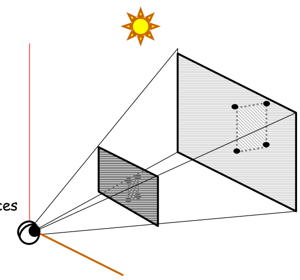
graphics pipeline 2

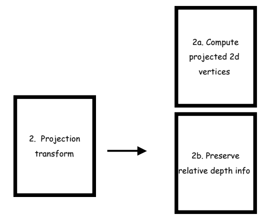
2. projection transform



transform vertices to (almost) projected vertices

10/27/2003

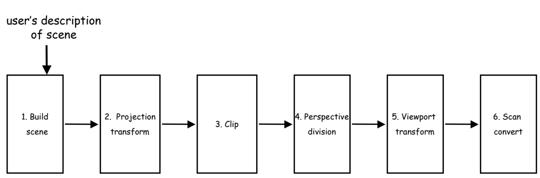
Projection ABC



linear transformation

10/27/2003

graphics pipeline

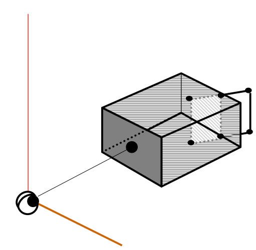


why keep depth info?

10/27/2003

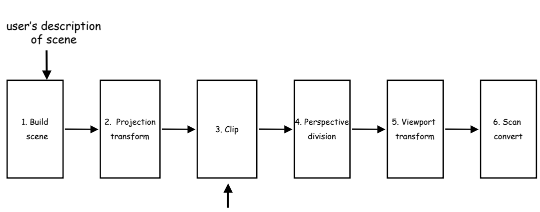
graphics pipeline 3

3. clip



10/27/2003

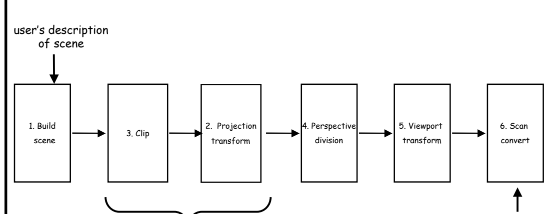
graphics pipeline



need relative depth info

10/27/2003

graphics pipeline



this may seem like the way to do it

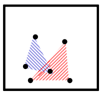
need relative depth info

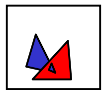
10/27/2003

scan conversion

2. Projection transform → 5. Scan convert →

need relative depth





graphics pipeline

user's description of scene

1. Build scene → 3. Clip → 2. Projection transform → 4. Perspective division → 5. Viewport transform → 6. Scan convert

this may seem like the way to do it

but it doesn't eliminate the need to keep depth info

need relative depth info

graphics pipeline

user's description of scene

1. Build scene → 2. Projection transform → 3. Clip → 4. Perspective division → 5. Viewport transform → 6. Scan convert

linear transforms can be collapsed into 1

$v \rightarrow Mv$ where $M=M_p\underbrace{M_vM_w}$

remain fixed for all vertices in scene

graphics pipeline

user's description of scene

1. Build scene → 2. Projection transform → 3. Clip → 4. Perspective division → 5. Viewport transform → 6. Scan convert

linear transforms can be collapsed into 1

$v \rightarrow Mv$ where $M=M_pM_vM_w$

optimized by use of hierarchical coordinates & matrix stacks

Projection ABC

2. Projection transform →

2a. Compute projected 2d vertices

2b. Preserve relative depth info

2c. Convert to canonical world

linear transformation

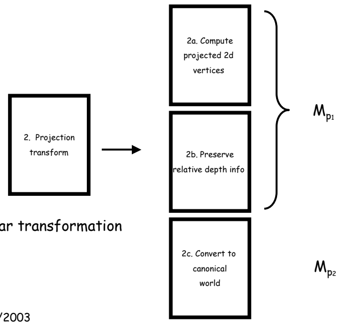
graphics pipeline

user's description of scene

1. Build scene → 2. Projection transform → 3. Clip → 4. Perspective division → 5. Viewport transform → 6. Scan convert

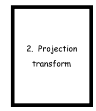
optimized for "canonical world"

Projection ABC



10/27/2003

Projection ABC



$u \rightarrow M_{p_2} M_{p_1} u$

or the shorthand

$u \rightarrow M_p u$

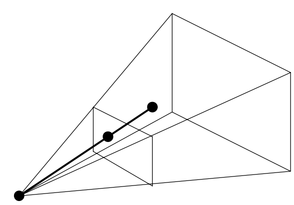
10/27/2003

actually … there are 2 types of projection

- perspective
- orthographic

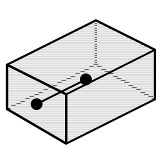
10/27/2003

perspective projection



10/27/2003

orthographic projection



10/27/2003

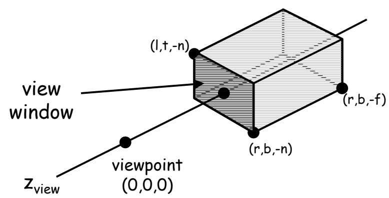
actually … there are 2 types of projection

- perspective
- **orthographic**

10/27/2003

orthographic view volume

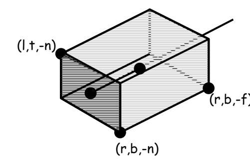
axes aligned parallelpiped



orthographic projection

Project vertex

$(x,y,z) \to (x,y,-n)$

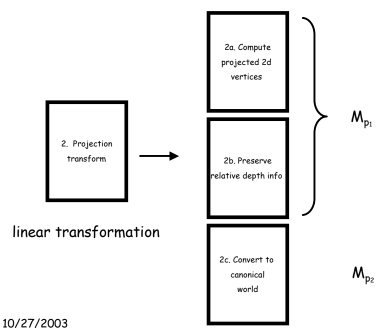


2a. compute projected 2d vertices: (x,y)

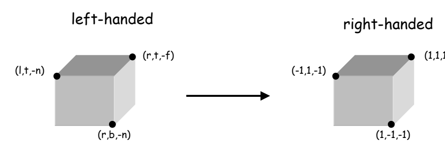
2b. preserve relative depth information: z

$M_{p_1} = I$

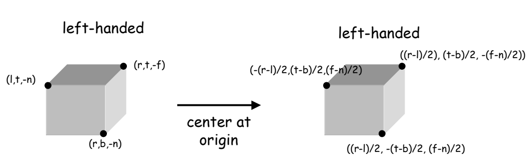
Projection ABC



orthographic M_{p_2}

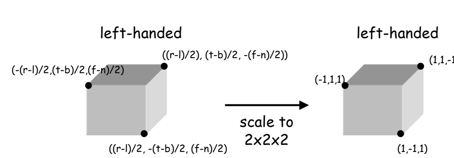


center at origin



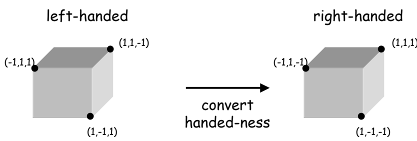
1. translate by (-(r+l)/2, -(t+b)/2, (f+n)/2))

scale to 2x2x2



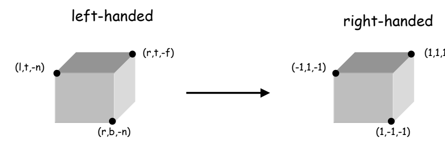
2. scale by (2/(r-l), 2/(t-b), 2/(f-n))

change handed-ness



3. scale by (0,0,-1)

orthographic M_{p_2}



1. translate by (-(r+l)/2, -(t+b)/2, (f+n)/2))

2&3. scale by (2/(r-l), 2/(t-b), -2/(f-n))

orthographic M_{p_2}

$$\begin{pmatrix} 2/(r-l) & 0 & 0 & 0 \\ 0 & 2/(t-b) & 0 & 0 \\ 0 & 0 & -2/(f-n) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -(r+l)/2 \\ 0 & 1 & 0 & -(t+b)/2 \\ 0 & 0 & 1 & (f+n)/2 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2/(r-l) & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2/(t-b) & 0 & -(t+b)/(t-b) \\ 0 & 0 & -2/(f-n) & -(f+n)/(f-n) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

scale translate M_{p_2}

orthographic projection matrix

$$M_{p_2} I = M_p = \begin{pmatrix} 2/(r-l) & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2/(t-b) & 0 & -(t+b)/(t-b) \\ 0 & 0 & -2/(f-n) & -(f+n)/(f-n) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

orthographic projection

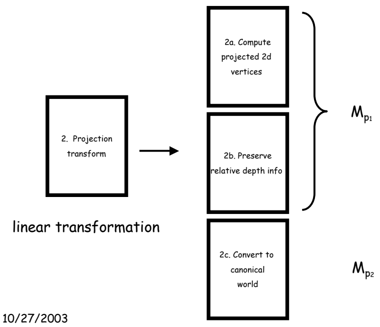
$$(x,y,z,1) \rightarrow (x',y',z',1)$$

where (x',y') is the vertex projected into a canonical 2x2 view window and z' is vertex's relative depth based on a canonical 2x2x2 world

actually … there are 2 types of projection

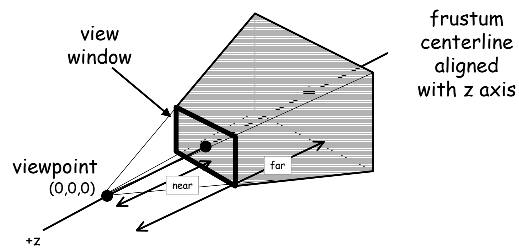
- **perspective**
- orthographic

Projection ABC



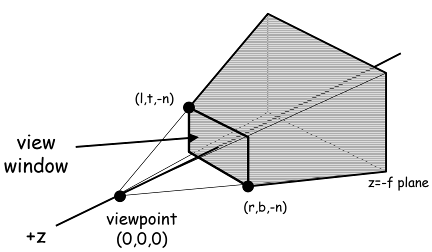
perspective view volume

in view coordinates

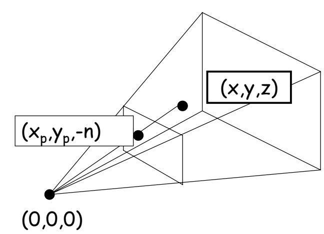


perspective view volume (frustum)

in view coordinates

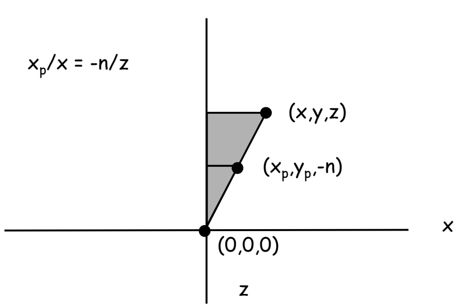


perspective projection

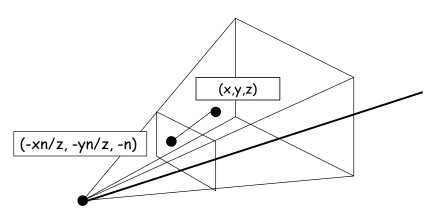


x-projection

$x_p/x = -n/z$



perspective projection



2a. compute projected 2d vertices: (-xn/z,-yn/z)

2b. preserve relative depth information: z

M_{p_1} = ?

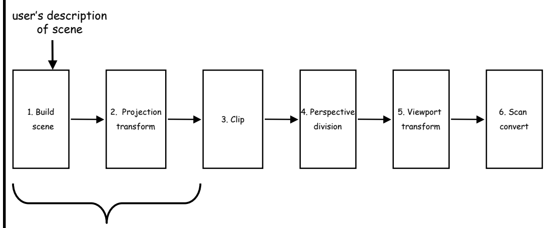
perspective M_{p_1}

what is wrong with this picture

$$\begin{pmatrix} -n/z & 0 & 0 & 0 \\ 0 & -n/z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} -nx/z \\ -ny/z \\ z \\ 1 \end{pmatrix}$$

M_{p_1} ?

graphics pipeline



linear transforms can be collapsed into 1

$v \rightarrow Mv$ where $M=M_p\underbrace{M_vM_w}$

remain fixed for all vertices in scene

perspective M_{p_1}

a little trick!

$$\begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ az-b \\ -z \end{pmatrix}$$

we'll specify a & b in a moment

remember scale factor -z in w-component

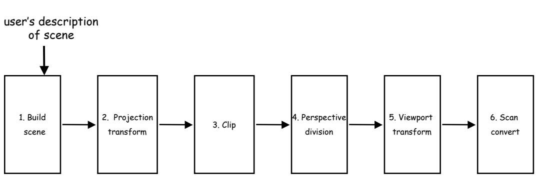
perspective M_{p_1}

a little trick!

$$\begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ az+b \\ -z \end{pmatrix} \longrightarrow \begin{pmatrix} -nx/z \\ -ny/z \\ -a - b/z \\ 1 \end{pmatrix}$$

later we'll normalize by w-component of vertex

graphics pipeline



later is step 4

perspective M_{p_1}

a little trick!

$$\begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ az+b \\ -z \end{pmatrix} \longrightarrow \begin{pmatrix} -nx/z \\ -ny/z \\ -a - b/z \\ 1 \end{pmatrix}$$

this is where the relative depth info will be

-b/z is (almost) as good as z for any nonzero constant b so let a=0 and b=fn!

perspective M_{p_1}

a little trick!

$$\begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & 0 & fn \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} nx \\ ny \\ fn \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -nx/z \\ -ny/z \\ -fn/z \\ 1 \end{pmatrix}$$

if the vertex has been clipped then

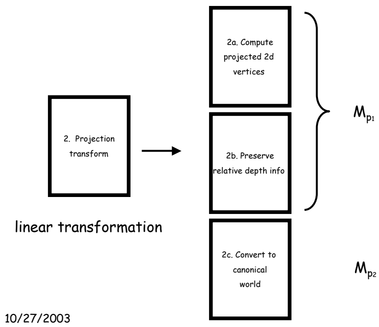
z takes on values in [-n,-f]

-fn/z takes on values in [n,f]

10/27/2003

(We switched to right-handed coordinates ... we'll fix this in a minute.)

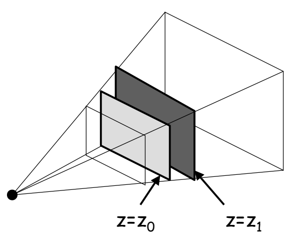
Projection ABC



10/27/2003

perspective projection

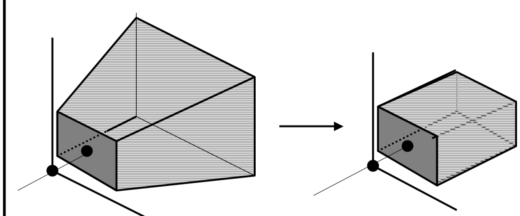
$(x,y,z) \rightarrow (-xn/z,\ \ -yn/z,\ \ -fn/z)$



depth-dependent scale

10/27/2003

depth dependent x,y scale



10/27/2003

perspective projection

1. M_{p_1}
2. perspective division
3. orthographic M_{p_2}

this would work

10/27/2003

alternative

1. M_{p_1}
2. orthographic M_{p_2}
3. perspective division

does this work?

sure – perspective division is multiplication by a scalar

(albeit a special one)

10/27/2003

perspective projection matrix

$$\begin{pmatrix} 2/(r-l) & 0 & 0 & -(r+l)/(r-l) \\ 0 & 2/(t-b) & 0 & -(t+b)/(t-b) \\ 0 & 0 & -2/(f-n) & -(f+n)/(f-n) \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n & 0 & 0 & 0 \\ 0 & n & 0 & 0 \\ 0 & 0 & 0 & fn \\ 0 & 0 & -1 & 0 \end{pmatrix} =$$

$$\begin{pmatrix} 2n/(r-l) & 0 & (r+l)/(r-l) & 0 \\ 0 & 2n/(t-b) & (t+b)/(t-b) & 0 \\ 0 & 0 & (f+n)/(f-n) & -2fn/(f-n) \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

except we're back in left-handed coordinates

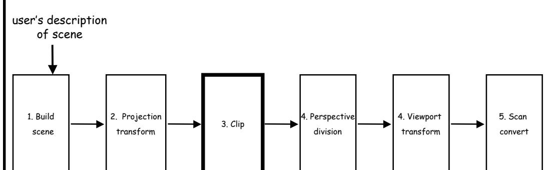
projection matrix: M_P

$$\begin{pmatrix} -2n/(r-l) & 0 & (r+l)/(r-l) & 0 \\ 0 & -2n/(t-b) & (t+b)/(t-b) & 0 \\ 0 & 0 & -(f+n)/(f-n) & -2fn/(f-n) \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

geometric primitives

object coordinates: v	description of vertex
world coordinates: $M_W v$	description of vertex situated in world
view coordinates: $M_V M_W v$	description of vertex in world as seen from a particular viewpoint
clip coordinates: $M_p M_V M_W v$	description of vertex seen from viewpoint in a normalized world

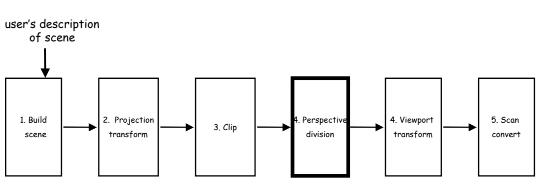
graphics pipeline



can we clip before perspective division?

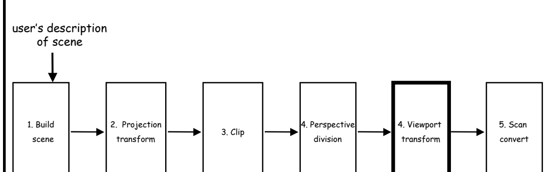
yes – we'll come back to this

graphics pipeline

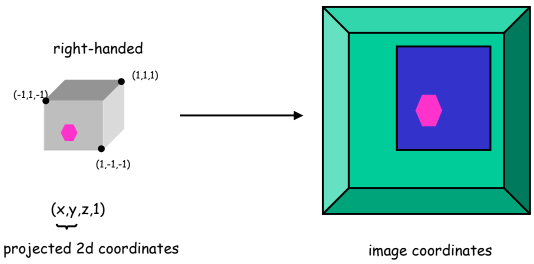


$(x,y,z,w) \rightarrow (x/w,y/w,z/w,1)$

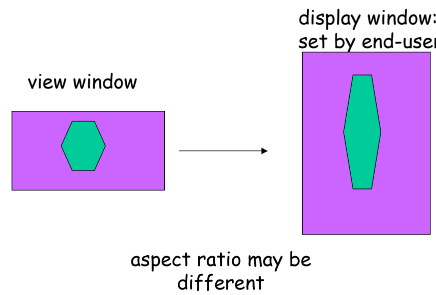
graphics pipeline



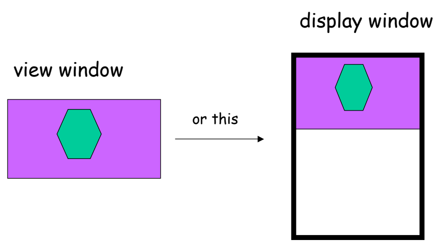
viewport transormation



viewport transformation



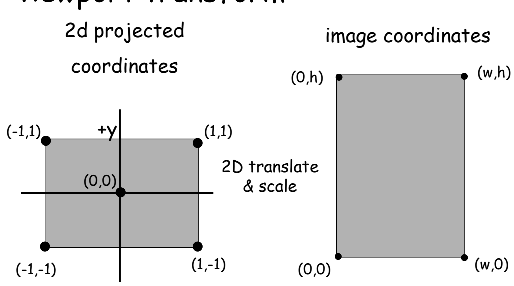
viewport transformation



viewport transformation

the user specifies how the projected world is mapped to the screen

viewport transform



We'll call this M_I

geometric primitives

object coordinates: v — description of vertex

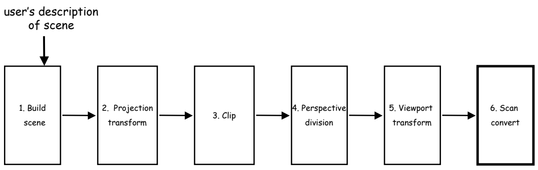
world coordinates: $M_W v$ — description of vertex situated in world

view coordinates: $M_V M_W v$ — description of vertex in world as seen from a particular viewpoint

clip coordinates: $M_p M_V M_W v$ — description of vertex seen from viewpoint in a normalized world

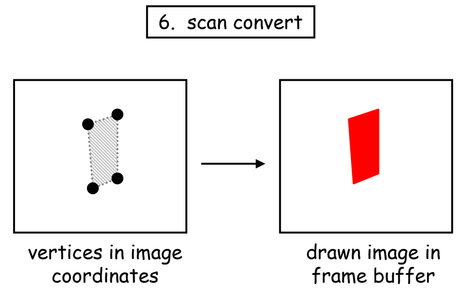
image coordinate $M_I M_p M_V M_W v$ — description of a vertex in image coordinates

graphics pipeline



10/27/2003

graphics pipeline 6



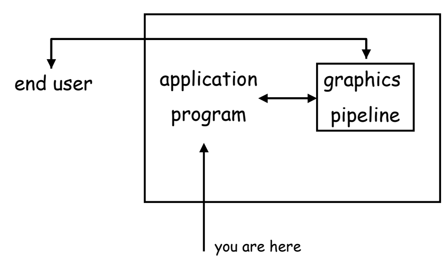
10/27/2003

pipeline

- we'll return to clipping and scan conversion next week
- for now, let's look at the pipeline transforms in openGL

10/27/2003

who's who



10/27/2003

application program

- primitive & their modeling transforms (M_w)
- viewpoint (M_v)
- projection type & view volume (M_p)
- viewport transformation (M_I)

10/27/2003