Select Tender Topics

(not a substitute for Recitation Recap, nor for studying on your own)

What did I need to know, in particular, from cs1566?
...**Everything**.

cs1566 Alum gone straight to Sony Playstation, about his job interview.

Recitation Tomorrow (and on your own)

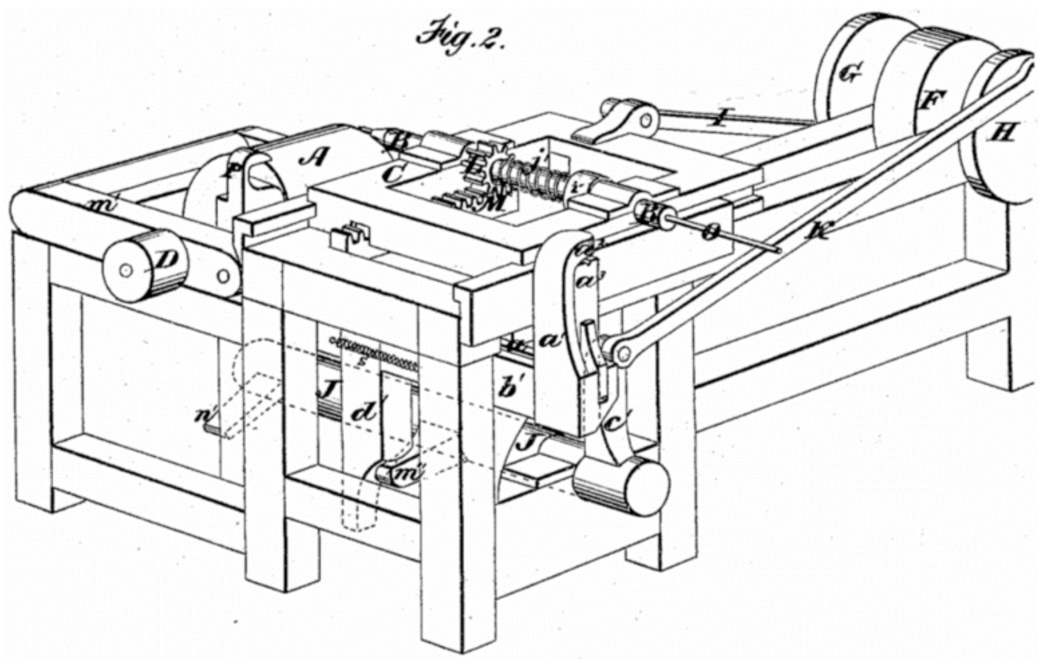
<u>Example</u> topics to recap:

- Graphics pipeline
  - mapping screen (mouse) to math drawing space and the other way around,…
- Dot products, cross products and their uses
- Vectors and vector manipulation
  - particle systems, collision detection,…
- Geometric transformations
  - homogeneous coordinates
  - composing transformations (order!)
  - generating 3D shapes
  - constructing an arbitrary axis rotation
  - …
- Camera, …
- Ray construction, …
- Ray-Object intersections, implicit reps,…

etc, etc

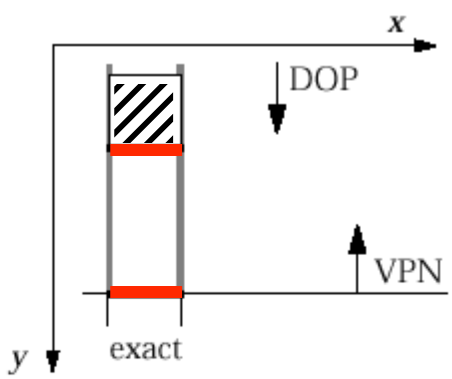
Projections

- know how to recognize them from 2D drawings, and how to generate a 2D drawing using a particular projection
- experiment with expressing projections mathematically

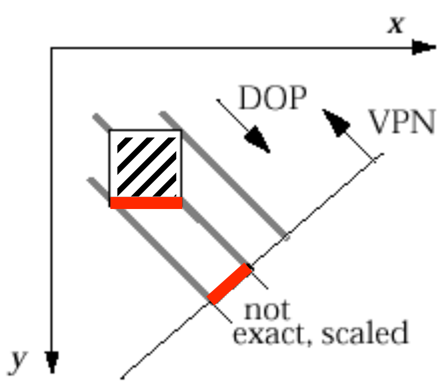


Overview of Parallel Projections

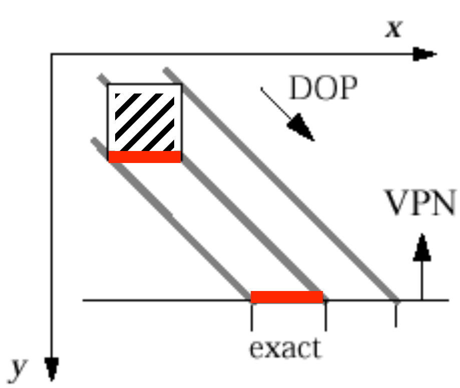
- Assume object face-of-interest lies in principal plane, i.e., parallel to *xy, yz,* or *zx* planes.
- DOP = Direction of Projection, VPN = View Plane Normal



- 1) Multiview Orthographic
  - VPN || a principal coordinate axis
  - DOP || VPN
  - shows single face, exact measurements

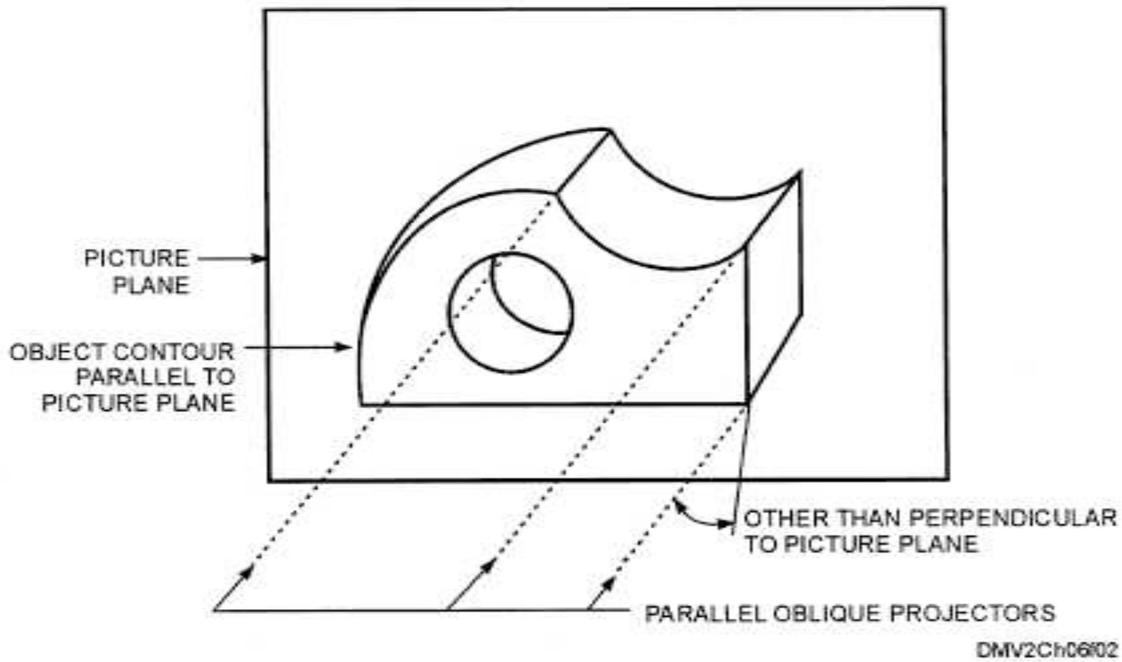


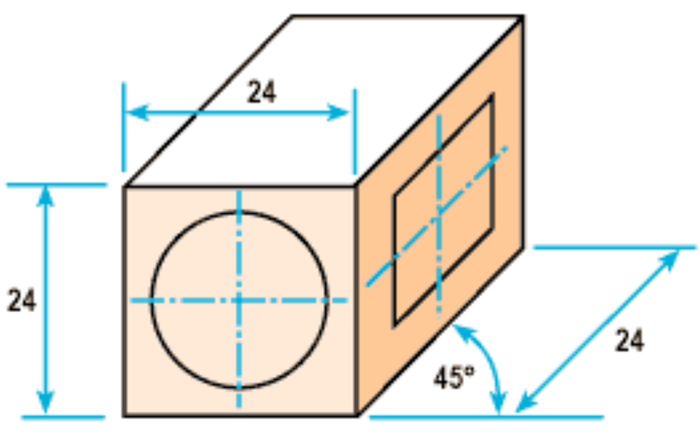
- 2) Axonometric
  - VPN $\nparallel$ a principal coordinate axis
  - DOP || VPN
  - adjacent faces, none exact, uniformly foreshortened (function of angle between face normal and DOP)



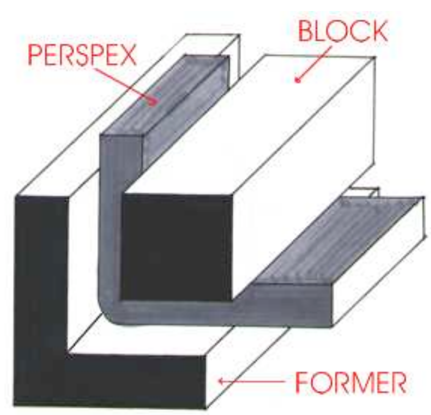
- 3) Oblique
  - VPN || a principal coordinate axis
  - DOP $\nparallel$ VPN
  - adjacent faces, one exact, others uniformly foreshortened

Oblique Projections

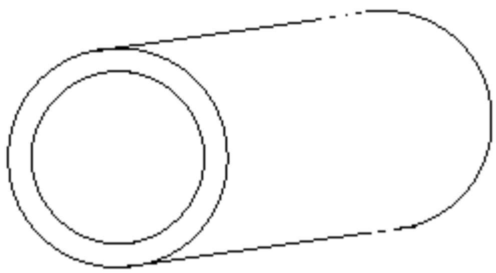




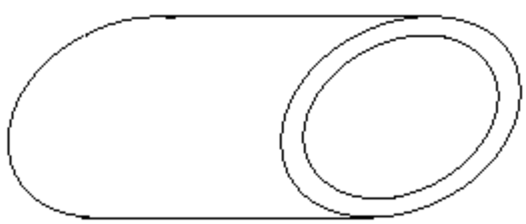
Cavalier projection



If It Were Oblique Projection



- Projection plane parallel to circular face

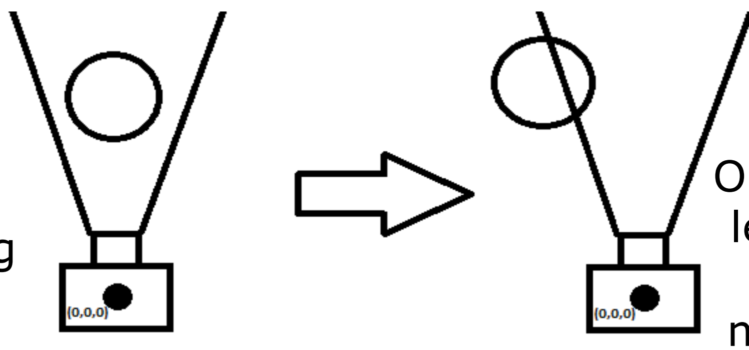


- Projection plane not parallel to circular face

Relative Motion

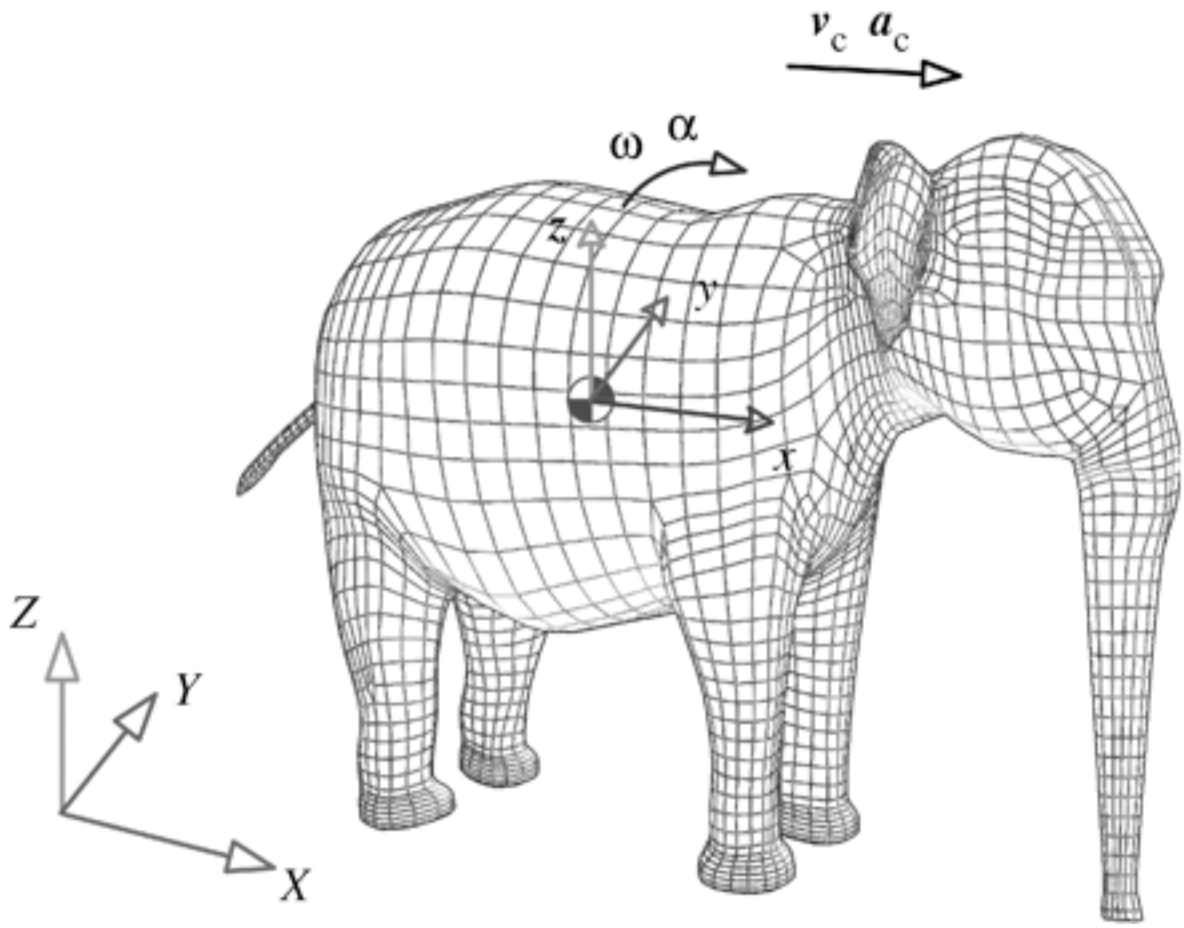
Translate “camera” to the right

Field of view in OpenGL can be thought of as view of camera at origin looking down –z axis



Object moves to left to simulate a camera moving to right

Aligning
Local Coordinate Systems



Two coordinate systems used to track elephant motions. A trackway coordinate system XYZ was defined by aligning the axes with the trackway. A body-fixed coordinate system xyz was defined to move with the elephant body. The derived CM velocity vc and acceleration ac, and torso angular velocity ω and angular acceleration ɑ were all expressed in the body-fixed coordinate system.

http://rsif.royalsocietypublishing.org/content/5/19/195/F2.expansion.html

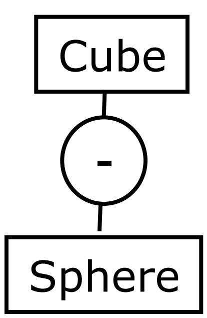
Ray-Object Intersection

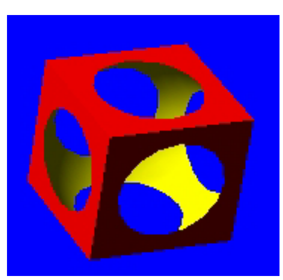
Implicit surface strategy summary

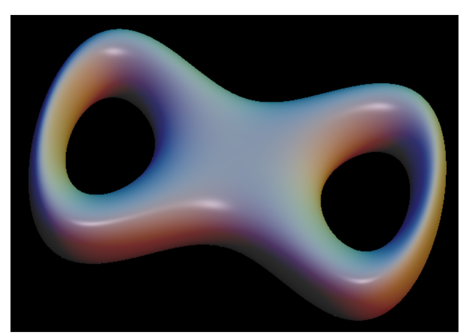
- Substitute ray (P + *t*d) into implicit surface equations and solve for *t*
  - surface you see “first” from eye point is at smallest non-negative *t*-value
  - testing for non-intersection, tangents etc

- For complicated objects (not defined by a single equation), write out a set of equalities and inequalities and then code as cases…

- Latter approach can be generalized cleverly to handle all sorts of complex combinations of objects
  - note we don’t need a polygonal representation of the object







F(x,y,z) = ((x^2*(1-x^2)-y^2)^2+0.5*z^2-f*(1+b*(x^2+y^2+z^2)) = 0