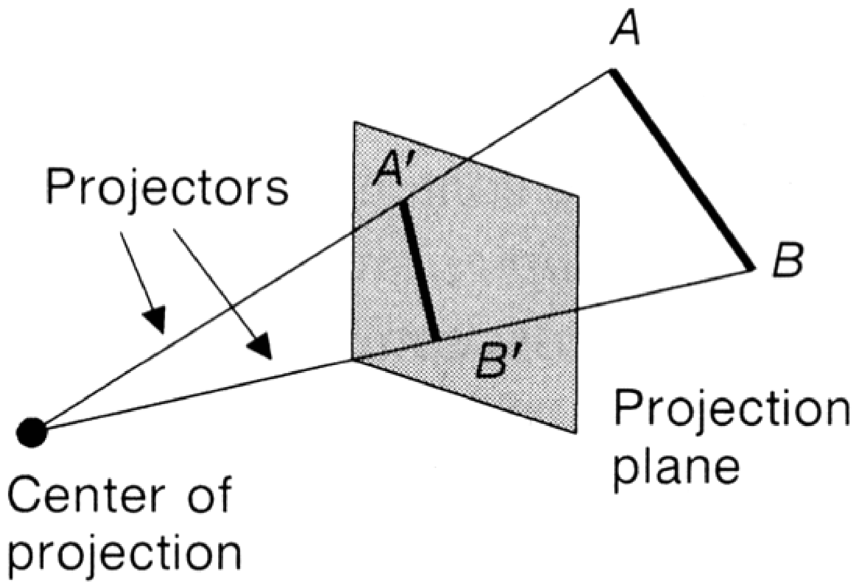
CS5182 Computer Graphics Projection and Clipping

2024/25 Semester A

City University of Hong Kong (DG)

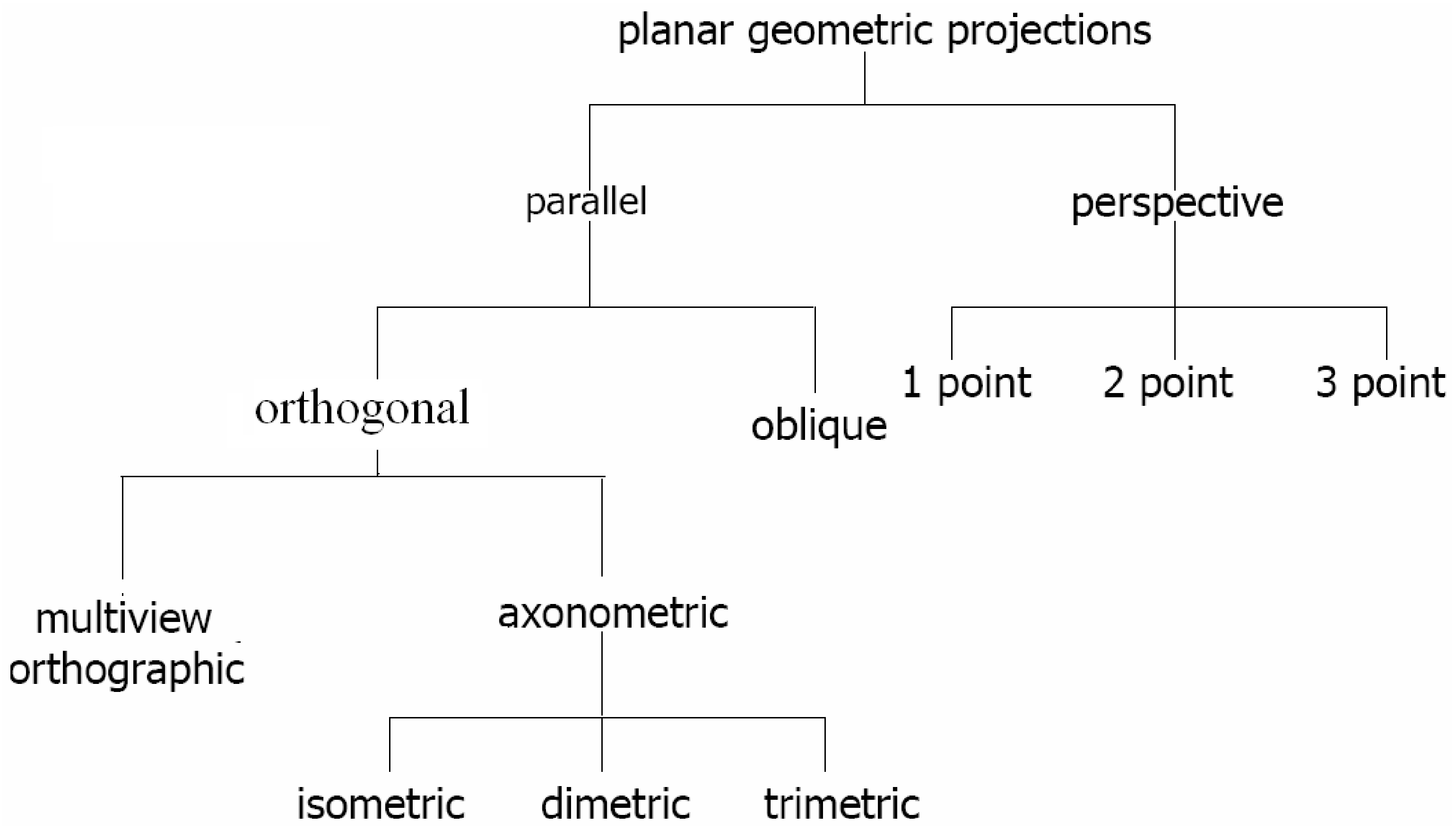
Projection

- Because existing computer screens can only display 2D images, we need to convert a 3D scene into a 2D image. To do this, we project each object in the scene onto a 2D plane.
  - Projection from 3D to 2D is defined by straight **projectors** emanating from the **center of projection** (COP), passing through each point of the object, and intersecting the **projection plane** to form a projection.



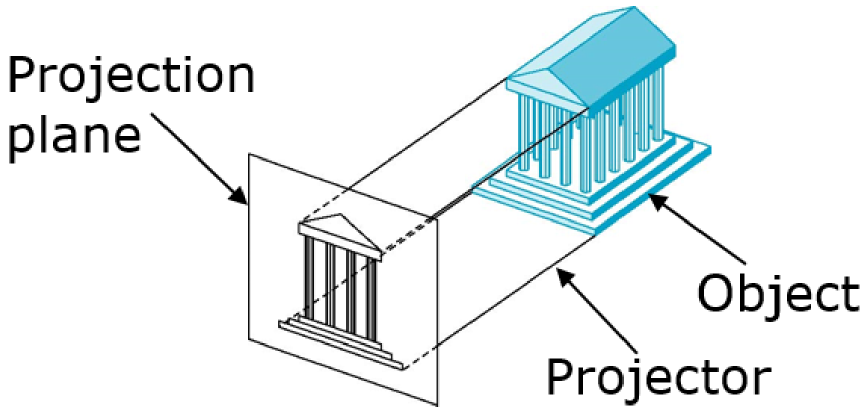
Projection

- Two types of planar projections
  - Parallel projection
  - Perspective projection



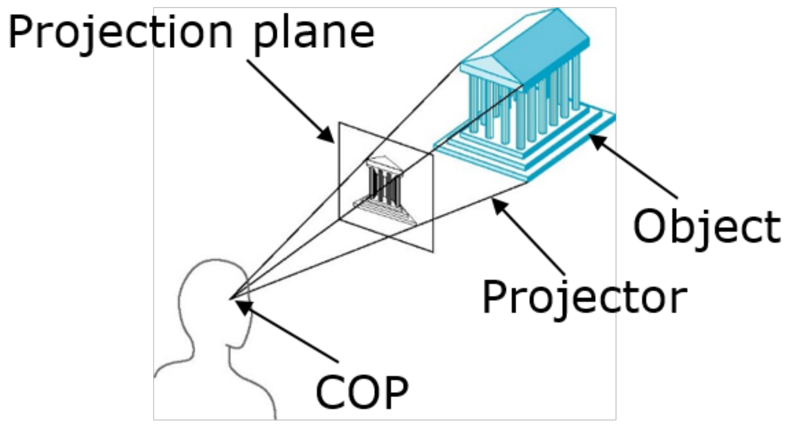
Parallel vs. Perspective Projection

- Parallel projection
  - The COP is at infinity.
  - All projectors are parallel.
  - Parallelism is preserved.



Parallel projection

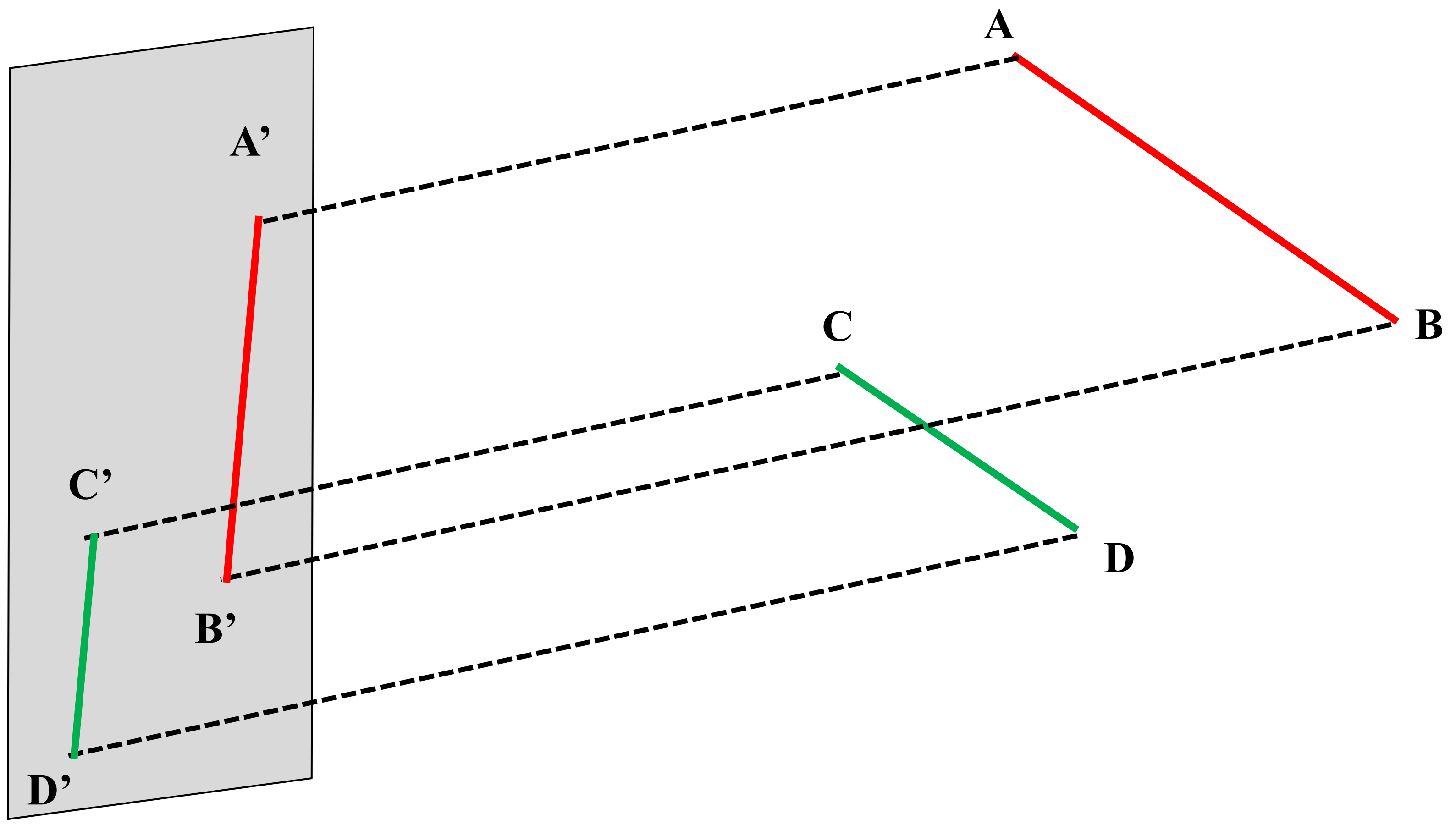
- Perspective projection
  - The COP is at a finite distance from the projection plane.
  - All projectors meet at COP.
  - Parallelism is not preserved in general.



Perspective projection

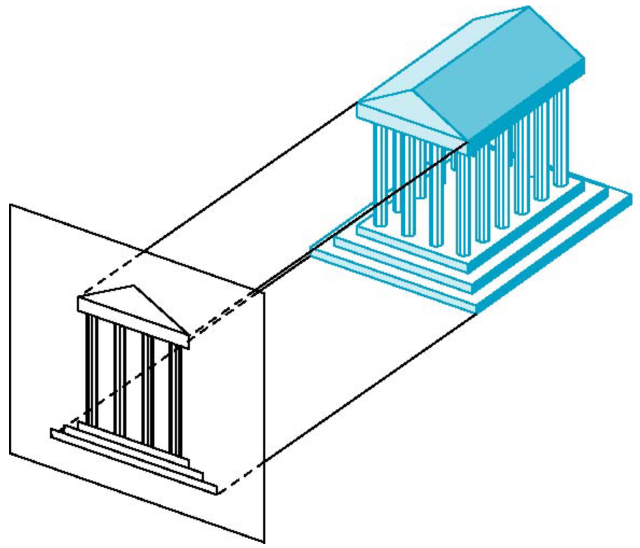
Parallel Projection

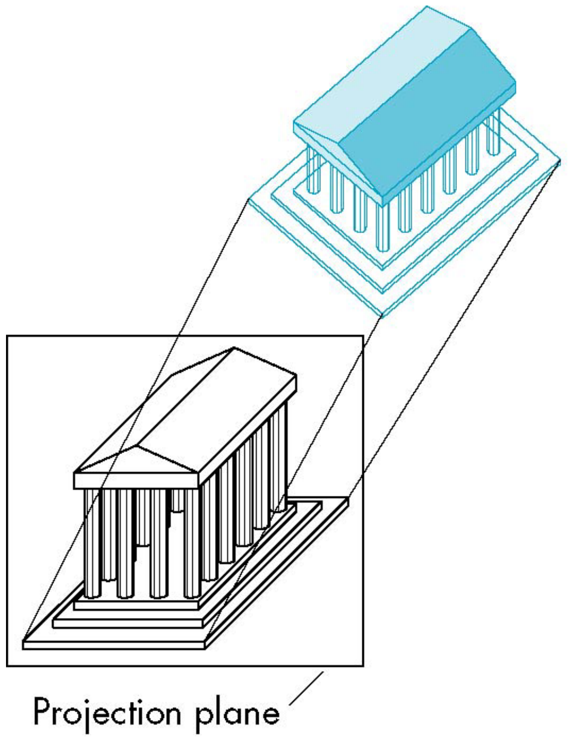
- Parallel projection preserves the parallelism, i.e., parallel lines remain parallel under parallel projection



Parallel Projection

- Two types of parallel projections
  - Orthographic projection if projectors are perpendicular to projection plane.
  - Oblique projection, otherwise.

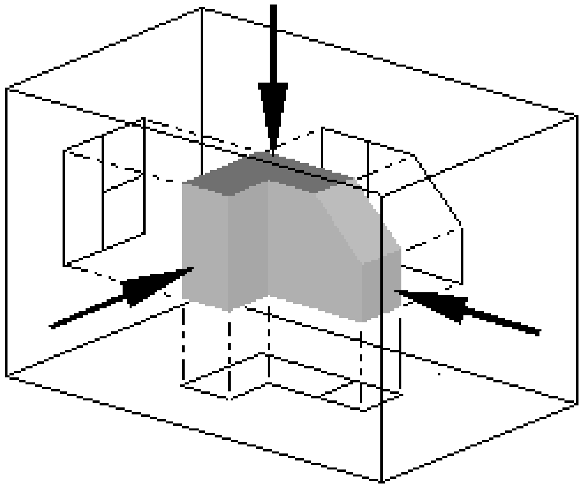

Orthographic projection

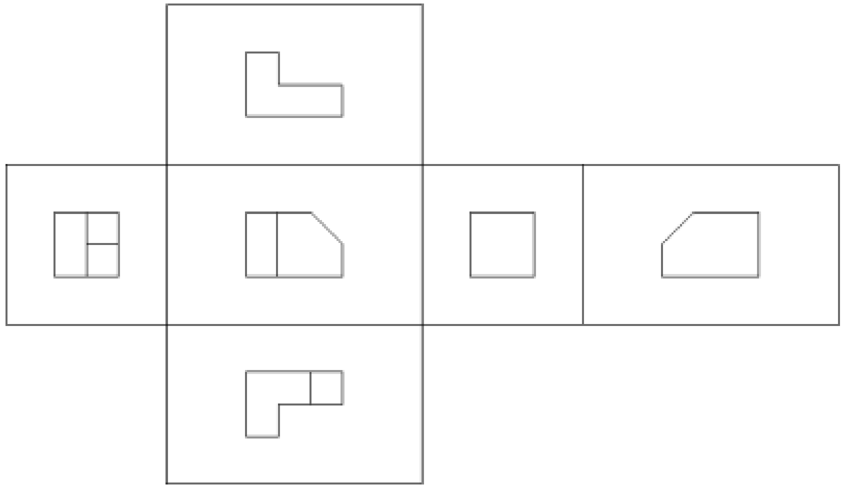


Oblique projection

Multi-view Orthographic Projection

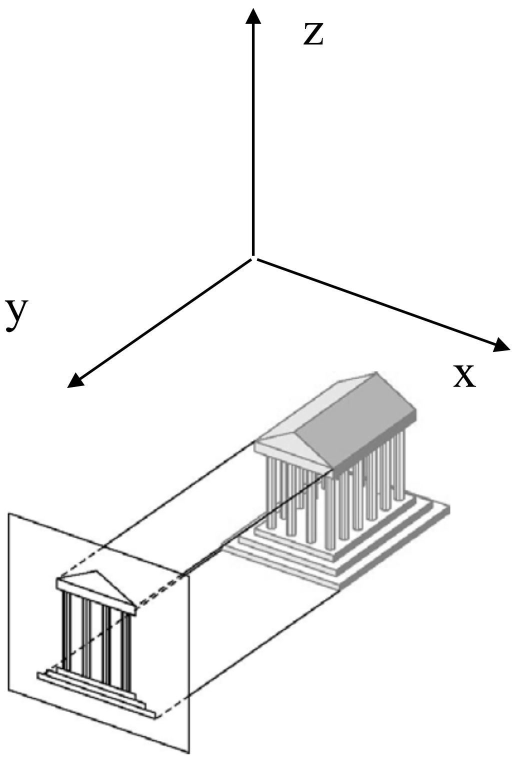
- Projection plane is parallel to one of the coordinate planes
- The appearances of views may be thought of as being *projected* onto planes that form a transparent "box" around the object. Commonly employed in engineering and architectural drawing. One can accurately measure the length if it is parallel to the projection plane



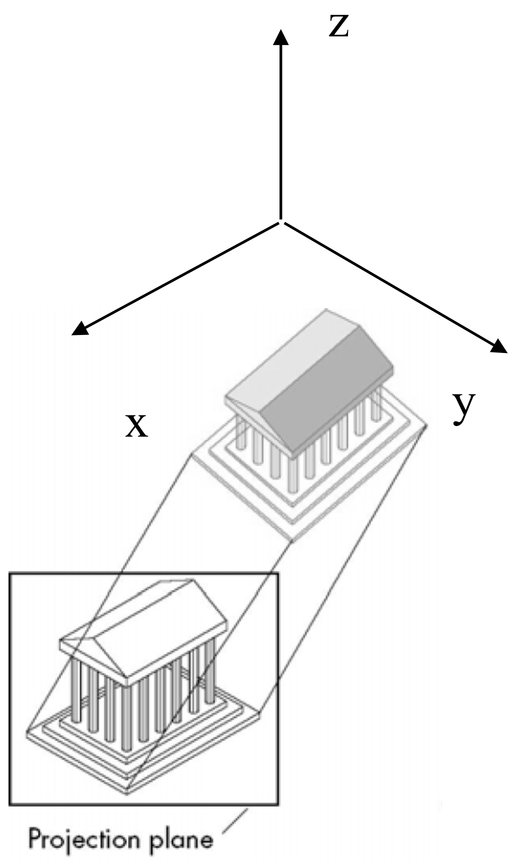


Axonometric Projection

- In contrast to the multiview orthographic projection, projection plane of axonometric projection is not parallel to any of the coordinate planes



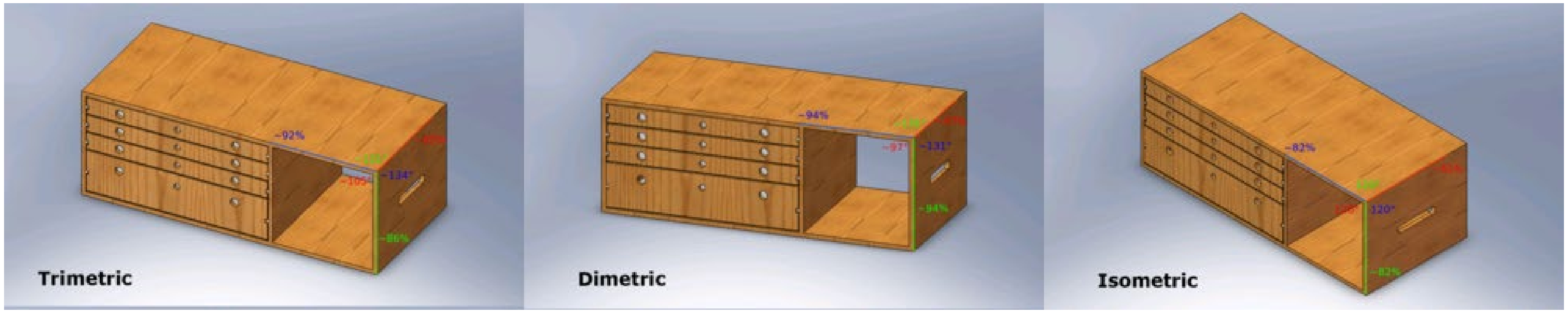
Multiview orthographic projection



Axonometric projection

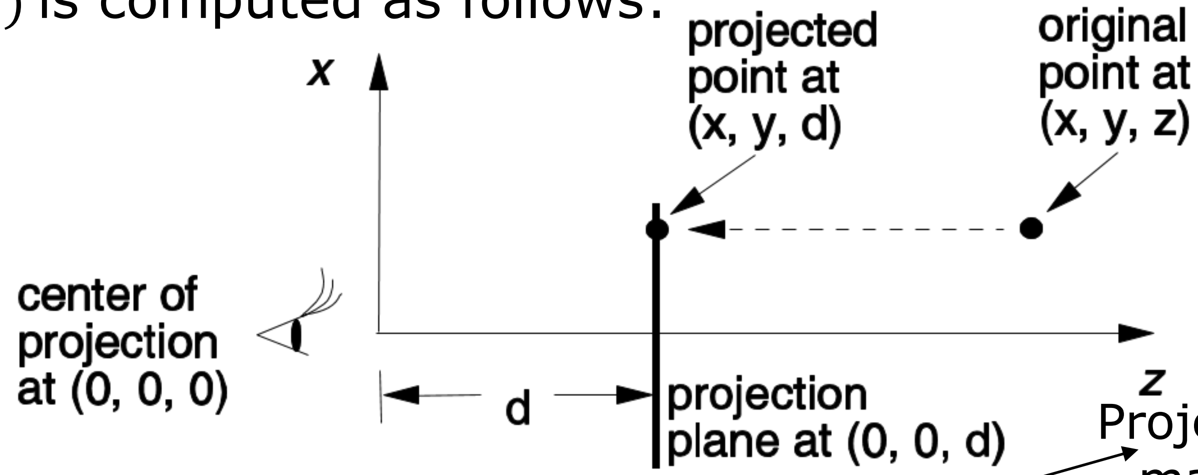
Axonometric Projection

- "Axonometric" means "to measure along axes"
- How many angles of a cube’s corner are equal?
  - **Trimetric**: angles between the three principal axes are different.
  - **Dimetric**: angles between two of the principal axes are equal.
  - **Isometric**: angles between all three principal axes are equal. The projection plane intersects each coordinate axis at the same distance from the origin.



Parallel Projection

- Projection matrix for the orthographic projection
  - Assume that the viewer is located at the origin, i.e., $(0,0,0)$, and the projection plane is located at $(0,0,d)$ and parallel to the *XY*-plane. Given a point (x,y,z), the projected point (x',y',z') is computed as follows:

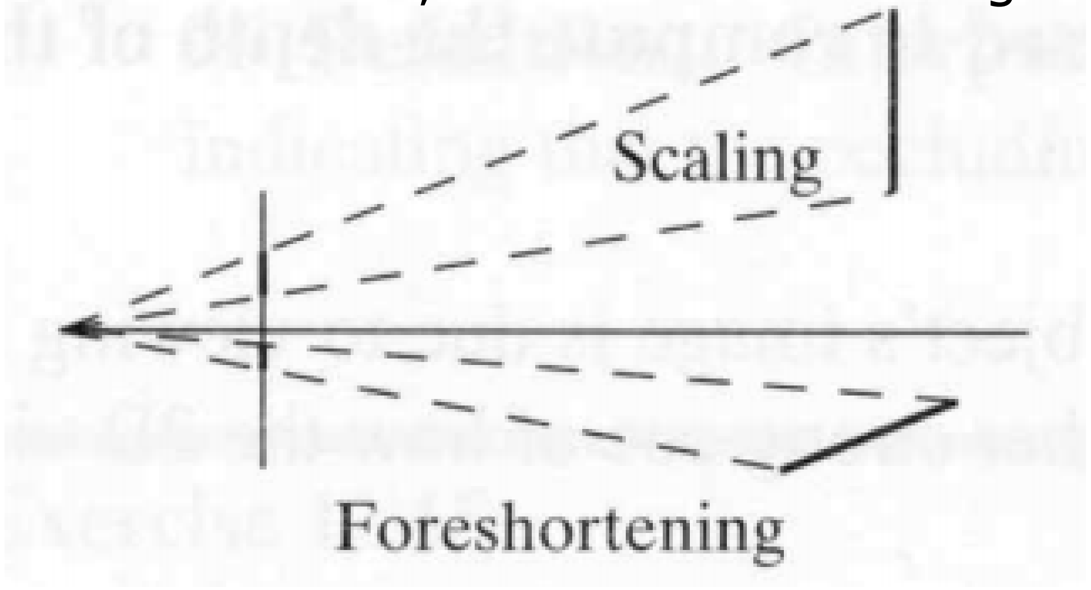

x
projected point at (x, y, d)
original point at (x, y, z)
center of projection at (0, 0, 0)
d
projection plane at (0, 0, d)
z

$$\begin{cases} x' = x \\ y' = y \\ z' = d \end{cases} \Rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \boxed{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Projection matrix

Perspective Projection

- Parallel projection does not give us depth information! For realistic three-dimensional viewing we'd like to require our objects to be displayed with proper perspective.
  - Objects are drawn smaller as their distance from the observer increases, which gives us the sense of depth.
  - If a line is parallel to the projection plane, the effect of perspective projection is a scaling; otherwise, the effect is a perspective distortion, called foreshortening.

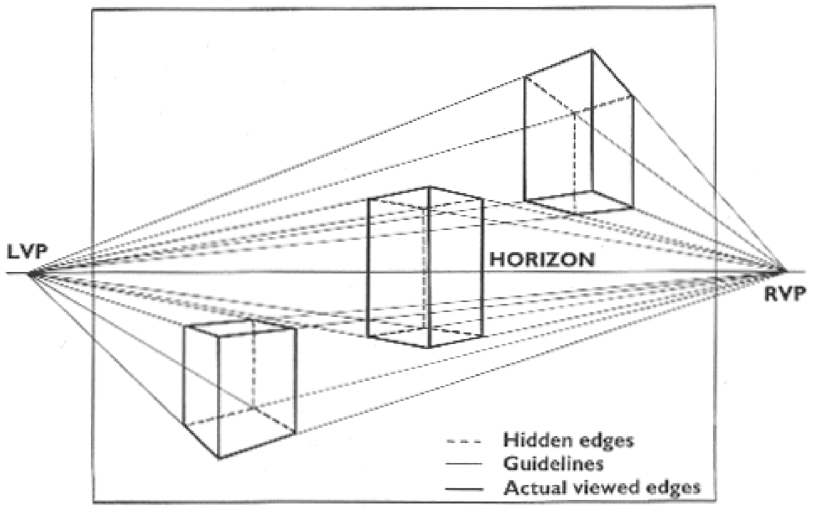


Perspective Projection

- In photography, perspective gives the impression of three dimensional depth and distance in an image
  - Close objects look bigger and distant objects look smaller.
  - Parallelism is **not** preserved in general! Parallel lines could meet on the projection plane.

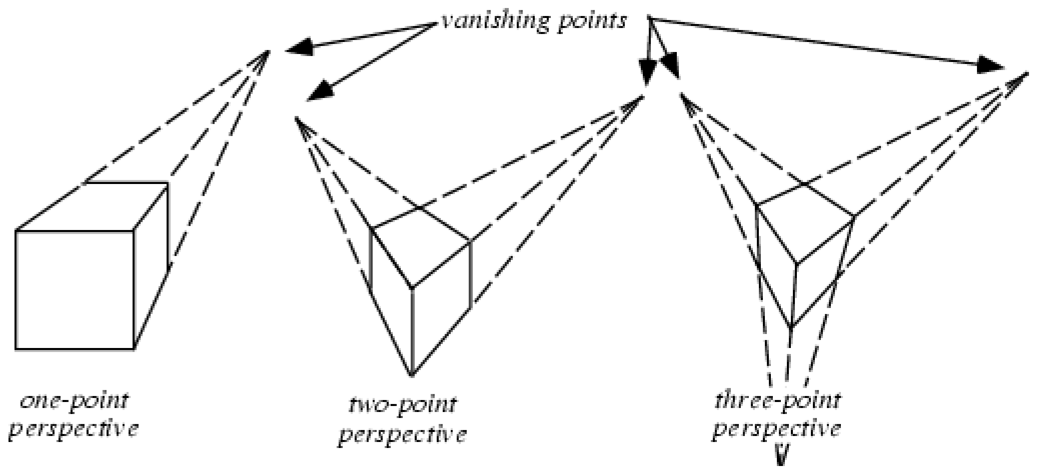
Perspective Projection

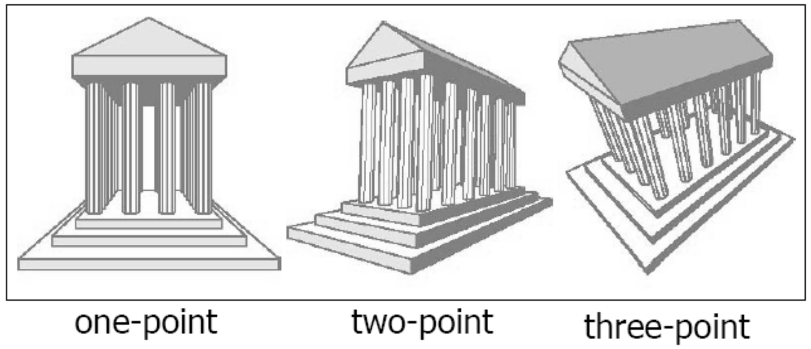
- Vanishing Points
  - Any set of parallel lines that are not parallel to the projection plane will converge to a vanishing point.
  - Each set of parallel lines meets at a different point, i.e., the vanishing point for this direction.
  - Sets of parallel lines on the same plane lead to collinear vanishing points: the horizon for that plane.
  - Any point on the horizon corresponds to a unique direction.



Perspective Projection

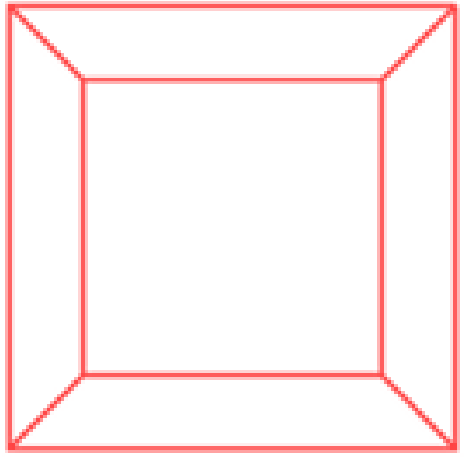
- One-point perspective
- Two-point perspective
- Three-point perspective

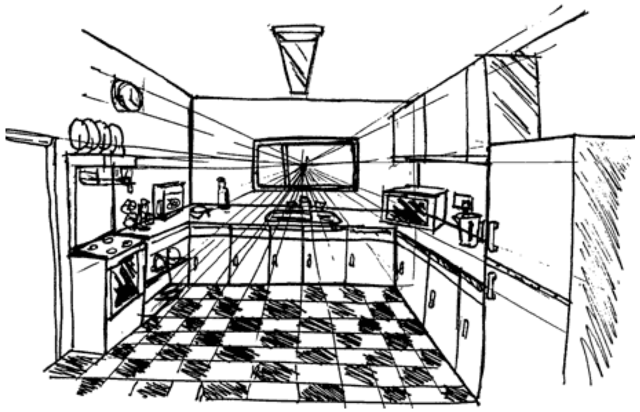


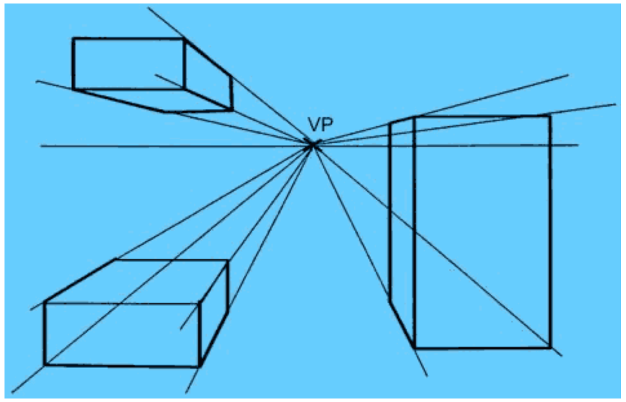


Perspective Projection

- One-Point Perspective Projection
  - The projection plane is parallel to **two** principal axes
  - All elements that are parallel to the projection plane are drawn as parallel lines. All elements that are perpendicular to the painting plate converge at a single point (a vanishing point) on the horizon.
  - One vanishing point

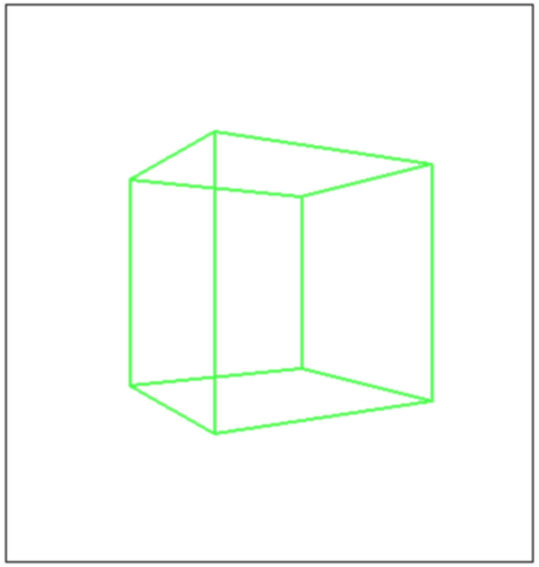


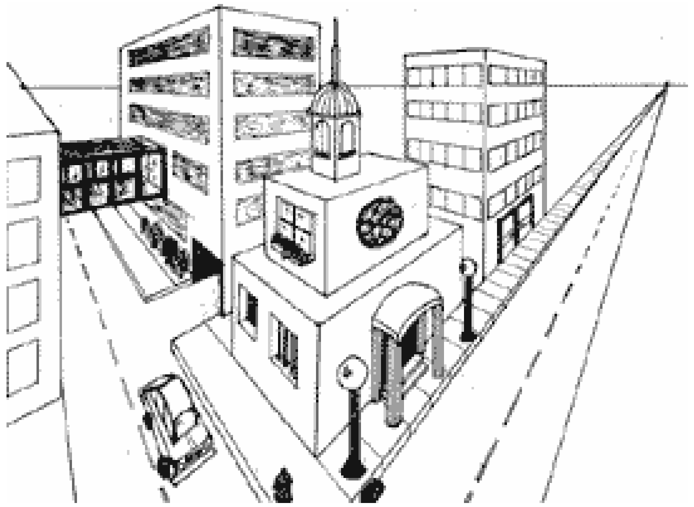




Perspective Projection

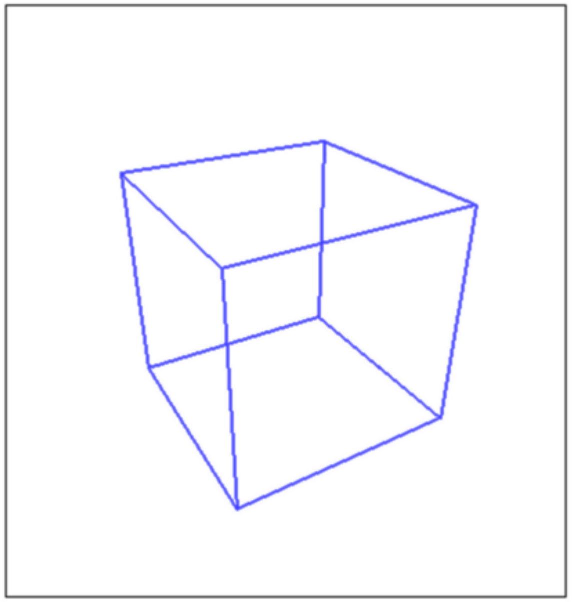
- Two-Point Perspective Projection
  - The projection plane is parallel to exactly **one** principal axis.
  - Two vanishing points

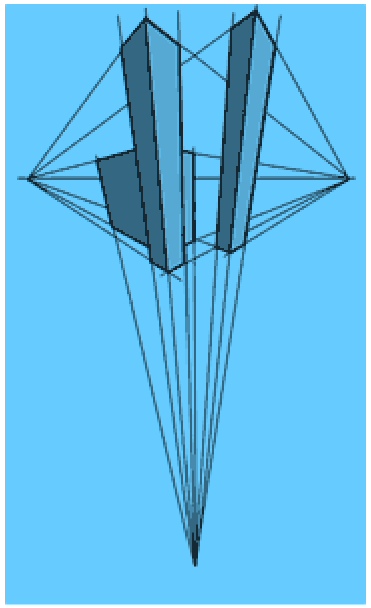




Three-Point Perspective Projection

- Three-Point Perspective Projection
  - The projection plane is **NOT** parallel to any of the principal axes.
  - Each of the three vanishing points corresponds to one of the three axes of the scene.

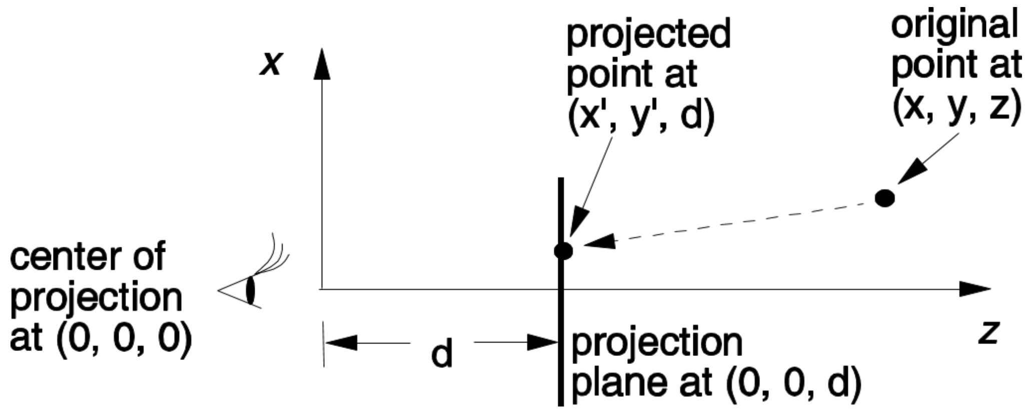




Perspective Projection

- Perspective projection matrix
  - Assume that the viewer is located at the origin, i.e., $(0,0,0)$, and the projection plane is located at $(0,0,d)$ and parallel to *XY*-plane. Given a point (x,y,z), the projected point (x',y',z') is computed as follows:

$$\begin{cases} x' = d \times x/z \\ y' = d \times y/z \\ z' = d \end{cases}$$


x
projected point at (x', y', d)
original point at (x, y, z)
center of projection at (0, 0, 0)
d
projection plane at (0, 0, d)
z

WARNING: These equations are **non-linear**, as z appears in the denominator! How can we convert them into a matrix?

Perspective Projection

- Homogeneous Coordinates
  - A 2D (resp. 3D) coordinate (x, y) (resp. (x, y, z)) can be written as (x_h, y_h, h) (resp. (x_h, y_h, z_h, h)) such that

$$x_h = hx \qquad y_h = hy \qquad z_h = hz$$

where h is a scaling factor. (x_h, y_h, h) (resp. (x_h, y_h, z_h, h)) is called a homogeneous coordinate. Typically, h is set to 1.

- Perspective Projection Matrix

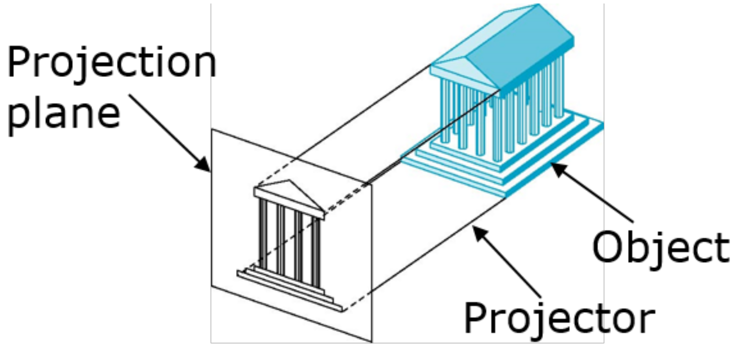
Projection matrix

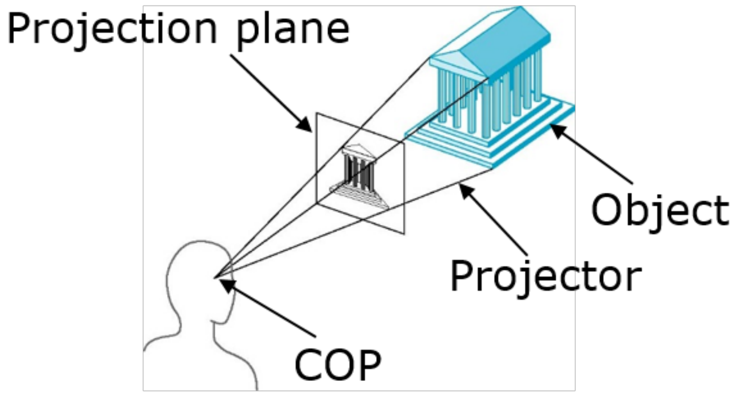
$$\begin{cases} x' = d \times x/z \\ y' = d \times y/z \\ z' = d \end{cases}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \frac{d}{z} \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix} = \frac{d}{z} \boxed{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Perspective vs. Parallel Projection

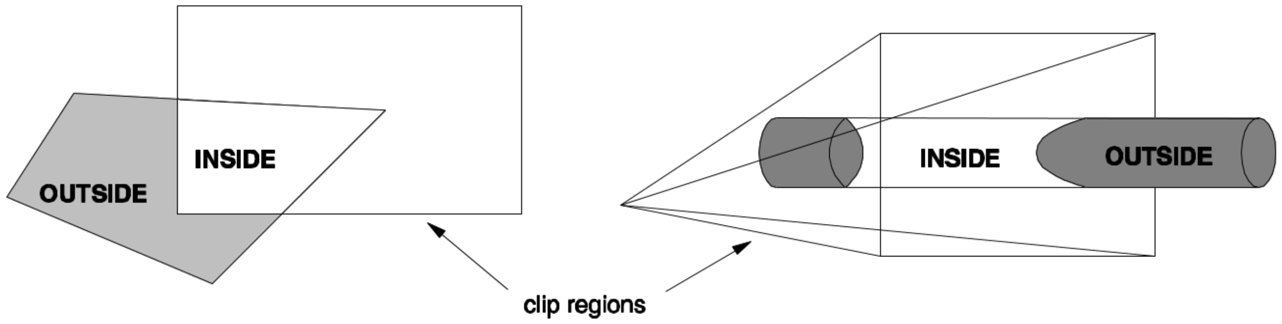
- Perspective projection
  - Similar visual effect to human visual system
  - Size of object varies inversely with distance from the center of projection.
  - Angles only remain intact for faces parallel to projection plane
- Parallel projection
  - Less realistic view because of no foreshortening
  - Parallel lines remain parallel.
  - Angles only remain intact for faces parallel to projection plane.





Clipping

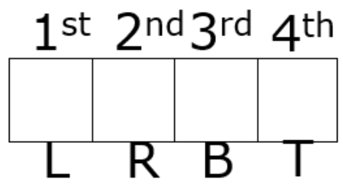
- Clipping is a process to determine the portion of an object lying inside (or outside) a region called the *clip region*.
  - A clip region is typically either a window on a screen (2D) or a view volume (3D).



  - In a window system, clipping helps make sure that the contents outside the window or in the obscured parts of a window do not show up on the screen.
  - In image generation, clipping helps remove objects (or parts of objects) lying outside the view volume, in earlier stage to save computation time.

Clipping

- The Cohen-Sutherland Line-Clipping Algorithm
  - This method aims at quickly identifying lines needed to be clipped and how they should be clipped.
  - Each of the two end points of a line to be clipped is assigned a 4-bit code word.

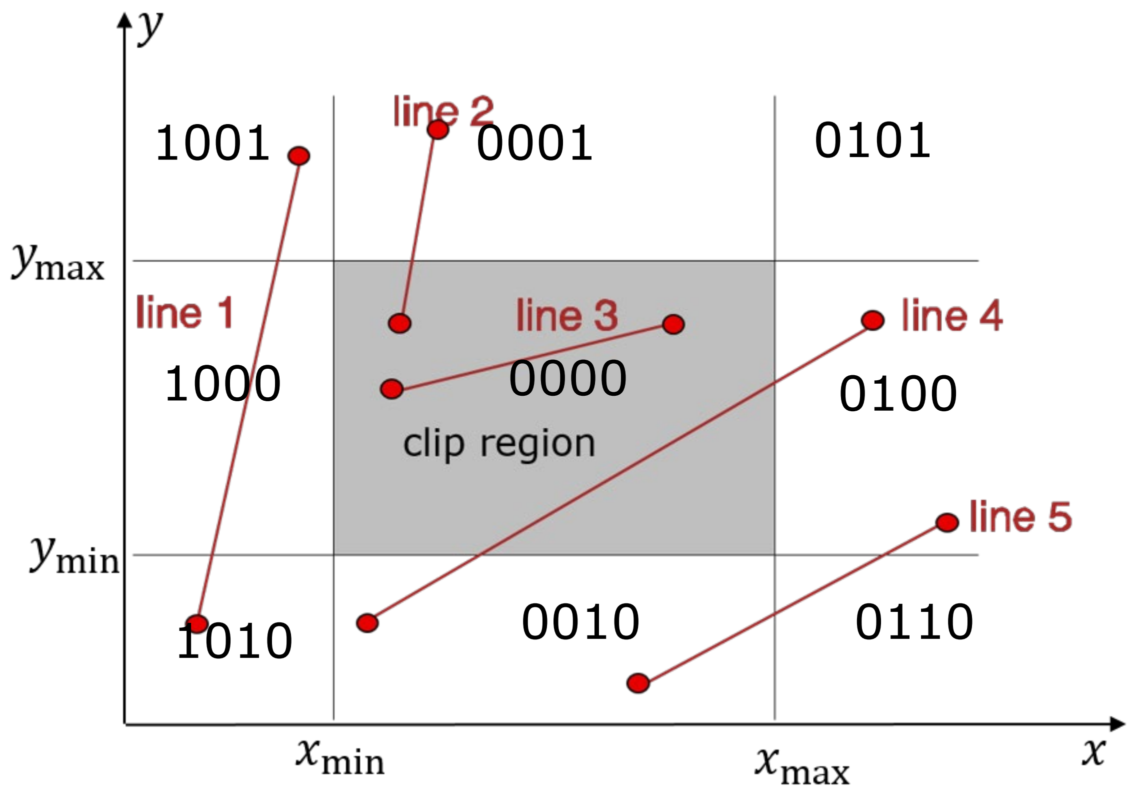


1st bit set 1, If $x < x_{\min}$

2nd bit set 1, If $x > x_{\max}$

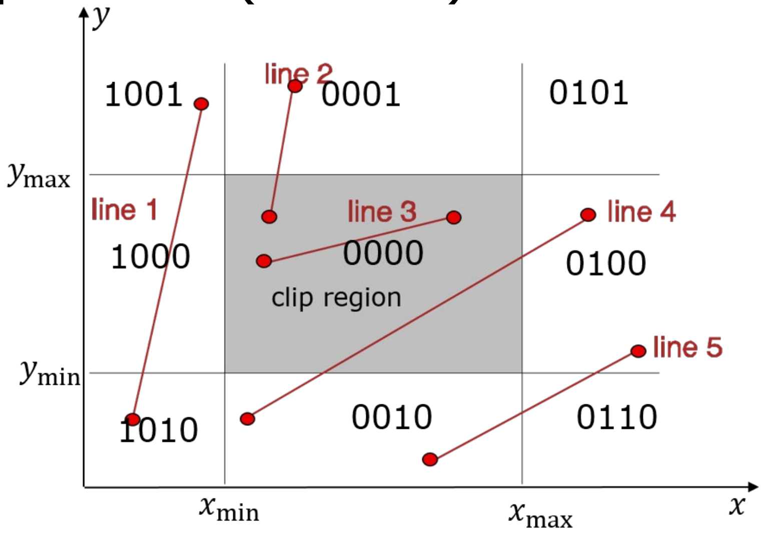
3rd bit set 1, If $y < y_{min}$

4th bit set 1, If $y > y_{\max}$



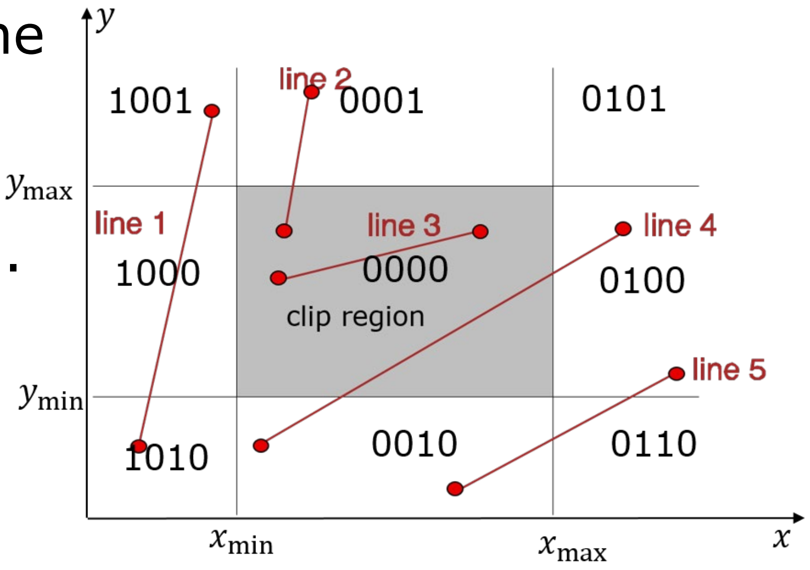
Clipping

- The Cohen-Sutherland Line-Clipping Algorithm
  - **Trivial accept**: If both ends of a line have the same code word as "0000", the line is completely inside the clip region and can be accepted without further operations. The logical **OR** (bitwise) is **zero**. (line 3)
  - **Trivial reject**: If the two code words of a line have the same bit set to "1", the line is completely outside and can be rejected. The logical **AND** operation (bitwise) is **not zero**. (line 1)
  - Otherwise, the line needs to be clipped. (lines 2, 4, 5)



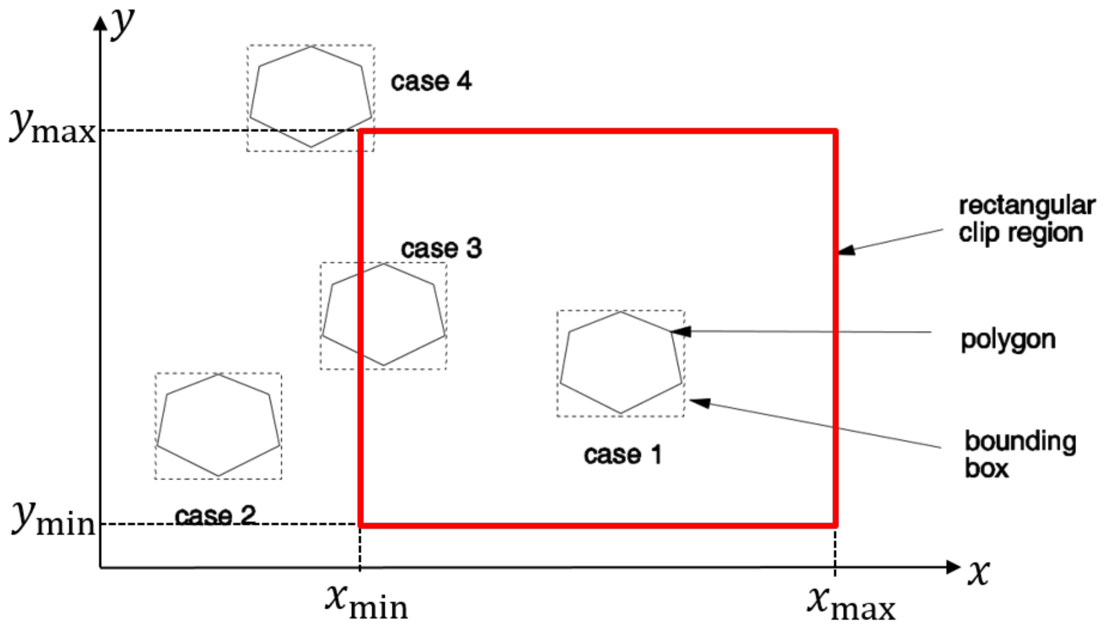
Clipping

- The Cohen-Sutherland Line-Clipping Algorithm
  - Find the boundary of the clip region that the line crossed (from the “1” bit(s) of the code word).
  - Compute the intersection point between this boundary with the line to be clipped.
  - Replace the end point of the line with this intersection point and re-compute the code word.
  - Repeat until the clipped line is either trivially accepted or trivially rejected.



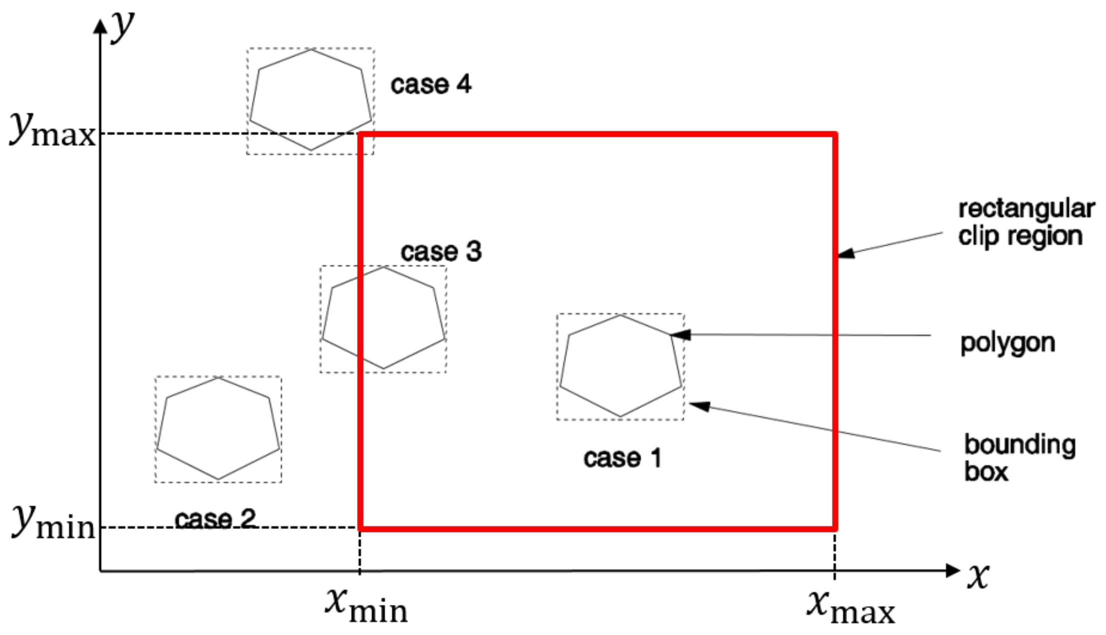
Clipping

- The Sutherland-Hodgman Polygon-Clipping Algorithm
  - Input: 2D polygon; Output: list of clipped vertices
  - If the bounding box is completely outside the clip region (case 2), the polygon is outside the clip region and no clipping is needed. (Compare the coordinates of the bounding box with those of the clip region)
  - If the bounding box is completely inside the clip region (case 1), the polygon is inside the clip region and no clipping is needed.



Clipping

- The Sutherland-Hodgman Polygon-Clipping Algorithm
  - Otherwise, the bounding box of the polygon overlaps with the clip region (case 3 or 4) and the polygon may be partly inside and partly outside of the clip region. So, we need to perform a polygon clipping operation against the rectangular clip region.



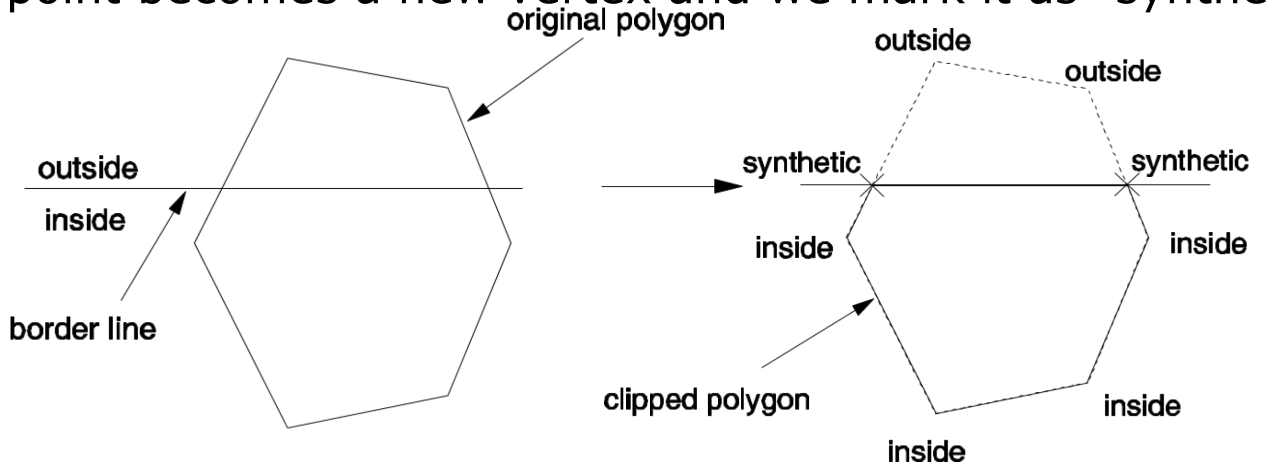
Clipping

- The Sutherland-Hodgman Polygon-Clipping Algorithm
  - The polygon clipping operation is as follows.

1) Using the first vertex as the current vertex. If the point is in the inside of the border line, mark it as "inside"; otherwise, mark it as "outside".

2) Check the next vertex and mark it as "inside" or "outside" accordingly.

3) Compare the current and the next vertices. If one is marked as "inside" and the other as "outside", the edge joining the two vertices crosses the border line. In this case, we need to compute where the edge intersects the border line (i.e., intersection between two lines). The intersection point becomes a new vertex and we mark it as "synthetic".



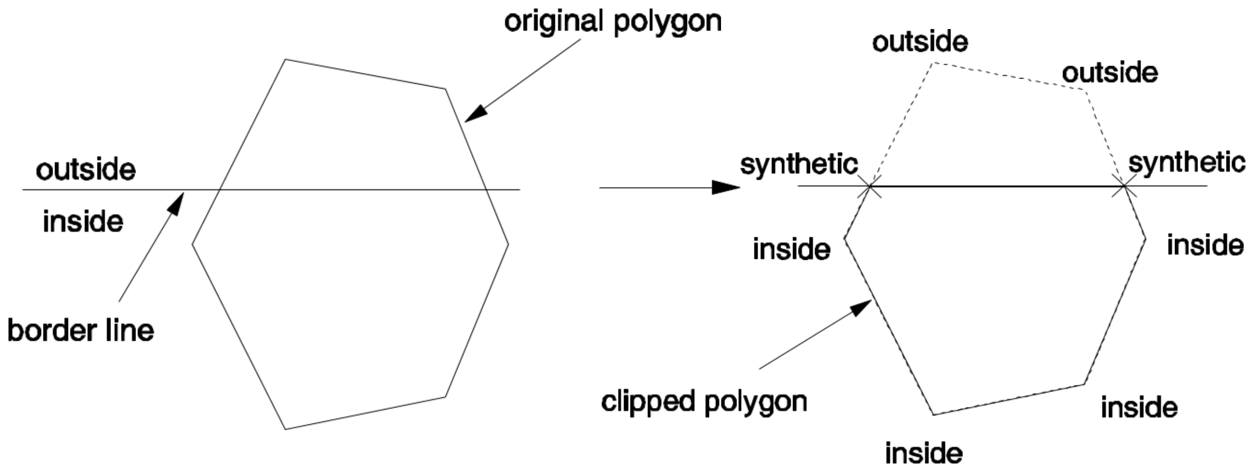
Clipping

- The Sutherland-Hodgman Polygon-Clipping Algorithm

  4) we set the next vertex as the current vertex and the one following it as the next vertex, and we repeat the same operations until all the edges of the polygon have been considered.

  5) After the whole polygon has been clipped by a border, we throw away all the vertices marked as “outside” while keeping those marked as “inside” or “synthetic” to create a new polygon.

  6) We repeat the clipping process with the new polygon against the next border line of the clip region


original polygon
outside
inside
border line
outside
outside
synthetic
synthetic
inside
inside
clipped polygon
inside
inside