Arch 481:
3D Modeling + Rendering

LeCorbusier's Villa Savoye

Handout

Online

University of Washington
Department of Architecture

Arch 481
3D Modeling and Rendering

ARCH 481: 3d Modeling and Rendering

http://online.caup.washington.edu/courses/arch481

When:	Tu/Th, 1:30 to 2:50	**Where:**	Digital Commons (GLD007F)
Credits:	3, Graded	**Prerequisite:**	Arch 380 (or permission)
Instructor:	Brian Johnson	**GSA:**	Nanching Tai
E-mail to:	brj@u.washington.edu	**E-mail to:**	tai@u.washington.edu
Office:	Condon 622	**Office:**	varies
Office Hours:	By appointment	**Office Hours:**	see web pages
Office phone:	206.543.2132	**Office phone:**	n/a

This is a course about Modeling and Rendering with computer graphics tools. There are two important things to keep in mind as you start this study, and they are encapsulated by these two quotes:

> *What remains hard is modeling. The structure inherent in three-dimensional models is difficult for people to grasp and difficult too for user interfaces to reveal and manipulate. Only the determined model three-dimensional objects, and they rarely invent a shape at the computer, but only record a shape so that analysis of manufacturing can proceed. The grand challenges to three-dimensional graphics are to make simple modeling easy and make complex modeling accessible to far more people.*
>
> *-Robert Sproull, keynote speech, SIGGRAPH 1990*

Robert Sproull is one of the "founding fathers" of computer graphics. The "Special Interest Group on Graphics" (SIGGRAPH) annual conference is the biggest computer graphics conference in the world. This guy knew what he was talking about in 1990, and it's still true today.

> *A journey of a thousand miles must begin with a single step.*
>
> *-Lao-tzu*

Lao-tzu was a Chinese philosopher. China is a big country. He knows what he's talking about too.

Introduction

This course is about *beginnings*, about preparation for the use of 3D computing during your professional career. It is not a course about a particular software package. You may not use the same programs in a few months or years as you use this quarter, not because they aren't good, but because change happens. From this point of view it is not so important what programs or commands you learn as it is what you learn from using those programs and commands. On the other hand, it's very hard to learn about this subject purely as "book learning". You need to *do* modeling and rendering as well as learn the concepts. The exercises and lectures are intended to work together to provide an enriched "soil" in which your professional experiences can flourish.

This is a fast-paced, "learning by doing" lecture & laboratory course. Each week lectures and demonstrations presented during the class sessions will prepare you to undertake a related modeling and/or rendering exercise. Background reference and online readings will explain the technical issues surrounding the exercises.

The weekly exercise is a very important element of the course. The lectures support and explain the exercises by providing material that may not be available in the readings and by providing an opportunity to ask questions. However, completing the exercise provides most of the actual learning experience. The exercises take time (5-10 hours/week) to complete and write up. That time must generally be spent on campus. In addition, most of the help is available during the day and early evening. Please consider these things as you set up your schedule for the quarter.

26.IX.06 1

ARCH 481: 3D MODELING AND RENDERING AUTUMN QUARTER, 2014

Lecture & Project Schedule

Dates	**LECTURE TOPIC**/Lab Subject	Due [†]	Weekly topic cluster
Th 9/25	**#1 Concepts**		3D modeling data and operations; Solids (geometry + topology); Mouse input. "Cameras:" projections from 3D to 2D, saving 2D images.
Tu 9/30	CONCEPT MODELS	#0	
Th 10/2	**#2 Lights + Shading**		Boolean operations. Lights, Shading, Shadows. Hidden line removal by 'back-siding' and depth sorting.
Tu 10/7	ADDING LIGHT & RENDERING SHADOWS	#1	
Th 10/9	**#3 Geometric Detail**		From 2D to 3D: extrusion, revolution, sweeps & lofting. Hidden-line removal by z-buffering. Symbols: instances *v.* copies
Tu 10/14	DETAILS AND ENTOURAGE	#2	
Th 10/16	**#4 Surface Detail**		Faux geometry: Smoothing (Gouraud/Phong), Texture maps (surface, solid & procedural textures, color, transparency & bump maps).
Tu 10/21	ADDING & CONTROLLING TEXTURES.	#3	
Th 10/23	**#5 Terrain** [brj @ acadia]		Landform modeling: contours, meshes or TINs. Building a site. Abstraction *v.* "reality". (fractals, meshes, randomness and irregularity)
Tu 10/28	MODELING TERRAIN	#4	
Th 10/30	**#6 Complex curvature**		Principles and introduction to modeling of curved objects beyond surfaces of revolution and sweeps. (NURBS, patches, handles, nodes)
Tu 11/4	NURBS, Splines, etc.	#5	
Th 11/6	**#7 Photorealism**		Rendering refraction, diffuse reflection, soft shadows, fog, etc.–. (global illumination, ray tracing, radiosity, physically-based rendering)
Tu 11/11	*(VETERANS DAY HOLIDAY)*		
Th 11/13	Raytracing, Radiosity, & time….	#6	Animation basics. ('tweening, interpolation, paths, previews, codecs) Playback technology bottlenecks, solutions and work-arounds.
Tu 11/18	**#8 Motion**		
Th 11/20	CAMERAS & MODELS IN MOTION		Designing & assembling presentations using video (codecs, transitions, intellectual property rights, audio)
Tu 11/25	**#9 Post-production**	#7	
Th 11/27	*(THANKSGIVING HOLIDAY)*		
Tu 12/2	WORK & POST PRODUCTION	#8	
Th 12/4	Course Wrap-up, Review, Evaluation		Review of course material. Time for feedback.
Tu 12/9	*(FINALS WEEK - NO CLASS THIS DAY)*		
Th 12/11	**Final Show & Tell** **4:30-6:00**	#9	Presentation of completed (Ex 9) projects. Guest critic(s) will be invited. *Attendance is expected.*

Notes:

[†] Assignments are due at the START OF CLASS on the indicated date. These dates and times should be reflected in the Dropbox "due dates".

Revised: 9/17/14

Arch 481 :

3D Modeling and Rendering

Model & Rendering by Roark Congdon, Aut 09.

Exercise Workbook

Autumn 2014

© by Brian Johnson
brj@u.washington.edu

Syllabus

University of Washington
Department of Architecture

Arch 481
3D Modeling and Rendering

ARCH 481: 3d Modeling and Rendering

http://online.caup.washington.edu/courses/arch481

When:	Tu/Th, 1:30 to 2:50	**Where**:	Digital Commons (GLD007F)
Credits:	3, Graded	**Prerequisite**:	Arch 380 (or permission)

Instructor:	Brian Johnson	**GSA**:	Nanching Tai
E-mail to:	brj@u.washington.edu	**E-mail to**:	tai@u.washington.edu
Office:	Condon 622	**Office**:	varies
Office Hours:	By appointment	**Office Hours**:	see web pages
Office phone:	206.543.2132	**Office phone**:	n/a

This is a course about Modeling and Rendering with computer graphics tools. There are two important things to keep in mind as you start this study, and they are encapsulated by these two quotes:

> *What remains hard is modeling. The structure inherent in three-dimensional models is difficult for people to grasp and difficult too for user interfaces to reveal and manipulate. Only the determined model three-dimensional objects, and they rarely invent a shape at the computer, but only record a shape so that analysis of manufacturing can proceed. The grand challenges to three-dimensional graphics are to make simple modeling easy and make complex modeling accessible to far more people.*
>
> *-Robert Sproull,*
> *keynote speech, SIGGRAPH 1990*

Robert Sproull is one of the "founding fathers" of computer graphics. The "Special Interest Group on Graphics" (SIGGRAPH) annual conference is the biggest computer graphics conference in the world. This guy knew what he was talking about in 1990, and it's still true today.

> *A journey of a thousand miles must begin with a single step.*
> *-Lao-tzu*

Lao-tzu was a Chinese philosopher. China is a big country. He knows what he's talking about too.

Introduction

This course is about *beginnings*, about preparation for the use of 3D computing during your professional career. It is not a course about a particular software package. You may not use the same programs in a few months or years as you use this quarter, not because they aren't good, but because change happens. From this point of view it is not so important what programs or commands you learn as it is what you learn from using those programs and commands. On the other hand, it's very hard to learn about this subject purely as "book learning". You need to *do* modeling and rendering as well as learn the concepts. The exercises and lectures are intended to work together to provide an enriched "soil" in which your professional experiences can flourish.

This is a fast-paced, "learning by doing" lecture & laboratory course. Each week lectures and demonstrations presented during the class sessions will prepare you to undertake a related modeling and/or rendering exercise. Background reference and online readings will explain the technical issues surrounding the exercises.

The weekly exercise is a very important element of the course. The lectures support and explain the exercises by providing material that may not be available in the readings and by providing an opportunity to ask questions. However, completing the exercise provides most of the actual learning experience. The exercises take time (5-10 hours/week) to complete and write up. That time must generally be spent on campus. In addition, most of the help is available during the day and early evening. Please consider these things as you set up your schedule for the quarter.

26.IX.06 1

Course description

Contact information

Grading

Resources

Expectations

Schedule

ARCH 481: 3D Modeling and Rendering — Autumn Quarter, 2014

Lecture & Project Schedule

Exercise Name

Weekly topics

Due Dates

Dates	**LECTURE TOPIC**/Lab Subject	Due †	**Weekly topic cluster**
Th 9/25	**#1 Concepts**		3D modeling data and operations; Solids (geometry + topology); Mouse input. "Cameras:" projections from 3D to 2D, saving 2D images.
Tu 9/30	Concept Models	#0	
Th 10/2	**#2 Lights + Shading**		Boolean operations. Lights, Shading, Shadows. Hidden line removal by 'back-siding' and depth sorting.
Tu 10/7	Adding Light & Rendering Shadows	#1	
Th 10/9	**#3 Geometric Detail**		From 2D to 3D: extrusion, revolution, sweeps & lofting. Hidden-line removal by z-buffering. Symbols: instances *v.* copies
Tu 10/14	Details and Entourage	#2	
Th 10/16	**#4 Surface Detail**		Faux geometry: Smoothing (Gouraud/Phong), Texture maps (surface, solid & procedural textures, color, transparency & bump maps).
Tu 10/21	Adding & Controlling Textures.	#3	
Th 10/23	**#5 Terrain** [brj @ acadia]		Landform modeling: contours, meshes or TINs. Building a site. Abstraction *v.* "reality". (fractals, meshes, randomness and irregularity)
Tu 10/28	Modeling Terrain	#4	
Th 10/30	**#6 Complex curvature**		Principles and introduction to modeling of curved objects beyond surfaces of revolution and sweeps. (NURBS, patches, handles, nodes)
Tu 11/4	NURBS, Splines, etc.	#5	
Th 11/6	**#7 Photorealism**		Rendering refraction, diffuse reflection, soft shadows, fog, etc.–. (global illumination, ray tracing, radiosity, physically-based rendering)
Tu 11/11	*(Veterans Day Holiday)*		
Th 11/13	Raytracing, Radiosity, & time.…	#6	Animation basics. ('tweening, interpolation, paths, previews, codecs) Playback technology bottlenecks, solutions and work-arounds.
Tu 11/18	**#8 Motion**		
Th 11/20	Cameras & Models in motion		Designing & assembling presentations using video (codecs, transitions, intellectual property rights, audio)
Tu 11/25	**#9 Post-production**	#7	

Workbook

Arch 481 :

3D Modeling and Rendering

Model & Rendering by Roark Congdon, Aut 09.

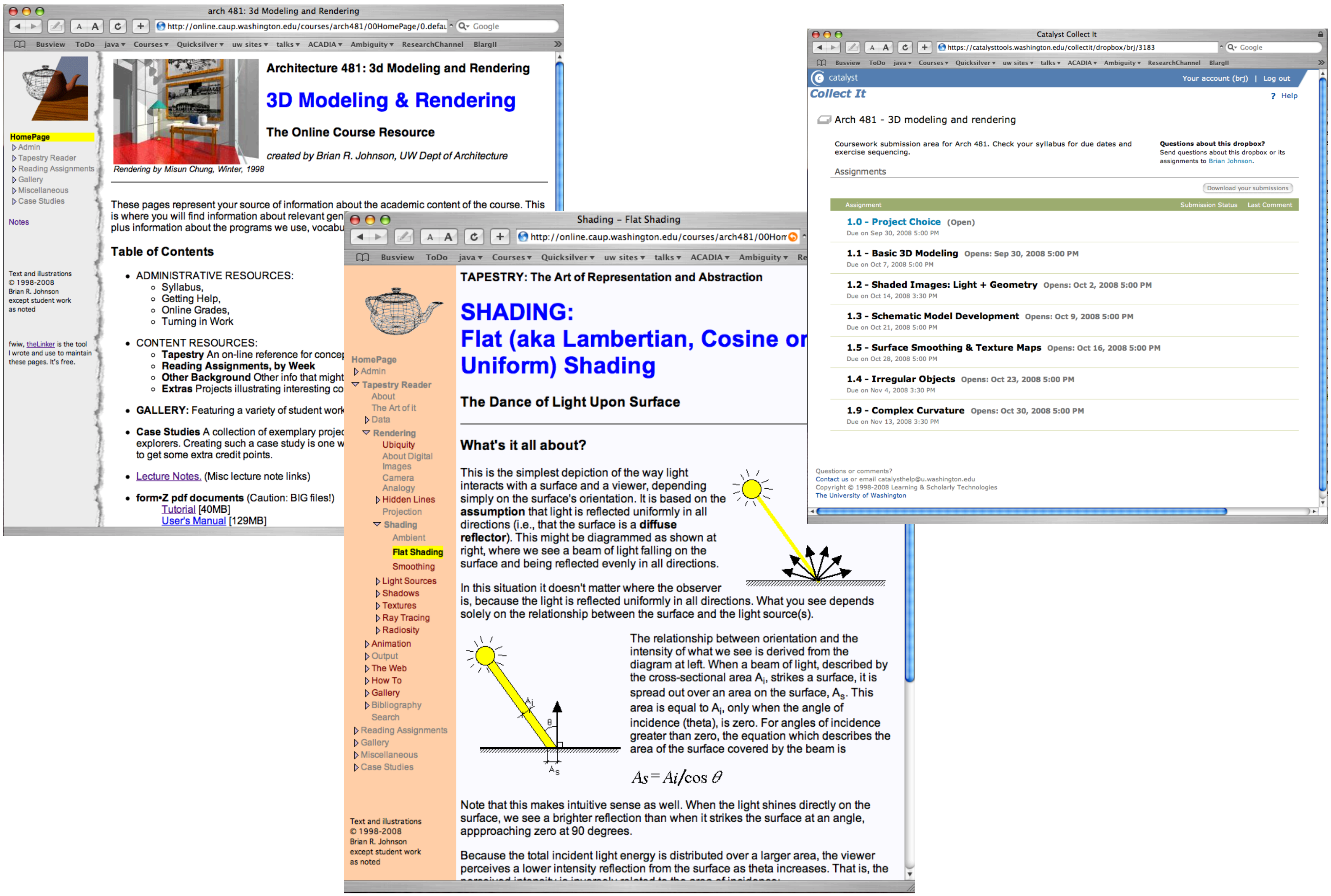
Exercise Workbook

Autumn 2014

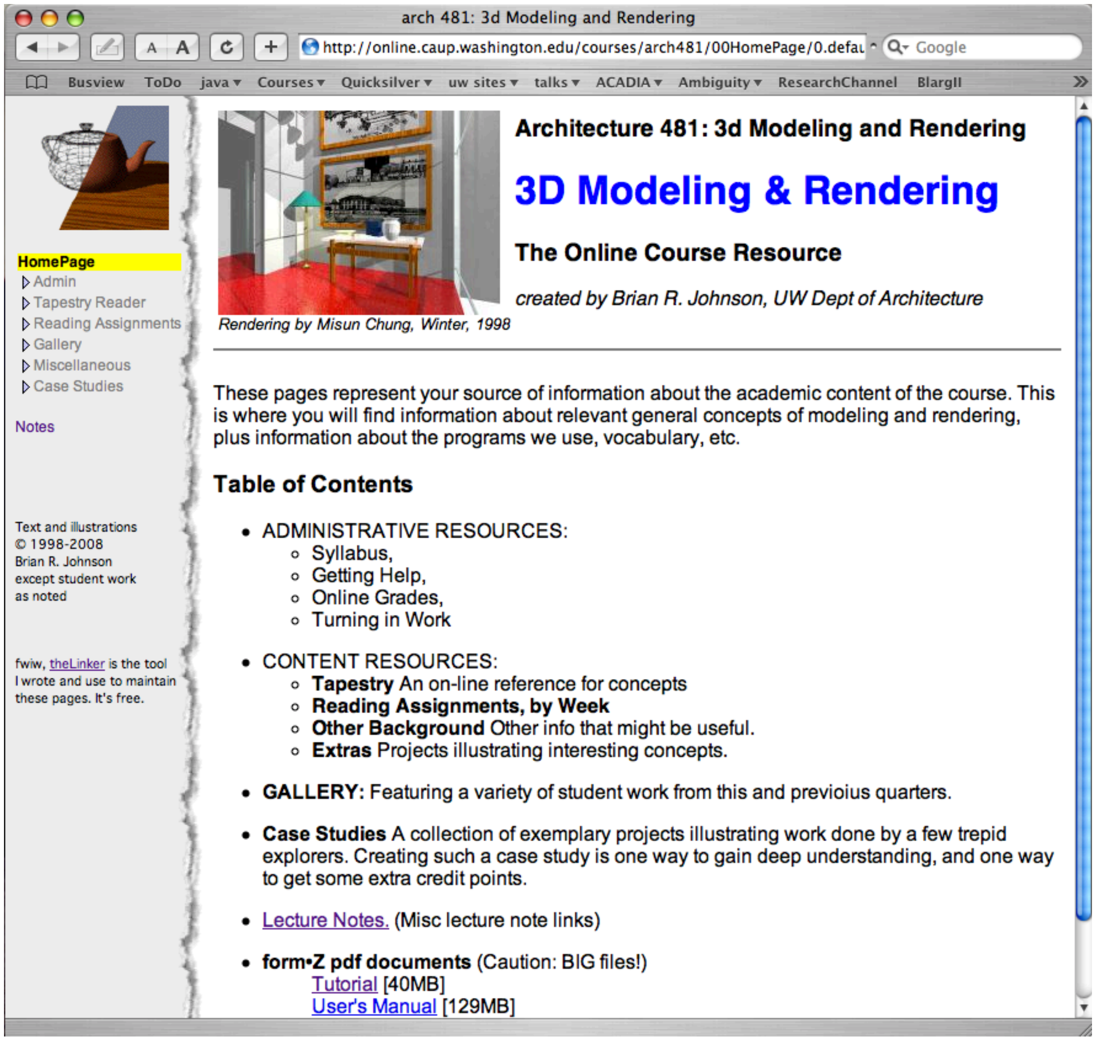
brj@u.washington.edu

- Exercises by Name & #
- Learning Goals
- Production Goals
- Instructions
- Hints
- Write-up Questions
- Extra-credit opportunities

Web pages


arch 481: 3d Modeling and Rendering
http://online.caup.washington.edu/courses/arch481/00HomePage/0.defau
Architecture 481: 3d Modeling and Rendering
3D Modeling & Rendering
The Online Course Resource
created by Brian R. Johnson, UW Dept of Architecture
Rendering by Misun Chung, Winter, 1998
HomePage
Admin
Tapestry Reader
Reading Assignments
Gallery
Miscellaneous
Case Studies
Notes
Text and illustrations © 1998-2008 Brian R. Johnson except student work as noted
These pages represent your source of information about the academic content of the course. This is where you will find information about relevant gen
plus information about the programs we use, vocabu
Table of Contents
ADMINISTRATIVE RESOURCES:
Syllabus,
Getting Help,
Online Grades,
Turning in Work
CONTENT RESOURCES:
Tapestry An on-line reference for concep
Reading Assignments, by Week
Other Background Other info that might
Extras Projects illustrating interesting co
GALLERY: Featuring a variety of student work
Case Studies A collection of exemplary projec explorers. Creating such a case study is one w to get some extra credit points.
Lecture Notes. (Misc lecture note links)
form•Z pdf documents (Caution: BIG files!)
Tutorial [40MB]
User's Manual [129MB]
Shading – Flat Shading
http://online.caup.washington.edu/courses/arch481/00Hom
TAPESTRY: The Art of Representation and Abstraction
SHADING:
Flat (aka Lambertian, Cosine or Uniform) Shading
The Dance of Light Upon Surface
What's it all about?
This is the simplest depiction of the way light interacts with a surface and a viewer, depending simply on the surface's orientation. It is based on the assumption that light is reflected uniformly in all directions (i.e., that the surface is a diffuse reflector). This might be diagrammed as shown at right, where we see a beam of light falling on the surface and being reflected evenly in all directions.
In this situation it doesn't matter where the observer is, because the light is reflected uniformly in all directions. What you see depends solely on the relationship between the surface and the light source(s).
The relationship between orientation and the intensity of what we see is derived from the diagram at left. When a beam of light, described by the cross-sectional area Ai, strikes a surface, it is spread out over an area on the surface, As. This area is equal to Ai, only when the angle of incidence (theta), is zero. For angles of incidence greater than zero, the equation which describes the area of the surface covered by the beam is
As = Ai/cos θ
Note that this makes intuitive sense as well. When the light shines directly on the surface, we see a brighter reflection than when it strikes the surface at an angle, appproaching zero at 90 degrees.
Because the total incident light energy is distributed over a larger area, the viewer perceives a lower intensity reflection from the surface as theta increases. That is, the
Tapestry Reader
Rendering
Shading
Ambient
Flat Shading
Smoothing
Light Sources
Shadows
Textures
Ray Tracing
Radiosity
Animation
Output
The Web
How To
Bibliography
Search
Catalyst Collect It
https://catalysttools.washington.edu/collectit/dropbox/brj/3183
Collect It
Arch 481 - 3D modeling and rendering
Coursework submission area for Arch 481. Check your syllabus for due dates and exercise sequencing.
Questions about this dropbox?
Assignments
1.0 - Project Choice (Open)
1.1 - Basic 3D Modeling Opens: Sep 30, 2008 5:00 PM
1.2 - Shaded Images: Light + Geometry Opens: Oct 2, 2008 5:00 PM
1.3 - Schematic Model Development Opens: Oct 9, 2008 5:00 PM
1.5 - Surface Smoothing & Texture Maps Opens: Oct 16, 2008 5:00 PM
1.4 - Irregular Objects Opens: Oct 23, 2008 5:00 PM
1.9 - Complex Curvature Opens: Oct 30, 2008 5:00 PM

Course web site



Copies of Handouts

Reading Assignments

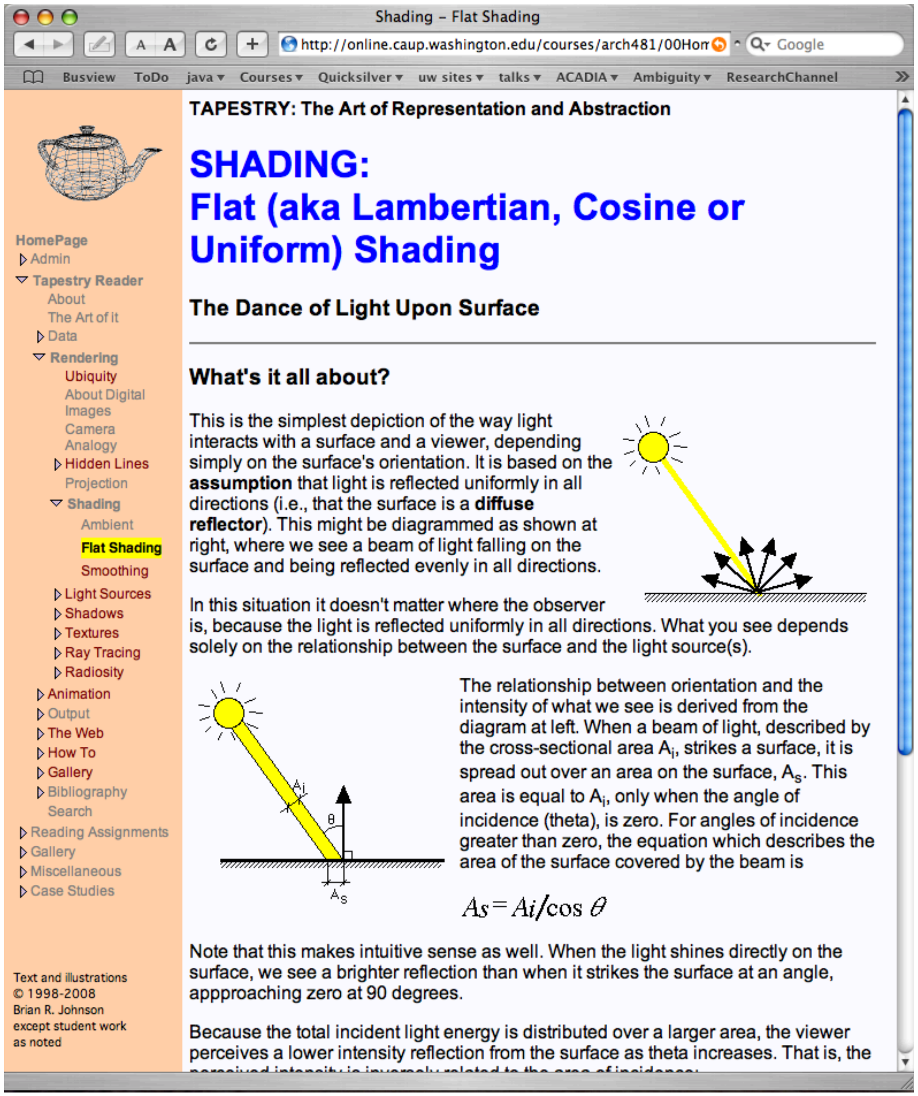
Tapestry Reader

Gallery of work

Misc. resources and links

Case Studies

Tapestry Reader



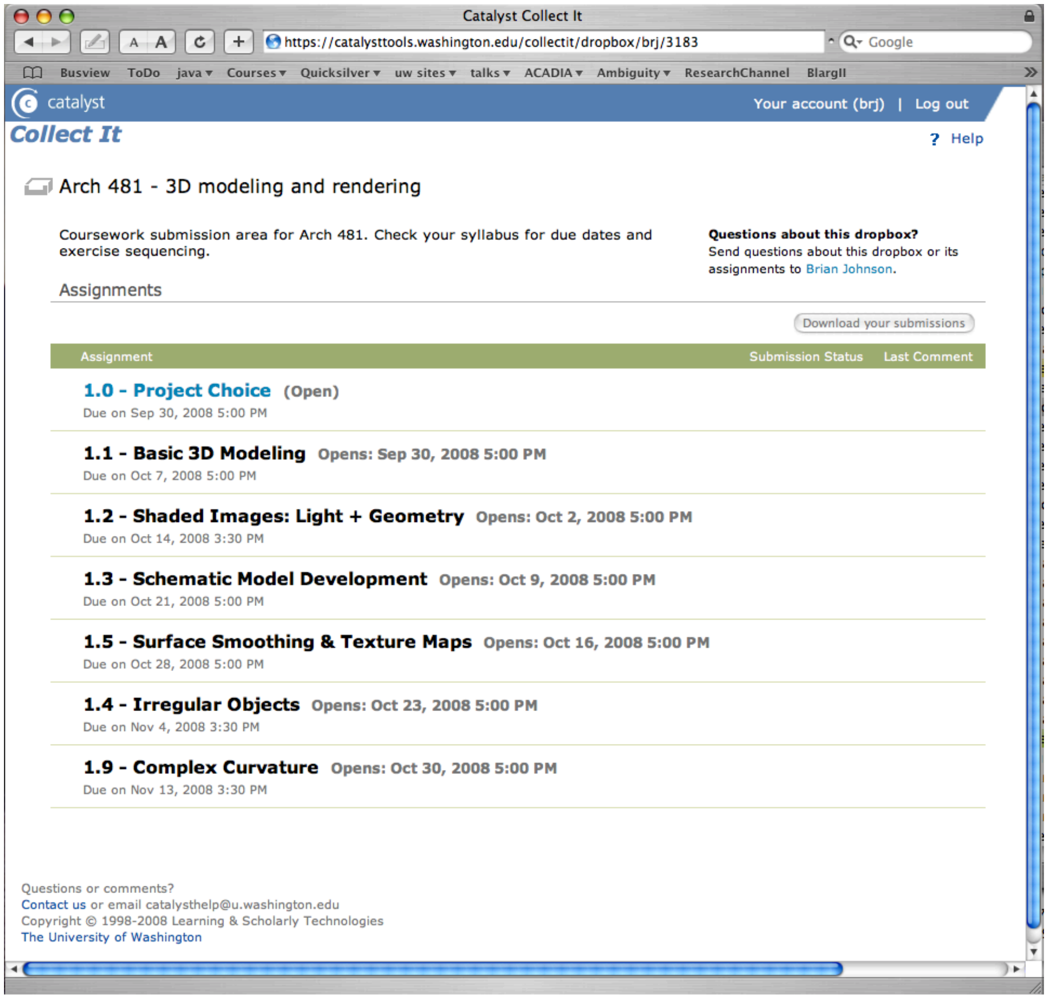
Hierarchical by topic

Graphic/Animated

Vocabulary

“How To” pages

Turning in completed work



Web-based

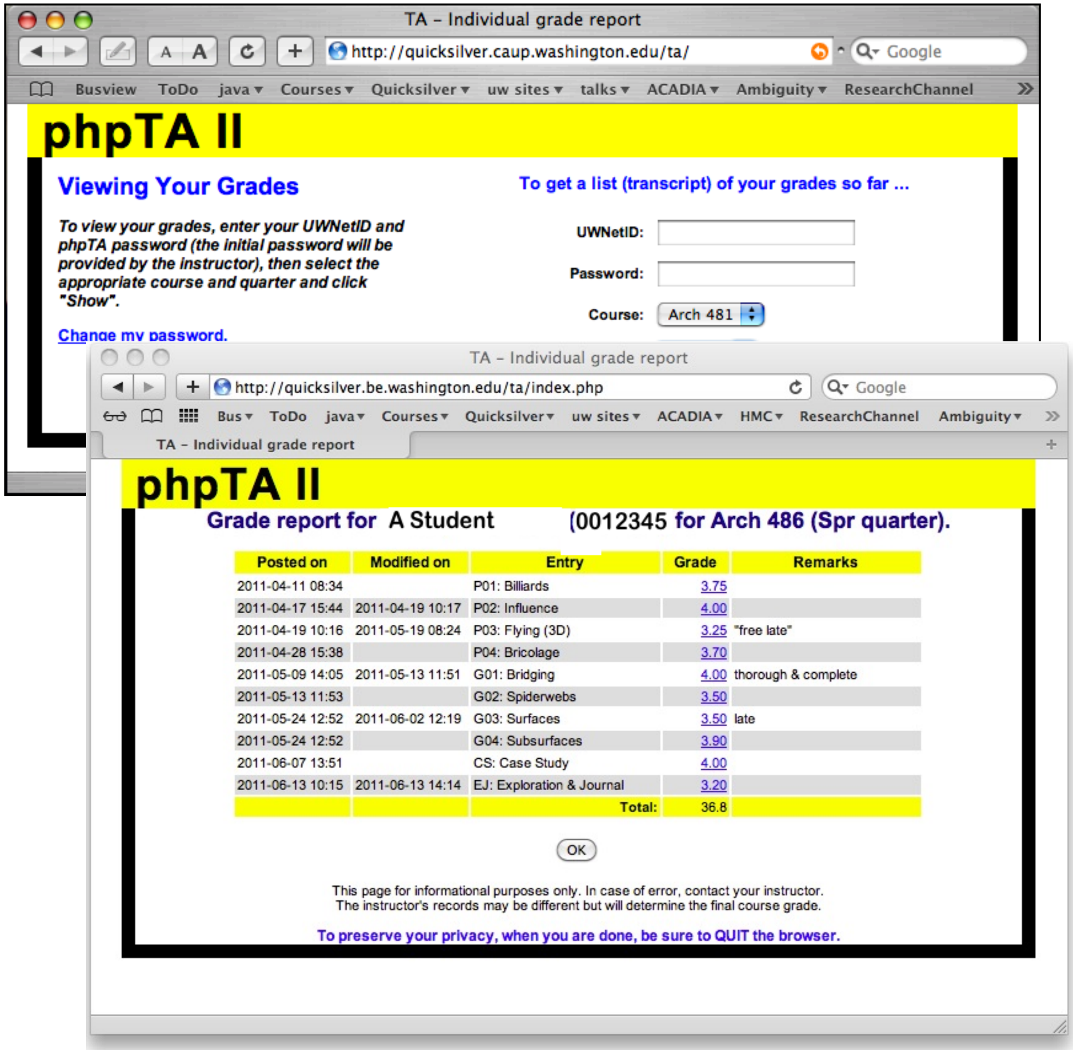
Linked from Course "Admin" area

Exercises "open" and "close" by dates.

An experiment--give feedback

Don't forget hardcopy of Write-up questions!

PhpTA (online grade report)



Uses "pubcookie" (i.e. your UWNetID name and password).

Is approximate--i.e. **not** the final record--(doesn't show late penalty deductions)

http://www.traveladventures.org/

Beginnings…

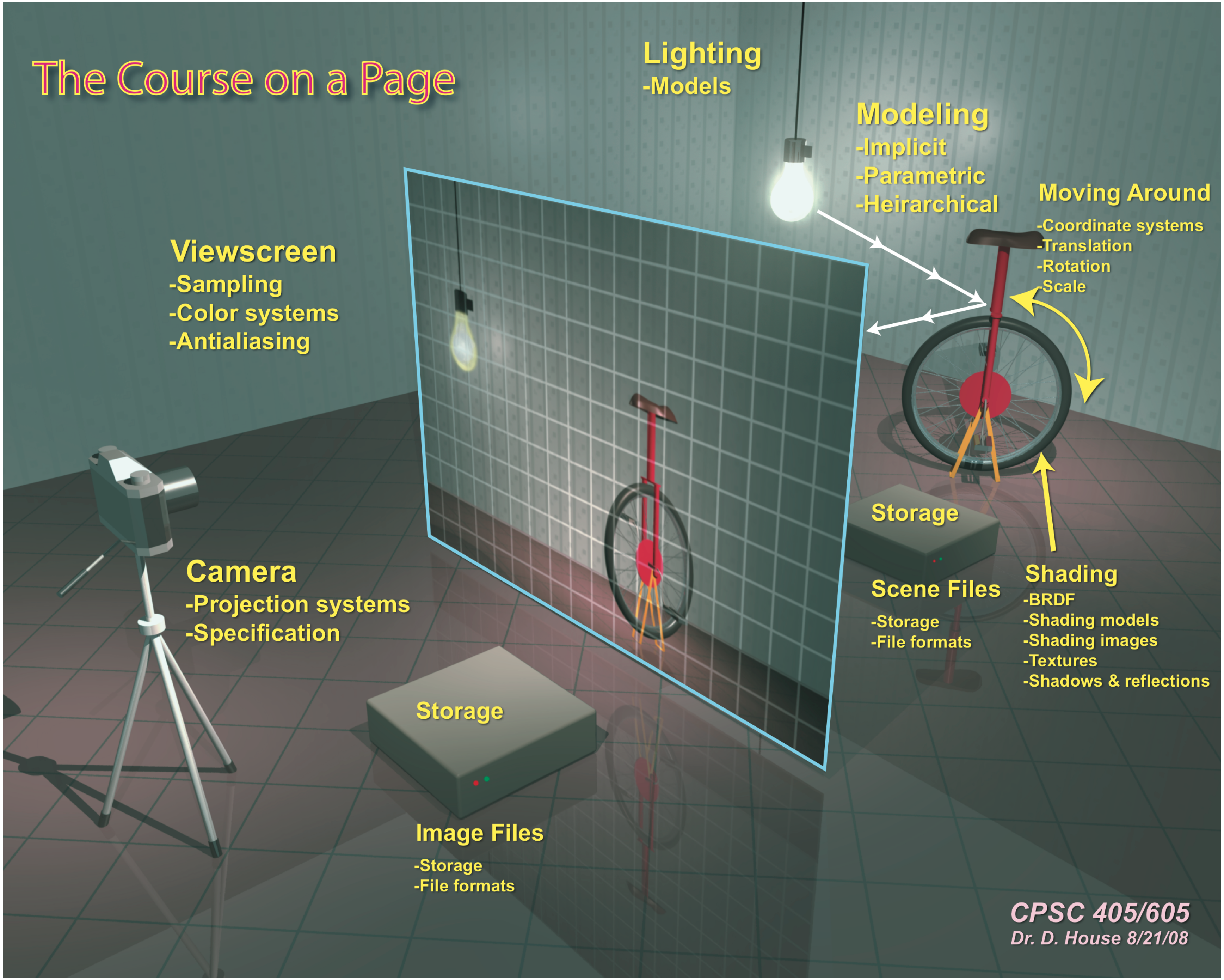
Destination: *unknown*

Path: *uneven*

Preparation: *some*

Motivation: *high*

When: *now!*


The Course on a Page
Lighting
-Models
Modeling
-Implicit
-Parametric
-Heirarchical
Moving Around
-Coordinate systems
-Translation
-Rotation
-Scale
Viewscreen
-Sampling
-Color systems
-Antialiasing
Camera
-Projection systems
-Specification
Storage
Scene Files
-Storage
-File formats
Shading
-BRDF
-Shading models
-Shading images
-Textures
-Shadows & reflections
Storage
Image Files
-Storage
-File formats
CPSC 405/605
Dr. D. House 8/21/08

storytelling

Images have meaning (“worth 1000 words”)

Rendering is turning data into image.

Data becomes image → Data is subjective.

You are <u>always</u> telling a story.

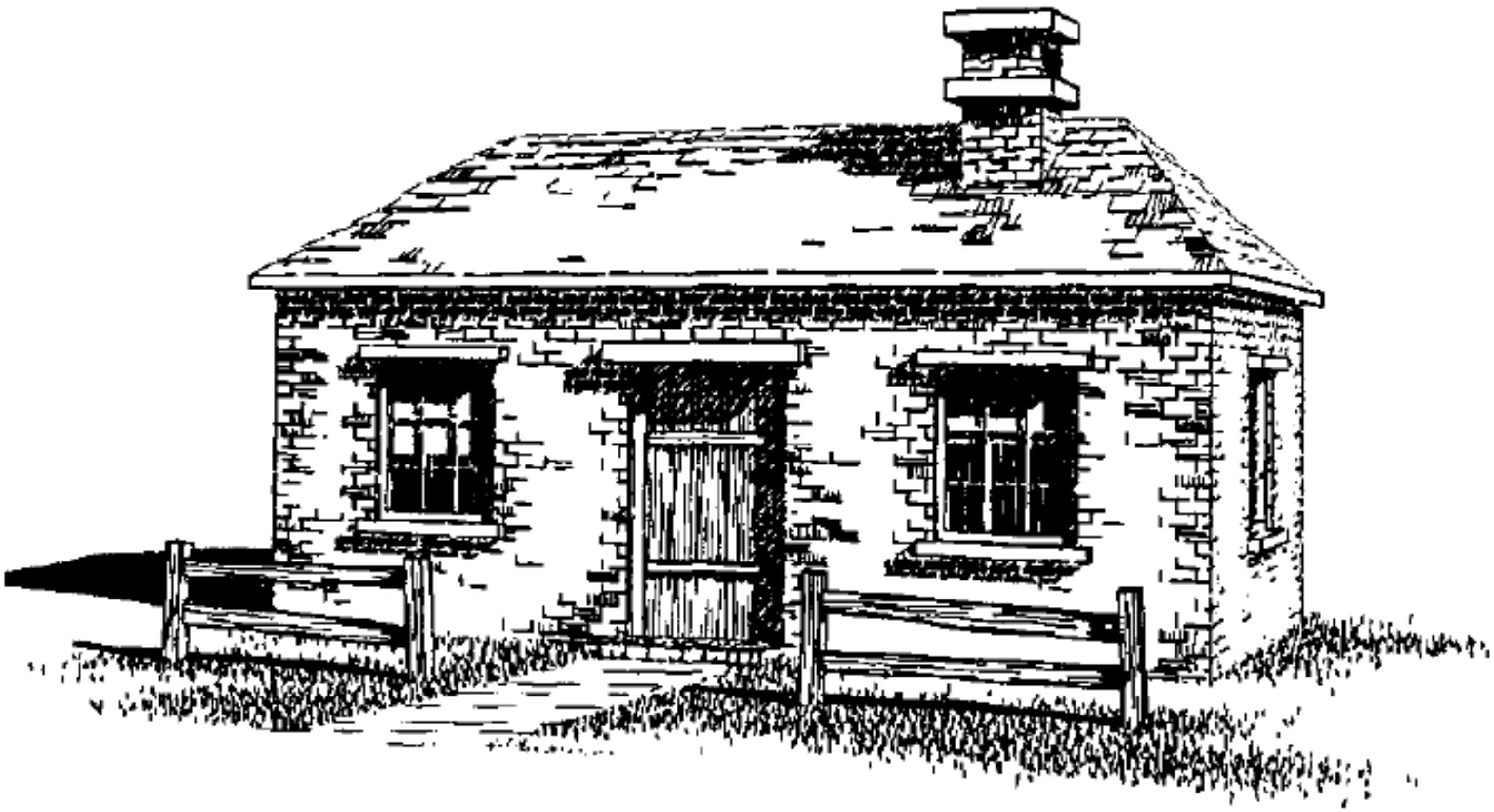
Think about the story you will tell.

Control the story you tell.

photo real

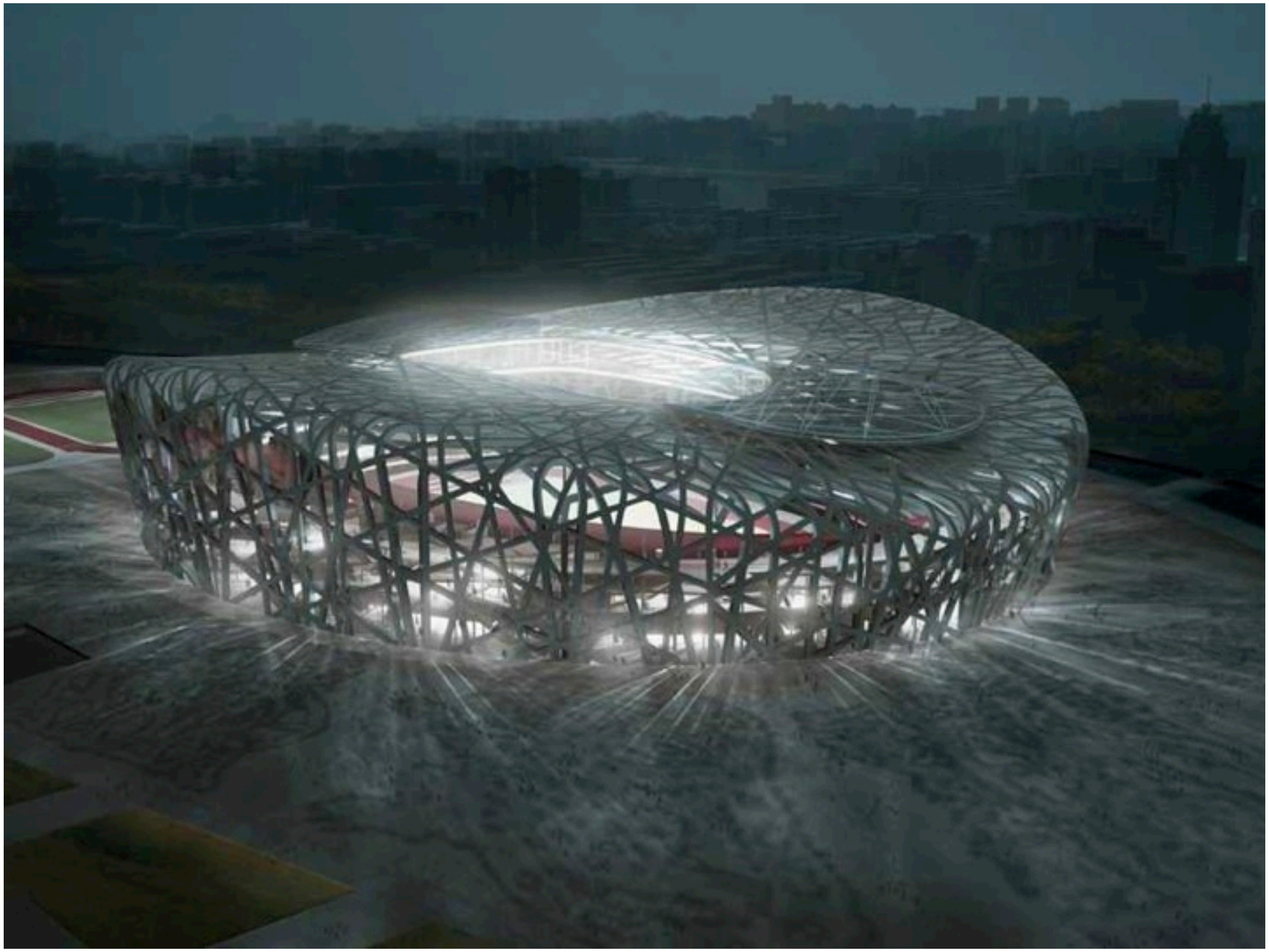
John Stout, Arch 481, Aut 2009

Non-photo-real

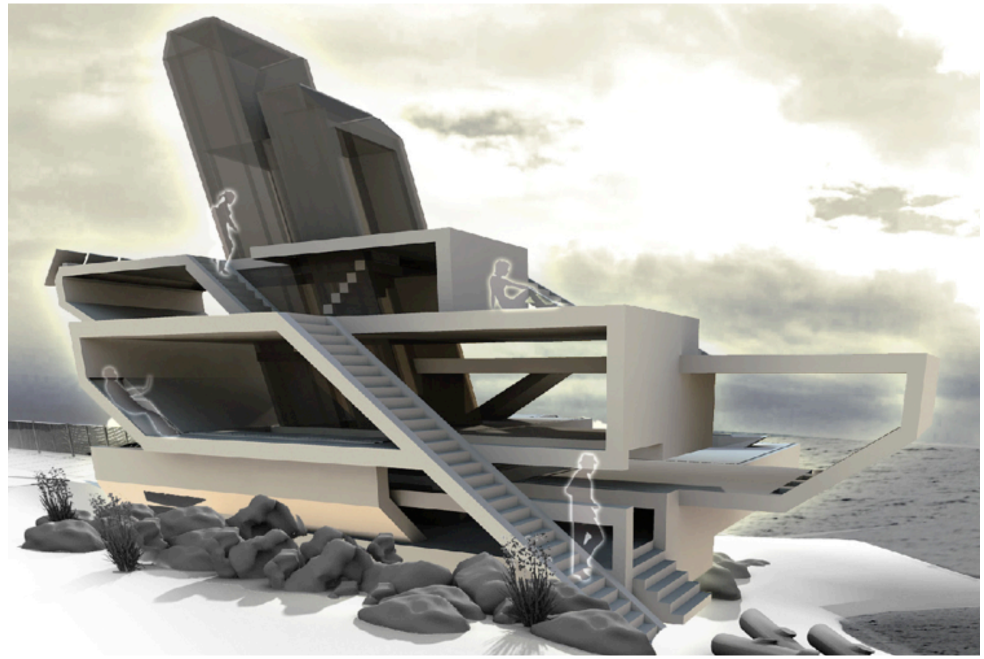


Winkenbach & Salesin, SIGGRAPH '94

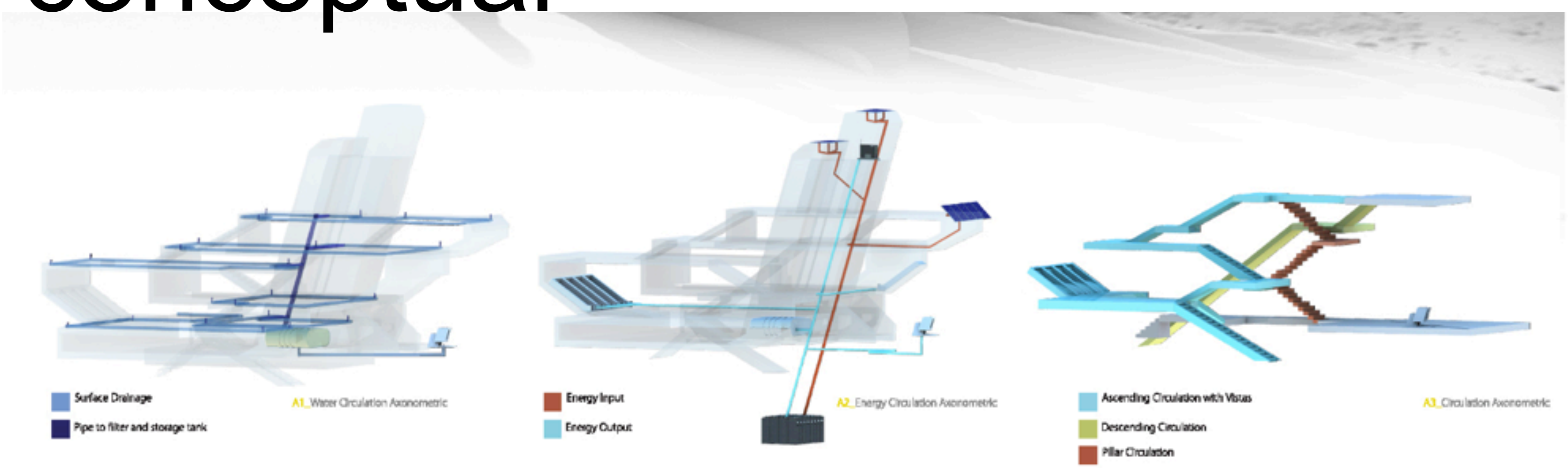
hyper real



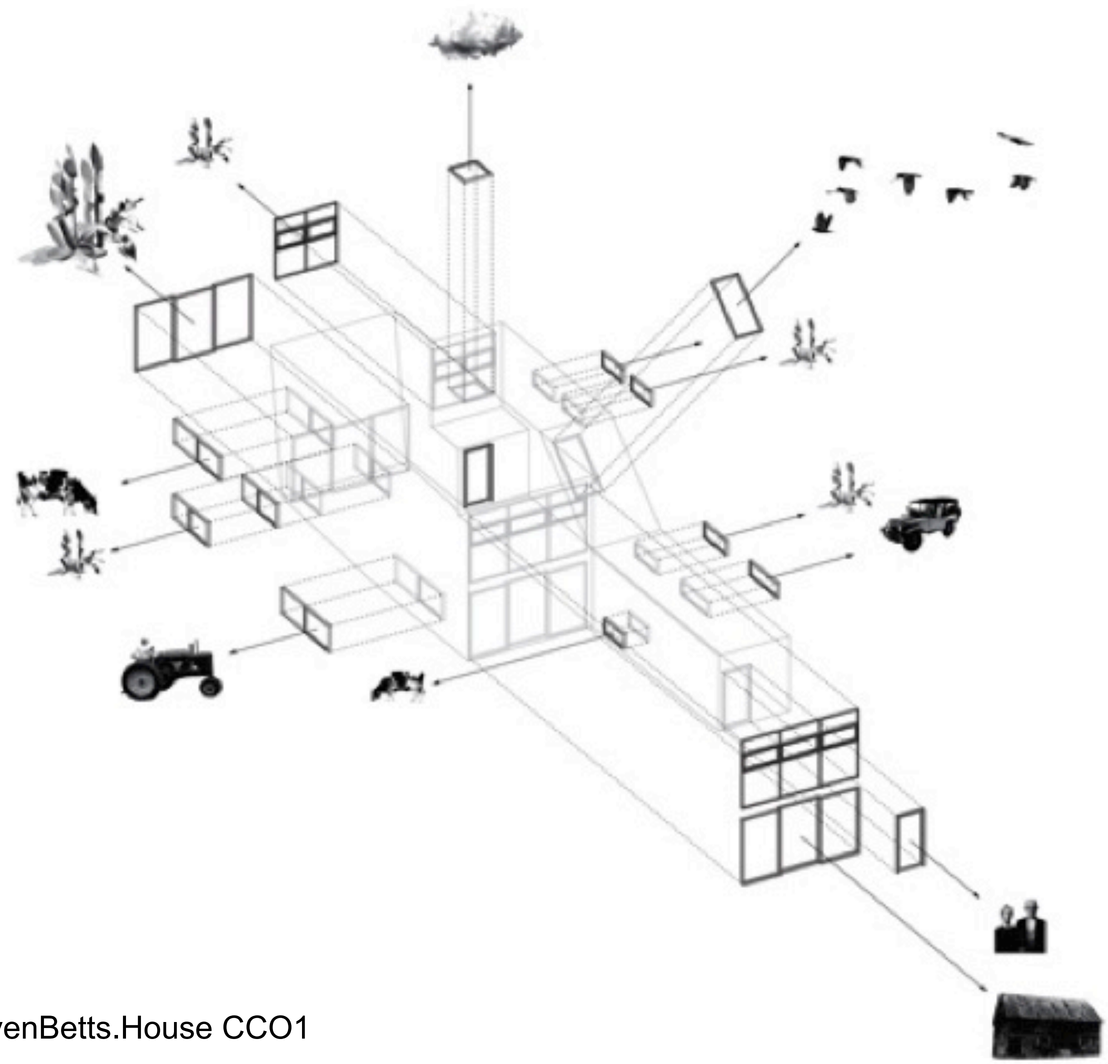
visual



conceptual



relationships



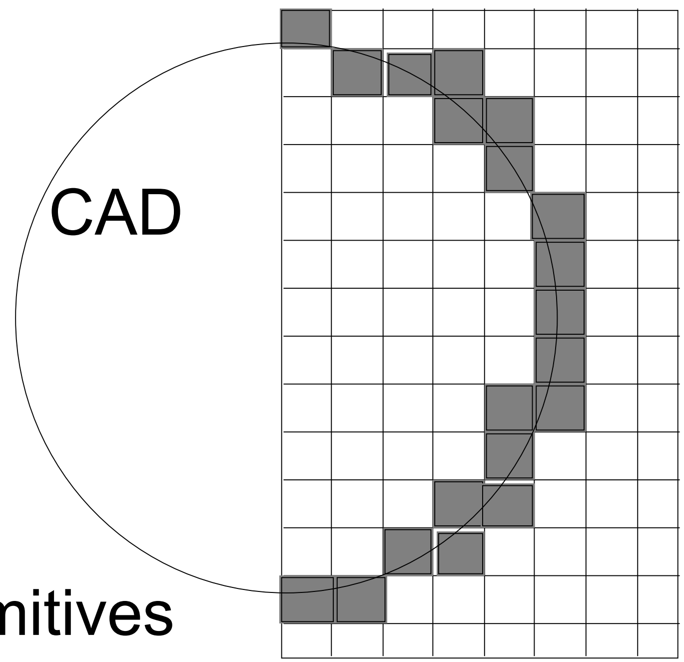
LevenBetts.House CCO1

Key Concepts Review

Object

Photoshop

.WRL



pixel

.JPG

drawing

painting

.DWG

primitives

lossy

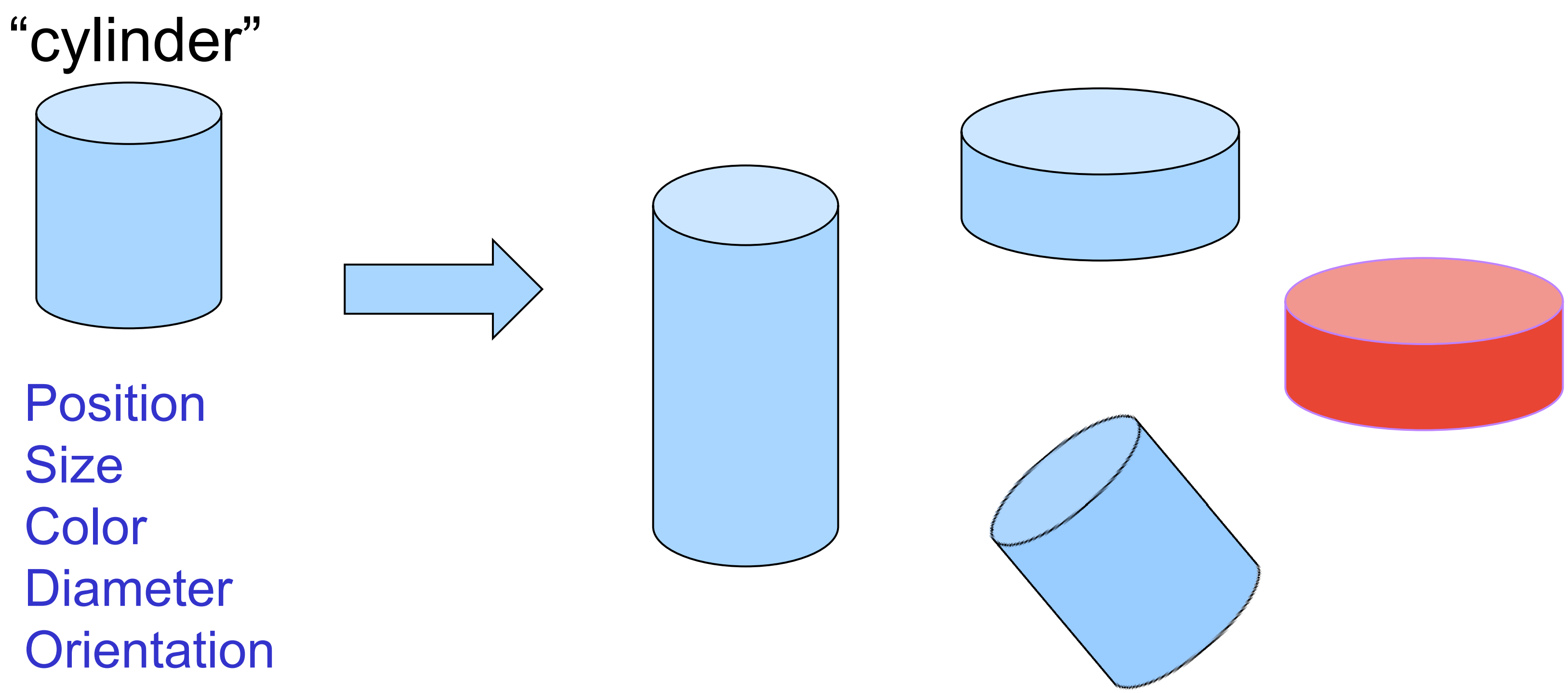
fax

attributes

Vector v. Raster

Primitives & Attributes: Instances

primitive **Instance variations**

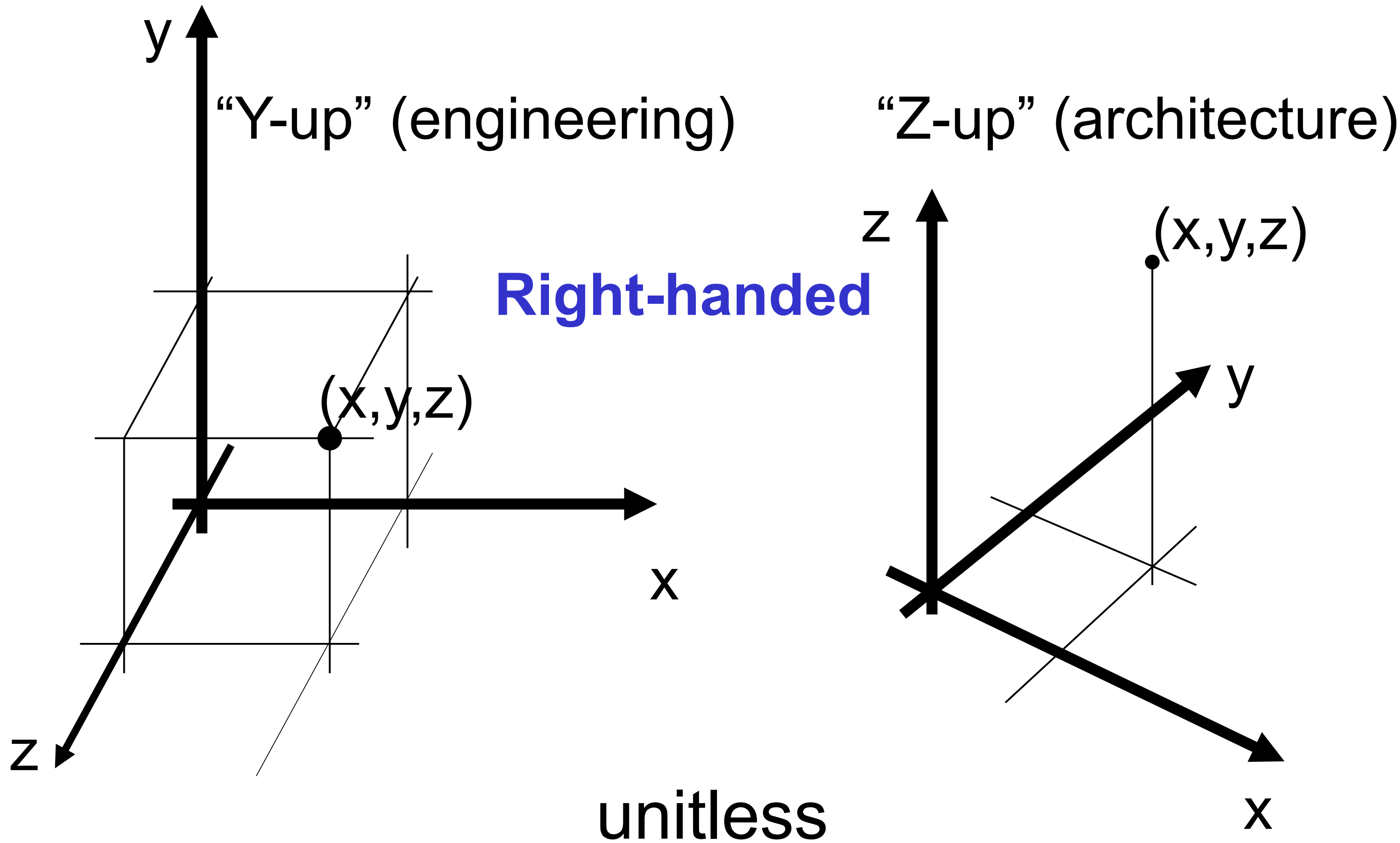


Primitives vs. pseudo-primitives

Attributes of primitives

- Varies by program
- Varies by primitive
- In Rhino …
  - Object Type
  - Name
  - Layer
    - Name
    - Visibility
    - Color
    - Line-color
    - (more)
  - Color
  - Visibility
  - Render mesh setting
  - Shadow casting/receiving
  - Texture/Material
  - Texture Mapping
  - Decals
  - Control points
  - Degree
  - Coordinates (points)
  - Analytic Properties
    - Planarity
    - Area & volume
    - Orientation/direction
- NOT (e.g. plane)

Coordinate Systems



The Pointing Problem

Mouse
2D

Model
3D

The Pointing Problem

Solutions:

Keyboard coordinates
Construction planes
Reference existing data
Multi-view input (multiple clicks – rare)

The influence extends to many 3D modeling operations that depend on 2D input (extrusion, revolution, sweeps, etc.)

Mouse is 2D -> Model is 2D?

2D drawings are ambiguous

- 2D *views* often combine data from multiple 3D *positions*

Making 3D Model with 2D points is hard

- Simply "clicking points" (raw pointing) places points on a plane
- Ortho, (grid) snap, etc. constrain point locations.

Find the right view

Multi-views

- Find the planar “control curve”
- Pick or construct a cplane for the curve

3D views

- “Disambiguate” snaps

Named views

- Camera info, not image, so re-rendering works

3D: right view, right osnap

Right view (no ambiguity)

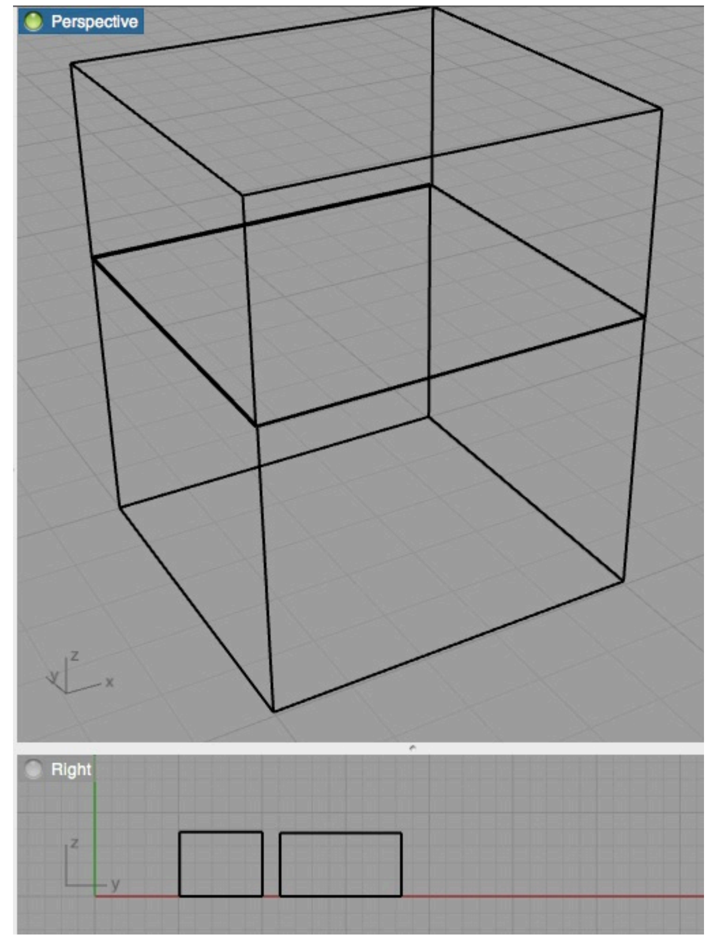
- Plan views have problems

Without osnap (at right)

- Stacked? Or not?

Osnap takes precedence

- Using osnaps, placing objects in perspective is usually quick and easy



Avoid “Clicks” + “Osnaps”

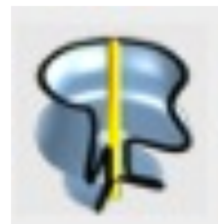
- Mixing Raw & Osnap points may be a bad idea
  - Raw pointing will reference points on a construction plane (cplane)
  - Using Osnaps you can make twisted shapes
  - Osnaps in ortho views can be ambiguous
- **To the rescue!**
- Rhino has “Project” (to cplane) option
- Rhino has “Planar” (temp cplane at first pt) option

Derivative Geometry

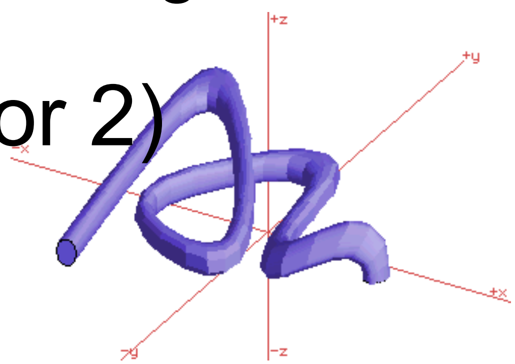
Arch 481

3D shapes, 2D “sections”

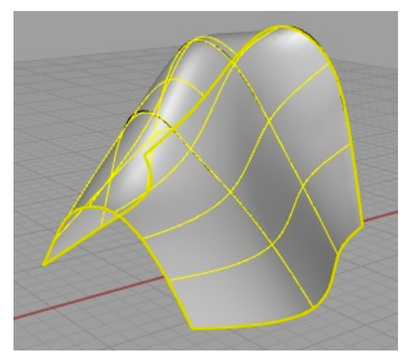
- Extrusion 
  - Section & displacement



- Revolution
  - Section, Axis & angle



- Sweep (1 rail or 2)
  - Section & path



- Lofting
  - Multiple sections (aka contours!)

The Camera Metaphor

In 3D programs the data is separate from the image. You produce images by “rendering” the data. The relationship of model data to rendered image is governed, in part, by the digital “camera” and its attributes, just as in the real world.

“Pinhole” Camera?

Most synthetic images mimic a pinhole camera:

- Perfect focus
- Infinite depth of field
- No spherical aberration
- No motion blur
- No lens-flare

Telephoto *v.* wide-angle lens

The mathematics of "projection":

- Projection refers to "squashing" 3D data to 2D
- In the computer, it's done with matrix math
- By varying the terms in the matrix, everything from a wide-angle view to a telephoto view can be generated. Normal "perspective" and "parallel" views are just variations.
- Usually, *points* are projected, and connected with straight *lines* in the image (no "fish eye")

Camera "View Vector"

A line from the camera to the center of view

Station Point

- Location (x,y,z) in space of the camera
- "z-up" direction

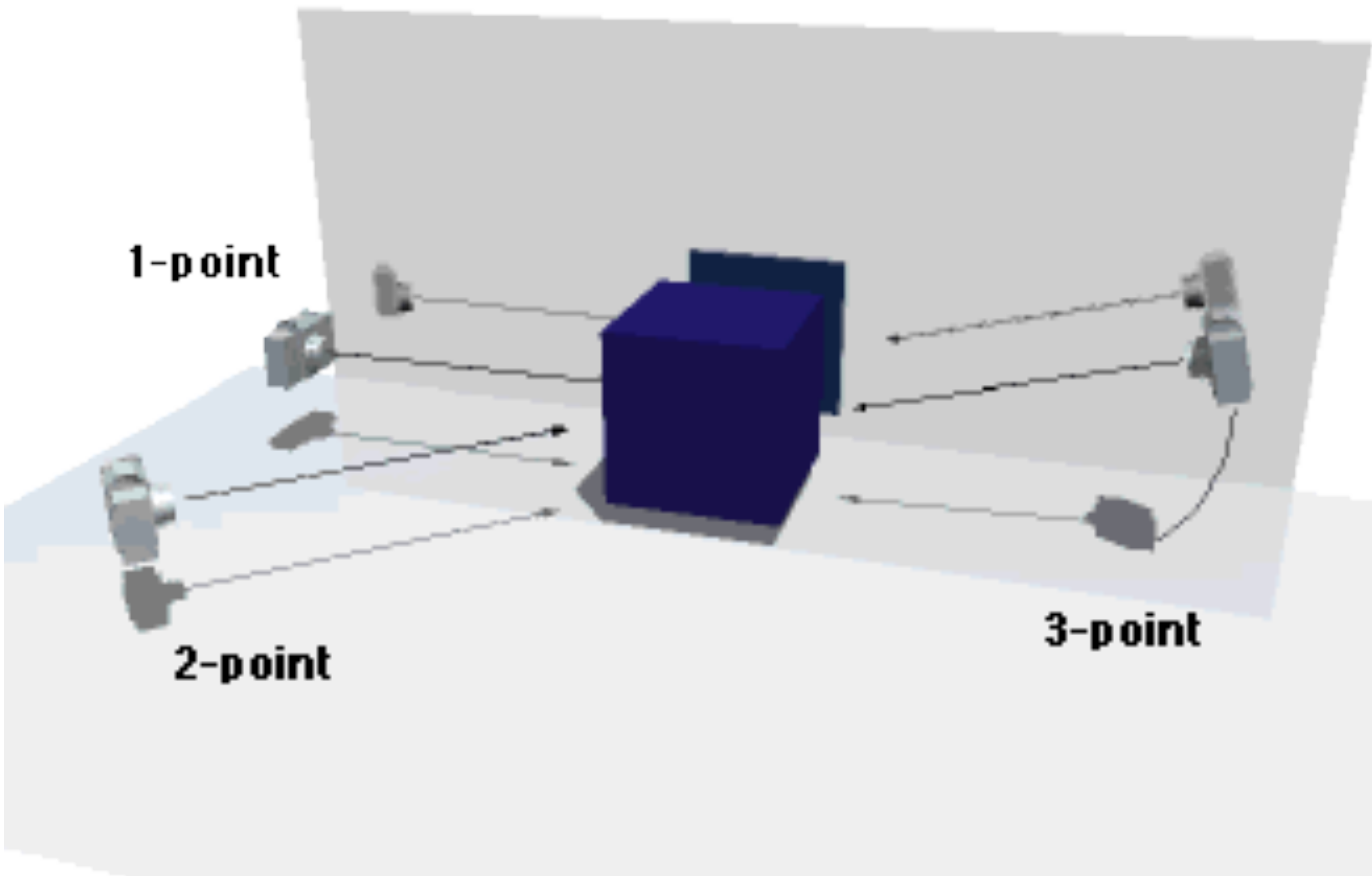
Focal Point

- Location (x,y,z) of the "center of view"

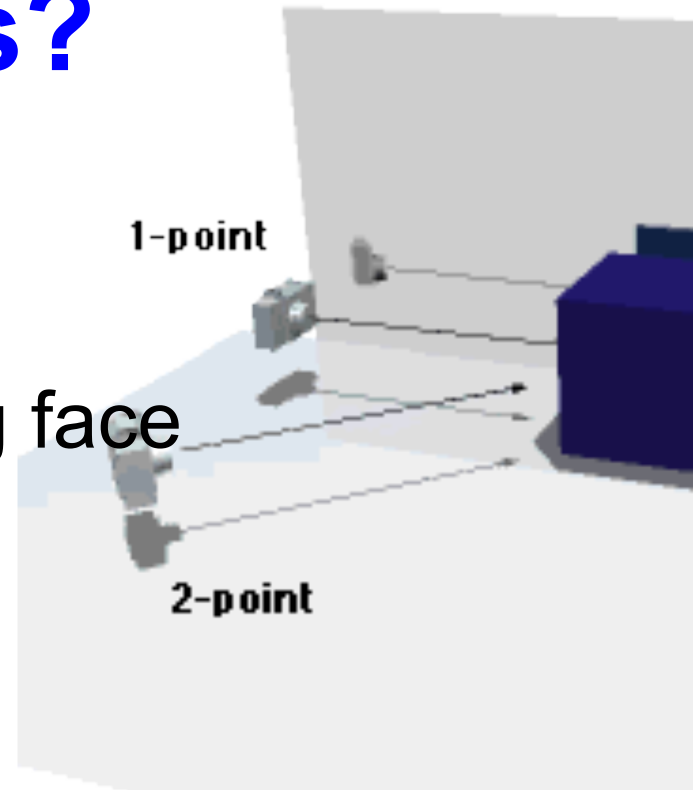
Cone of Vision

- Angle of view frustrum (perspective)
- Sets zoom/scale of window (ortho)

Traditional Views?



Traditional Views?



"One Point"

- VV perpendicular to building face

"Two Point"

- VV parallel to ground

"Three Point"

- VV unconstrained

Elevation, etc.

- One-point, with parallel projection

Symbols / Blocks

Arch 481

Most 3D assemblies repeat parts

Avoid copy & paste!

Best Practice

- Construct just one!
- Duplicate through instancing
  - Separate model?
  - Named construction planes?

Different Kinds of Symbols

- **Full Size**
  - Manufactured items (cars, water-closets)
  - Limited size variation
- **Unit Size**
  - Variable use size (trees)
  - Insertion scale sets size
- **Insertion Point**
  - Placement handle, might be snapped
  - Can simplify placement

Symbols: Copies *v.* Instances

Copies (copy & paste, “copy”, etc.)

- Repeated geometry = repeated data
- Manual change propagation

Instances (blocks, symbols, etc.)

- Name subdrawings
- Repeated geometry = repeated name references
- Automatic change propagation

Symbols: Attributes

Original geometry attributes

- Location relative to global origin
- Layer, color, linestyle (“object attributes”)

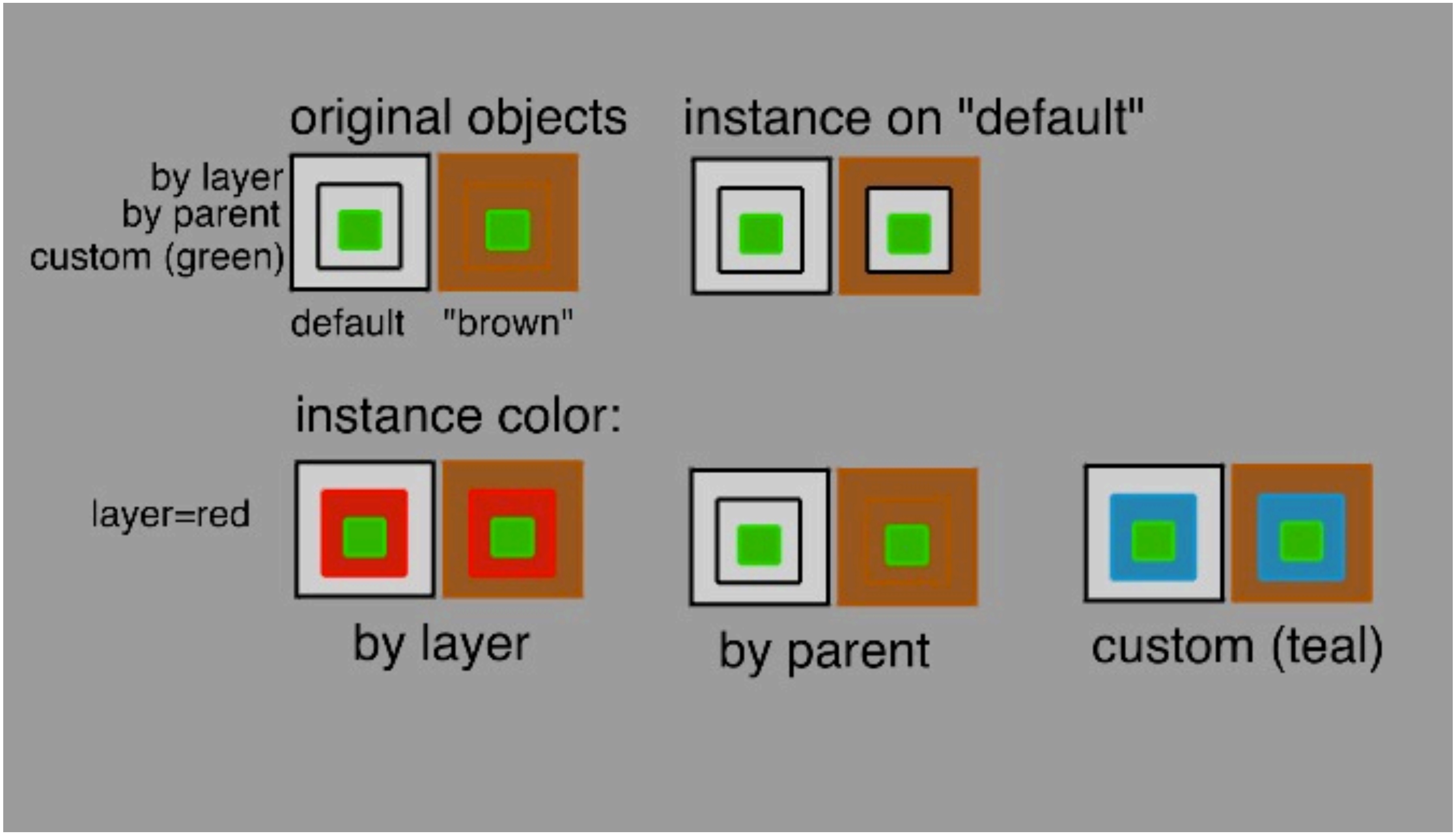
Instance attributes

- Each instance ***is an object*** in it’s own right

Predicting appearance … can be a challenge

- Interaction of original attributes & instance attributes

Symbols: Instance Attributes



Blocks vs. Files

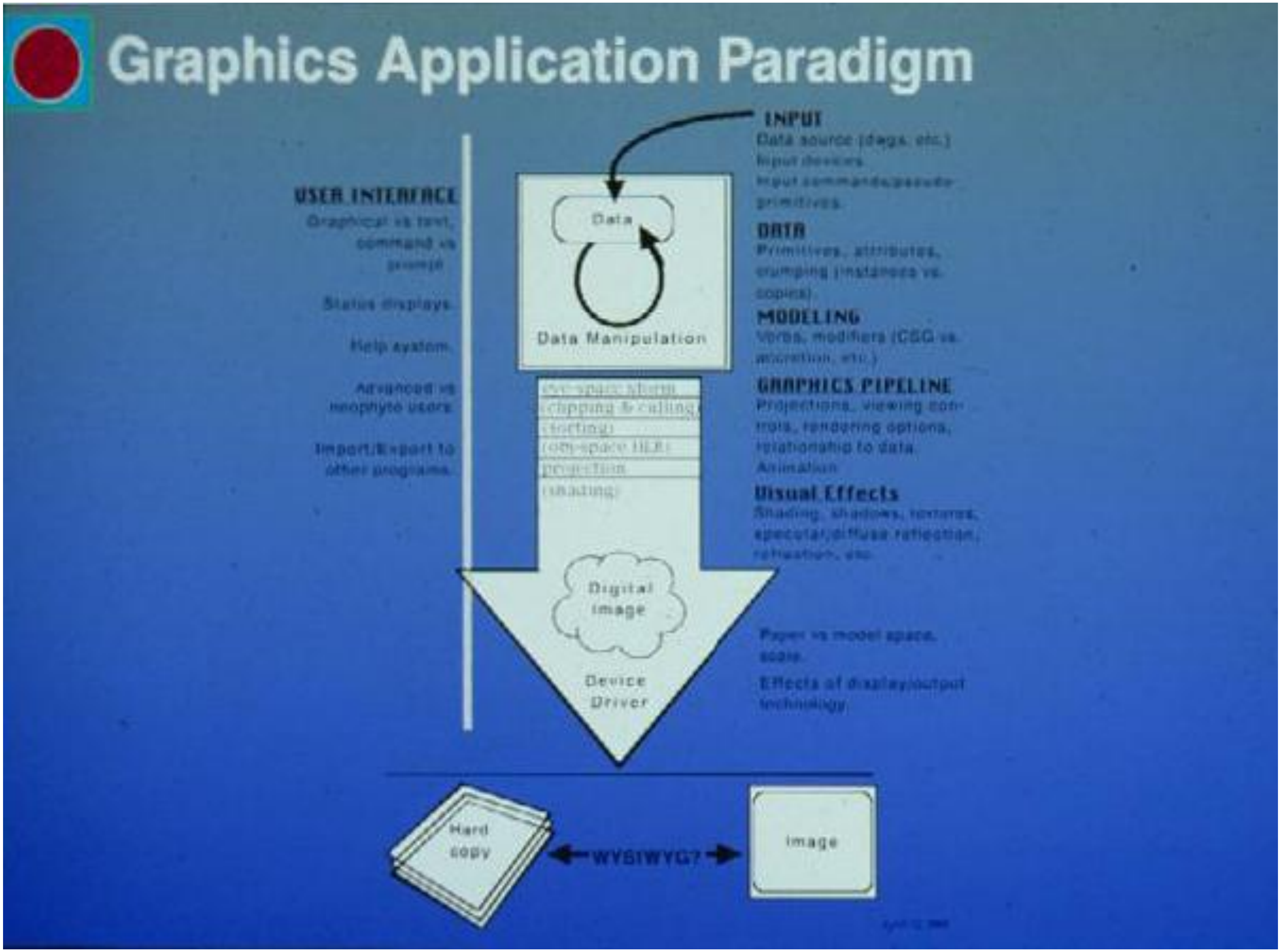
- *Existing geometry within a file* may be made into a block and inserted (insert).
- An *existing model* (.3dm) file may be inserted, creating a block and instance (insertFile).
- Internally defined blocks may be *redefined* (block), and are saved with the current file.
- External files may optionally be “*linked*” and automatically “re-inserted” at times, facilitating change propagation.

Some cautions

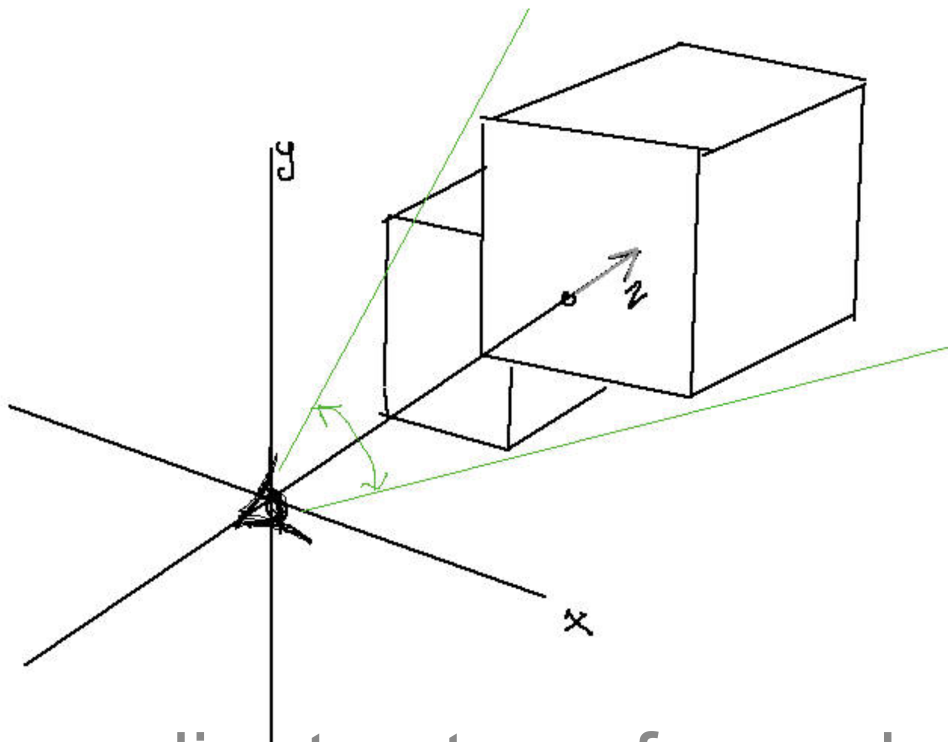
- Inserted file: origin becomes insertion pt.
- Editing = place, explode, edit, re-define
  - Mark insertion-point before re-defining or things will move!
- Redefining the block does not change instance references to the block.
- Blocks defined in the inserted file become blocks in the receiving file (libraries?).

2: The Graphics Pipeline

Arch 481

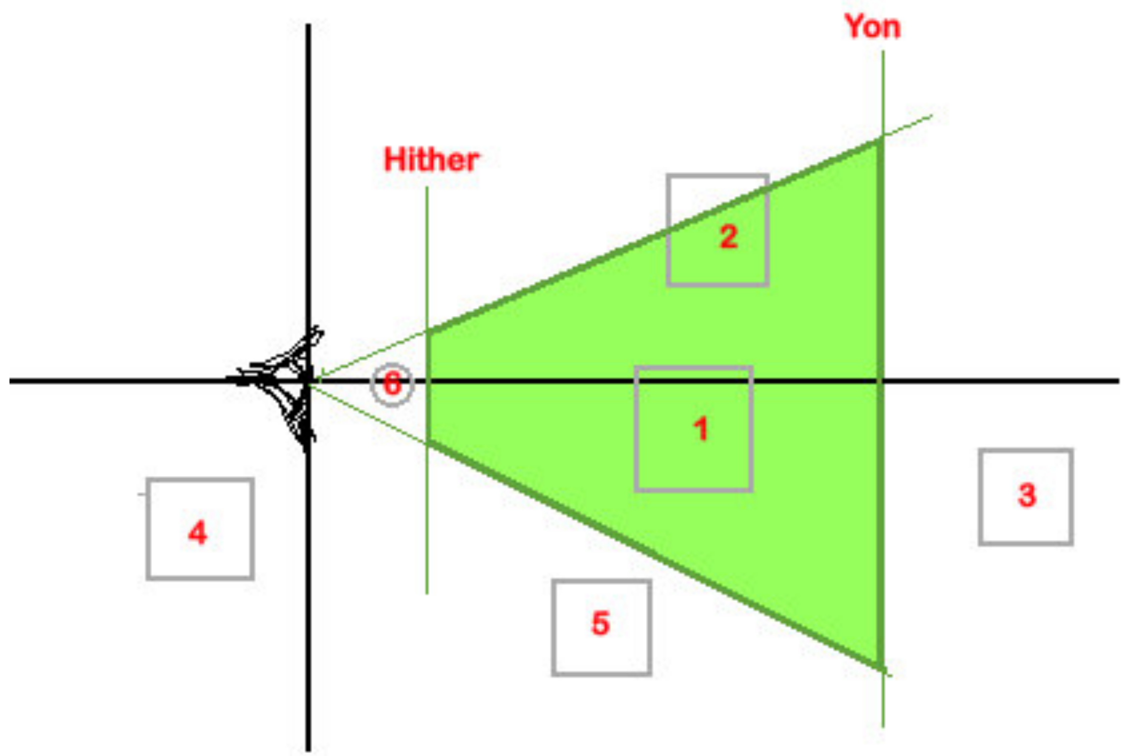

Graphics Application Paradigm
USER INTERFACE
Graphical vs text, command vs prompt
Status displays
Help system
Advanced vs neophyte users
Import/Export to other programs
INPUT
Data source (digs, etc.)
Input devices
Input commands/pseudo-primitives
DATA
Primitives, attributes, clumping (instances vs. copies)
MODELING
Verbs, modifiers (CSG vs. accretion, etc.)
GRAPHICS PIPELINE
Projections, viewing controls, rendering options, relationship to data.
Animation
Visual Effects
Shading, shadows, textures, specular/diffuse reflection, refraction, etc.
Data
Data Manipulation
eye-space xform
clipping & culling
(sorting)
(obj-space HLR)
projection
(shading)
Digital image
Device Driver
Paper vs model space, scale.
Effects of display/output technology
Hard copy
WYSIWYG?
image

Eye-space transform



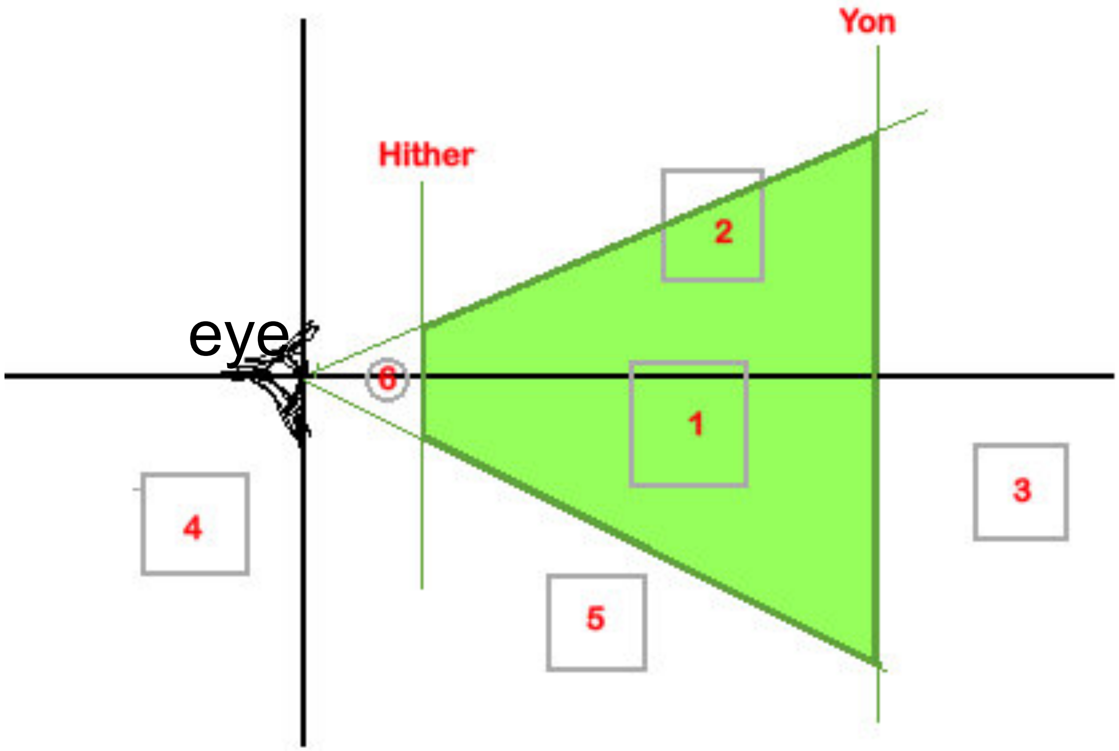
Model coordinates transformed so eye is at origin, with “image-x-axis” to the right and “image-y-axis” up (sometimes called u,v).

Clipping



Projection flattens ALL of 3D space onto the image plane, whether data is behind you (4) or off to the side (5). Clipping reduces it …

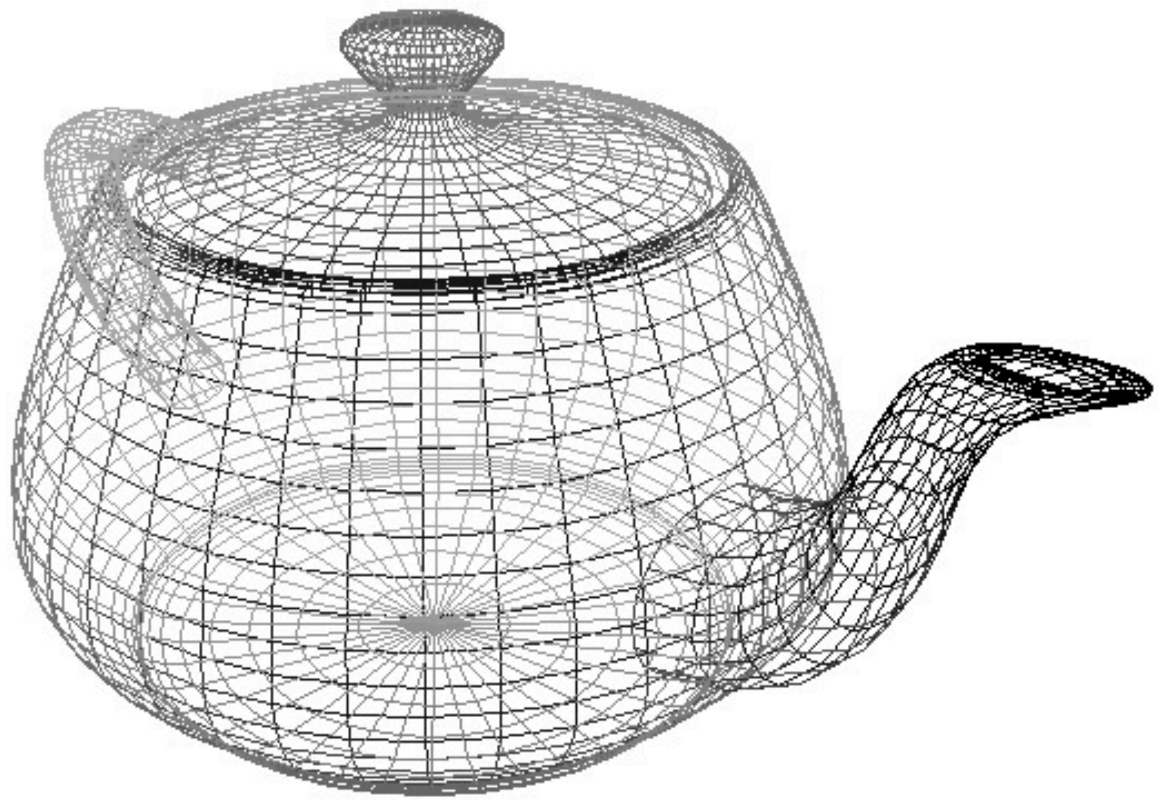
Clipping



Regular clipping will remove items 4 and 5 and trim 2. "Hither and Yon" clipping, if available, will not draw items 6 or 3 either.

Hidden Surfaces

This image uses "atmospheric perspective" (far lines are gray, near lines are black) to help viewers understand the model.



The default is for EVERYTHING to be visible (wireframe). "Hidden" is a decision the program must *make*.

Hidden Surfaces

Multiple strategies have been developed to address this problem. Different renderers use different schemes or combinations of:

- Culling
- Depth sorting
- Z-buffer

Hidden Surface Strategies

Thinking in MODEL SPACE

- We see the *surfaces that face* us
  - Discard surfaces facing away (**culling**)
  - Fails with one-sided sheets
  - Fails in a concave world
- We see only the *surfaces that are closest* to us.
  - Relies on the “destructive write” computer displays
  - Sort surfaces by depth in scene (**depth sorting**)
  - There is a time-cost when sorting complex scenes
  - Fails when surfaces intersect or overlap
- Other: scan-line, etc.

Hidden Surface Strategies

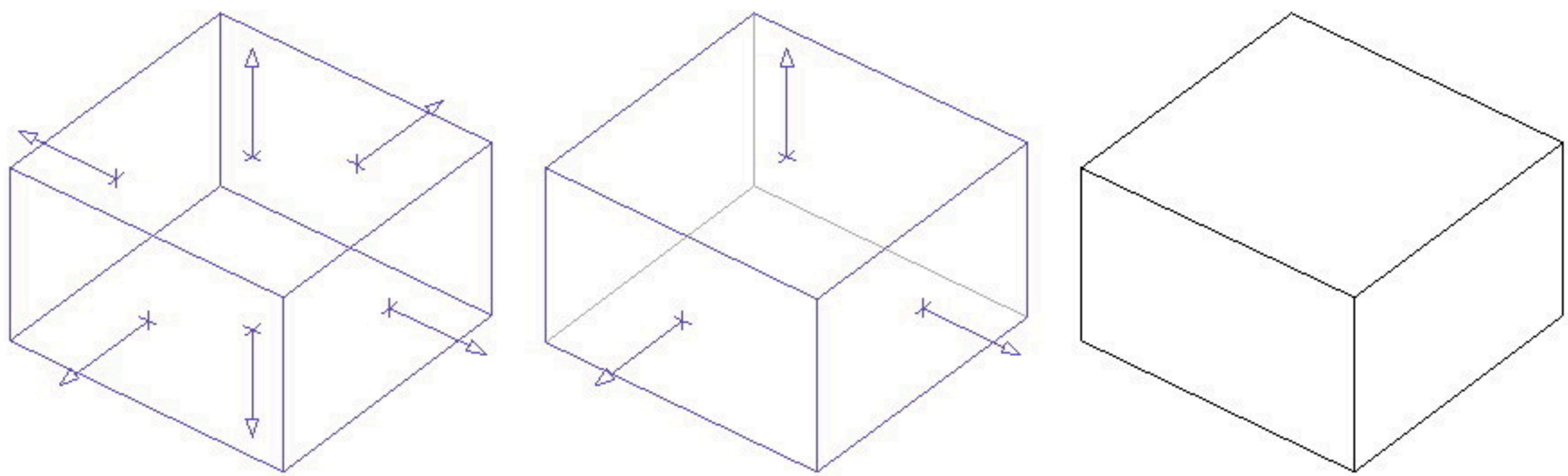
Thinking in IMAGE SPACE – "Z-buffering"

- Answer the "closest surface" question on a *per pixel basis?*
- The graphics pipeline "paints" polygons to the screen one at a time as raster data.
- In *eye space* the "z-coordinate" is depth into the scene.
- Knowing the depth at the corners allows us to interpolate depth at every pixel.
- Storing the depth at each pixel allows us to decide what is visible.

#1: “Culling” Hidden Surfaces

Surfaces have *orientation* (surface normals)

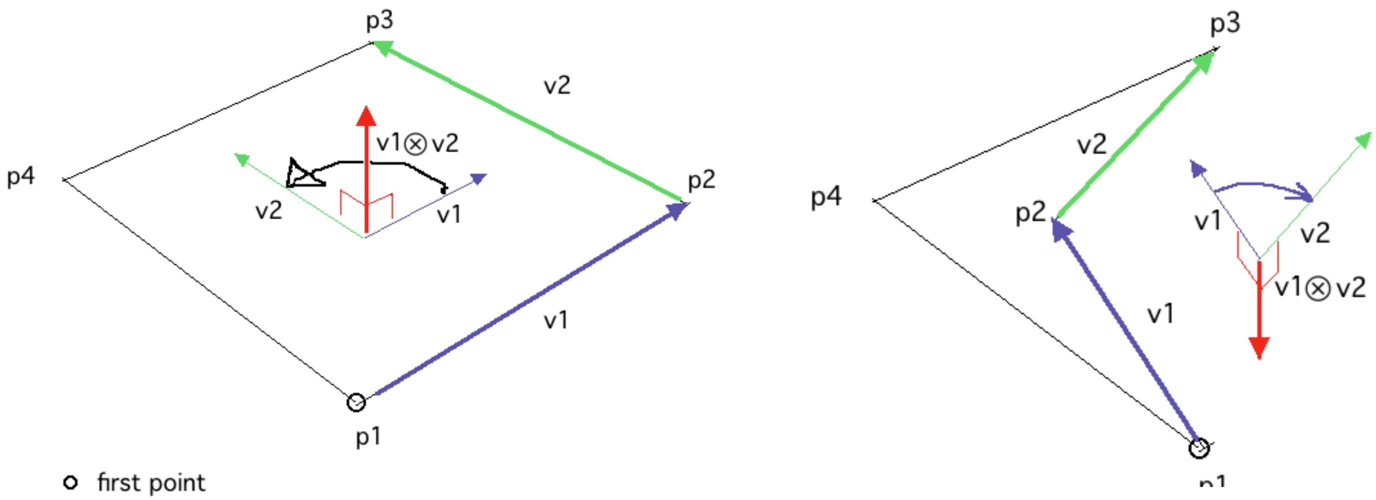
- **Normal**: a vector perpendicular to something
- You won’t see what doesn’t face you



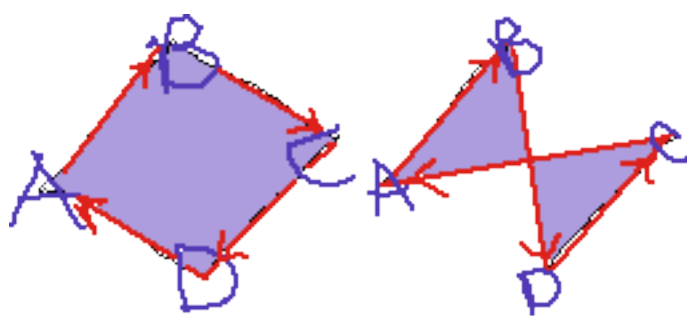
All faces and normals   Faces facing the Camera   Finished Image

What can go wrong?

Orientation is computed from edges:

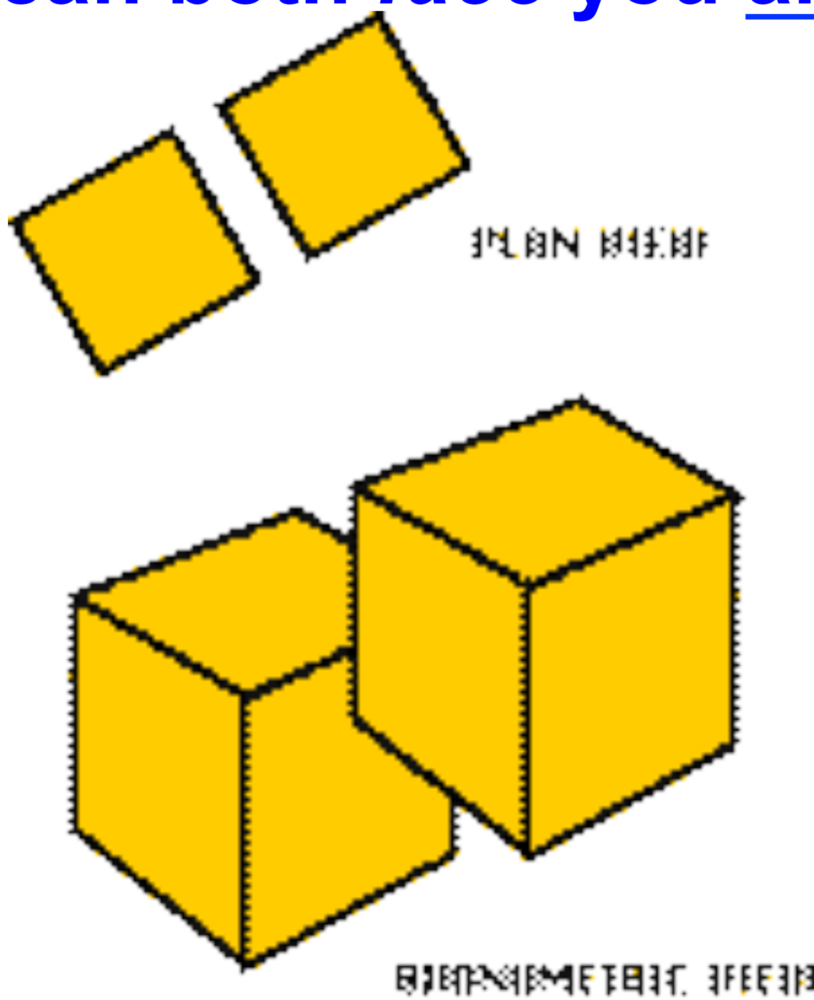


Polygons can get twisted or warped



What else can go wrong?

Polygons can both *face* you <u>and</u> *overlap*:



#2: “Depth Sorting” Surfaces

Eye space

- first step in graphics pipeline
- “+x” to the right, “+y” is up
- “+z” is distance away from the eye

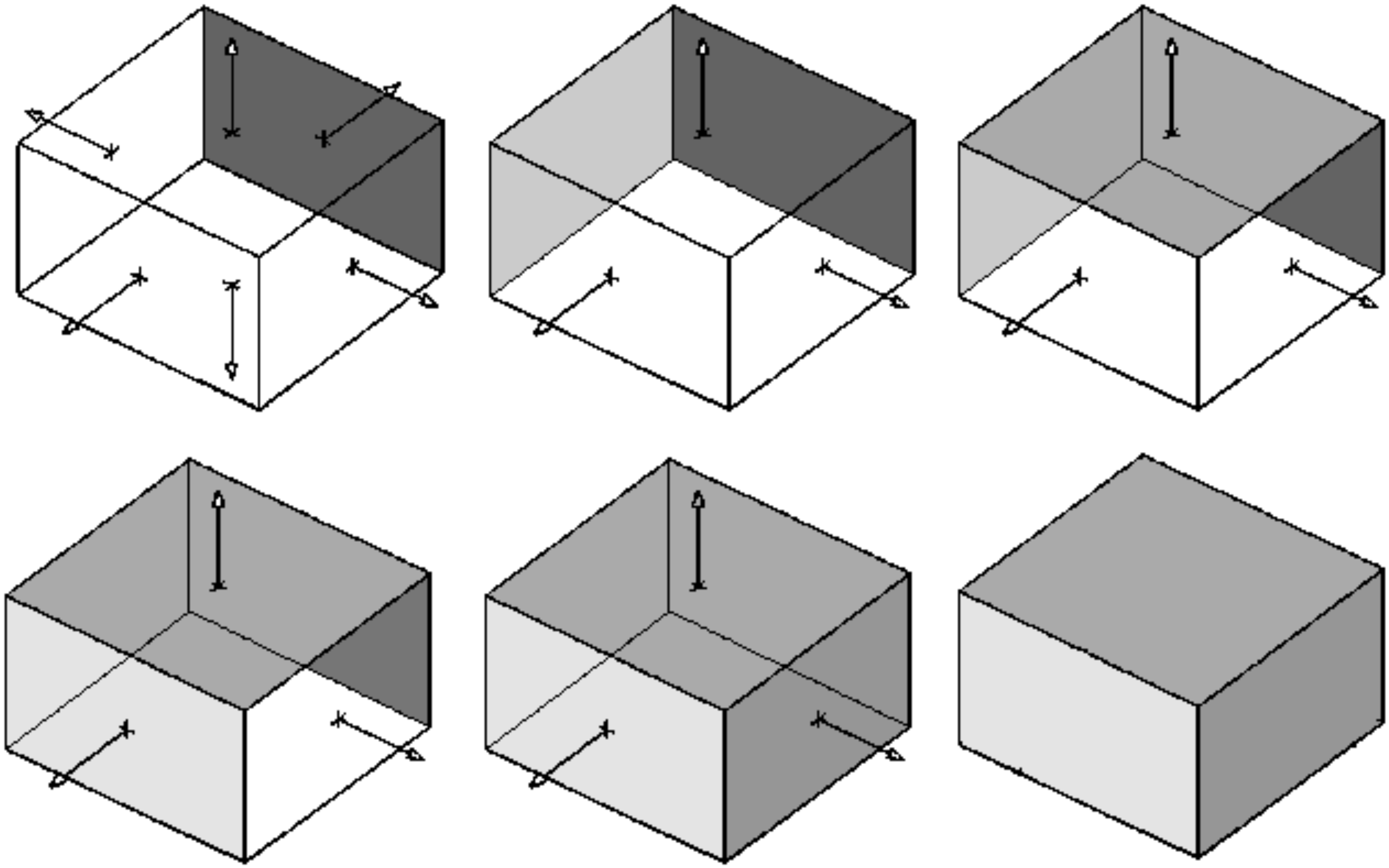
Raster screens support “destructive write”

- New stuff, drawn over old stuff, completely replaces it.

→ Draw from furthest away to nearest!

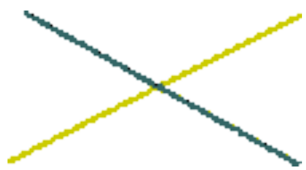
#2: “Depth Sorting” Surfaces

Draw the most distant face first, based on view point, then the closer ones.

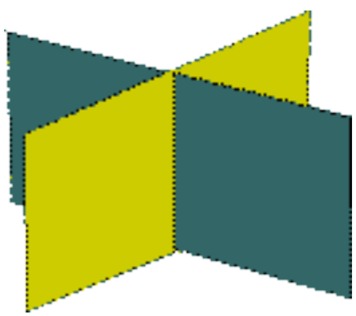


"Depth Sorting" errors…

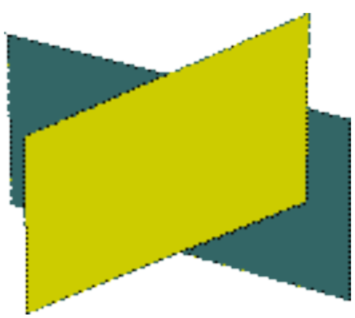
Because it works with whole polygons…



Intersecting polygons like this
(top view)



... that should look like this …
(axo view)



… will look like this instead!

#3: *Z-buffering*

Works with pixels, not polygons …

… works in “screen space” not “model space”

Screen space is NOT infinite (yay!)

Raster screen determines needed accuracy

Most uses will be raster.

Hidden Surfaces: #3 Z-buffering