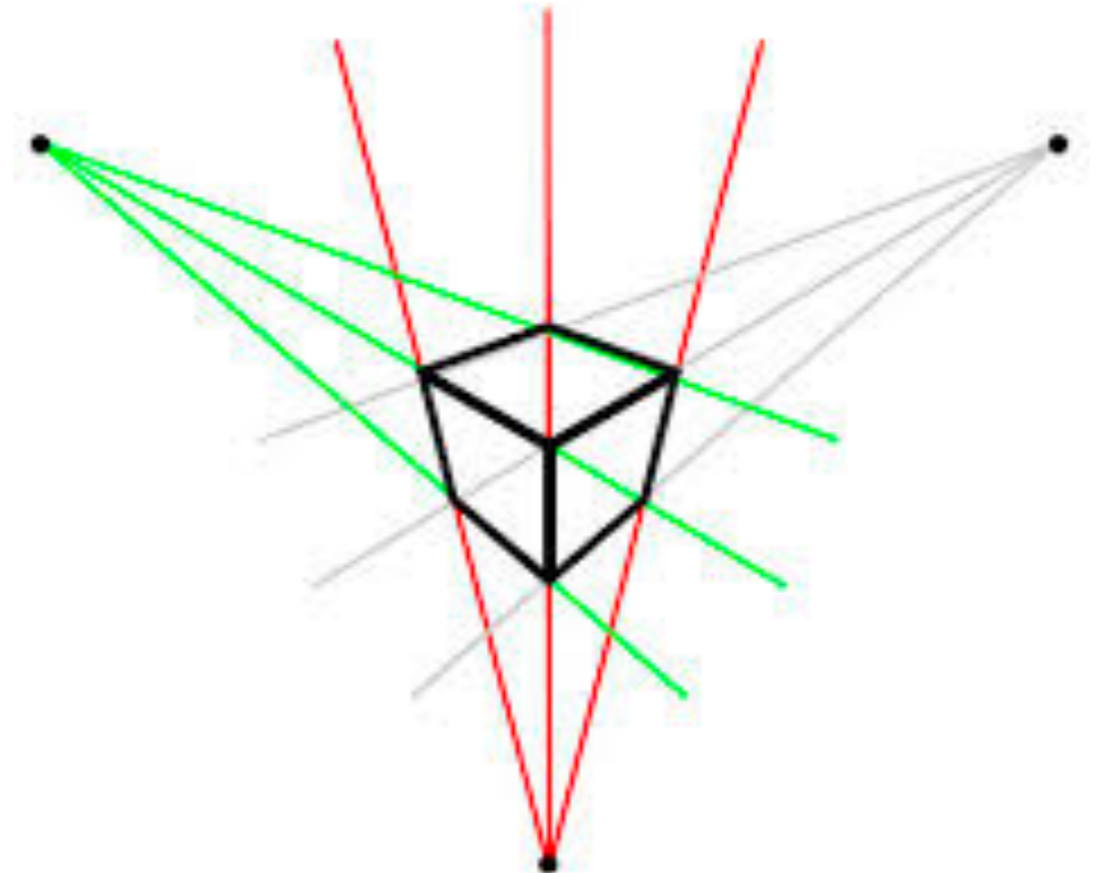
=

vector.tutsplus.com

Perspective projection

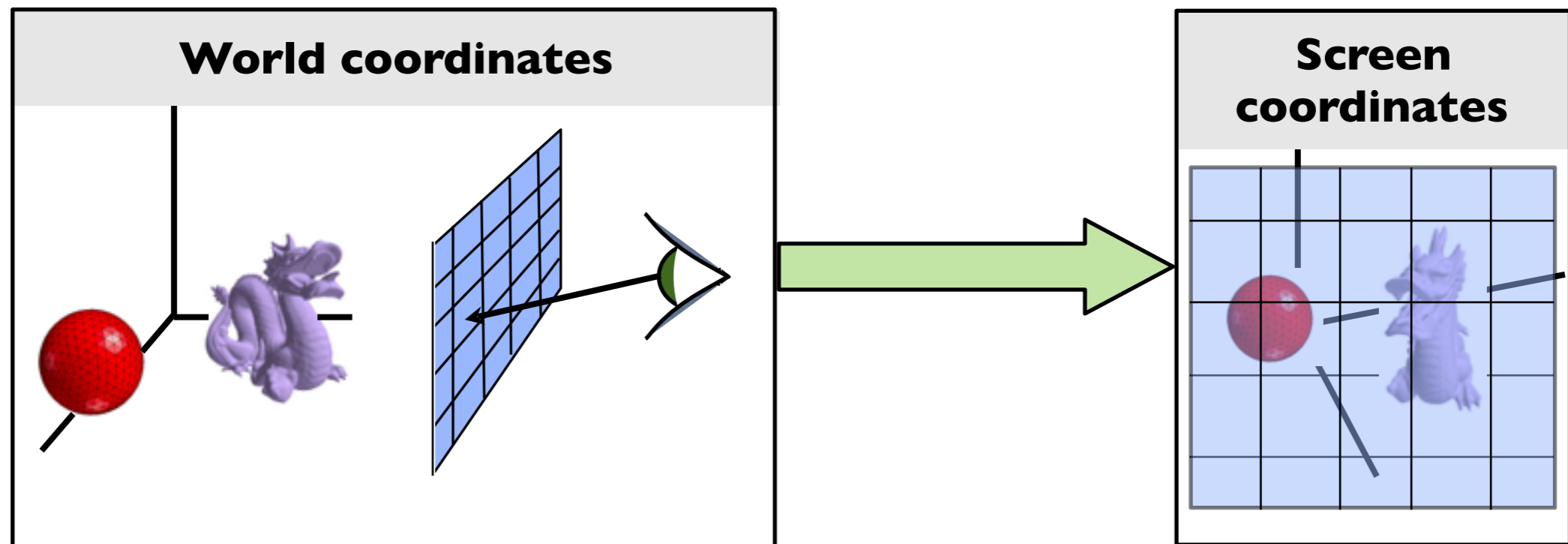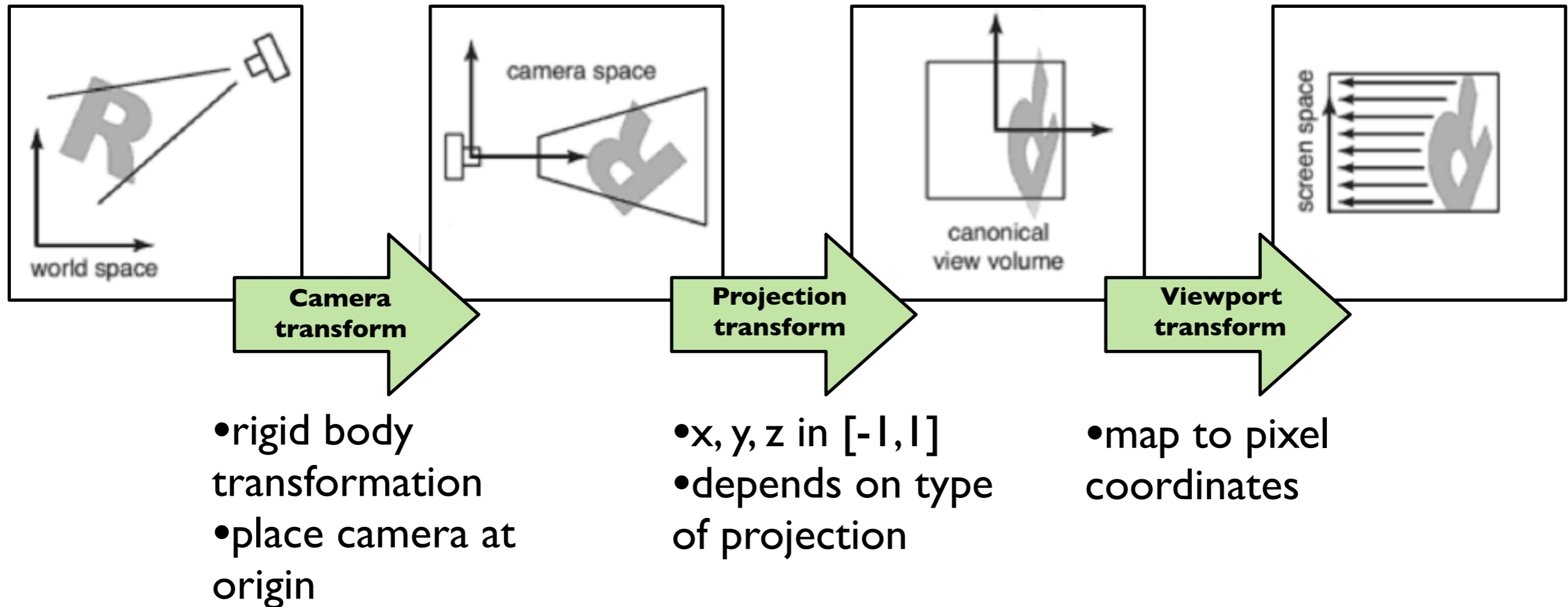www.artyfactory.com

two-point perspective　　three-point perspective

Viewing transformations

- Map objects from their 3D locations to their positions in a 2D view

Decomposition of viewing transforms

world space → **Camera transform** → camera space → **Projection transform** → canonical view volume → **Viewport transform** → screen space

Camera transform
- rigid body transformation
- place camera at origin

Projection transform
- x, y, z in [-1,1]
- depends on type of projection

Viewport transform
- map to pixel coordinates
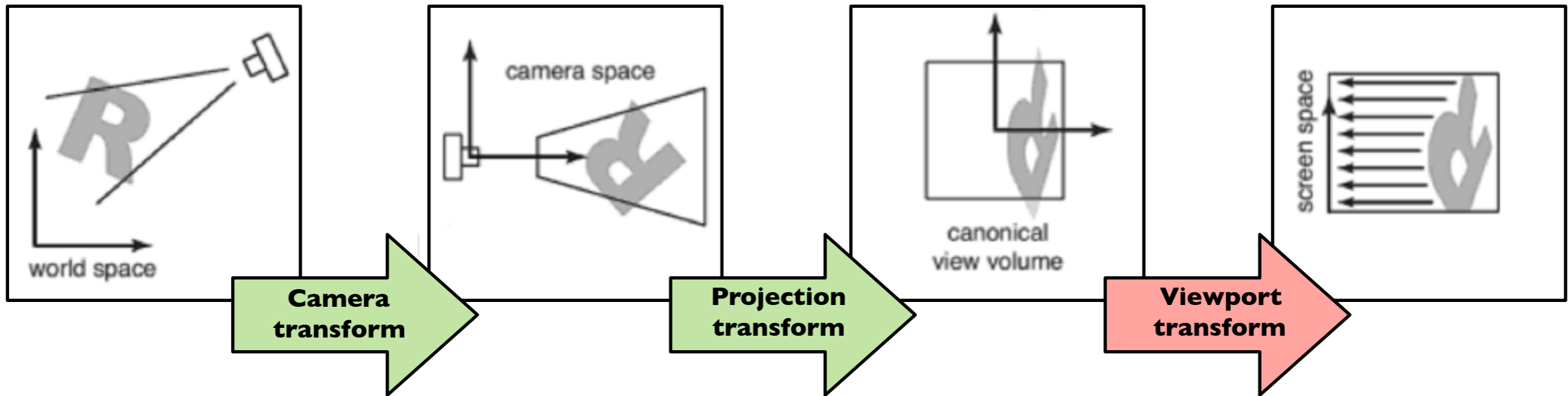
Viewing transforms depend on: camera position and orientation, type of projection, field of view, image resolution

Viewport transform

world space → **Camera transform** → camera space → **Projection transform** → canonical view volume → **Viewport transform** → screen space
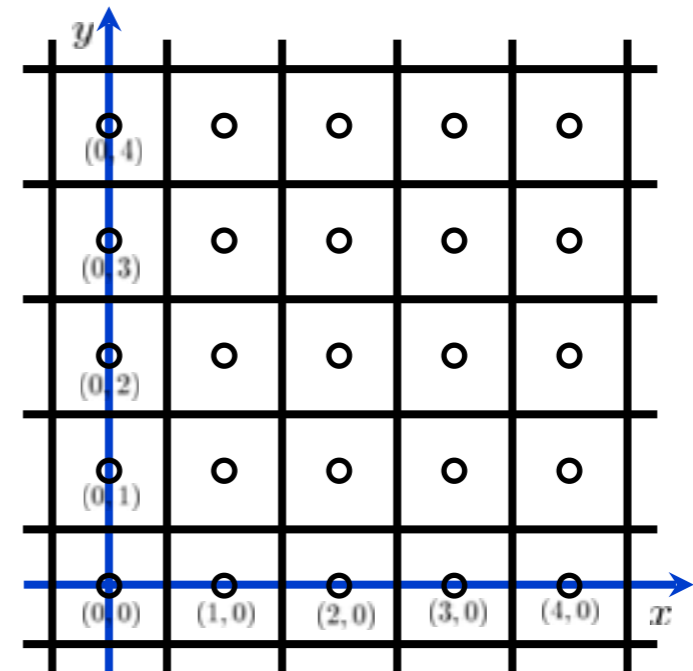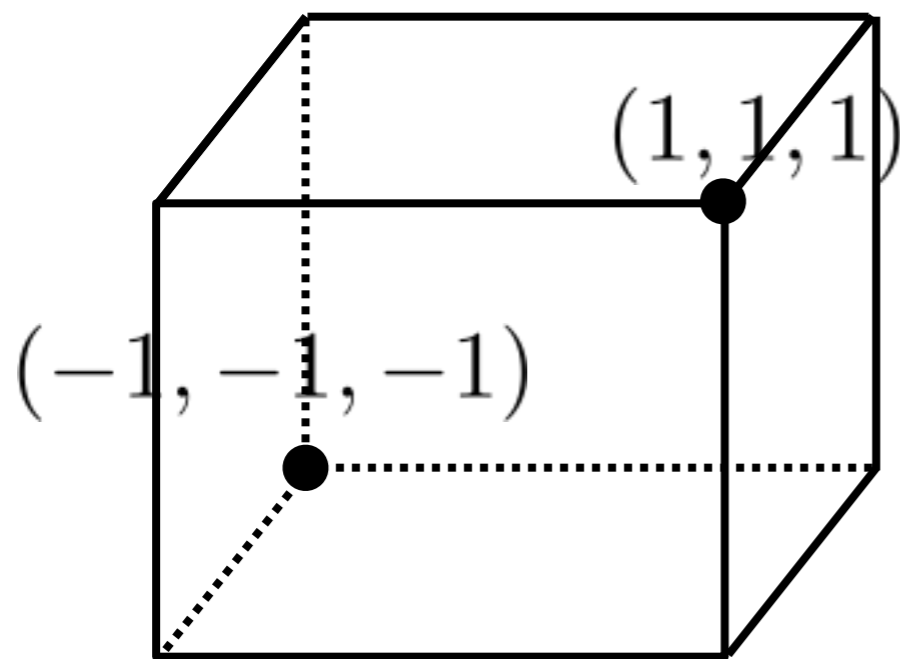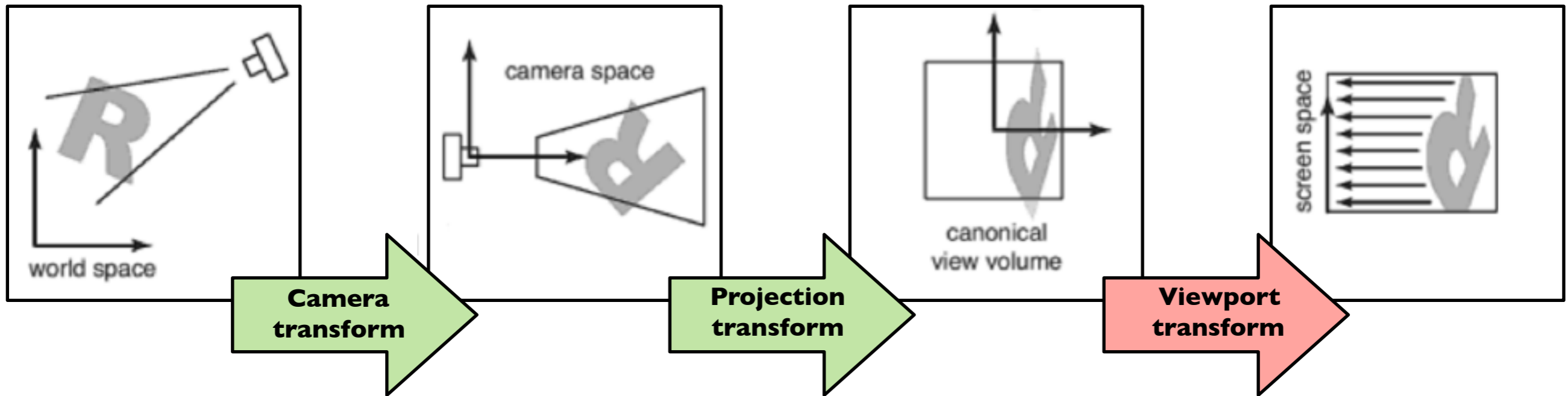
$$(x, y, z) \to (x', y', z')$$

$$(x, y, z) \in [-1, 1]^3$$

$$x' \in [-.5, n_x - .5]$$
$$y' \in [-.5, n_y - .5]$$

Viewport transform

camera space

canonical view volume

screen space

world space

Camera transform

Projection transform

Viewport transform

$(1, 1, 1)$

$(-1, -1, -1)$

M_{vp}

<whiteboard>

$(0, 4)$ $(0, 3)$ $(0, 2)$ $(0, 1)$ $(0, 0)$ $(1, 0)$ $(2, 0)$ $(3, 0)$ $(4, 0)$

y

x

n_y

n_x

Orthographic Projection Transform

Camera transform

Projection transform

Viewport transform

camera space

canonical view volume

screen space

world space

(r, t, n)

(l, b, f)

$(1, 1, 1)$

$(-1, -1, -1)$

M_{orth}

<whiteboard>