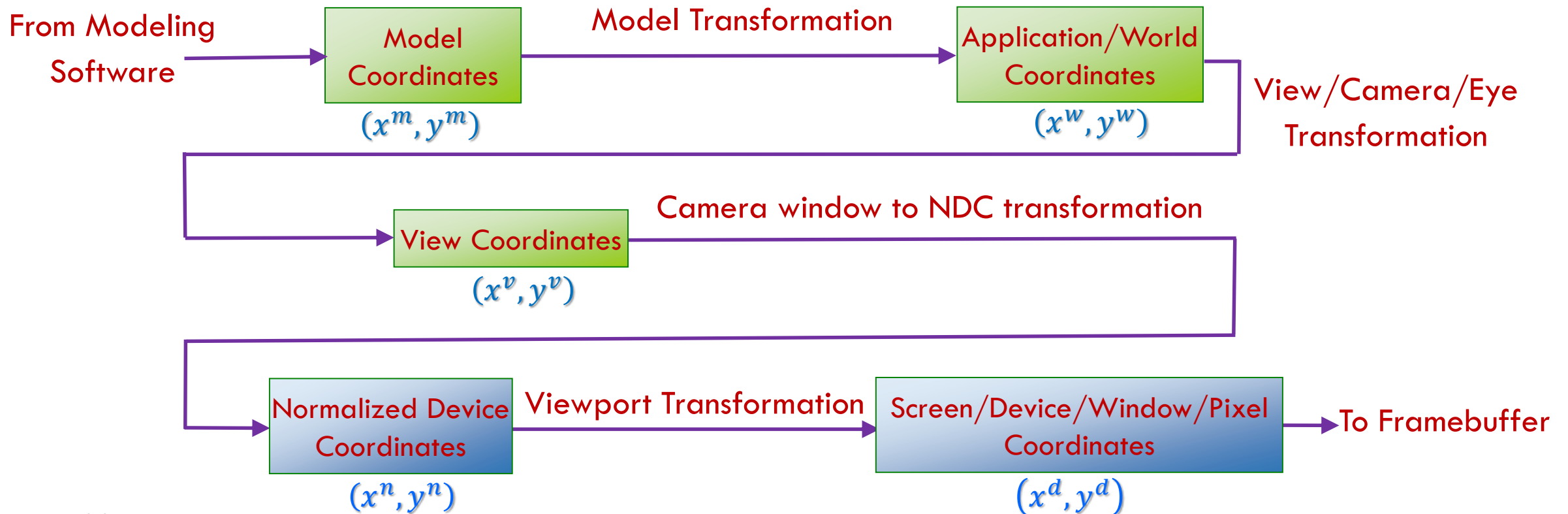# Introduction to Computer Graphics
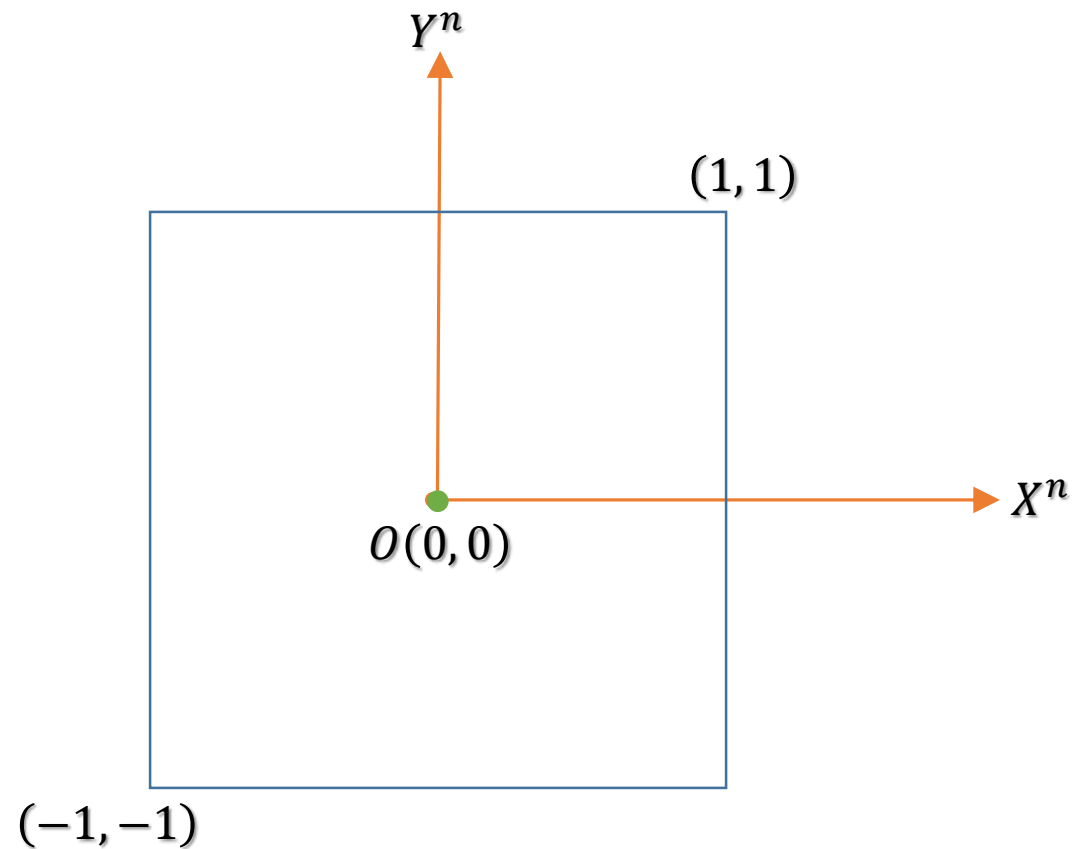
Viewport Transformation

Prasanna Ghali

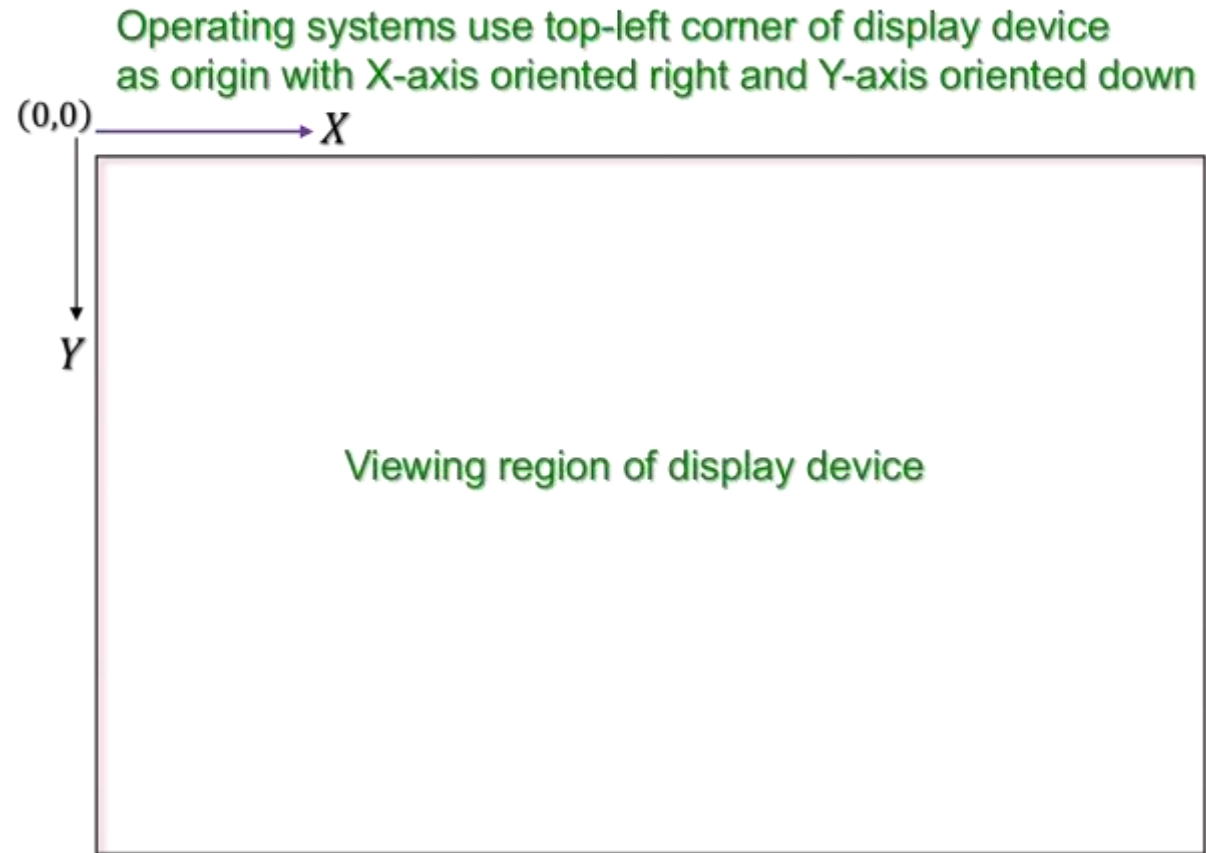# 2D Rendering Pipe: Coordinate Systems

- Rendering pipeline for 2D applications: different coordinate systems that 2D objects [and their primitives] must transition thro' in rendering pipeline

From Modeling Software → Model Coordinates (x^m, y^m) → Model Transformation → Application/World Coordinates (x^w, y^w) → View/Camera/Eye Transformation → View Coordinates (x^v, y^v) → Camera window to NDC transformation → Normalized Device Coordinates (x^n, y^n) → Viewport Transformation → Screen/Device/Window/Pixel Coordinates (x^d, y^d) → To Framebuffer

Normalized Device Coordinates

Y^n

$(1, 1)$

X^n

$O(0, 0)$

$(-1, -1)$

Display Coordinate System

Operating systems use top-left corner of display device
as origin with X-axis oriented right and Y-axis oriented down

(0,0) ⟶ X

Y

Viewing region of display device

Window in Display Device

Operating systems use top-left corner of display device
as origin with X-axis oriented right and Y-axis oriented down

(0,0) ──────────────→ X

(0,0) ──────────────→ X

Y

Y

Window's position in display device
is defined by its top-left corner

Viewing region of window

Window's height

Window's width
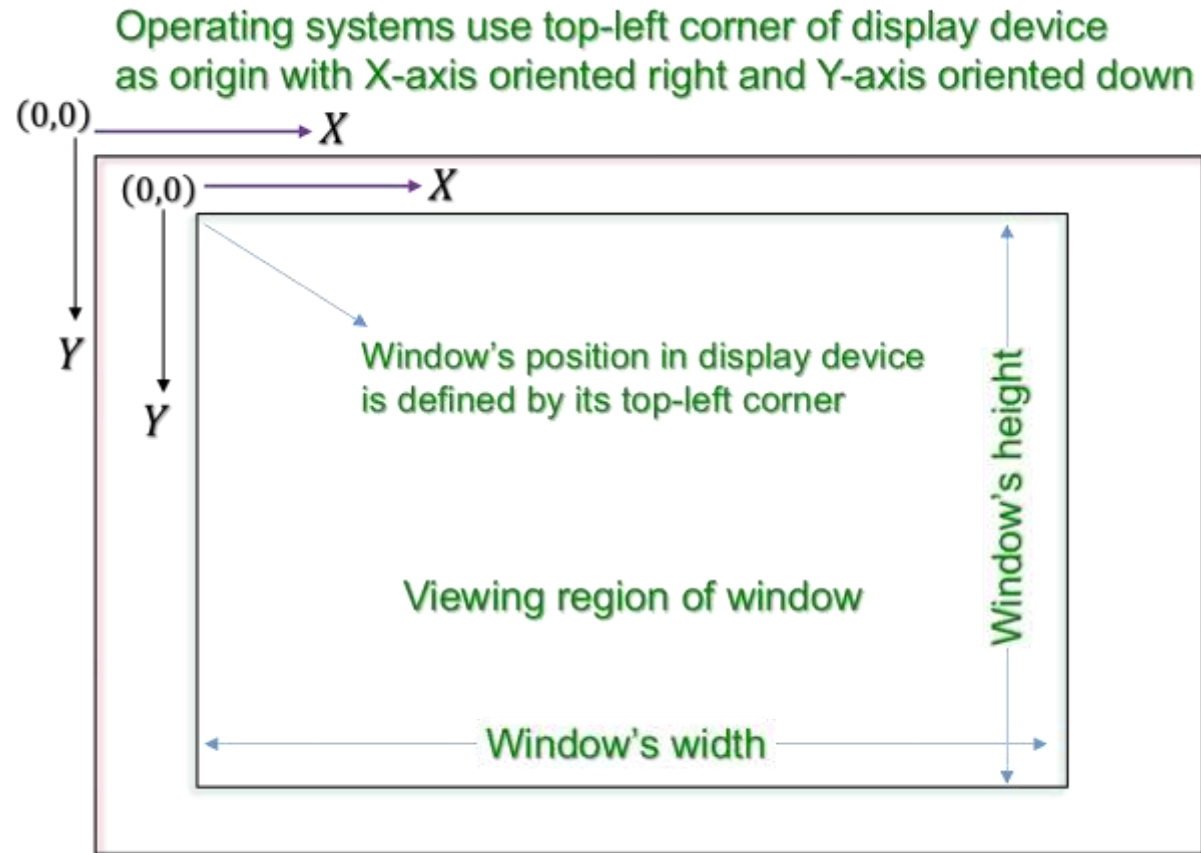
OpenGL Window Coordinate System

Operating systems use top-left corner of display device as origin with X-axis oriented right and Y-axis oriented down

$(0,0)$ → X

$(0,0)$ → X

Y

Y

Window's position in display device is defined by its top-left corner

Viewing region of framebuffer (or window)

Y^d

OpenGL uses window's bottom-left corner as origin with Y-axis oriented up

$(0,0)$ → X^d

Window height
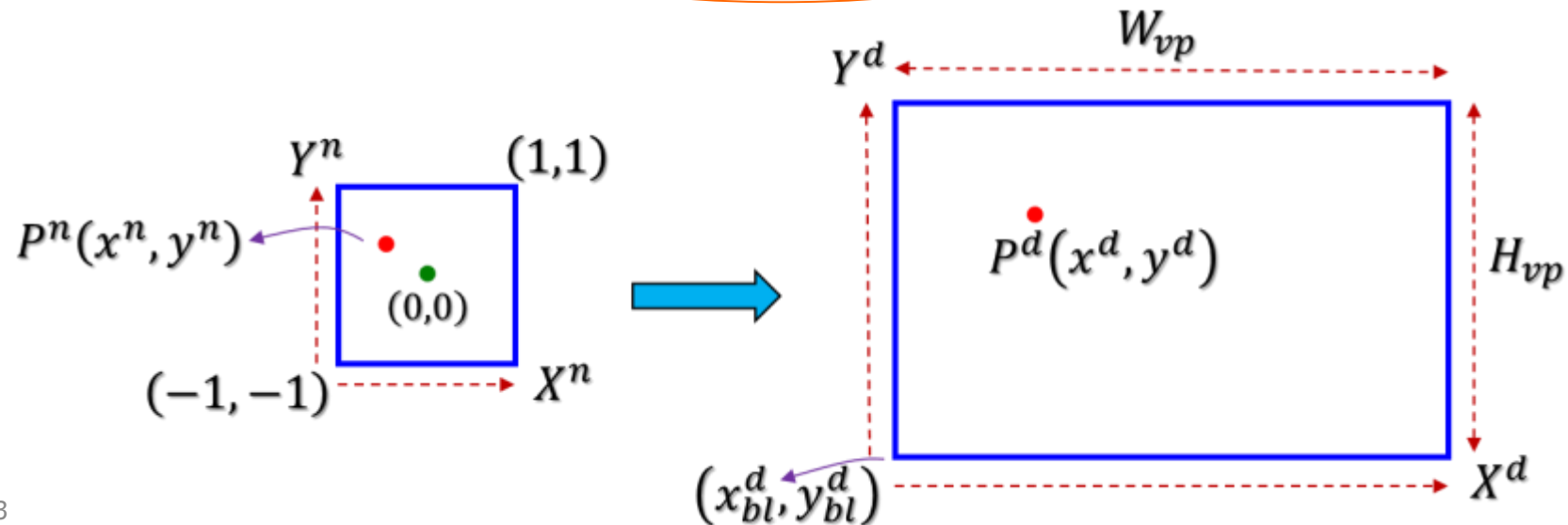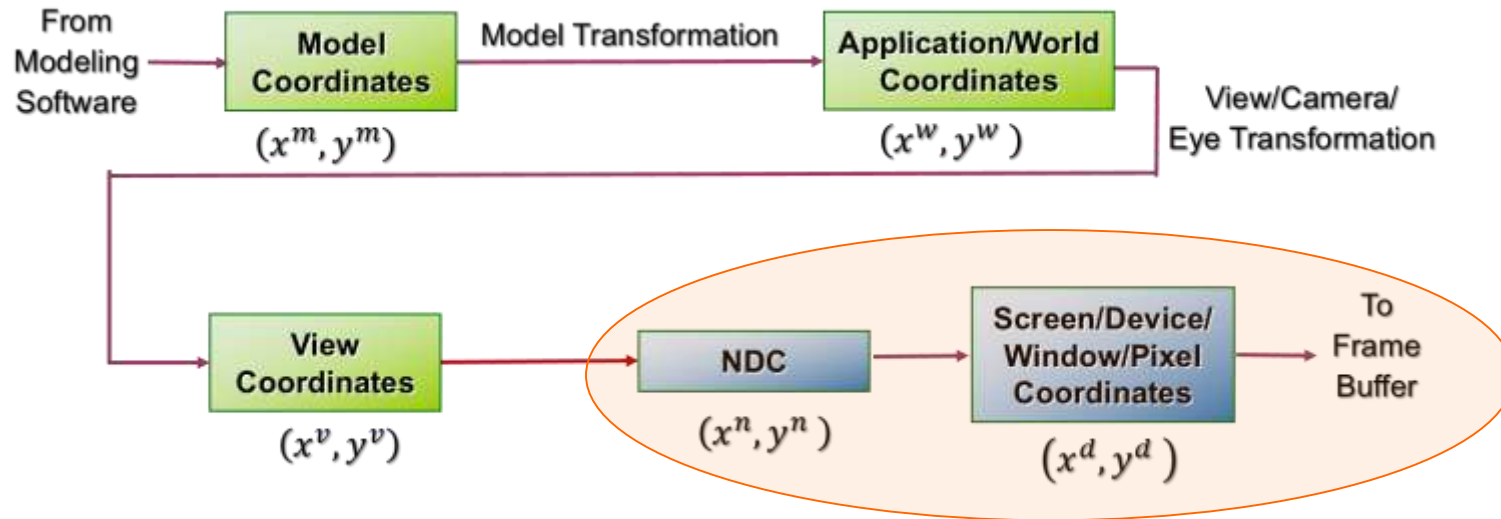
Window width

Superscript d indicates window (or device) coordinate system
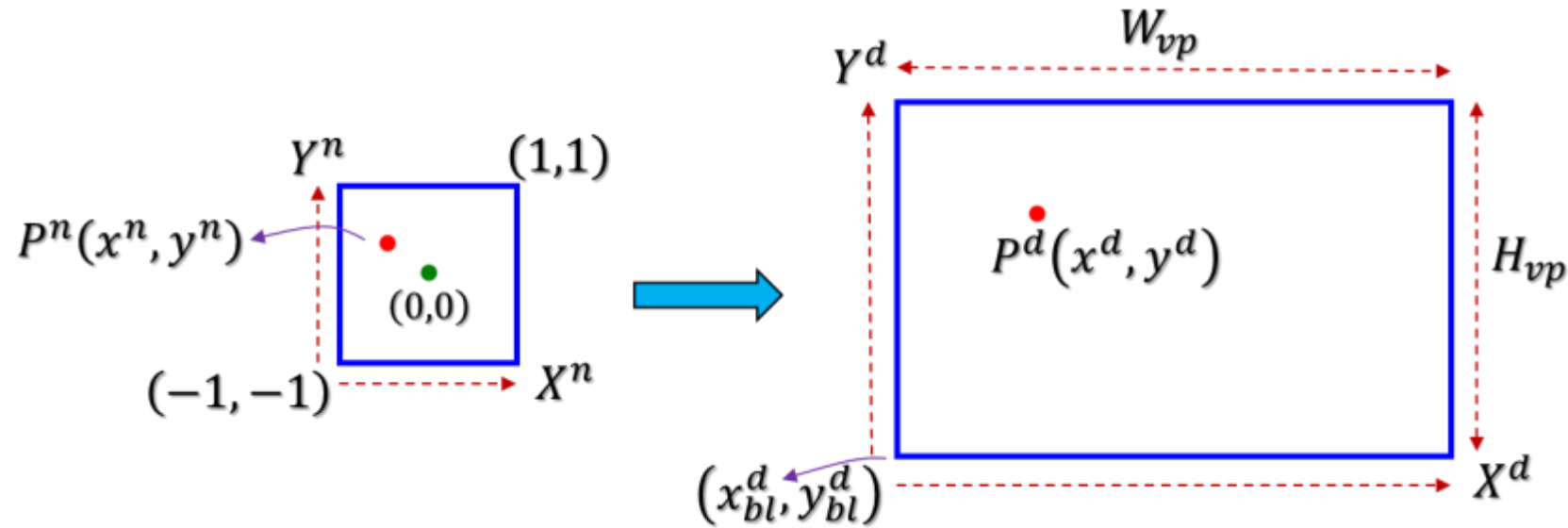
OpenGL Viewport in Window Coordinate System
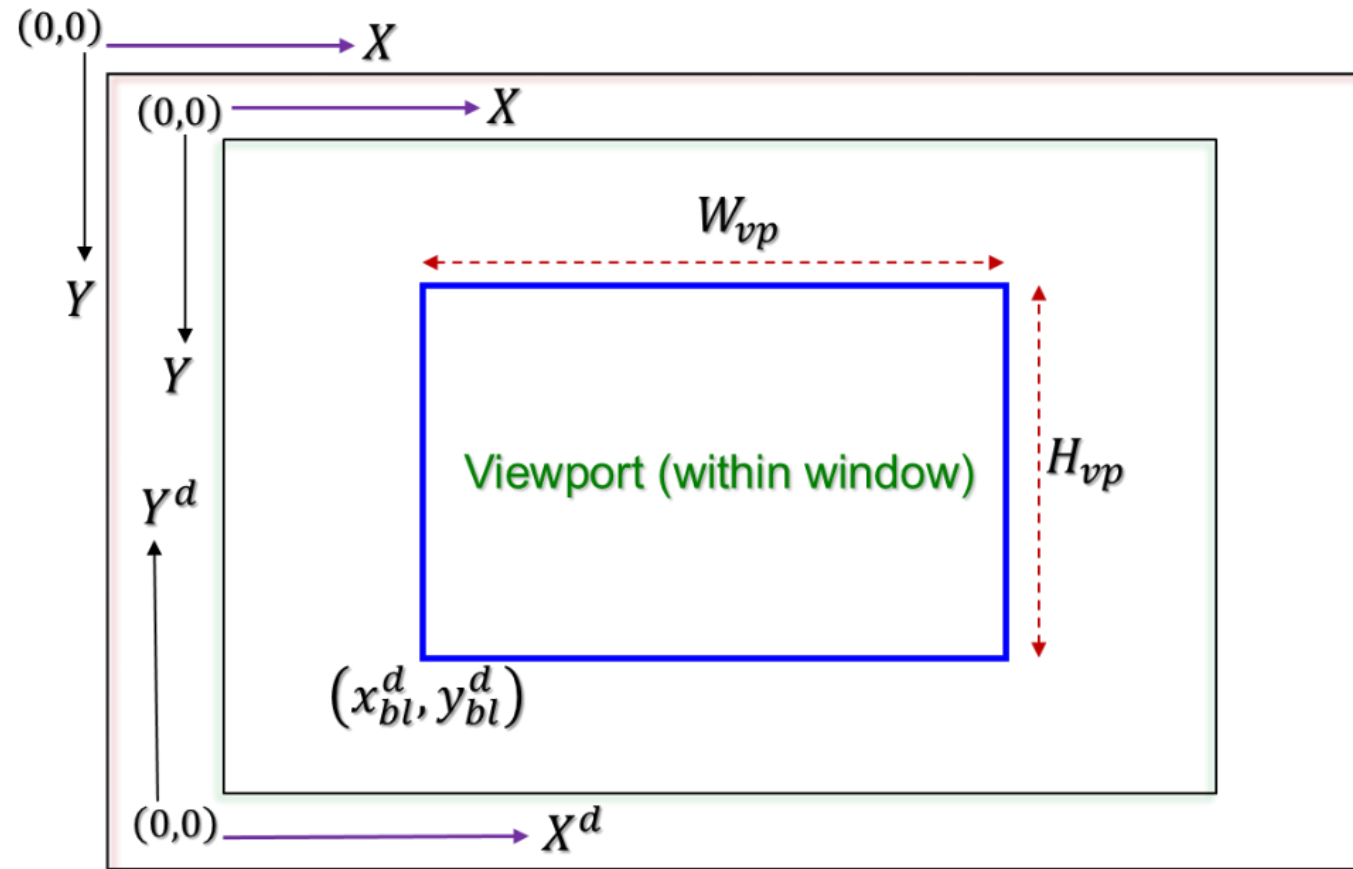
Viewport Transformation

Viewport Transformation Matrix

$$
\mathbf{M}_{viewport} = \begin{bmatrix} \dfrac{W_{vp}}{2} & 0 & x_{bl}^d + \dfrac{W_{vp}}{2} \\[2ex] 0 & \dfrac{H_{vp}}{2} & y_{bl}^d + \dfrac{H_{vp}}{2} \\[2ex] 0 & 0 & 1 \end{bmatrix}
$$