

Computer Graphics

Lecture 19

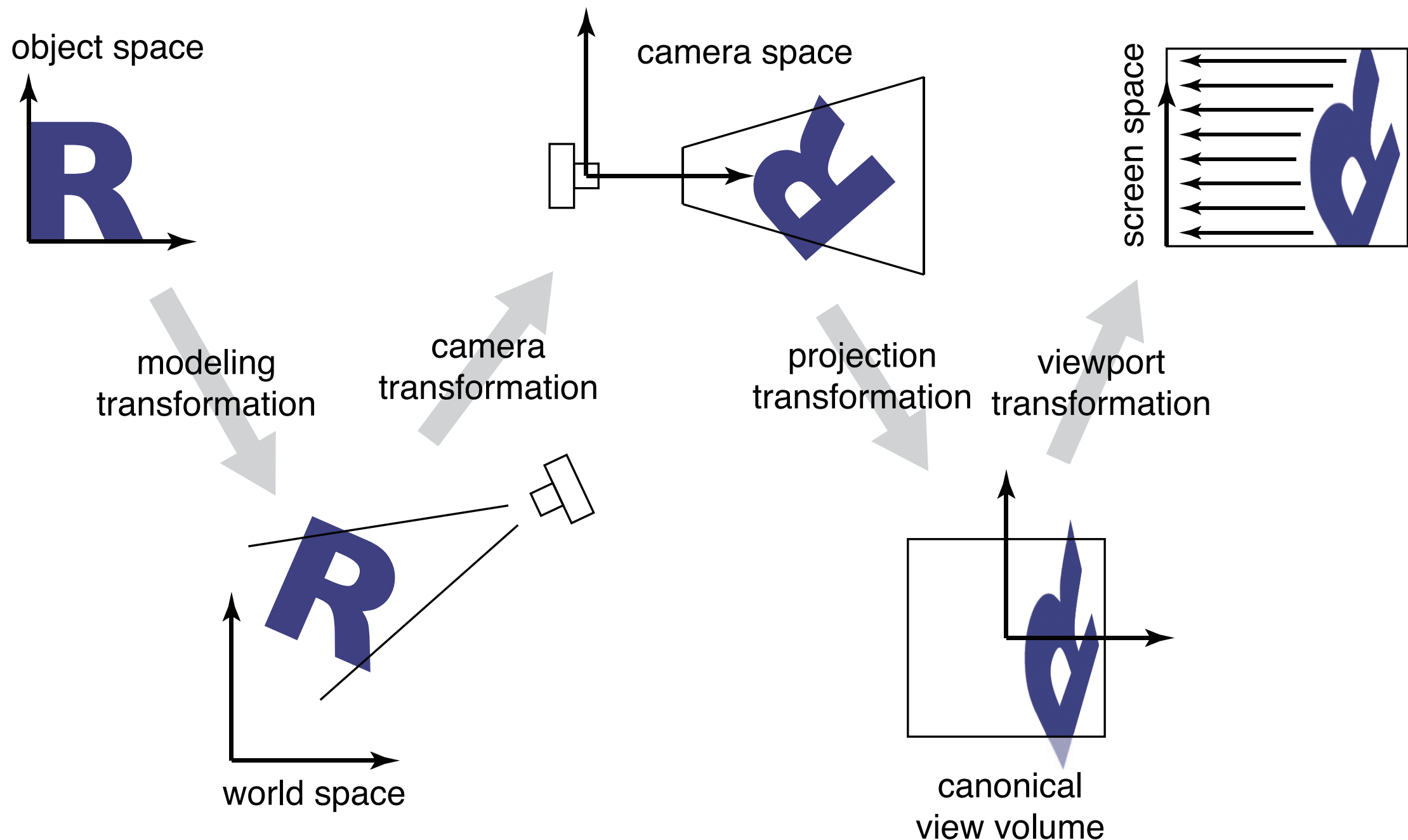
Viewing Transformations - 2

Announcements

- "Midterm" exam is out one week from Friday
- Takehome; 20% of your grade, inspired by HW
- Upshot: if you got HW problems wrong, make sure you know how to get them right. You have nearly 2 weeks, so now's the time to start reviewing.
- HW1 out today or tomorrow; aiming for quick grading turnaround on HW2 and HW3

Viewing Transformations: Overview

A standard sequence of transforms to go from **object (model) space** to **screen (image) space**



A Wireframe Rendering Algorithm

Form matrices $M_{vp}, M_{proj}, M_{cam}, M_{model}$

$M \leftarrow M_{vp}M_{proj}M_{cam}M_{model}$

for each line segment $\mathbf{a}_i, \mathbf{b}_i$:

$\mathbf{p} \leftarrow M\mathbf{a}_i$

$\mathbf{q} \leftarrow M\mathbf{b}_i$

`draw_line(p, q)`

Viewing Transformations: Demo

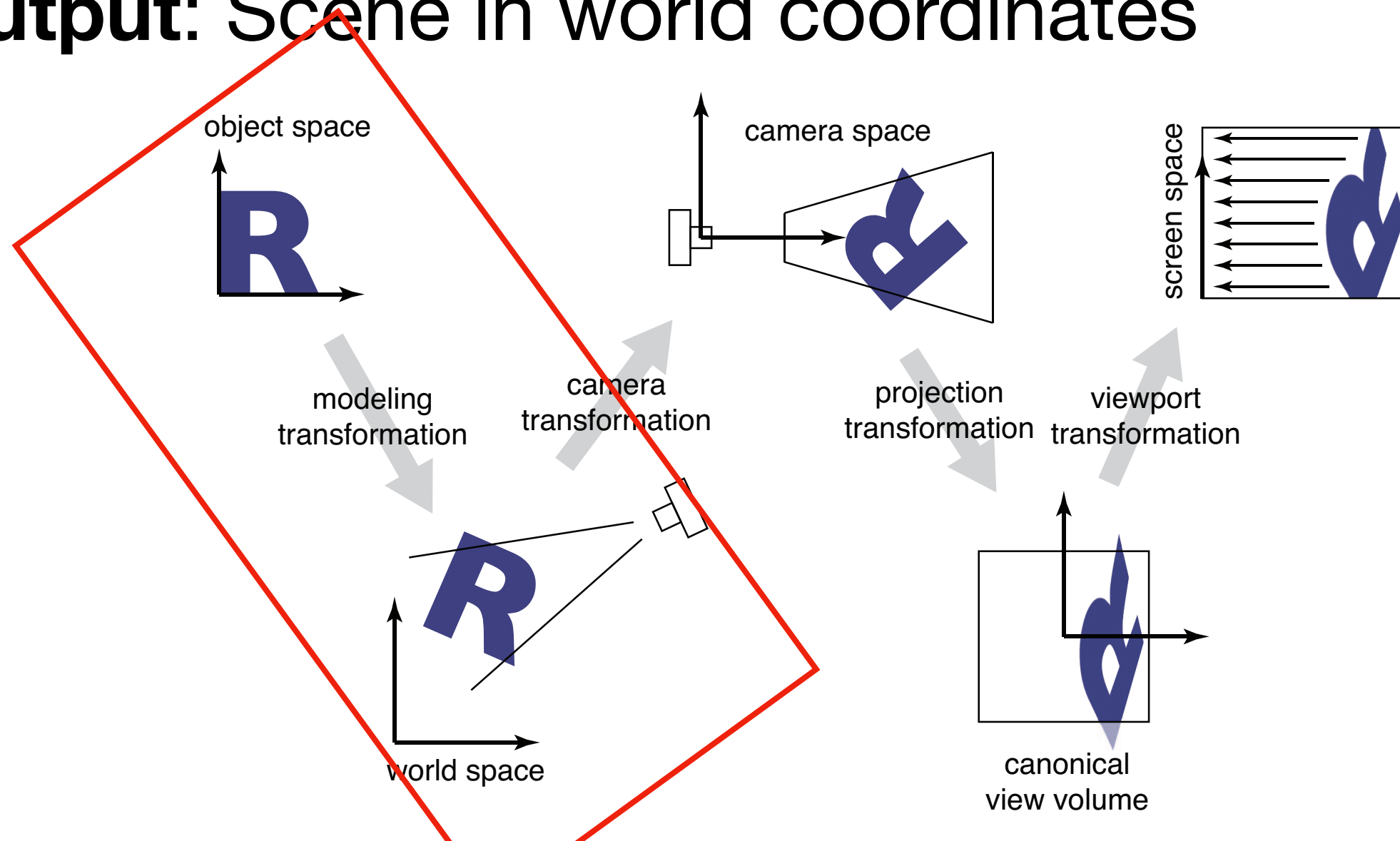
- https://www.cs.cornell.edu/courses/cs4620/2020fa/demos_cs4620/view_explore/view_explore.html

Model Matrix

Input: Scene in model coordinates

Parameters: Pose, scale, etc of model in scene

Output: Scene in world coordinates

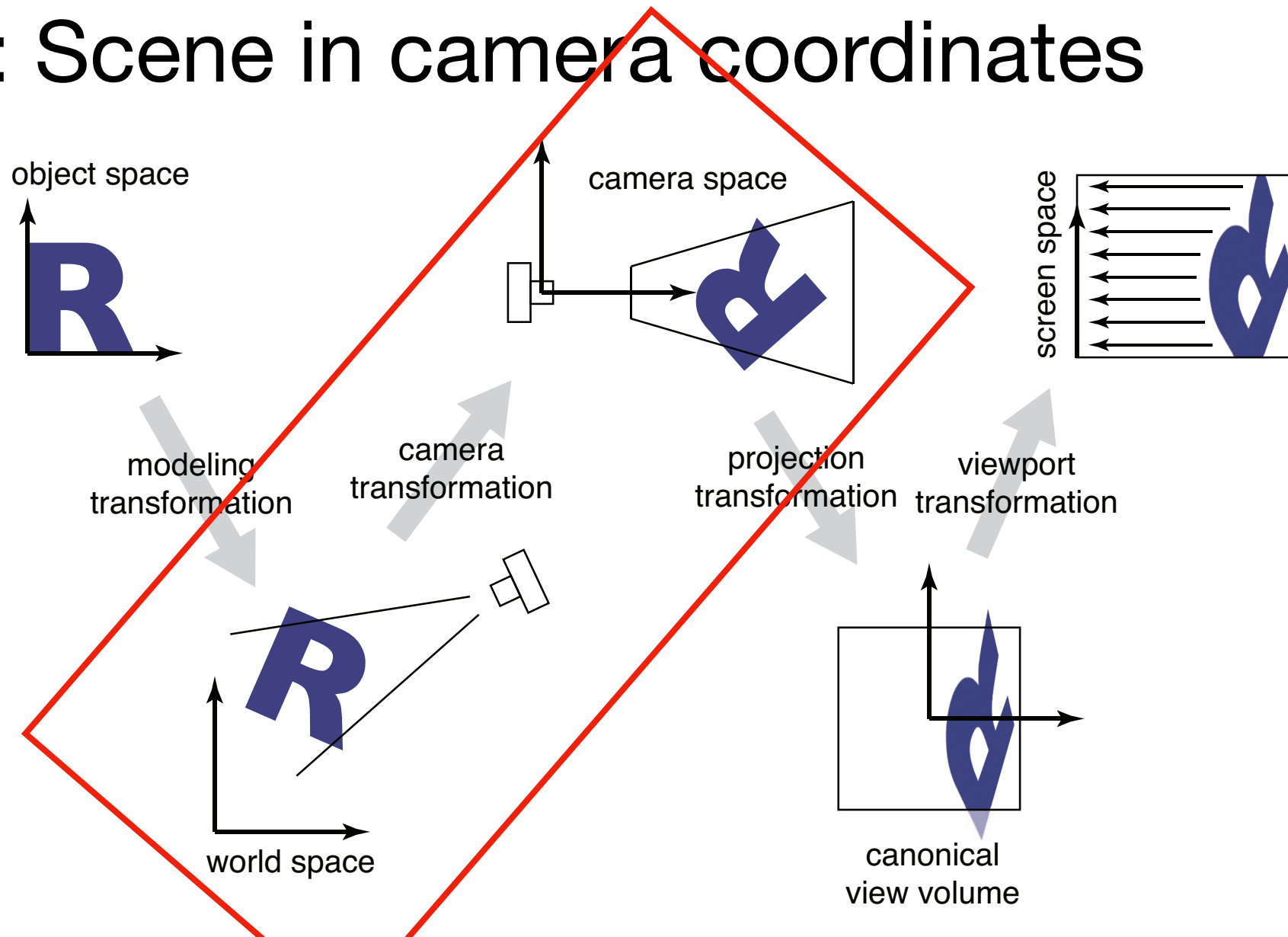


Camera Matrix

Input: Scene in world coordinates

Parameters: Camera frame (\mathbf{u} , \mathbf{v} , \mathbf{w} , \mathbf{e})

Output: Scene in camera coordinates

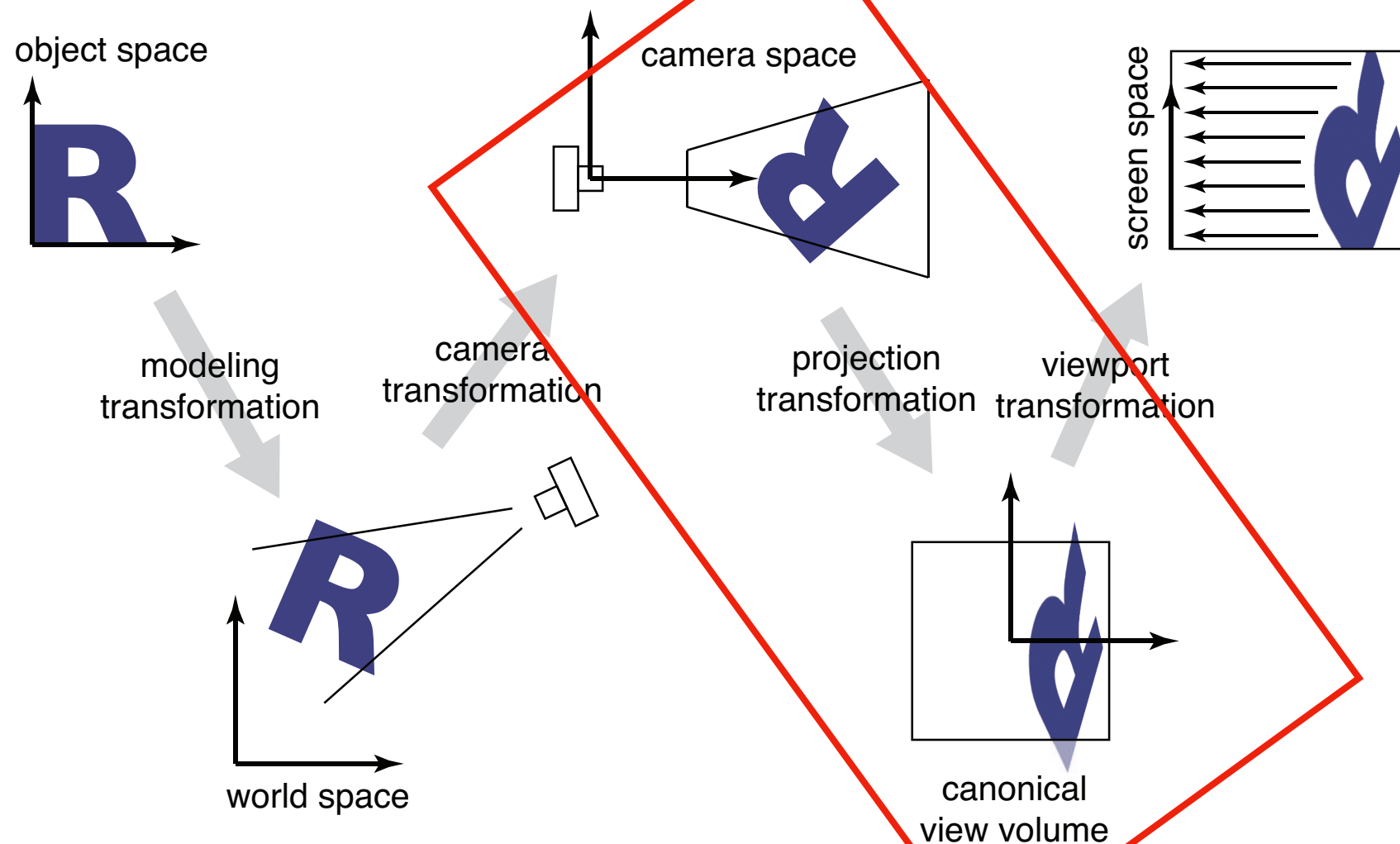


Projection Matrix - Orthographic

Input: Scene in (canonically-posed) camera coordinates

Parameters: Orthographic viewport dimensions

Output: Normalized device coordinates

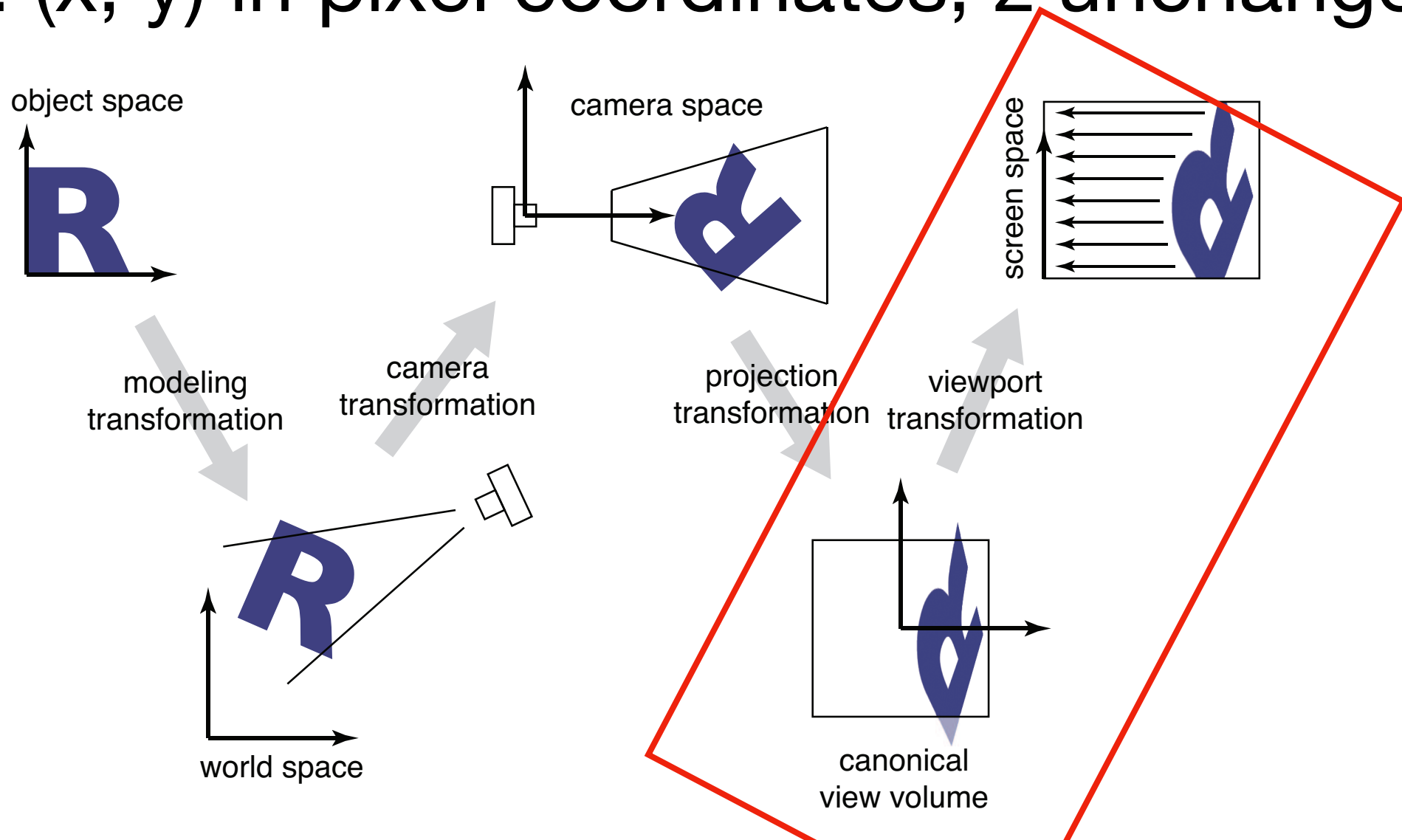


Viewport Matrix

Input: Scene in the canonical view volume

Parameters: W, H (image dimensions in pixels)

Output: (x, y) in pixel coordinates; z unchanged



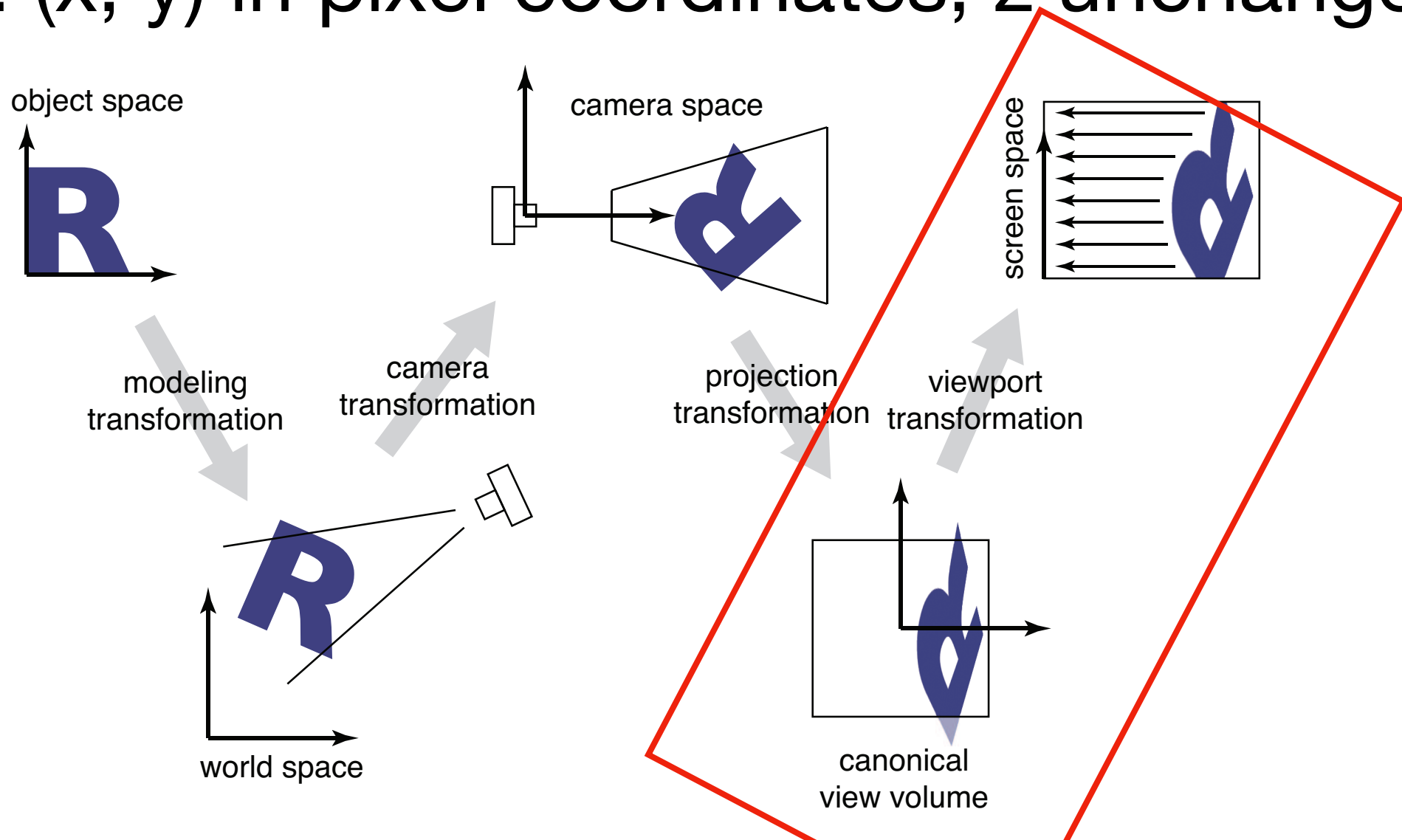
Let's build it

Viewport Matrix

Input: Scene in the canonical view volume

Parameters: W, H (image dimensions in pixels)

Output: (x, y) in pixel coordinates; z unchanged

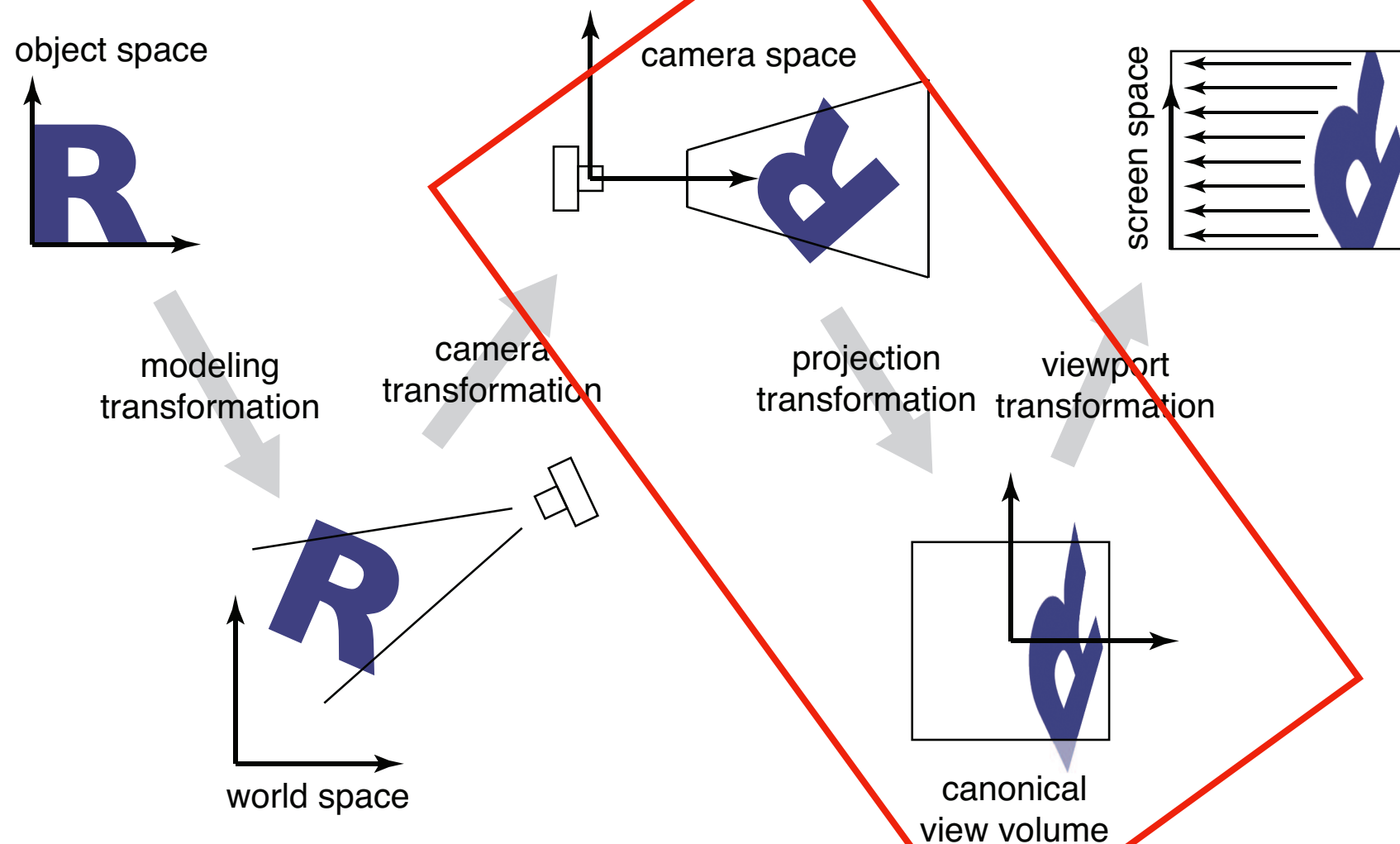


Projection Matrix - Orthographic

Input: Scene in (canonically-posed) camera coordinates

Parameters: Orthographic viewport dimensions

Output: Normalized device coordinates

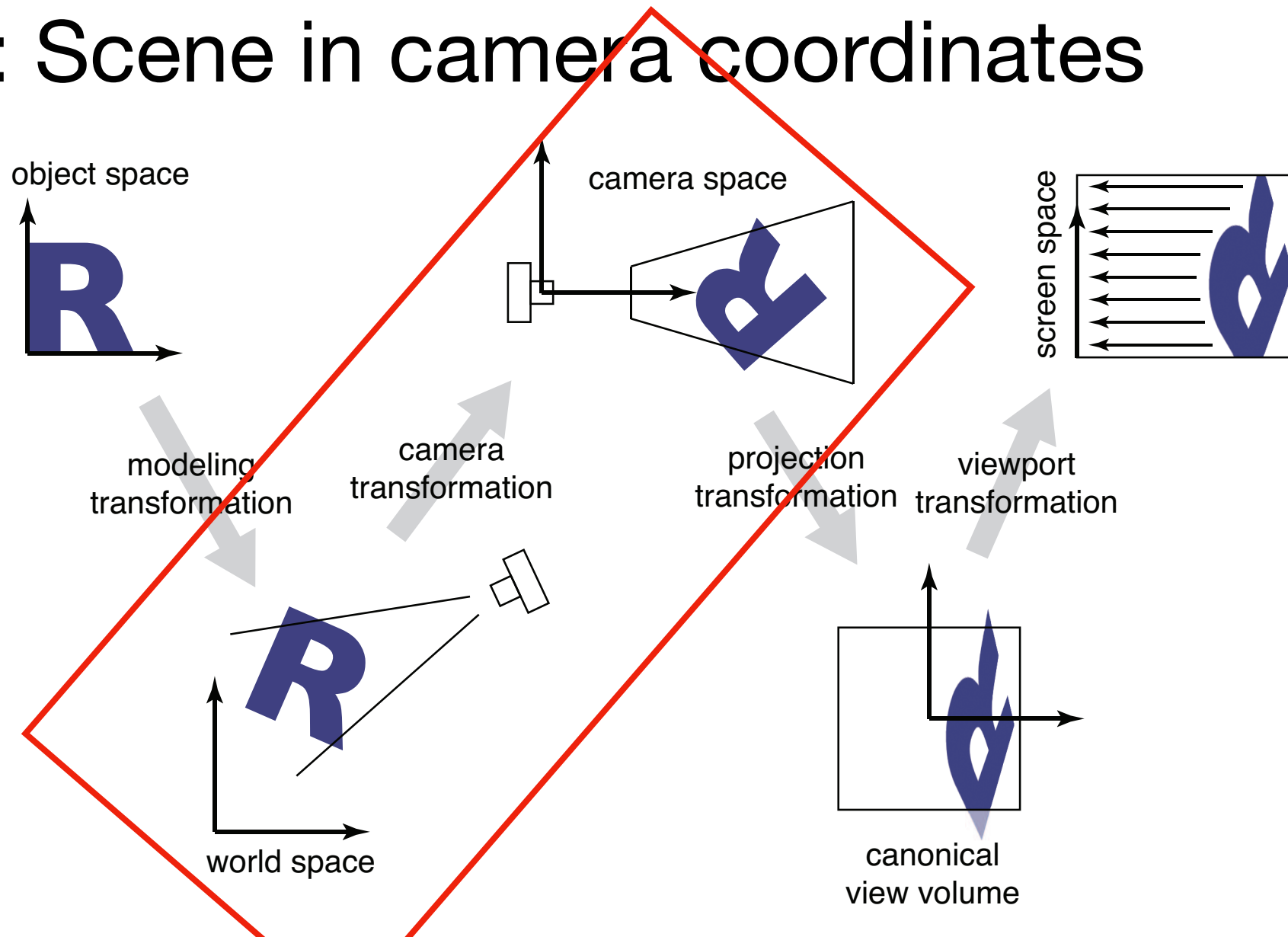


Camera Matrix

Input: Scene in world coordinates

Parameters: Camera frame (\mathbf{u} , \mathbf{v} , \mathbf{w} , \mathbf{e})

Output: Scene in camera coordinates

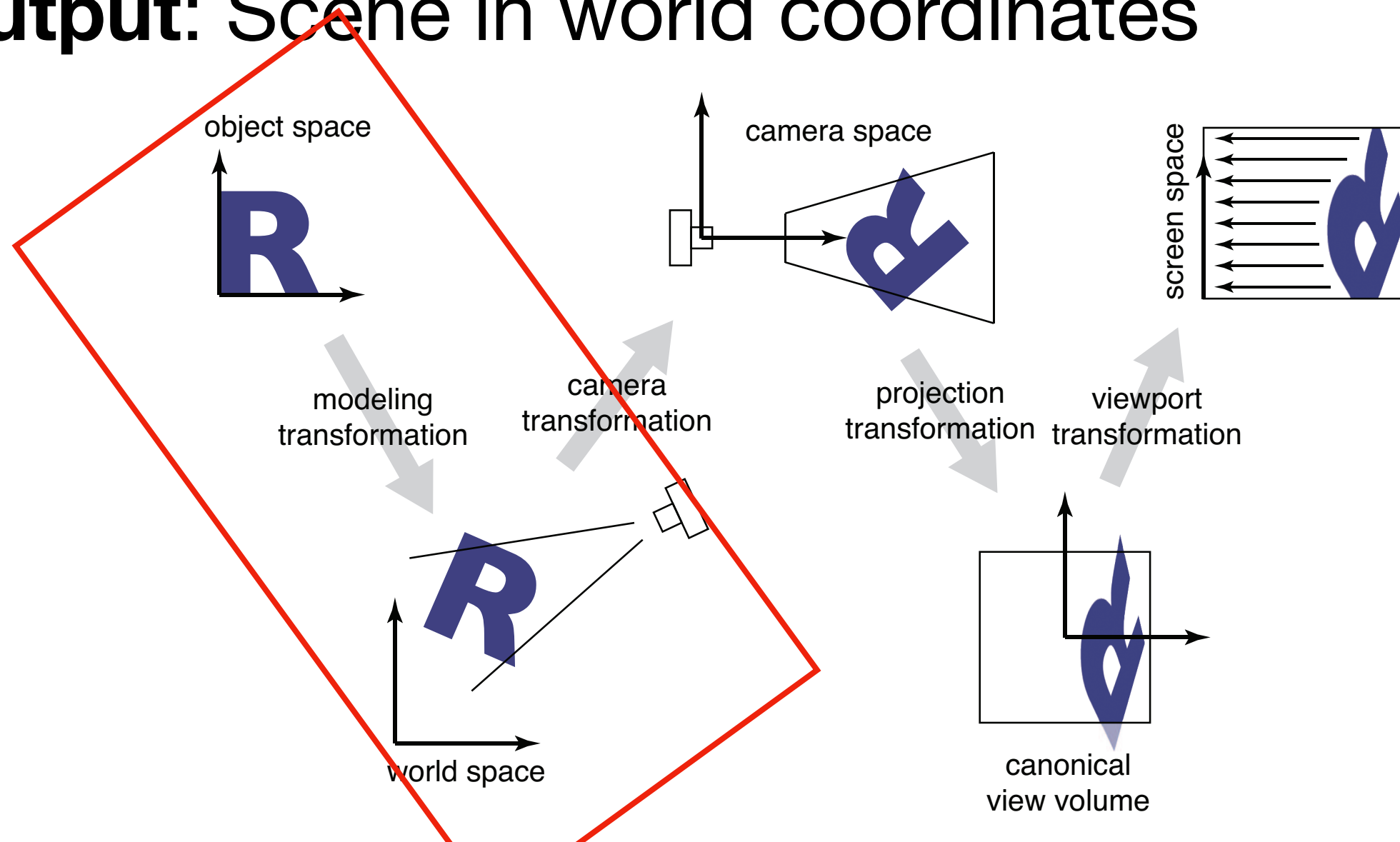


Model Matrix

Input: Scene in model coordinates

Parameters: Pose, scale, etc of model in scene

Output: Scene in world coordinates



Bonus Problem

Build a **model transformation** that takes our canonical $2 \times 2 \times 2$ cube, scales it to 40×40 and centers it at $x=0$, $y=1$, $z=-4$, rotated 30 degrees around the **y** axis.