Introduction to 2D and 3D Computer Graphics

Mastering 2D & 3D Computer Graphics Pipelines

Mastering 2D & 3D Graphics

Overview of 2D & 3D Pipelines

- What are pipelines?
- What are the fundamental components of pipelines?
- Example pipelines...
  - ...2D graphic object pipeline
  - ...3D graphic object pipeline
  - ...raster pipeline
  - ...bitmapped, HLHSR, illumination implications

Overview of 2D & 3D Pipelines

What are Pipelines?

- Are the fundamental concept around which graphics systems are based
- Represent conceptual models
- Describe the interaction and the flow of data for all functions
- Illustrate how the applications can create, save, modify, draw, and display pictures

Overview of 2D & 3D Pipelines

What are Pipelines?

- Illustrate how applications can create pictures
- Illustrate the use of functions to set up the environment
- Illustrate how the environment affects pictures being created
- Illustrate the interactions and the relationships between functions

Overview of 2D & 3D Pipelines

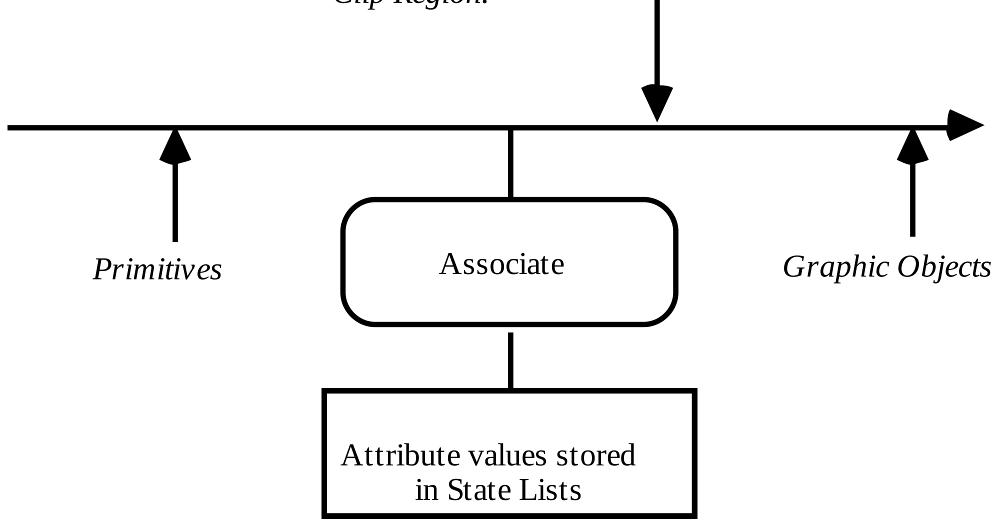
What are the fundamental components?

- Common components:
  - Attribute Association
  - Display-List Storage
  - Bundled Attribute Association
  - Modeling
  - Viewing & Clipping
  - Rendering
    - with many sub-components!

Overview of 2D & 3D Pipelines

Attribute Association: pipeline illustration

Graphic objects with
associated attributes;
for example: *Primitive Attributes,*
ASF Bundle Indicators,
Bundle Indices,
Width/Size Specification Mode,
Object Clipping Mode,
Transparency,
Drawing Mode,
Auxiliary Colour,
Pick Identifier,
Transformation (initially identity),
Clip Identifier, and
Clip Region.



Overview of 2D & 3D Pipelines

Bundled Attribute Association

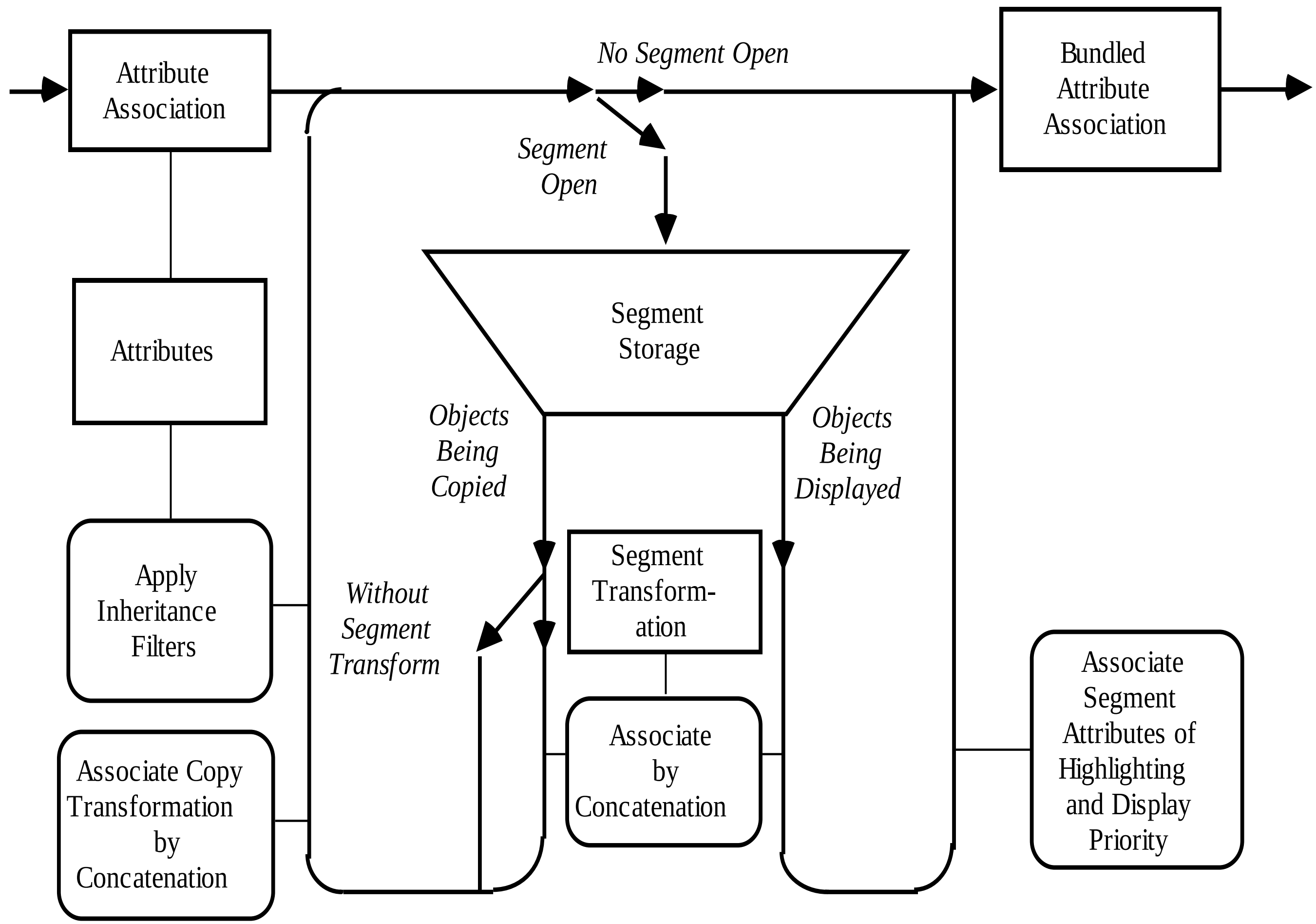
- Attributes set and accessed through tables...
  - ...are called bundled attributes
  - ...are grouped into bundle tables
  - ...can change the attributes associated with a graphic object as it flows along the pipeline

Overview of 2D & 3D Pipelines

Display-List Storage

- Segment models...(e.g., CGI, GKS)
  - ...describe the mechanism for saving graphic objects
- Structure models...(e.g., PHIGS)
  - ...describe the mechanism for saving primitive and complex elements
- Macro & procedure models...(e.g., POSTSCRIPT®)
  - …describe the mechanism for saving macros, procedures, & subroutines in memory for later use

One Approach to Display Lists *using segments*

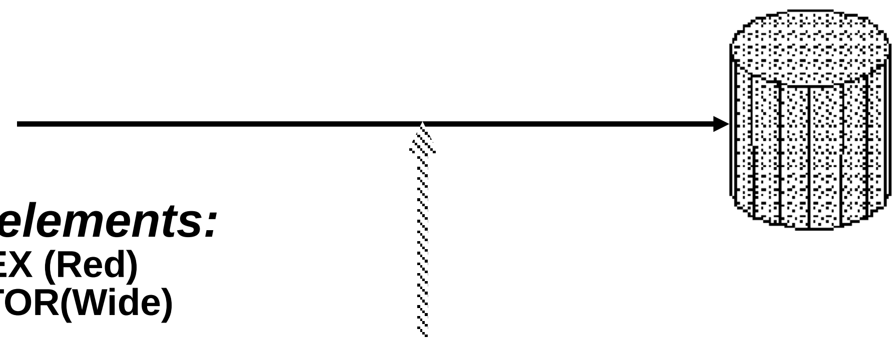


Or, with Structures...

Structure Creation: data definition process

Workstation Independent

Application
Program:



...functions that create elements:

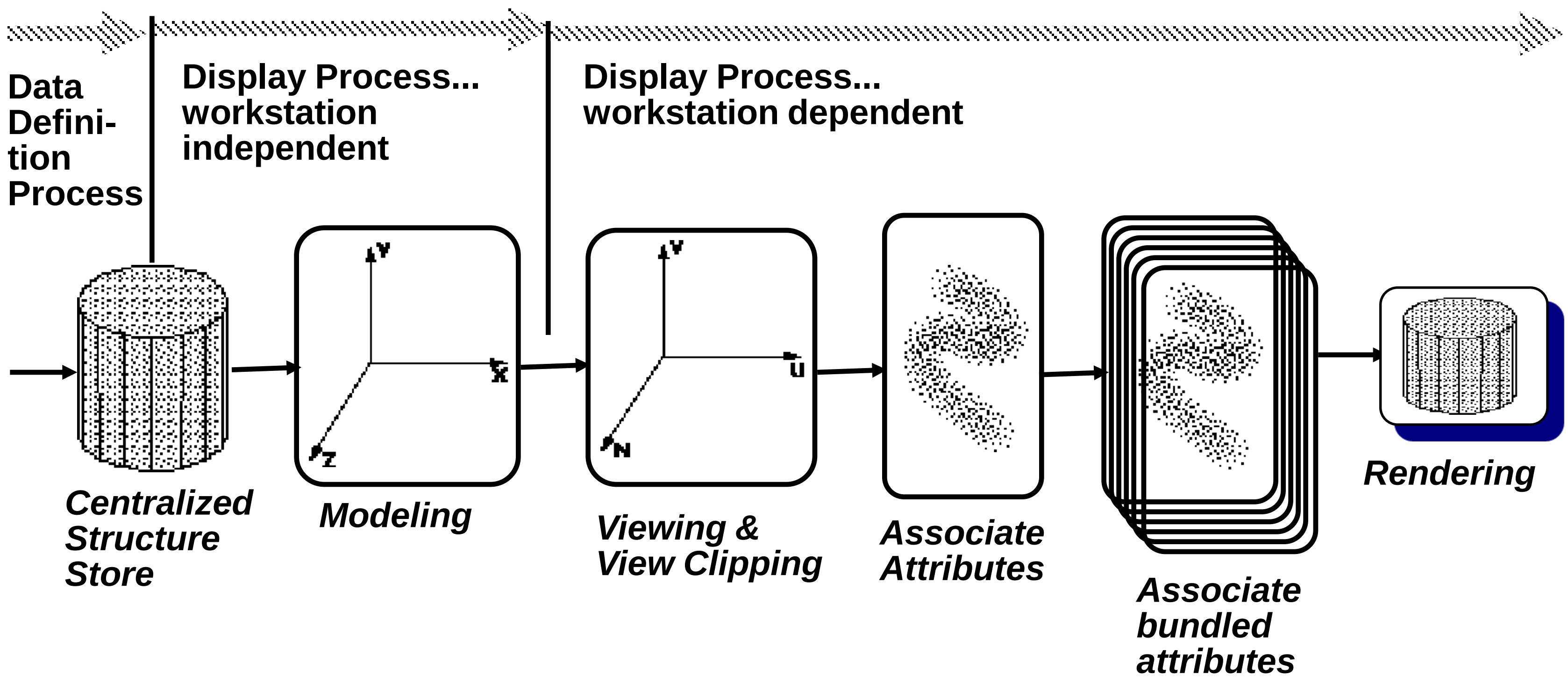
SET POLYLINE COLOUR INDEX (Red)
SET LINEWIDTH SCALE FACTOR(Wide)
SET LINETYPE(Dashed)
POLYLINE 3 ( **)**

...these functions cause elements to flow along the pipeline and into centralized structure storage:

set polyline colour index (red)
set linewidth scale factor (wide)
set linetype (dashed)
polyline 3 ( ***)***

Overview of 2D & 3D Pipelines

Example: 3D graphic object pipeline



Overview of 2D & 3D Pipelines
Modeling

- Modeling transformations...
  - ...allow objects to be defined in their own coordinate system
- Through modeling...
  - ...objects are transformed into world coordinates by scaling, rotation, and/or translation

Overview of 2D & 3D Pipelines

Modeling

- This allows all objects to initially be defined as centered around the origin...
  - ...and then modeling transformations can position, orient, and scale them into their correct geometric relationships
- This stage in the pipeline is independent of the display characteristics

Overview of 2D & 3D Pipelines

Modeling

- The modeling transformation...
  - ...maps modeling coordinates into world coordinates
  - ...is defined by transformation matrices inserted into a structure network
  - ...allows the picture to be composed of separate parts, each defined within its own modeling coordinate system
  - ...supports relative positioning of these parts when they are mapped to world coordinates

Overview of 2D & 3D Pipelines

Modeling

- Modeling Coordinates (called MC) are.
  - ...3D right-handed device-independent coordinates
  - ... specified by a homogeneous transformation matrix
- When modeling transformations are not inserted into a structure...
  - ...modeling coordinates are the same as world coordinates

Mastering 2D & 3D Graphics
Coordinate Spaces

- 2D transformations are represented by 3X3 matrices using homogeneous coordinates...
- 3D transformations are represented by 4x4 matrices using homogeneous coordinates in 3-space

Mastering 2D & 3D Graphics

Coordinate Spaces: homogeneous coordinates

- What are homogeneous coordinates?
- Well...instead of representing a point as (x,y,z)...
  - ...we represent it as (x,y,z,W) ...but we most commonly refer to it as (x/W, y/W, z/W, 1)
- When we transform a point to this form...
  - ...it is called homogenizing

Mastering 2D & 3D Graphics

Coordinate Spaces: homogeneous coordinates

- This means...that each point in 3-space is represented by a line through the origin in 4-space!
- So...homogenizing allows us to use points that form a 3D subspace of 4-space (i.e., W=1)

Mastering 2D & 3D Graphics

Coordinate Spaces: right-handed

- Most graphics systems use the right-handed coordinate system for the definition of objects
- In right-handed systems...
  - ...positive rotations occur when looking from the positive axis to the origin
  - ...a 90˚ counterclockwise rotation will transform one positive axis into the other

Mastering 2D & 3D Graphics

Coordinate Spaces: right-handed

- For example, to rotate around the x axis...
  ...the positive rotation is from y to z
- Or, to rotate around the y axis...
  - ...the positive rotation is from z to x
- And lastly, to rotate around the z axis...
  - ...the positive rotation is from x to y

Mastering 2D & 3D Graphics

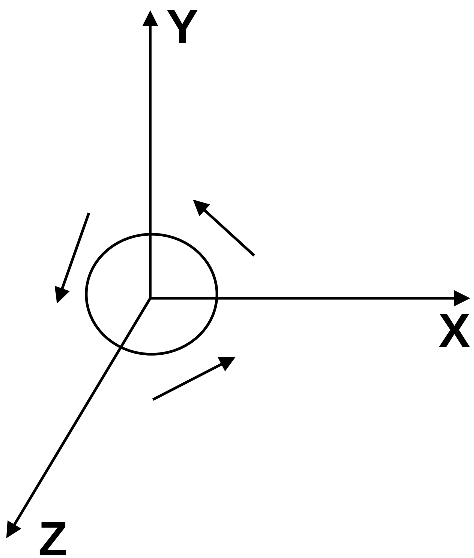
Coordinate Spaces: left-handed

- Some graphics systems use the left-handed coordinate system for viewing objects...
  - Left-handed systems may seem more natural... ...as larger z values appear further from the operator
- In left-handed systems...
  - ...positive rotations occur when looking from the positive axis to the origin
  - ...a 90° clockwise rotation will transform one positive axis into the other

Mastering 2D & 3D Graphics

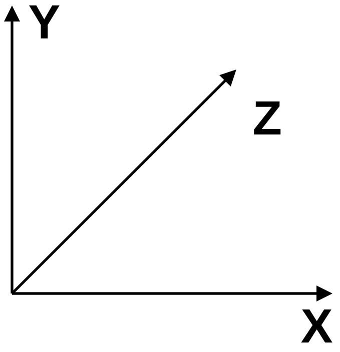
Coordinate Spaces

Right-Handed



Z is coming out of the display towards you

Left-Handed



Z is heading into the display (or away from you)

Mastering 2D & 3D Graphics

Linear Transformations

- A set of vertices or 3D points belonging to an object...
  - ...can be transformed into another set of points ...using linear transformations

Mastering 2D & 3D Graphics
Linear Transformations

- Translation:
- $$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{bmatrix}$$
- Translation means...
  - $x' = x + Tx \quad y' = y + Ty \quad z' = z + Tz$

Mastering 2D & 3D Graphics

Linear Transformations

- Scaling: Sx, Sy, and Sz are scaling factors
- $$[x'\ y'\ z'\ 1] = [x\ y\ z\ 1] \begin{matrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$
- When Sx = Sy = Sz, there is uniform scaling...
  - ...otherwise scaling occurs only along the axes specified

$x' = xSx \quad y' = ySy \quad z' = zSz$

Mastering 2D & 3D Graphics

Linear Transformations

- To rotate an object in three-dimensional space, an axis of rotation must be specified...
  - ...it is best to use one of the coordinate axes
- Counterclockwise rotation about each of the axes uses (around X, Y, Z respectively)
- 1 0 0 0 cos° 0 -sin° 0 cos° sin° 0 0
- 0 cos° sin° 0 0 1 0 0 -sin° cos° 0 0
- 0 -sin° cos° 0 sin° 0 cos° 0 0 0 1 0
- 0 0 0 1 0 0 0 1 0 0 0 1

Mastering 2D & 3D Graphics

Linear Transformations

- Around the Z axis, this means:
  - $x' = x\cos^{\circ} - y\sin^{\circ}$
  - $y' = x\sin^{\circ} + y\cos^{\circ}$
  - $z' = z$

Mastering 2D & 3D Graphics

Linear Transformations

- When objects are not at the origin...
  - ...a simple rotation will not rotate the object in place
- Instead, we must...
  - ...first translate the object to the origin,
  - ...apply the desired rotation, and
  - ...translate the object back to its original position

Mastering 2D & 3D Graphics

Linear Transformations

- This means for a rotation around the Z axis (when the Z axis passes through the point (Tx, Ty, 0))...
  - $x' = x\cos^\circ - y\sin^\circ - Tx\cos^\circ + Ty\sin^\circ + Tx$
  - $y' = x\sin^\circ + y\cos^\circ - Tx\sin^\circ - Ty\cos^\circ + Ty$
  - $z' = z$

Mastering 2D & 3D Graphics

Concatenating Transformations

- Useful in describing objects that have a natural hierarchy
- For example, a stylization of a human figure used in animation would...
  - ...have a set of local coordinate systems based on limb joints
  - ...describe a hierarchy based on these joints
  - ...allow the hand to move with respect to the wrist joint and the lower arm to move with respect to the elbow joint

Modeling Transformations

Composition: Concatenating Xforms

- With Preconcatenate...
  - ...the specified transformation is applied to the object first
  - ...if you preconcatenate a rotation, that rotation will be applied to the object before the other transformations
- Current xform x New (L'=LxM)

Modeling Transformations

Composition: Concatenating Xforms

- With Postconcatenate...
  - ...the specified transformation is applied to the object after the current existing local transformation
- New x Current Xform (L'=MxL)

Modeling Transformations

Composition: Inheritance

- Inheritance of modeling transformations means...
  - ...a parent's composite modeling transformation becomes the initial child's global modeling transformation
  - ...a child's initial local modeling transformation is the identity matrix
  - ...descendant structures do not affect parent structures

Modeling Transformations

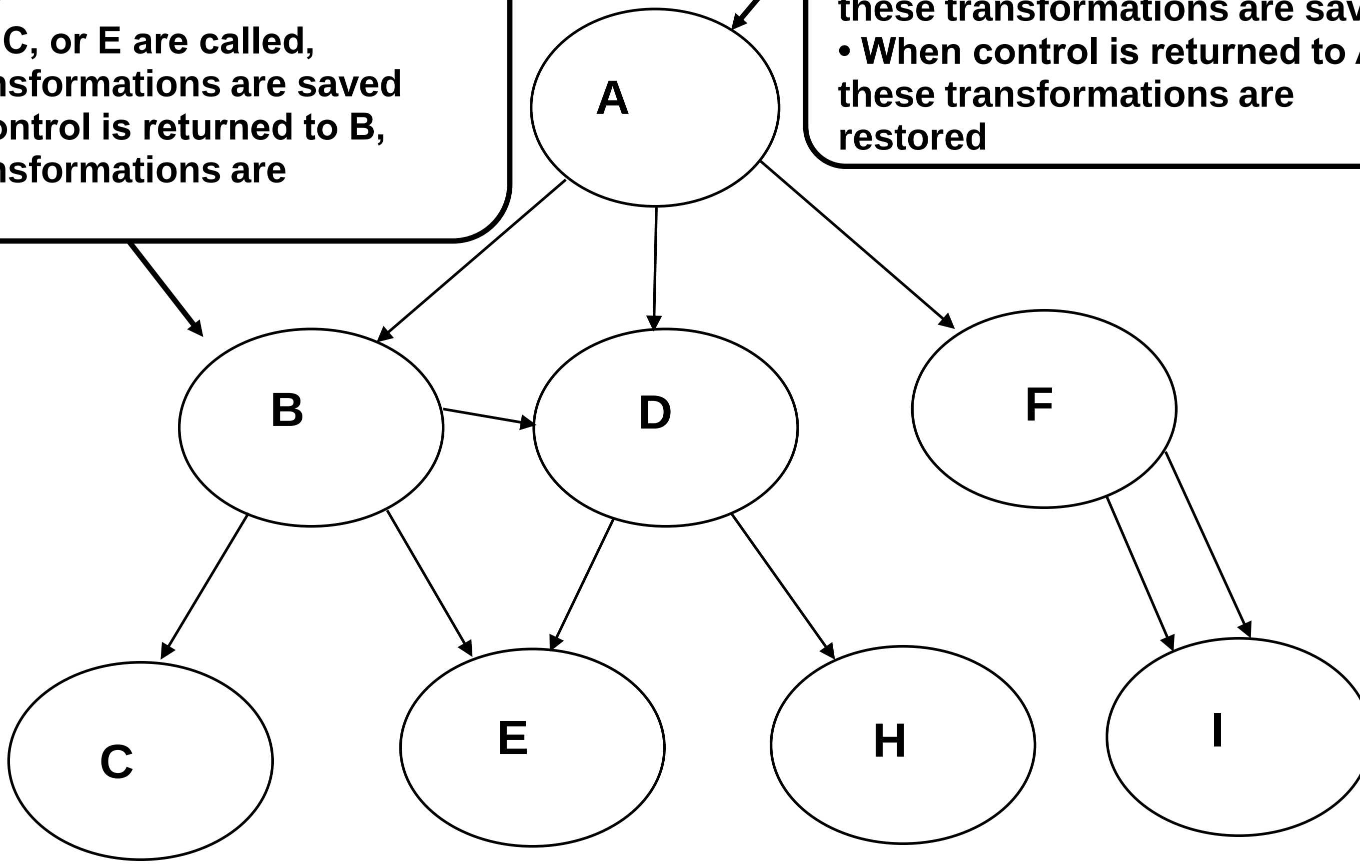
Composition: Inheritance

- Modifications...
  - to the global transformation only have effect within the structure in which they are encountered
  - This allows structure hierarchies to be built with movable parts...
  - ...since the motion of the parent structure is passed to its children via inheritance of the composite modeling transformation, the parts are kept connected

Inheritance...

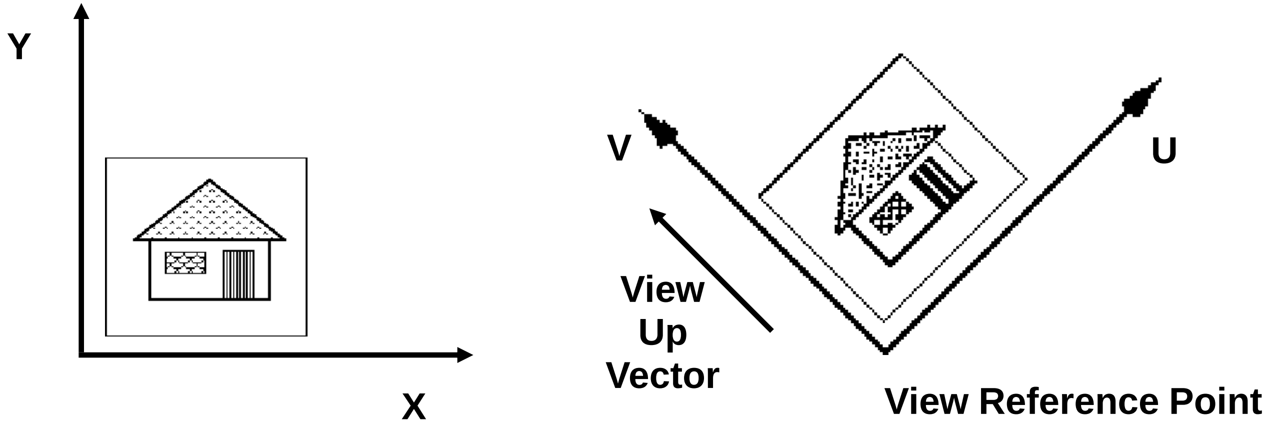
- When B is called, A's composite transformation becomes B's global transformation
- B's initial local transformation is identity
- When D, C, or E are called, these transformations are saved
- When control is returned to B, these transformations are restored

- Initial identity local & global transformations
- When B, D, or F are called, these transformations are saved
- When control is returned to A, these transformations are restored



2D Viewing Transformations
View Orientation Transformation

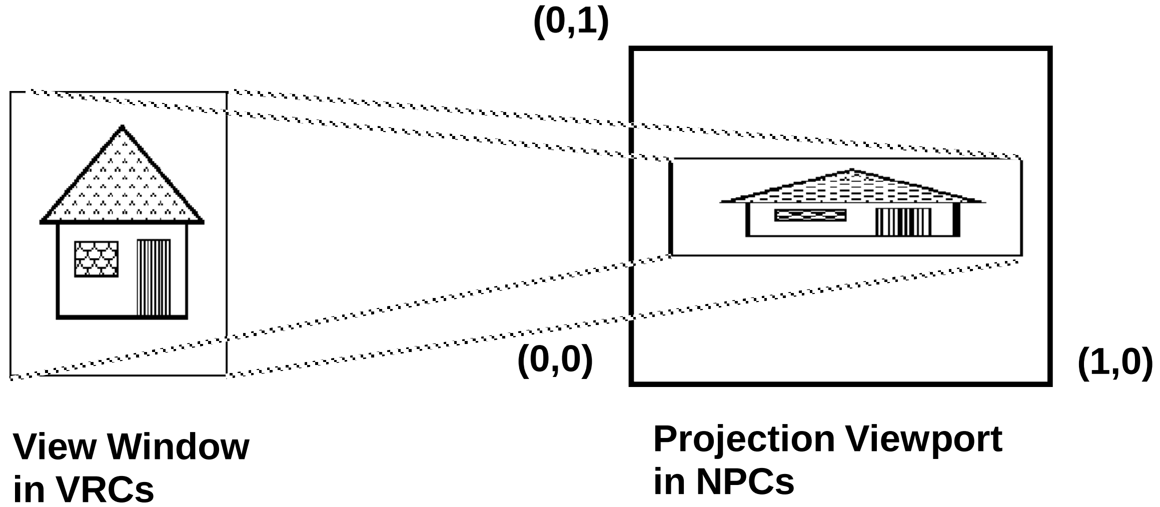
- View orientation transformation...
  - ...defines the position and orientation of view reference coordinates relative to world coordinates



2D Viewing Transformations

View Mapping Transformation

- View mapping transformation...
  - ...defines the mapping of view reference coordinates to normalized projection coordinates, allowing for distortion...

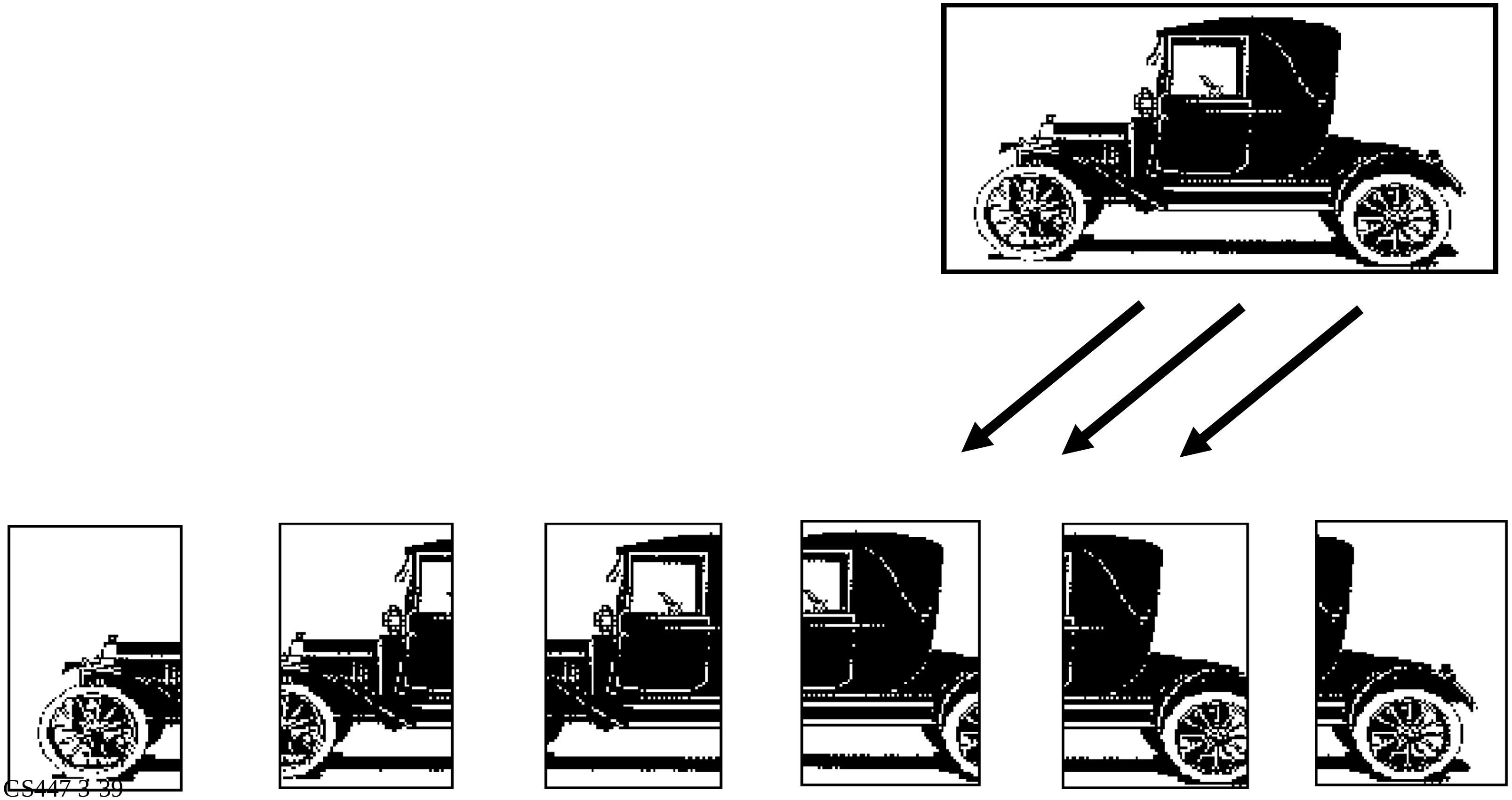


2D Viewing Transformations
View Mapping Transformation

- Allows dynamic horizontal and vertical panning...
  - ...by continuously adding to both sides of the view window, or
  - ...by changing the view reference point and leaving the view window unchanged
- Remember, panning is achieved by changing the window location!

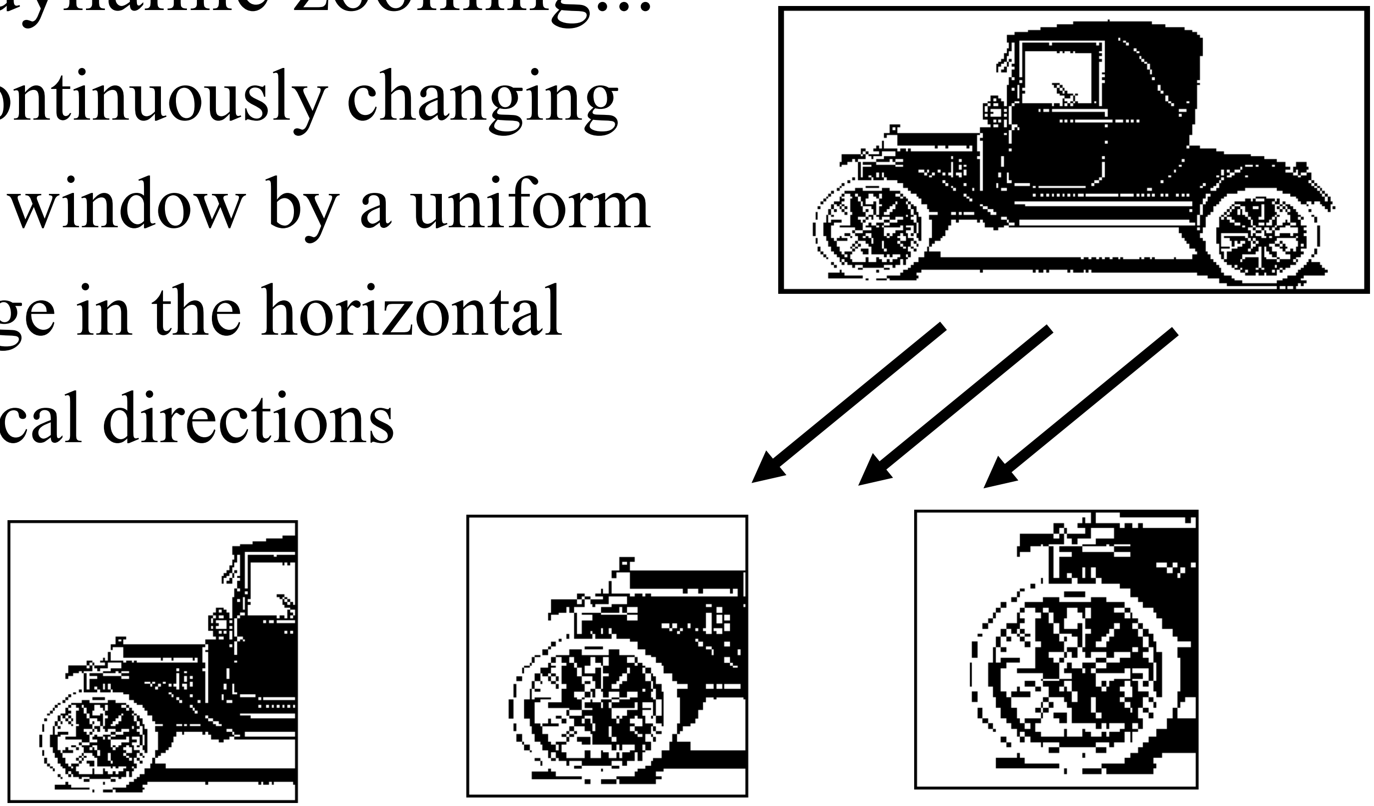
2D Viewing Transformations

View Mapping Transformation



2D Viewing Transformations
View Mapping Transformation

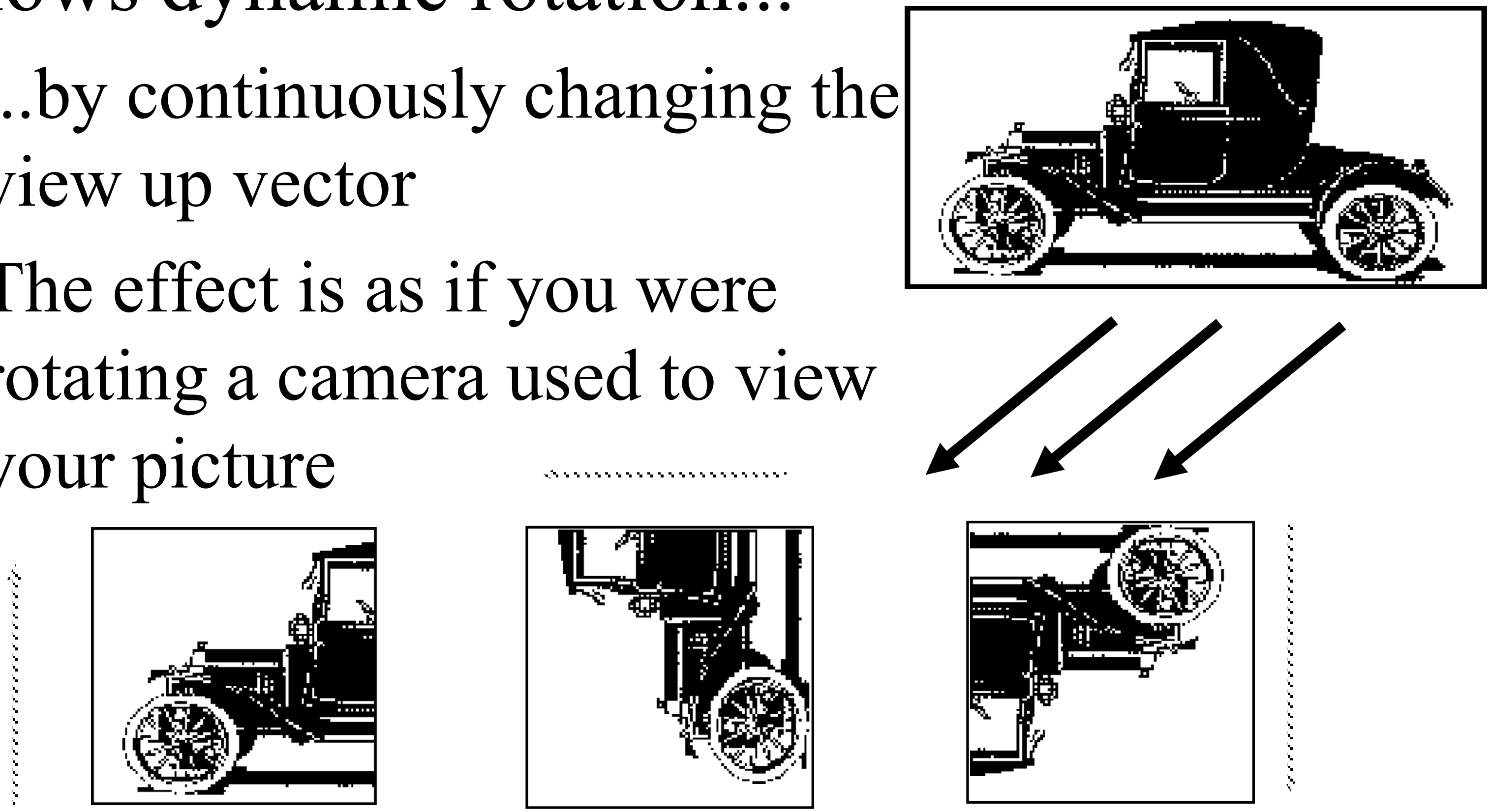
- Allows dynamic zooming...
  - ...by continuously changing the view window by a uniform percentage in the horizontal and vertical directions



2D Viewing Transformations

View Mapping Transformation

- Allows dynamic rotation...
  - ...by continuously changing the view up vector
  - The effect is as if you were rotating a camera used to view your picture



Overview of 2D & 3D Pipelines

3D Viewing & Clipping

- Three operations are associated with 3D viewing...
  - ...view orientation
  - ...view mapping ...XYZ clipping
- View orientation...
  - ...maps world coordinates to view reference coordinates
  - ...transforms and defines the position and orientation of the VRCs relative to world coordinates

Overview of 2D & 3D Pipelines

3D Viewing & Clipping

- View mapping...
  - ...maps view reference coordinates to normalized projection coordinates
    ...specifies a 3D volume of VRC to be mapped to a 3D volume of NPC (a unit cube extending from 0.0 to 1.0 in X,Y,Z)
- XYZ clipping...
  - ...is performed on the XY boundary (or clip rectangle), and front and back planes

Overview of 2D & 3D Pipelines
Rendering

- Is the general process for changing a 2D or 3D graphic object to a shaded 2D projection onto a drawing surface
- Rendering involves...
  - ...setting up data structures for polygon mesh models to ensure that they will contain all of the information required in the shading process

Overview of 2D & 3D Pipelines
Rendering

- Rendering also involves...
  - ...setting up data structures for polygon mesh models to ensure that they will contain all of the information required in the shading process
  - ...culling back-facing polygons (3D Only)
  - ...applying a hidden surface removal algorithm(3D Only) ...scan converting or rasterizing polygons: converting an object's vertices into a set of pixel coordinates & shading

Overview of 2D & 3D Pipelines

Example: 2D graphic object pipeline