CS 4204 Computer Graphics

Clipping and Class Viewing

Yong Cao

Virginia Tech

References:
Interactive Computer Graphics, Fourth Edition, Ed Angle

Objectives

Introduce basic implementation strategies

Clipping

Overview

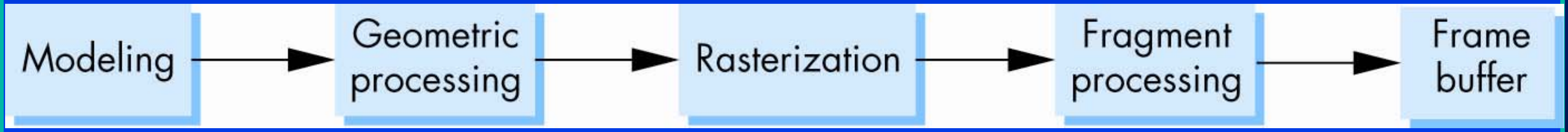
At end of the geometric pipeline, vertices have been assembled into primitives

Must clip out primitives that are outside the view frustum

- Algorithms based on representing primitives by lists of vertices

Must find which pixels can be affected by each primitive

- Fragment generation
- Rasterization or scan conversion



Required Tasks

Clipping

Rasterization or scan conversion

Transformations

Some tasks deferred until fragement processing

- Hidden surface removal
- Antialiasing

Clipping

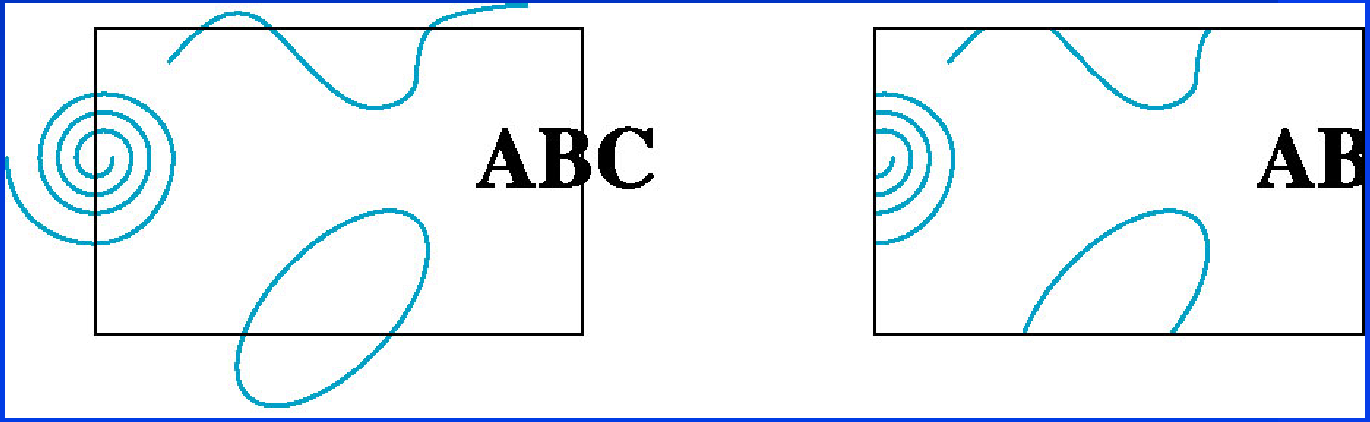
2D against clipping window

3D against clipping volume

Easy for line segments polygons

Hard for curves and text

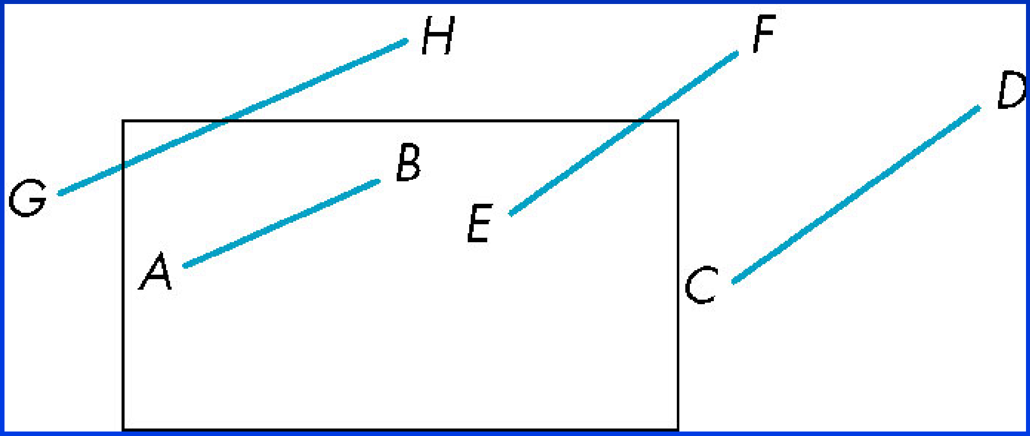
- Convert to lines and polygons first



Clipping 2D Line Segments

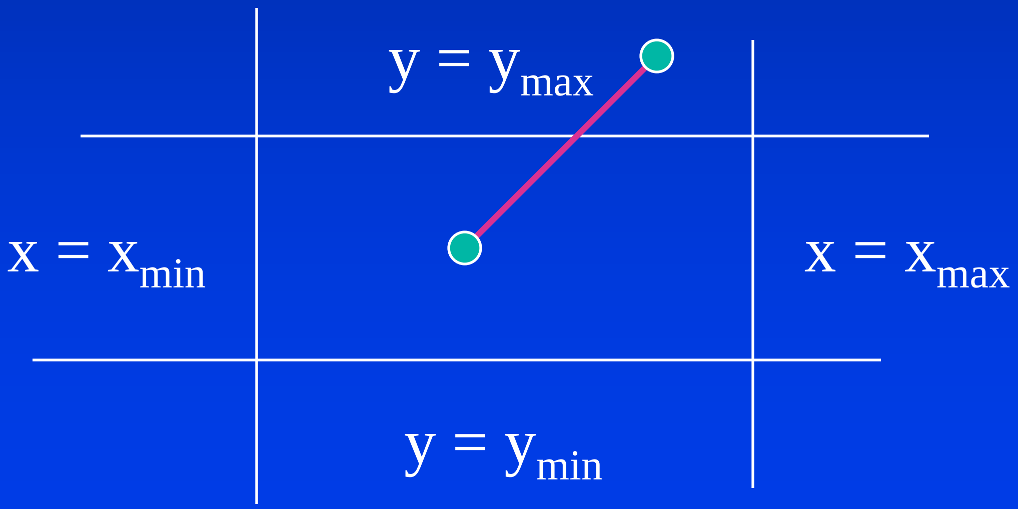
Brute force approach: compute intersections with all sides of clipping window

- Inefficient



Cohen-Sutherland Algorithm

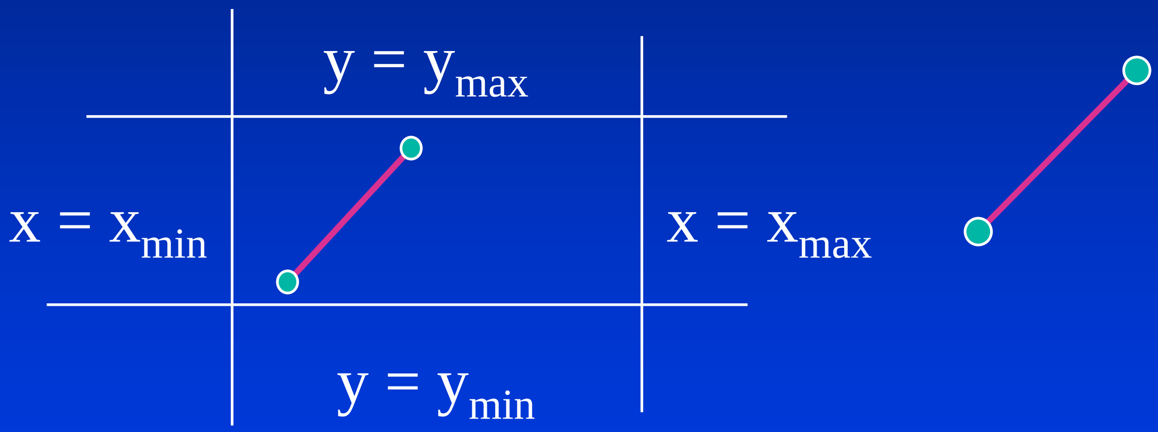
- *Idea: eliminate as many cases as possible without computing intersections*
- *Start with four lines that determine the sides of the clipping window*



The Cases

Case 1: both endpoints of line segment inside all four lines

- Draw (accept) line segment as is



Case 2: both endpoints outside all lines and on same side of a line

- Discard (reject) the line segment

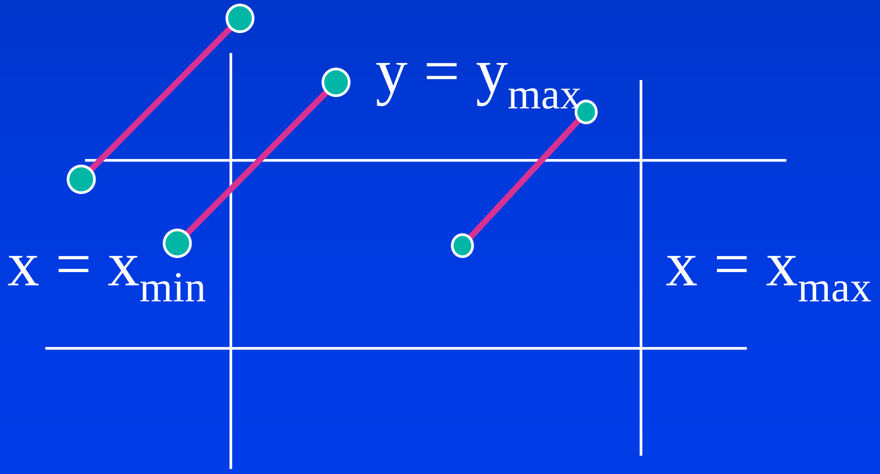
The Cases

Case 3: One endpoint inside, one outside

- Must do at least one intersection

Case 4: Both outside

- May have part inside
- Must do at least one intersection

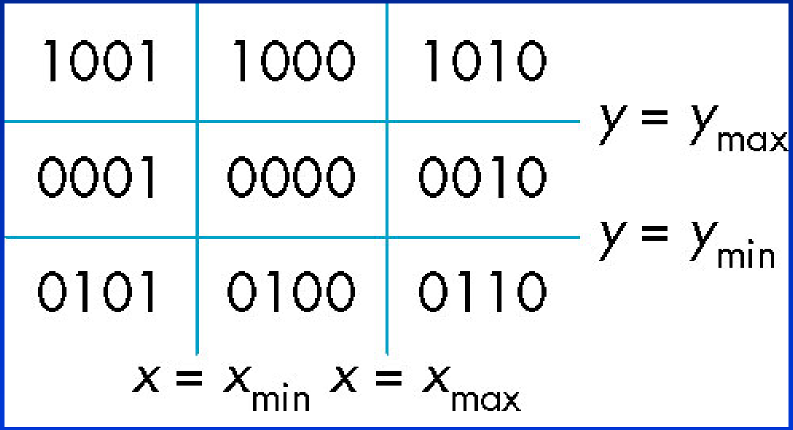


Defining Outcodes

For each endpoint, define an outcode

$b_0 b_1 b_2 b_3$

$b_0 = 1$ if $y > y_{max}$, 0 otherwise
$b_1 = 1$ if $y < y_{min}$, 0 otherwise
$b_2 = 1$ if $x > x_{max}$, 0 otherwise
$b_3 = 1$ if $x < x_{min}$, 0 otherwise



Outcodes divide space into 9 regions

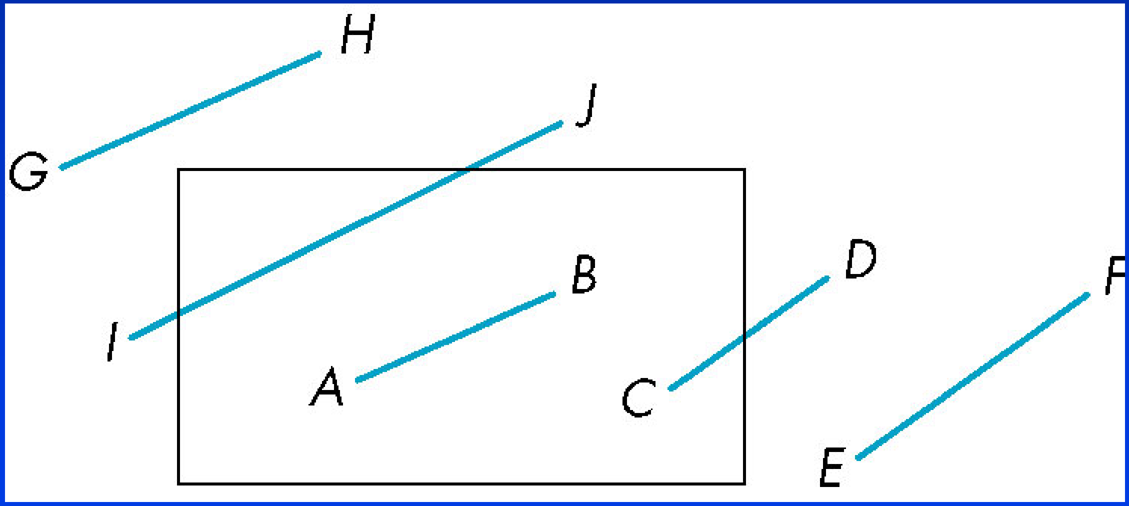
Computation of outcode requires at most 4 subtractions

Using Outcodes

Consider the 5 cases below

AB: outcode(A) = outcode(B) = 0

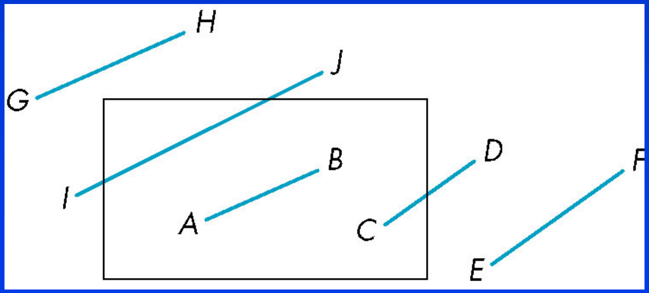
- Accept line segment



Using Outcodes

CD: outcode (C) = 0, outcode(D) $\neq$ 0

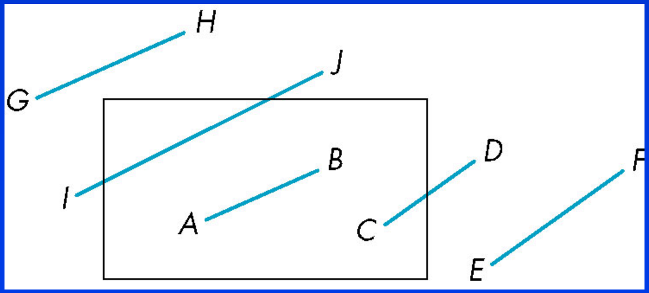
- Compute intersection
- Location of 1 in outcode(D) determines which edge to intersect with
- Note if there were a segment from A to a point in a region with 2 ones in outcode, we might have to do two intersections



Using Outcodes

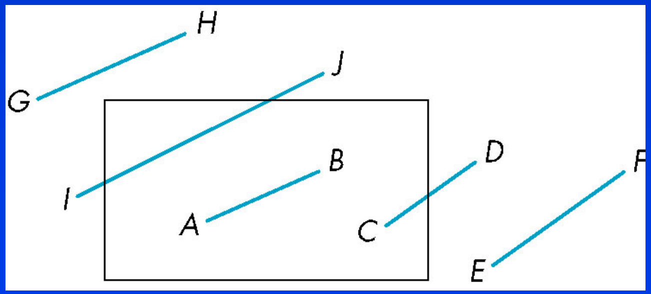
EF: outcode(E) logically ANDed with outcode(F) (bitwise) $\neq 0$

- Both outcodes have a 1 bit in the same place
- Line segment is outside of corresponding side of clipping window
- reject



Using Outcodes

- *GH and IJ: same outcodes, neither zero but logical AND yields zero*
- *Shorten line segment by intersecting with one of sides of window*
- *Compute outcode of intersection (new endpoint of shortened line segment)*
- *Reexecute algorithm*



Efficiency

In many applications, the clipping window is small relative to the size of the entire data base

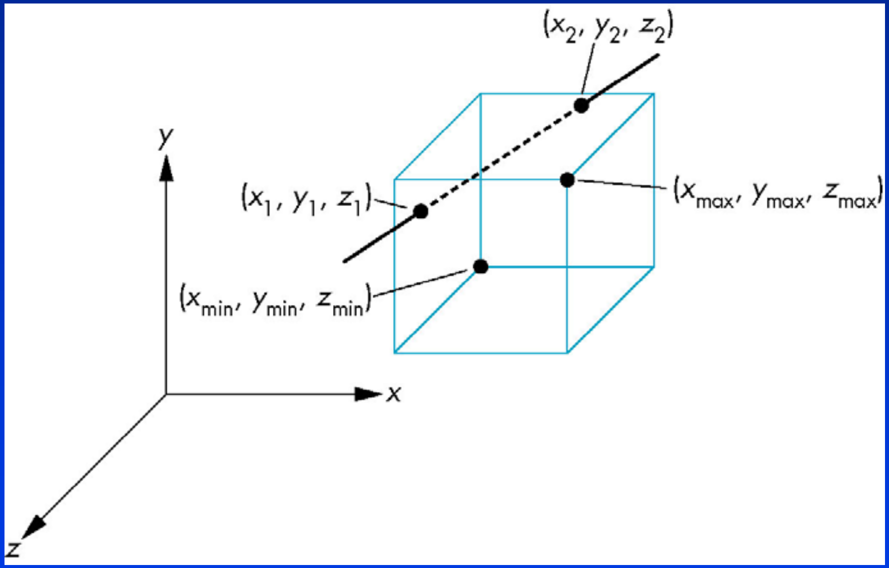
- Most line segments are outside one or more side of the window and can be eliminated based on their outcodes

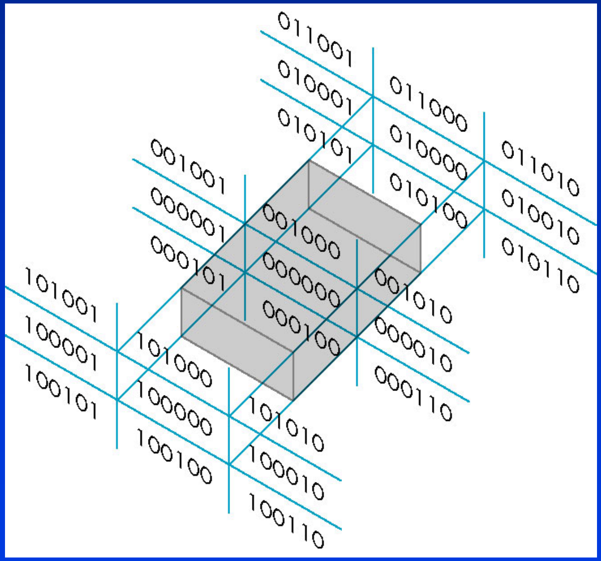
Inefficiency when code has to be reexecuted for line segments that must be shortened in more than one step

Cohen Sutherland in 3D

Use 6-bit outcodes

When needed, clip line segment against planes

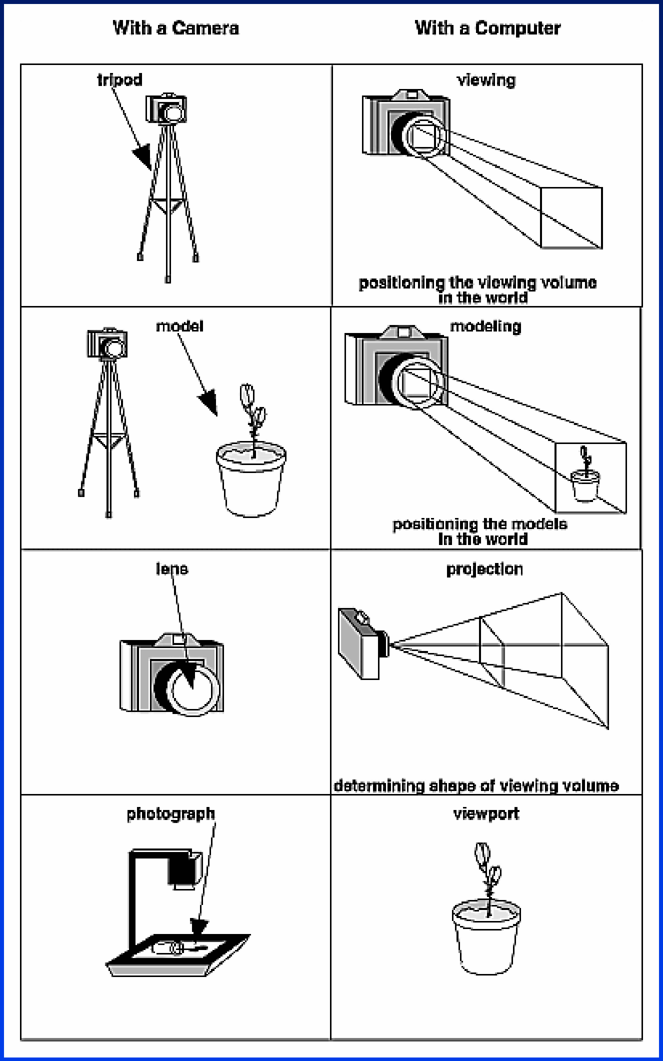




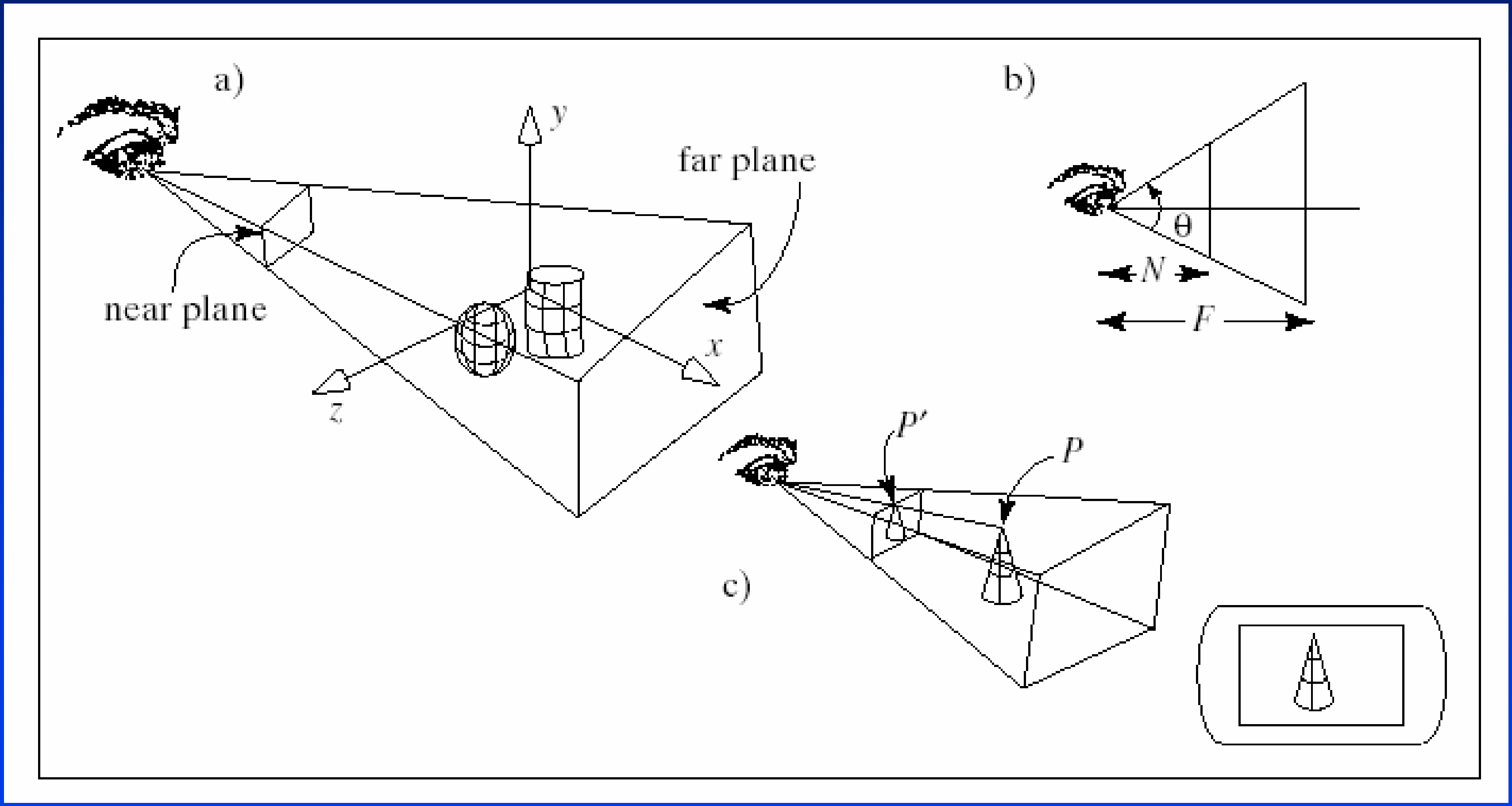
Viewing and Projection

Camera Analogy:

1. ***Set up your tripod and pointing the camera at the scene (viewing transformation).***
2. ***Arrange the scene to be photographed into the desired composition (modeling transformation).***
3. ***Choose a camera lens or adjust the zoom (projection transformation).***
4. ***Determine how large you want the final photograph to be - for example, you might want it enlarged (viewport transformation).***



Projection transformations



Introduction to Projection Transformations

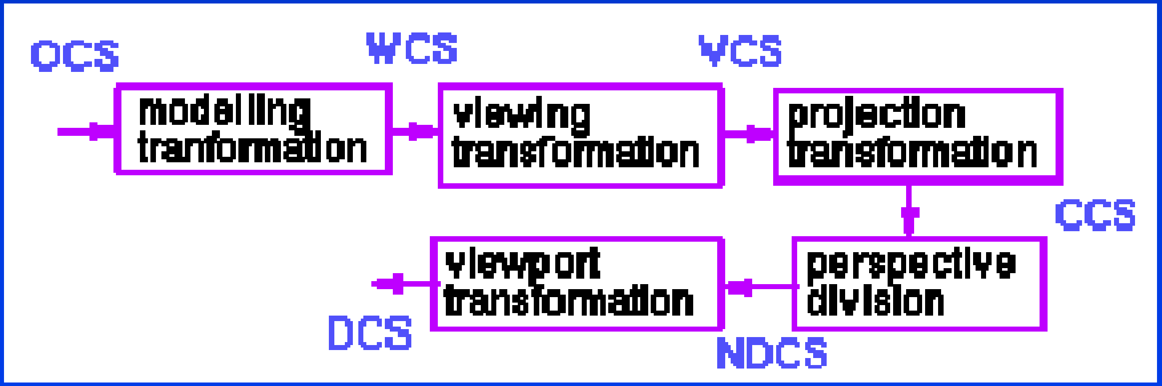
Mapping: $f : R^n \rightarrow R^m$

Projection: $n > m$

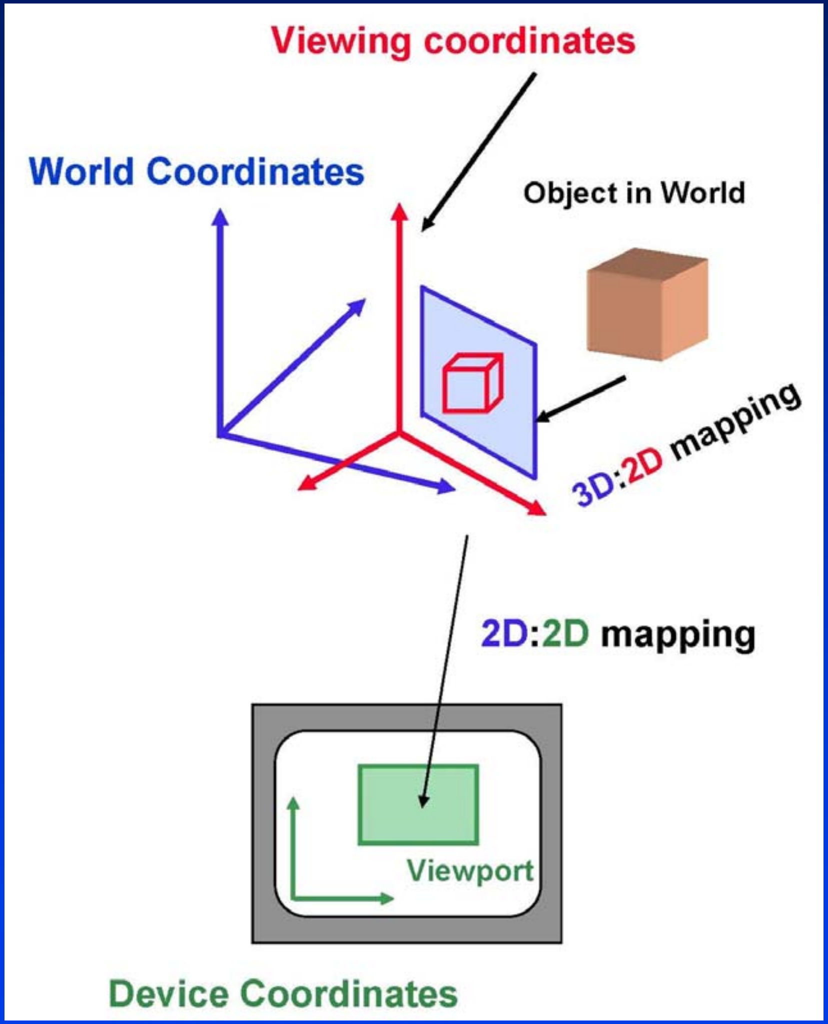
Planar Projection: Projection on a plane.

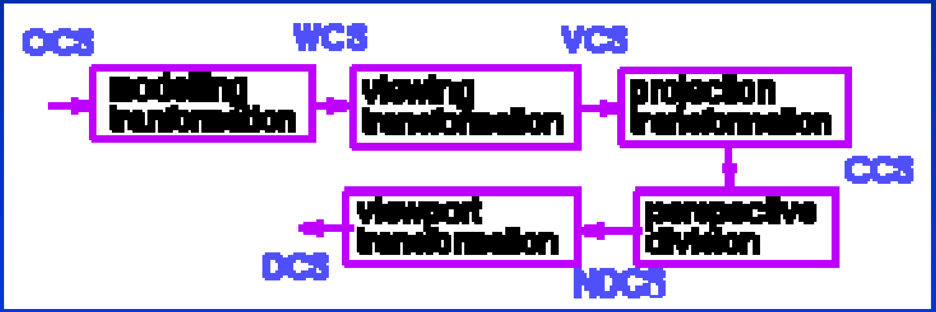
$R^3 \rightarrow R^2$ or

$R^4 \rightarrow R^3$ homogenous coordinates.



Introduction to Projection Transformations





Objectives

- *Introduce the classical views*
- *Compare and contrast image formation by computer with how images have been formed by architects, artists, and engineers*
- *Learn the benefits and drawbacks of each type of view*

Classical Viewing

Viewing requires three basic elements

- One or more objects
- A viewer with a projection surface
- Projectors that go from the object(s) to the projection surface

Classical views are based on the relationship among these elements

- The viewer picks up the object and orients it how she would like to see it

Each object is assumed to constructed from flat **principal faces**

- Buildings, polyhedra, manufactured objects

Planar Geometric Projections

Standard projections project onto a plane

Projectors are lines that either

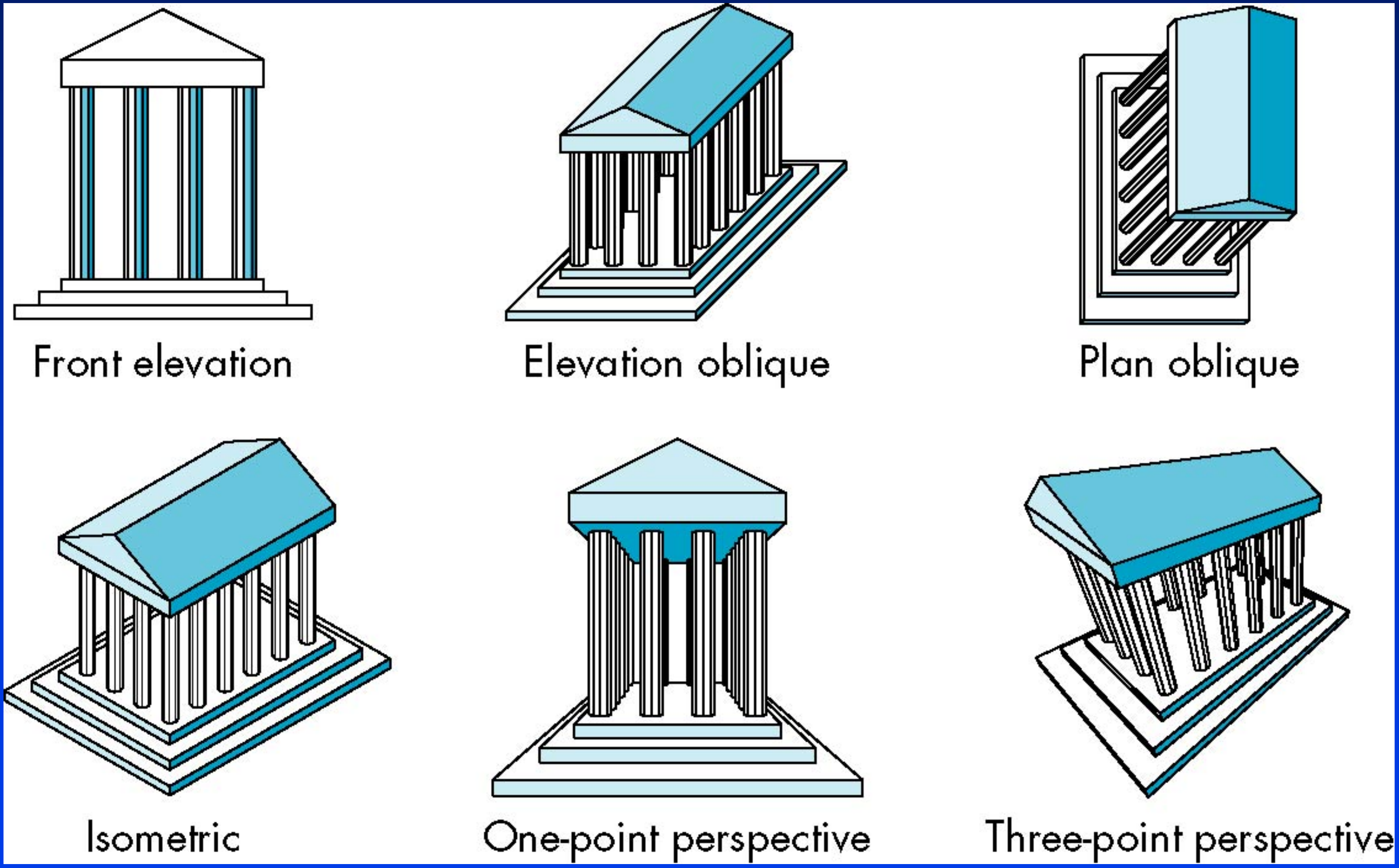
- converge at a center of projection
- are parallel

Such projections preserve lines

- but not necessarily angles

Non-planar projections are needed for applications such as map construction

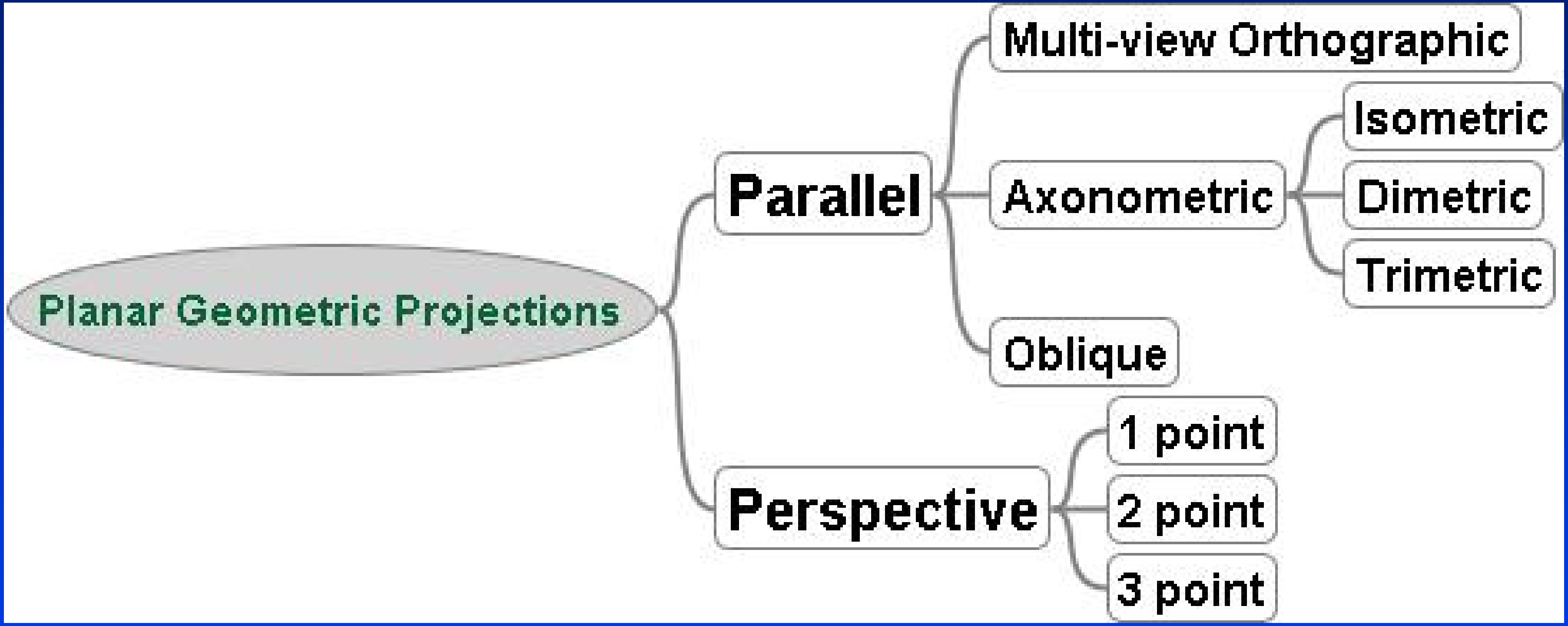
Classical Projections



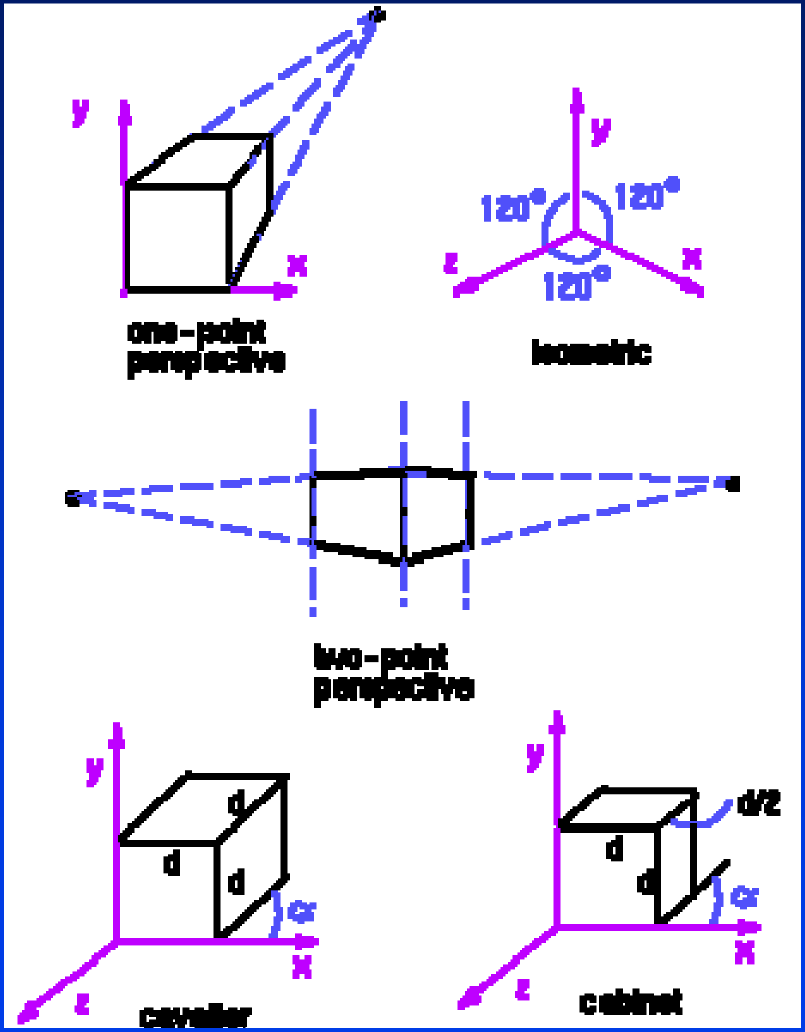
Perspective vs Parallel

- *Computer graphics treats all projections the same and implements them with a single pipeline*
- *Classical viewing developed different techniques for drawing each type of projection*
- *Fundamental distinction is between parallel and perspective viewing even though mathematically parallel viewing is the limit of perspective viewing*

Taxonomy of Planar Geometric Projections

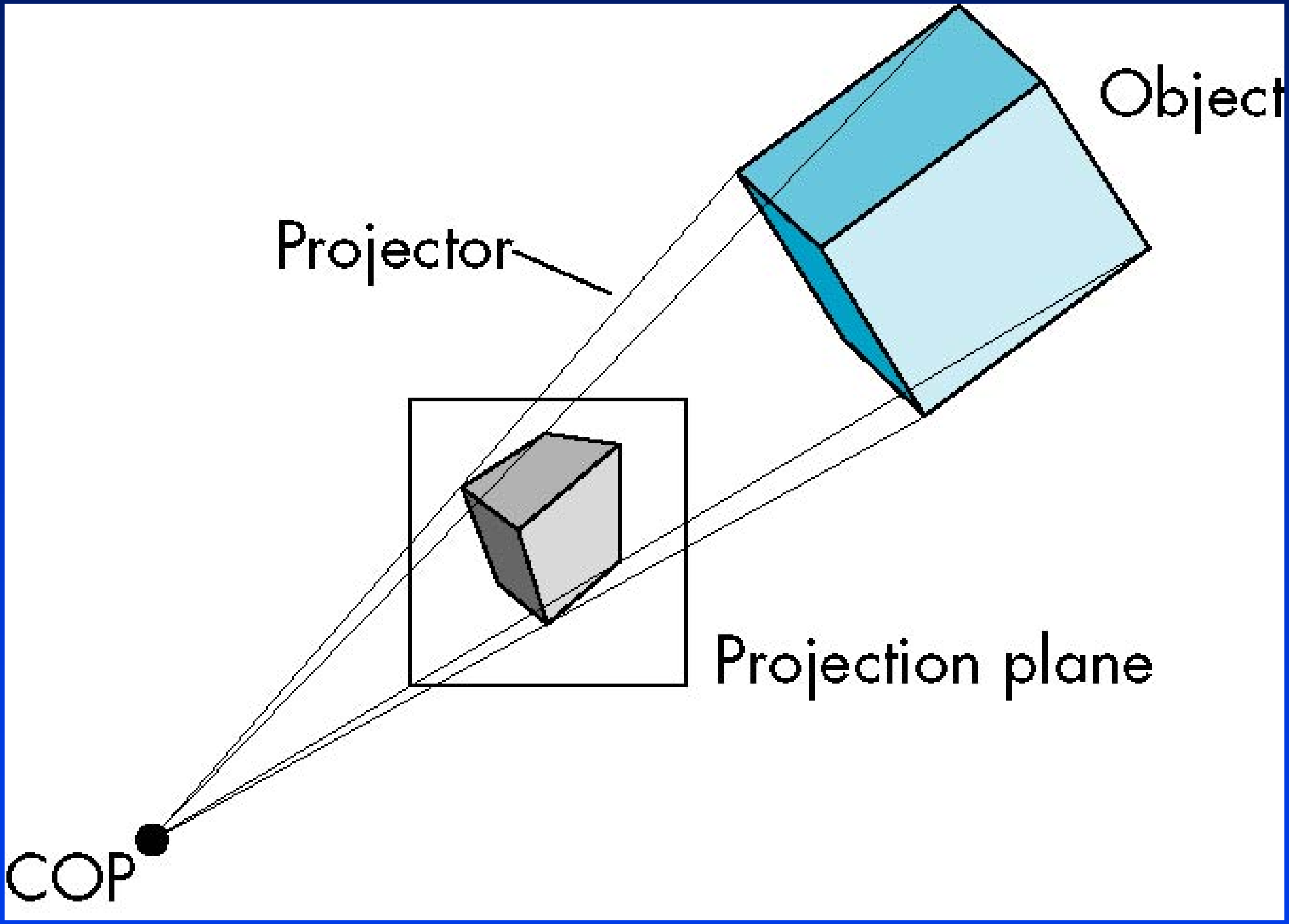


Examples

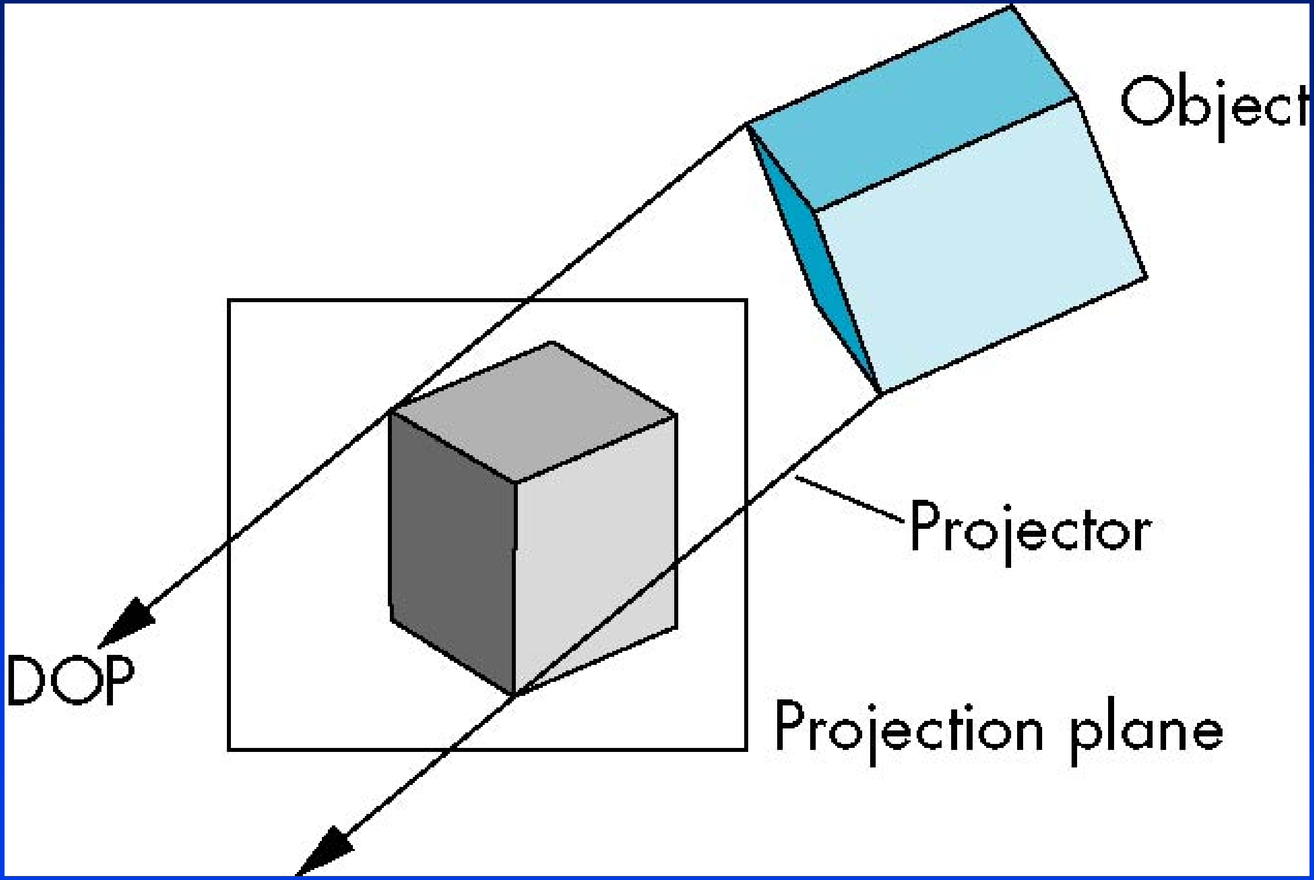


y
x
one-point
perspective
y
120°
120°
120°
z
x
isometric
two-point
perspective
y
d
d
d
α
x
z
cavalier
y
d/2
d
d
α
x
z
cabinet

Perspective Projection

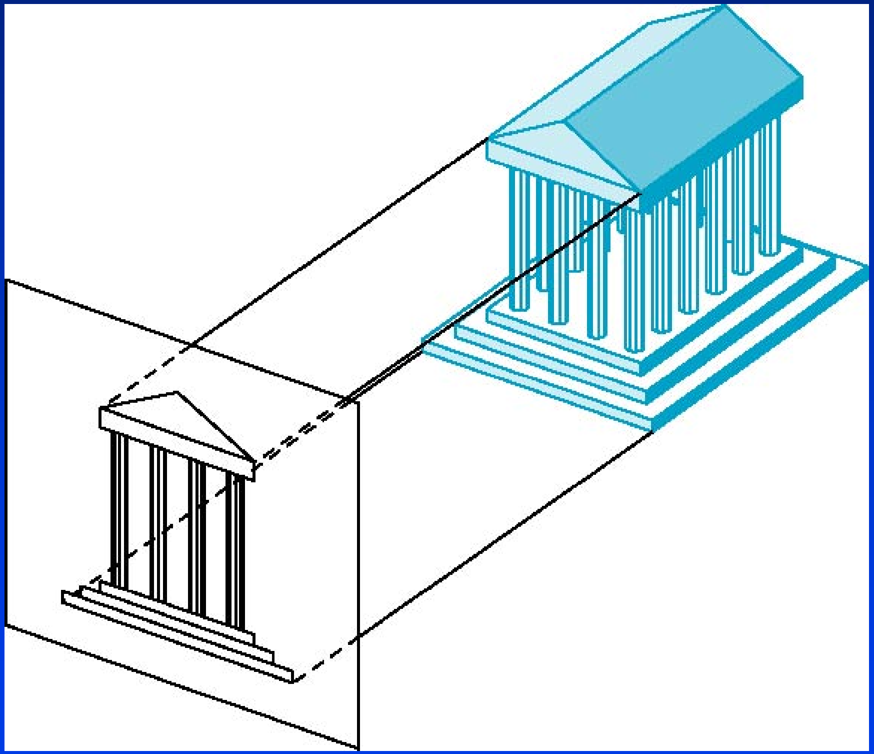


Parallel Projection



Orthographic Projection

Projectors are orthogonal to projection surface

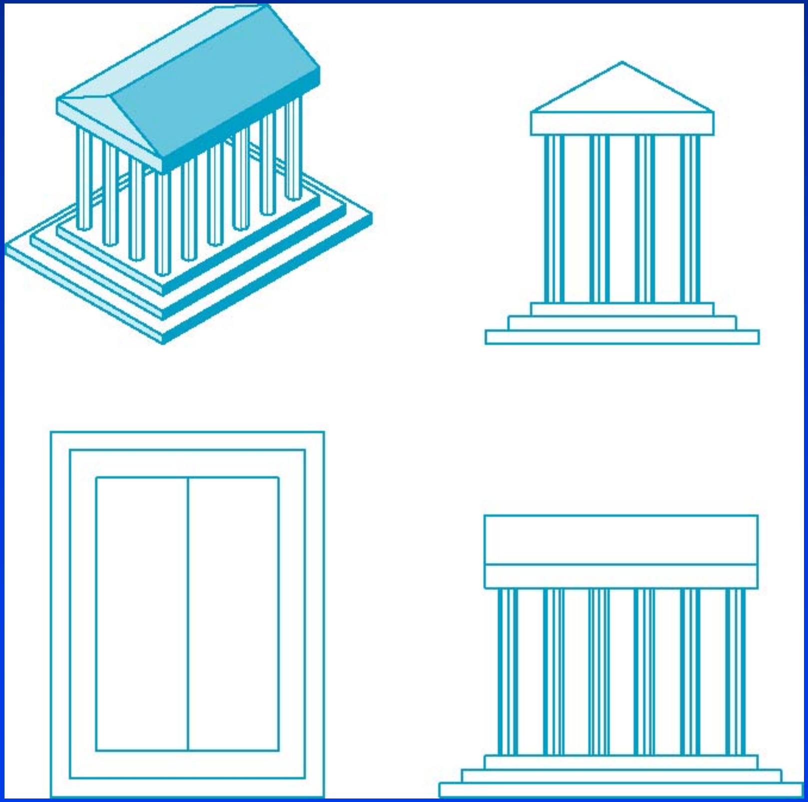


Multiview Orthographic Projection

Projection plane parallel to principal face

Usually form front, top, side views

isometric (not multiview orthographic view)



front

in CAD and architecture, we often display three multiviews plus isometric

top

side

Advantages and Disadvantages

Preserves both distances and angles

- Shapes preserved
- Can be used for measurements
  - *Building plans*
  - *Manuals*

Cannot see what object really looks like because many surfaces hidden from view

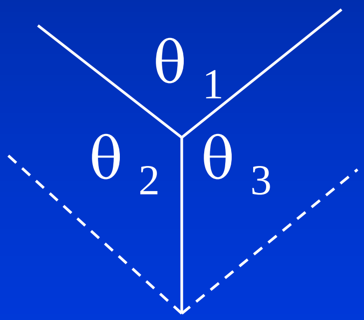
- Often we add the isometric

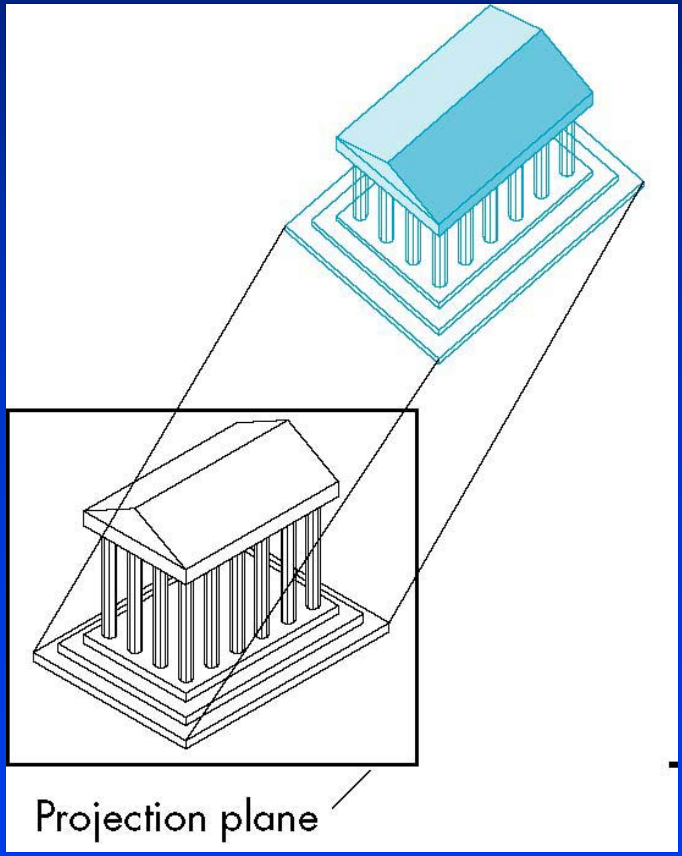
Axonometric Projections

Allow projection plane to move relative to object

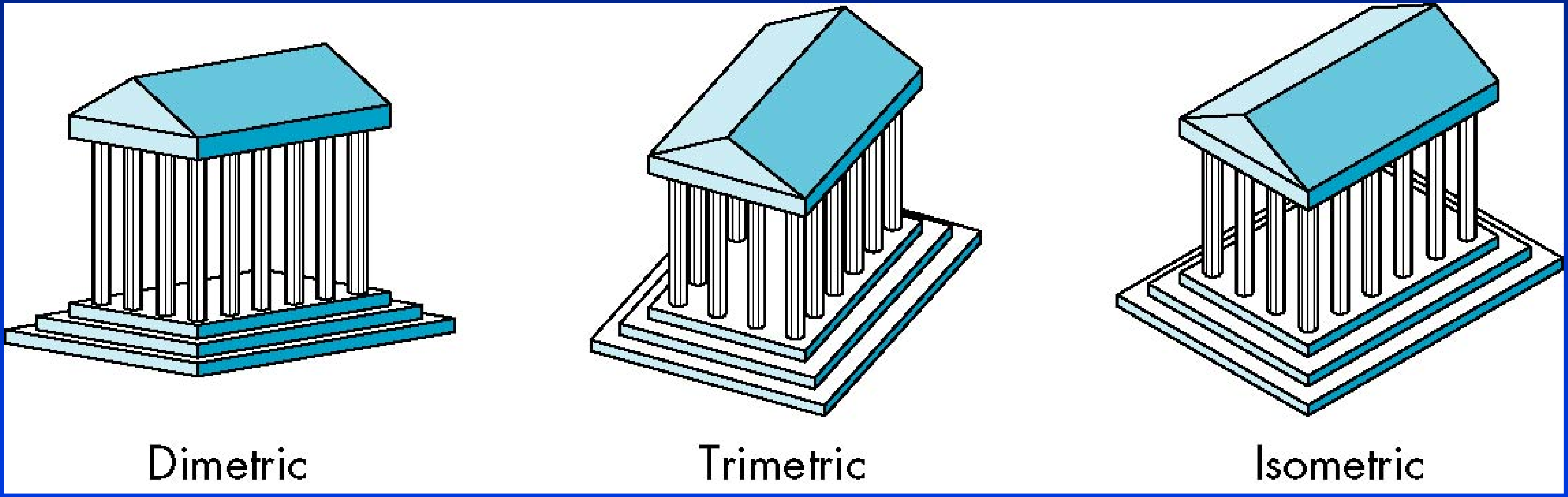
classify by how many angles of a corner of a projected cube are the same

none: trimetric
two: dimetric
three: isometric





Types of Axonometric Projections



Advantages and Disadvantages

Lines are scaled (**foreshortened*****) but can find scaling factors***

Lines preserved but angles are not

- Projection of a circle in a plane not parallel to the projection plane is an ellipse

Can see three principal faces of a box-like object

Some optical illusions possible

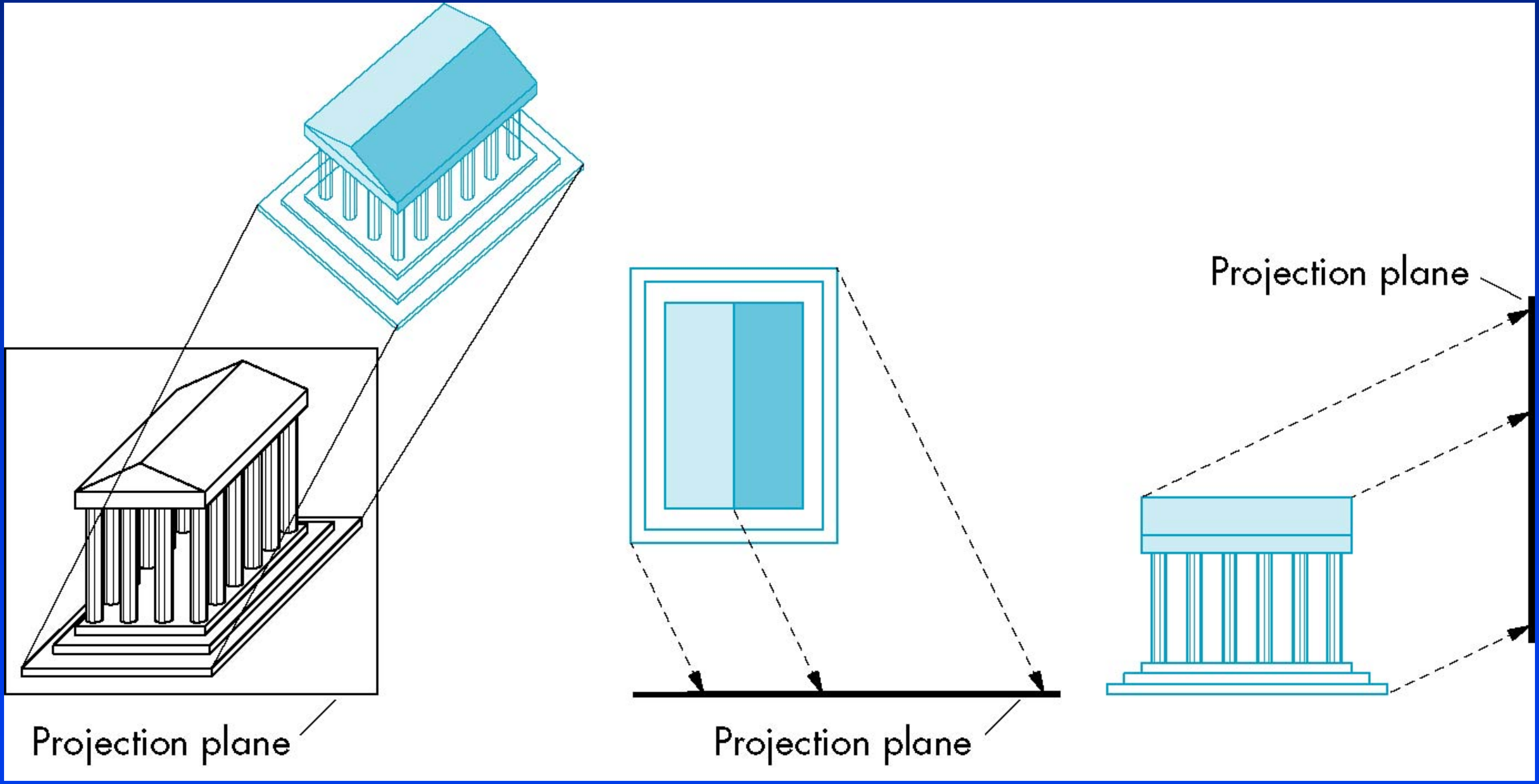
- Parallel lines appear to diverge

Does not look real because far objects are scaled the same as near objects

Used in CAD applications

Oblique Projection

Arbitrary relationship between projectors and projection plane

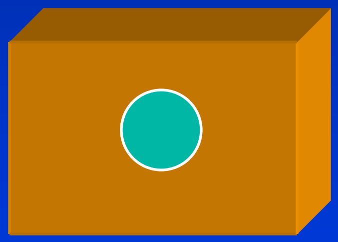


Advantages and Disadvantages

Can pick the angles to emphasize a particular face

- Architecture: plan oblique, elevation oblique

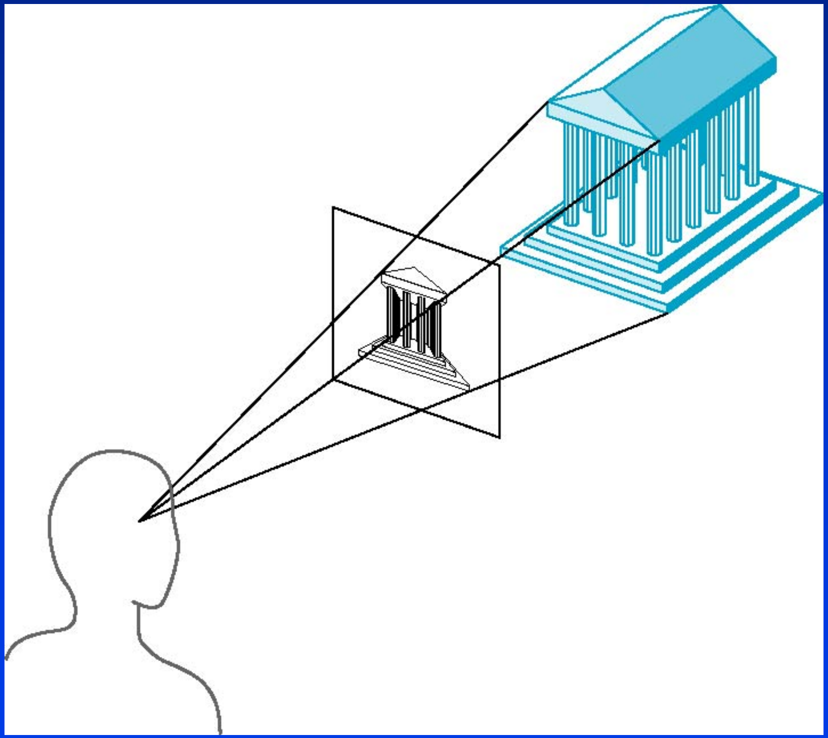
Angles in faces parallel to projection plane are preserved while we can still see "around" side



In physical world, cannot create with simple camera; possible with bellows camera or special lens (architectural)

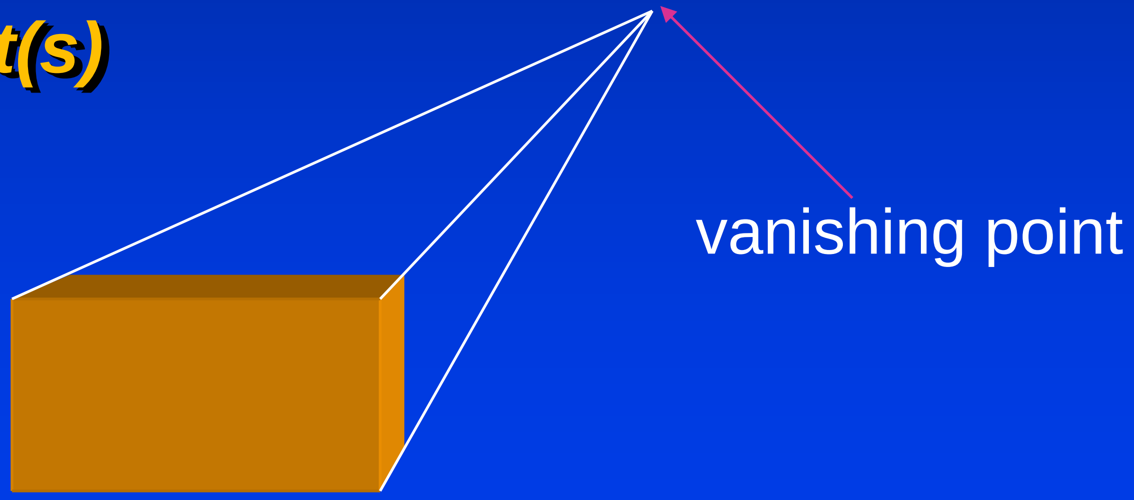
Perspective Projection

Projectors converge at center of projection



Vanishing Points

- ***Parallel lines (not parallel to the projection plan) on the object converge at a single point in the projection (the vanishing point)***
- ***Drawing simple perspectives by hand uses these vanishing point(s)***



Three-Point Perspective

No principal face parallel to projection plane

Three vanishing points for cube

Two-Point Perspective

On principal direction parallel to projection plane

Two vanishing points for cube

One-Point Perspective

One principal face parallel to projection plane

One vanishing point for cube

Advantages and Disadvantages

- ***Objects further from viewer are projected smaller than the same sized objects closer to the viewer (*diminution*)***
    - *Looks realistic*
- ***Equal distances along a line are not projected into equal distances (*non-uniform foreshortening*)***
- ***Angles preserved only in planes parallel to the projection plane***
- ***More difficult to construct by hand than parallel projections (but not more difficult by computer)***