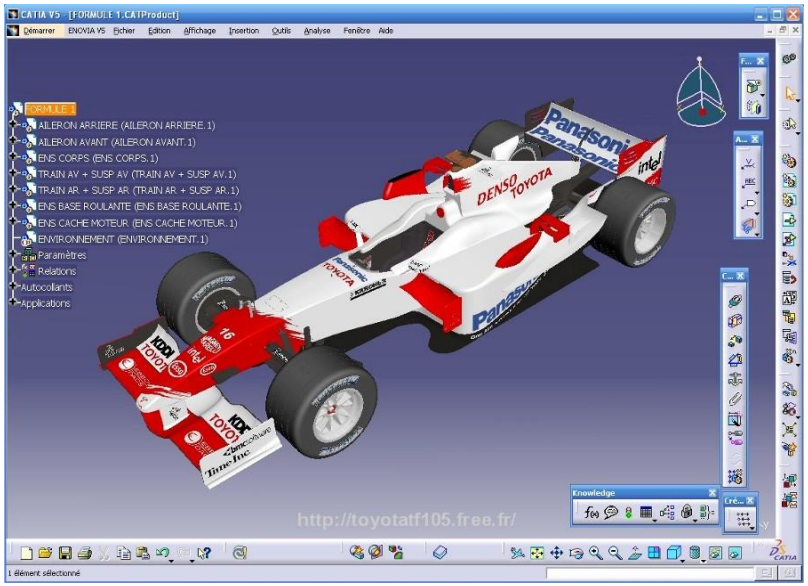
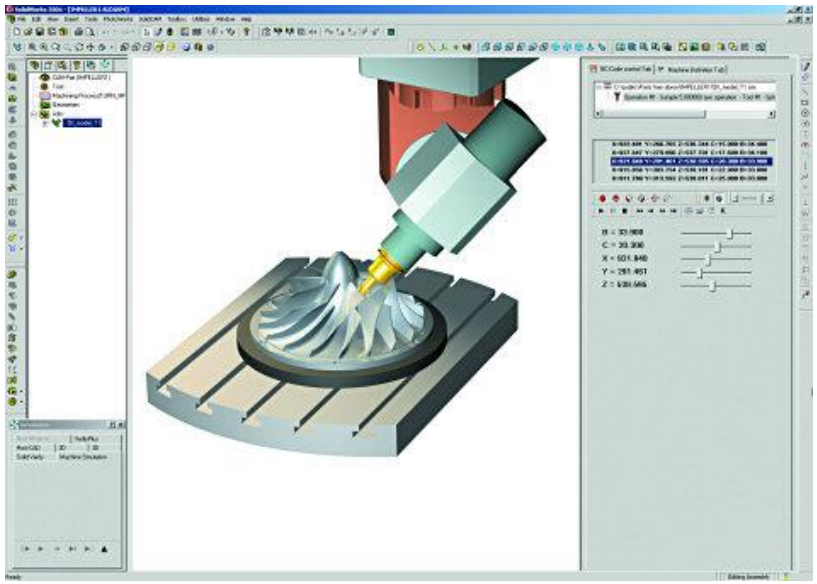
L1 - Introduction

- Contents
  - Introduction of CAD/CAM system
  - Components of CAD/CAM systems
  - Basic concepts of graphics programming





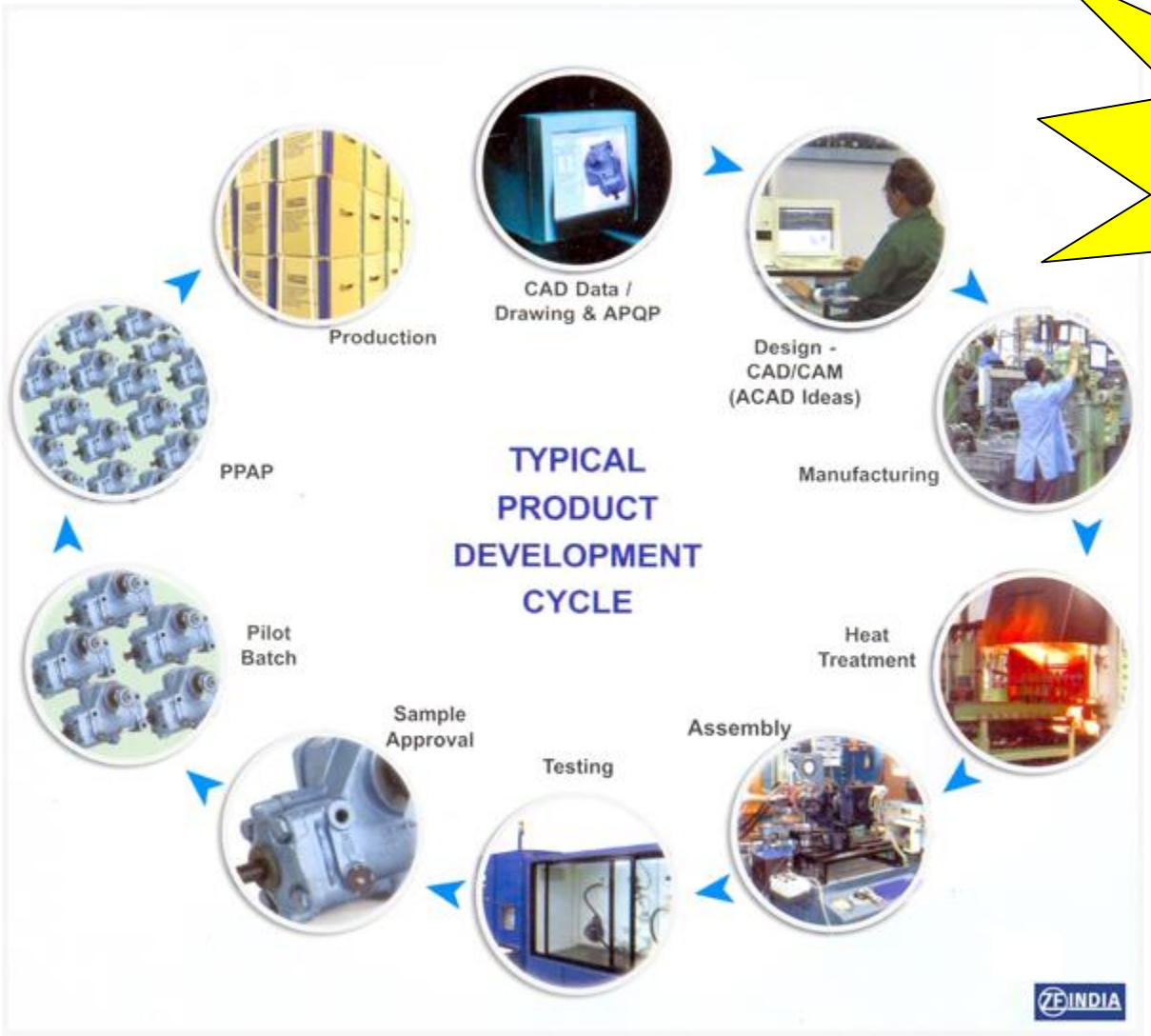
Definitions

- Computer-Aided Design (CAD)
  - The technology concerned with the use of computer system to assist in the creation, modification, analysis, and optimization of a design.
  - A computer program that embodies computer graphics and an application program facilitating engineering functions in the design process
  - From geometric tools for manipulating shapes (at one extreme) to those for analysis and optimization (at the other extreme)
  - Basic role of CAD is to define the geometry of design – because the geometry of the design is essential to all the subsequent activities in the product cycle

Definition (cont.)

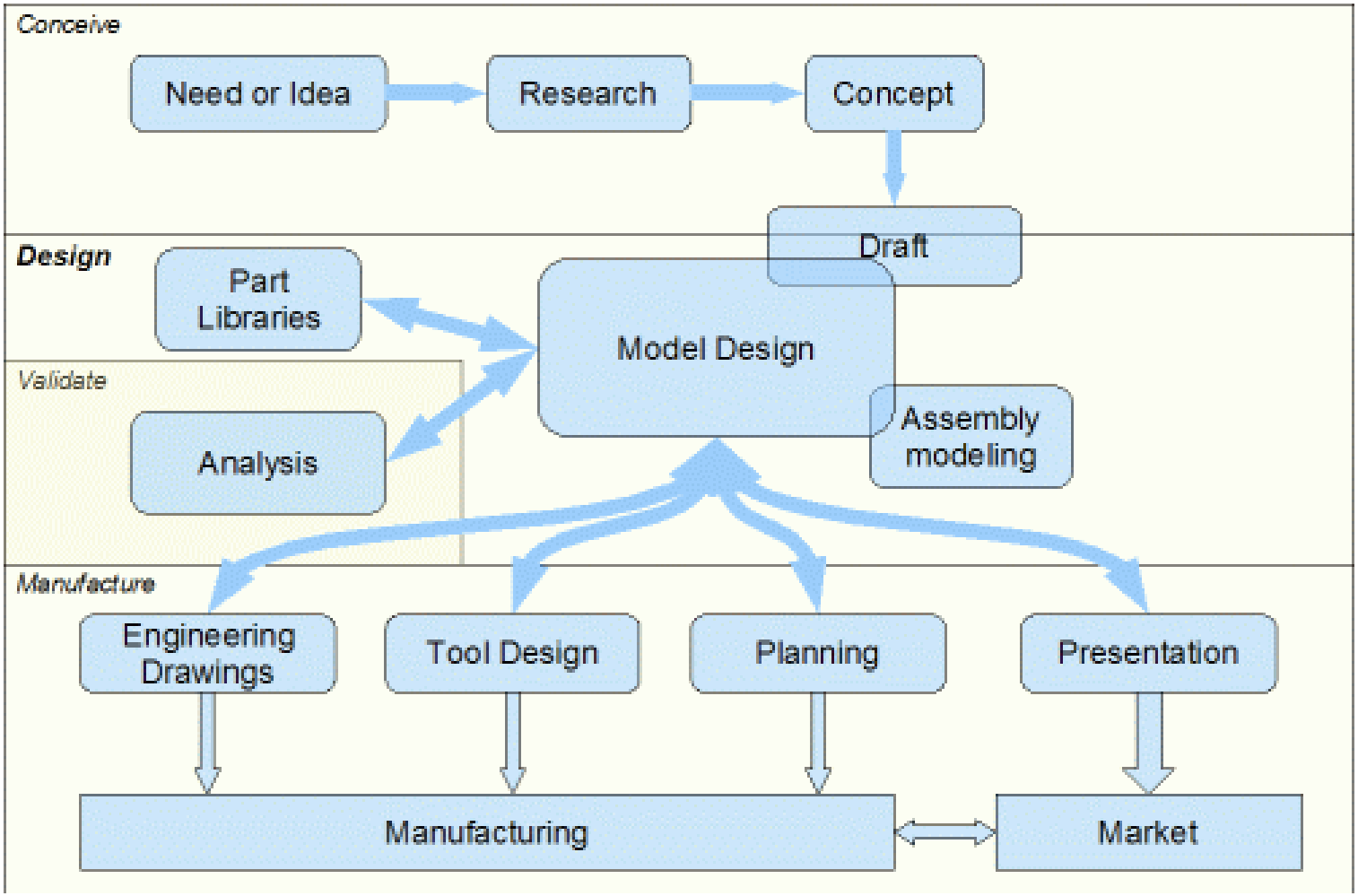
- Computer-Aided Manufacturing (CAM)
  - The technology concerned with the use of computer systems to plan, manage, and control manufacturing operations through either direct or indirect computer interface with the plant's production resources.
  - One of the most mature areas of CAM is numerical control (NC)
    - Control a machine tool for grinding, cutting, milling, punching, bending or turning raw stock into a finished part
  - Another significant CAM function is the programming of robots
    - May operate in a work-cell arrangement, selecting and positioning tools and work-pieces for NC machines

Modern Product Cycle



CAD/CAM plays a very important role

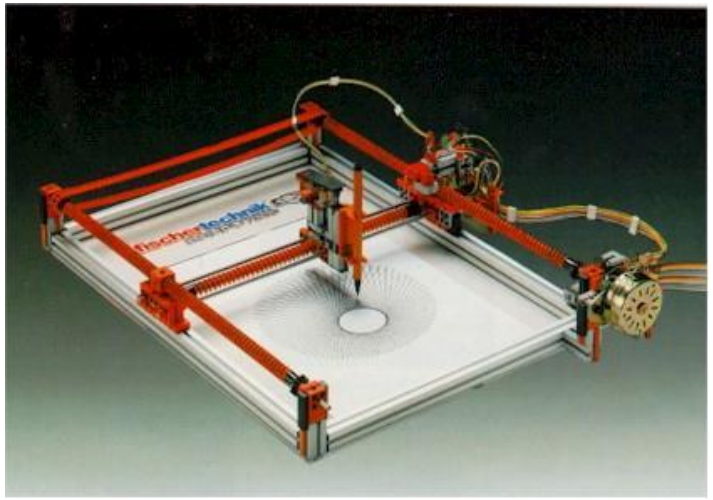
Modern Product Cycle (cont.)



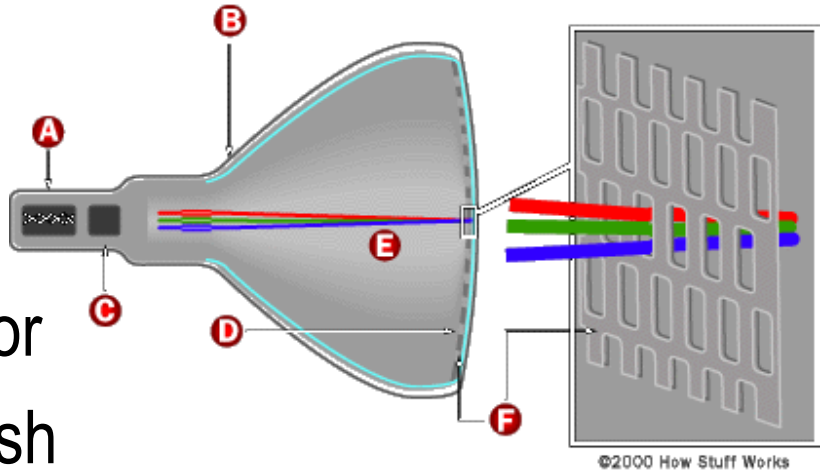
Using CAD/CAM/CAM System for Product Development – Some Examples

- A story of engineering design team (http://www.youtube.com/watch?v=1PoGXiNz_0k)
- NX7 product development productivity redefined (http://www.youtube.com/watch?v=JV4bJGtiuqc)
- Jewelry industry (http://www.youtube.com/watch?gl=US&feature=player_embedded&v=Hu_krFkDyl8)

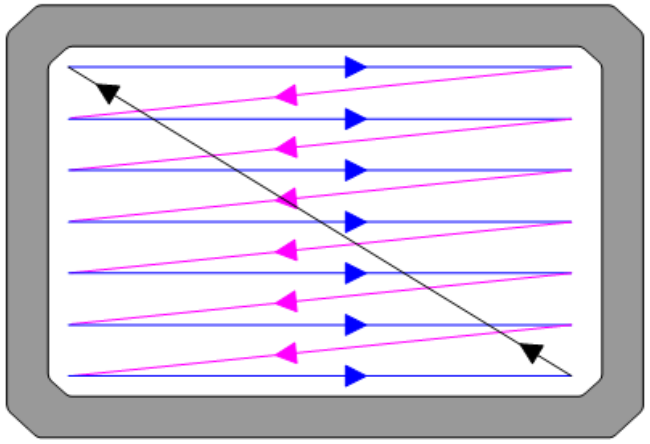
Hardware Components of CAD/CAM Systems



Raster Graphics Devices

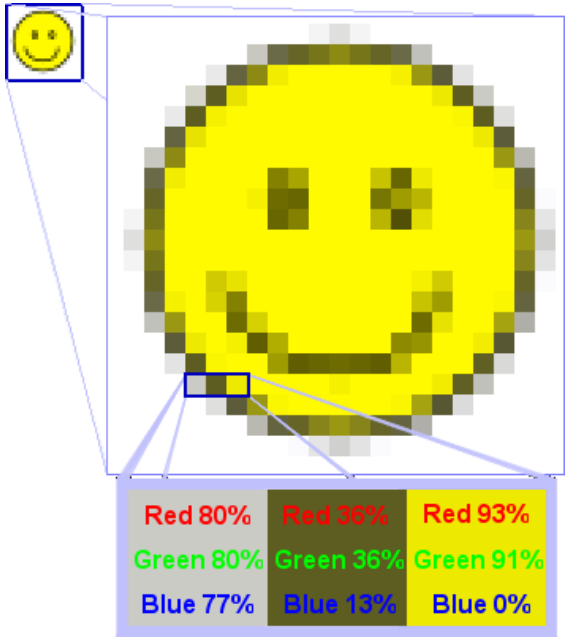


- *Cathode Ray Tube* (CRT) monitor
- Scanning pattern for raster refresh

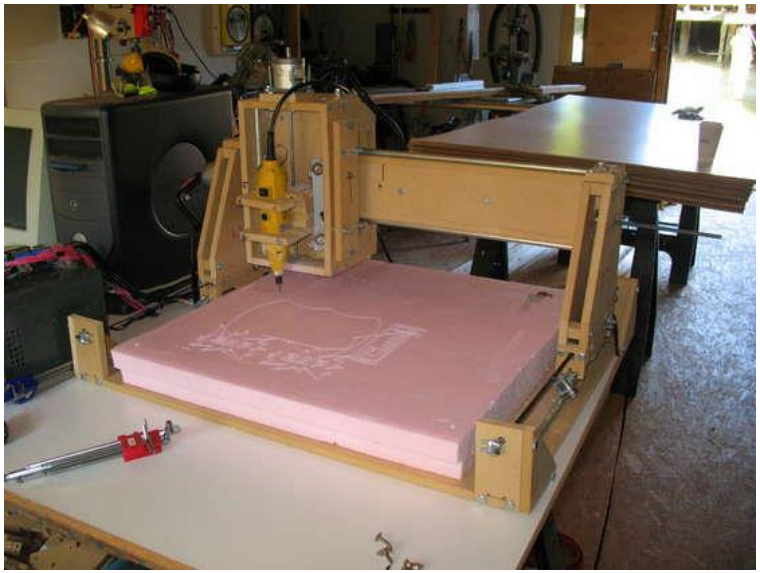


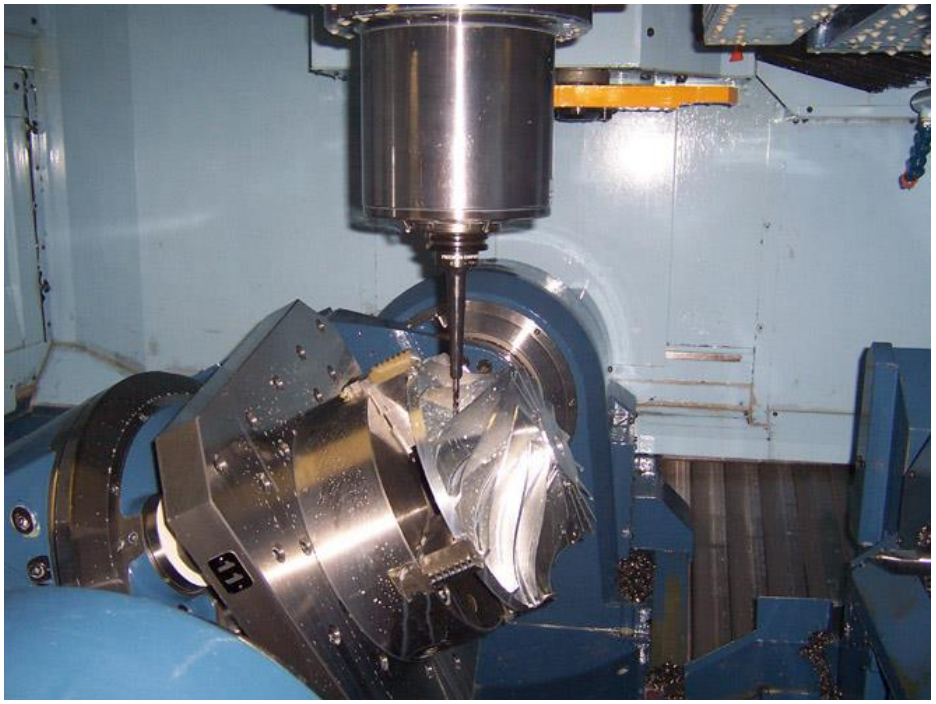
Path of electron beam when using a raster scan technique
(looking at the front of the screen)

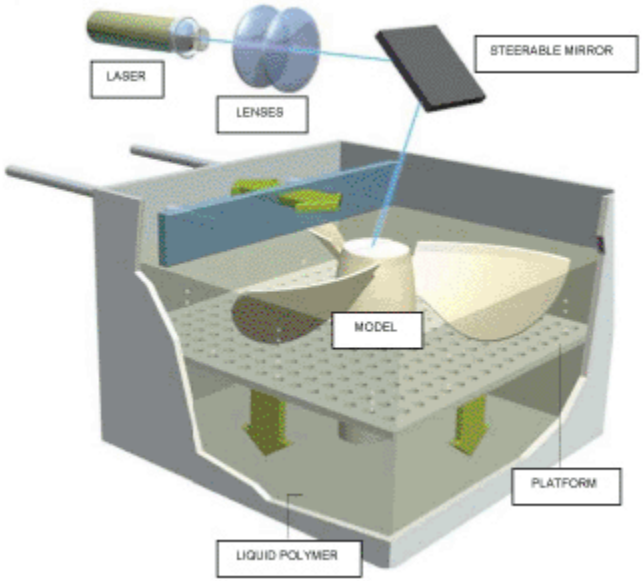
A raster graphics image, or bitmap, is a data structure representing a generally rectangular grid of pixels, or points of color, Viewable via a monitor, paper, etc

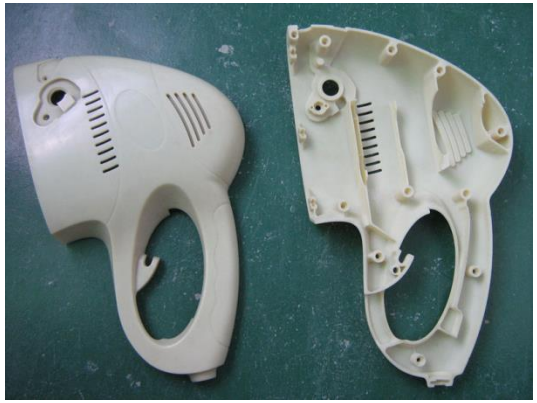


Manufacturing Components of CAD/CAM

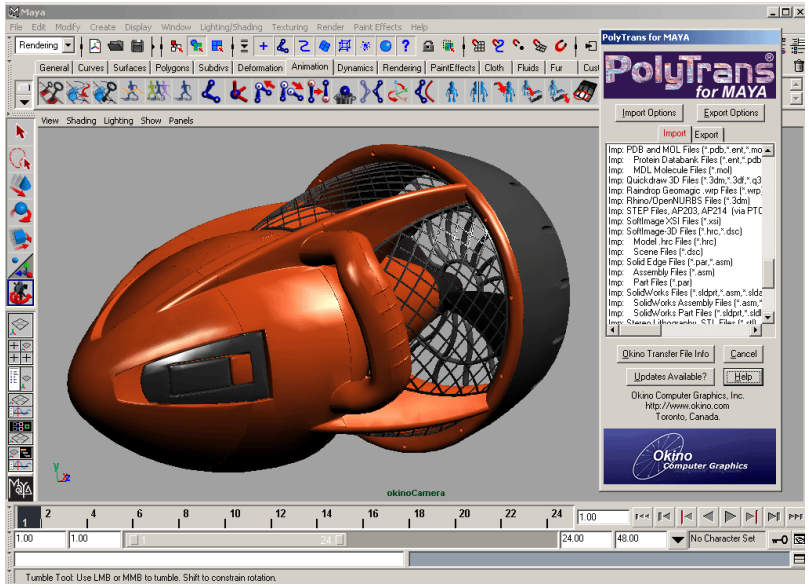


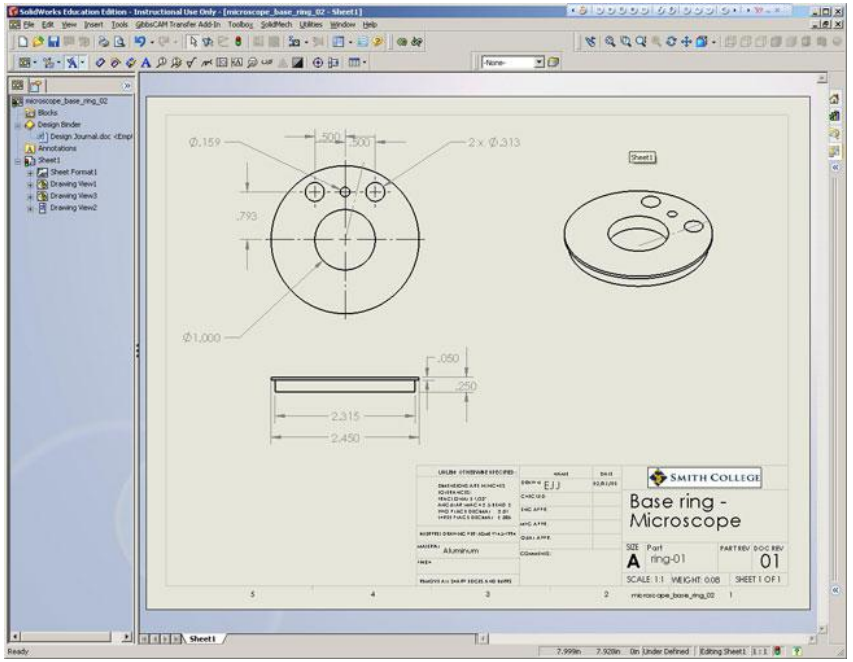


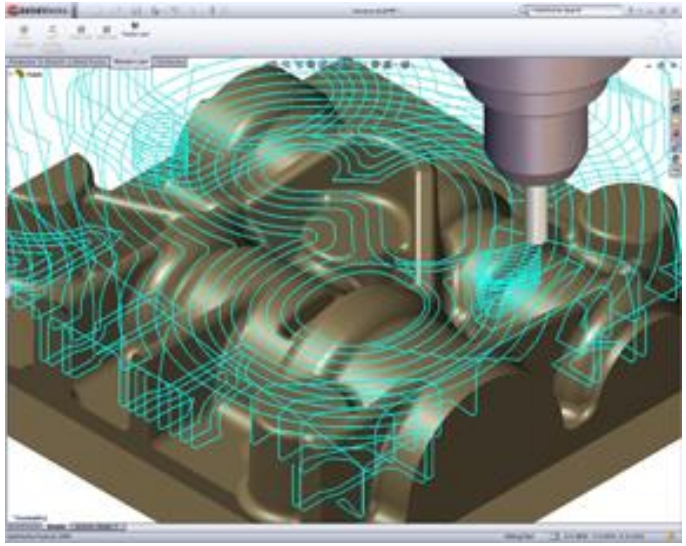


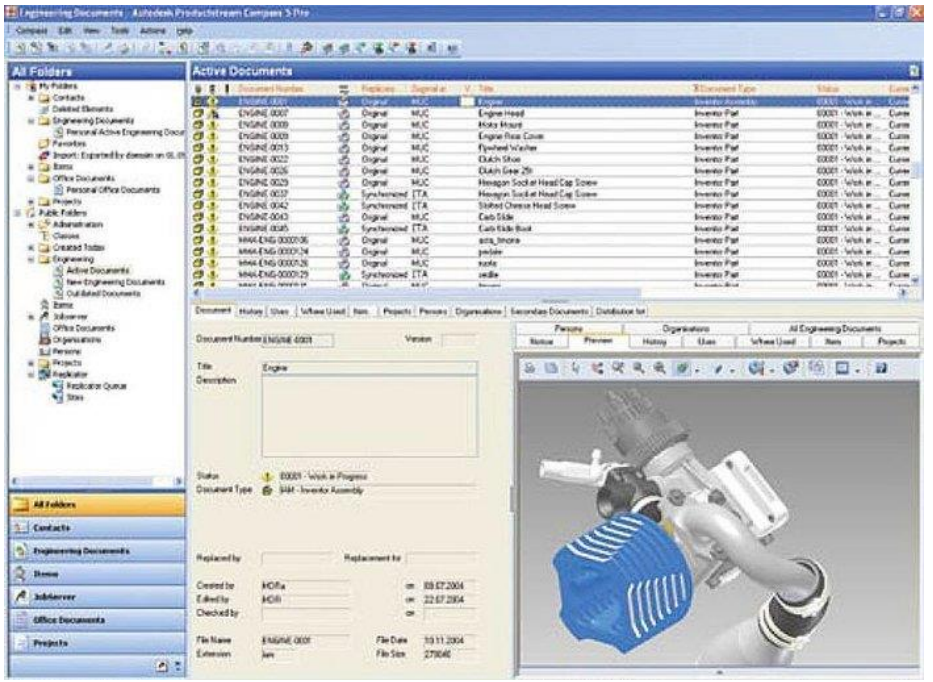


Software Components









Basic Concepts of Graphics Programming

- World coordinate system
  - The unique reference coordinate system used to describe what the world of interest looks like.
- Model coordinate system
  - Each object in the world has its own reference coordinate system, i.e., its model coordinate system. Two models' coordinate systems can be different
- Viewing coordinate system
  - A reference coordinate system used for projecting all the points in the three-dimensional objects onto the computer monitor (two-dimensional) to be seen by human eyes. By convention, in a viewing coordinate system, the z-axis is the viewing direction and is perpendicular to the monitor screen.

Coordinate Systems Conversion

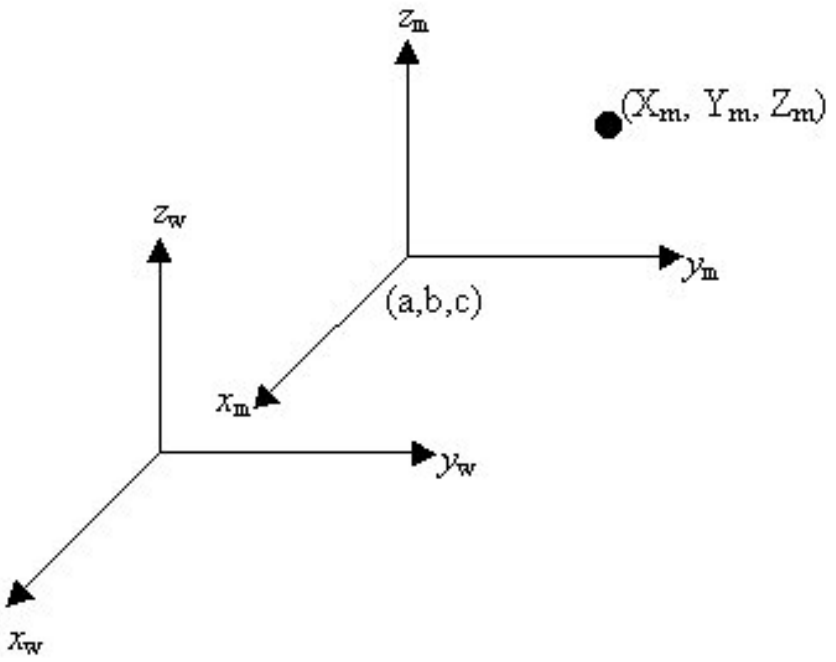
- Model to World coordinate system
  - Different parts of their own MCS assembled together with a single WCS.
- World to View coordinate system
  - Convert WCS to VCS, and project the object onto the computer screen (parallel and perspective projection).
- VCS to virtual device coordinate system
  - Convert the rectangle of the screen in VCS to a normalized 2D coordinate system (u,v).
- VDCS to specific device coordinate system

Transformation Matrix

- To convert a point in one coordinate system to another (for example, from MCS to WCS), a 4x4 matrix is used to achieve the task.
- Different types of transformation matrix
  - Translation
  - Rotation about one coordinate axis
  - Rotation about an arbitrary coordinate axis
  - Composition of translations and rotations
  - Others

Translation

- The x-y-z axes of MCS are parallel to the x-y-z axes of WCS

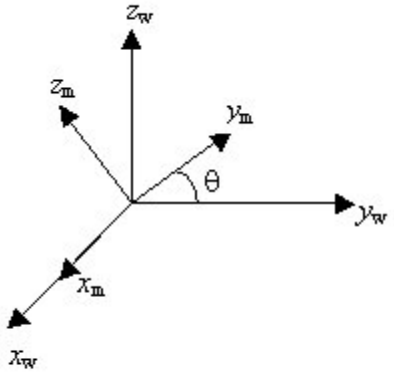


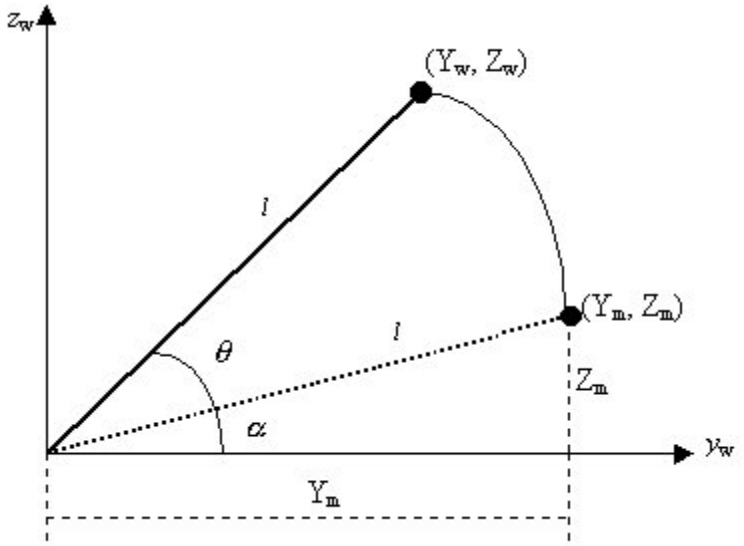
$$X_w = X_m + a$$
$$Y_w = Y_m + b$$
$$Z_w = Z_m + c$$

$$\begin{bmatrix} Xw \\ Yw \\ Zw \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix}$$

Rotation about the x-axis

- MCS is rotated about the x-axis of WCS, with the origin unchanged.





$$X_w = X_m$$

$$Y_w = l\cos(\theta+\alpha) = l\,(\cos\theta\cos\alpha - \sin\theta\sin\alpha) = l\cos\alpha\cos\theta - l\sin\alpha\sin\theta$$
$$= Y_m \cos\theta - Z_m \sin\theta$$

$$Z_w = l\sin(\theta+\alpha) = l\,(\sin\theta\cos\alpha + \cos\theta\sin\alpha) = l\cos\alpha\sin\theta + l\sin\alpha\cos\theta$$
$$= Y_m \sin\theta + Z_m \cos\theta$$

$$\begin{bmatrix} Xw \\ Yw \\ Zw \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix}$$

Rotation about the Principle Axes

- MCS is rotated about the x-, y- or z-axis of WCS, with the origin unchanged.

$$Rot(x,\theta)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & -\sin\theta & 0\\ 0 & \sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

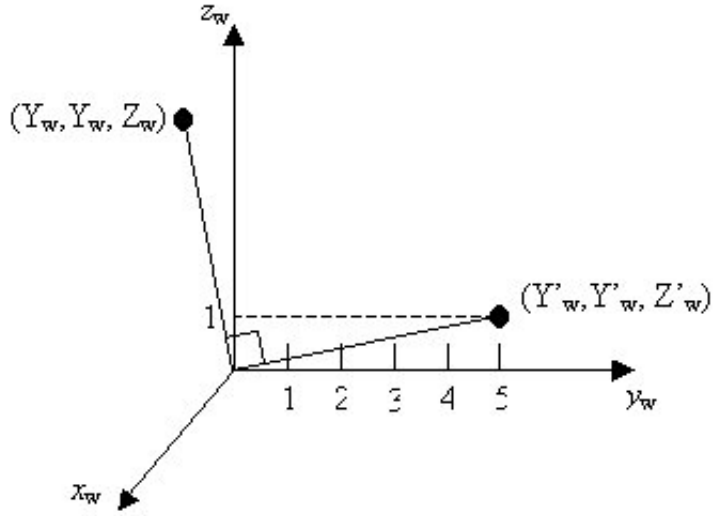
$$Rot(y,\theta)=\begin{bmatrix}\cos\theta & 0 & \sin\theta & 0\\ 0 & 1 & 0 & 0\\ -\sin\theta & 0 & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$$Rot(z,\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0 & 0\\ \sin\theta & \cos\theta & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Composition

- Series of translations and rotations concatenated together in a single matrix

A point (0, 0, 1) is translated by 5 units in the y direction and then rotated by 90 degree about the x-axis. What is its coordinates after these two transformations?



$$[X'_w, Y'_w, Z'_w\ 1]^T = \text{Trans}(0,5,0) \bullet [0\ 0\ 1\ 1]^T = [0\ 5\ 1\ 1]^T$$

$$[X_w, Y_w, Z_w\ 1]^T = \text{Rot}(x, 90°) \bullet [0\ 5\ 1\ 1]^T$$

Or in concatenation form:

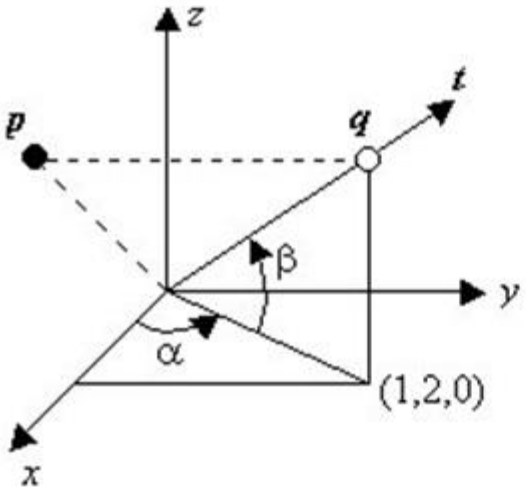
$$[X_w, Y_w, Z_w\ 1]^T = \text{Rot}(x, 90°) \bullet \text{Trans}(0,5,0) \bullet [0\ 0\ 1\ 1]^T$$

How about switch the order of the transformations? That is, first we rotate and then translate, will the new point be the same?

Rotation about an Arbitrary Axis

- MCS is rotated about an arbitrary vector in WCS, with the origin unchanged.

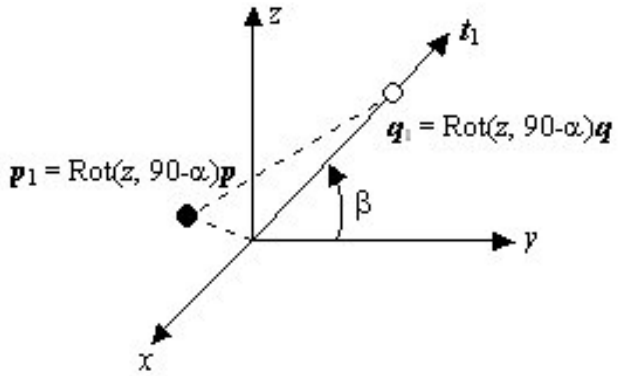
Axis t goes through the origin and a point $q = (1,2,1)$. Point $p = (0, -1,1)$ is rotated about t by an angle θ. What is the coordinates of p after rotation?



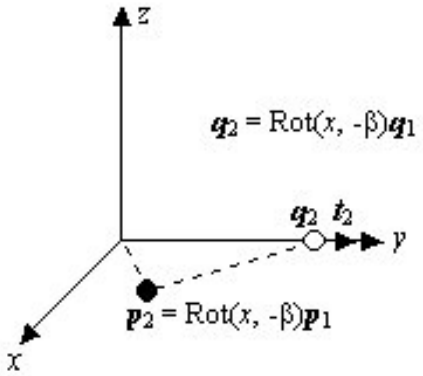
$$\alpha = \text{acos}(1/\sqrt{1^2+2^2})$$

$$\beta = \text{acos}(\sqrt{1^2+2^2}/\sqrt{1^2+2^2+1^2})$$

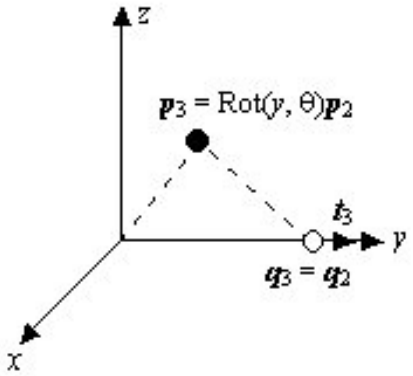
Solution: consider the rigid triangle of <origin, p, q>. Rotate this triangle so that t aligns with one of the three principle axes; then rotate p about the aligned principle axis. After this rotation, transform the new triangle back to align with the original t.



Step 1. Rot(z, 90-α) so ***t*** lies in *y*-*z* plane



Step 2. Rot(x,-β) so ***t*** aligns with *y*-axis



$p_3 = \text{Rot}(y,\theta) \bullet \text{Rot}(x,-\beta) \bullet \text{Rot}(z,90-\alpha) \bullet p$

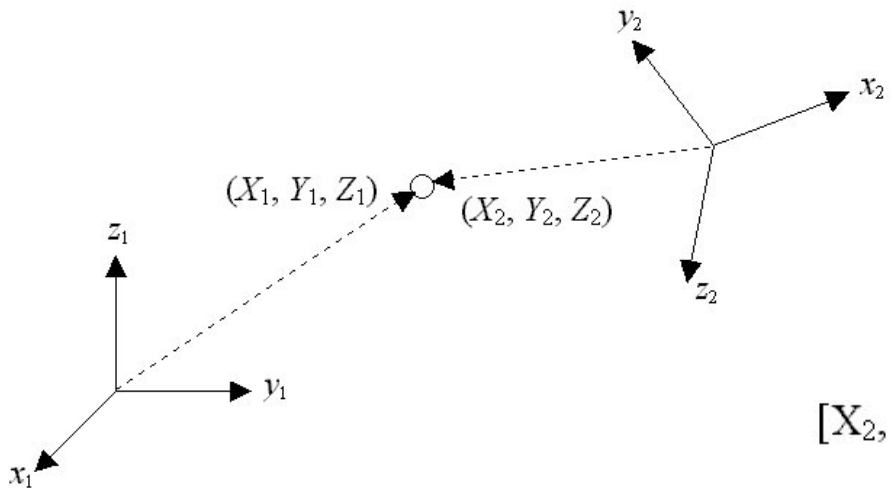
Step 3. Rot(*y*, θ) (***q***2 remains unchanged)

After transform the triangle <origin, ***p***3, ***q***3> back to align with the axis t, the coordinates of the ***p*** after rotated about t by an angle q is:

$$p' = \text{Rot}(z, -(90-\alpha)) \bullet \text{Rot}(x, \beta) \bullet p_3$$
$$= \text{Rot}(z, -(90-\alpha)) \bullet \text{Rot}(x, \beta) \bullet \underline{\text{Rot}(y, \theta)} \bullet \text{Rot}(x, -\beta) \bullet \text{Rot}(z, 90-\alpha) \bullet p$$

Mapping

- The 4x4 transformation matrix T1-2 between two given coordinate systems
- Two methods to compute:
  - Calculate the product of the transformation matrices, or
  - Solving the linear equations to find the elements in T1-2 by selecting 4 pairs of corresponding points in the two coordinate systems.



$$[X_2, Y_2, Z_2\ 1]^T = T_{1\text{-}2} \bullet [X_1, Y_1, Z_1\ 1]^T$$

Other Transformation Matrices

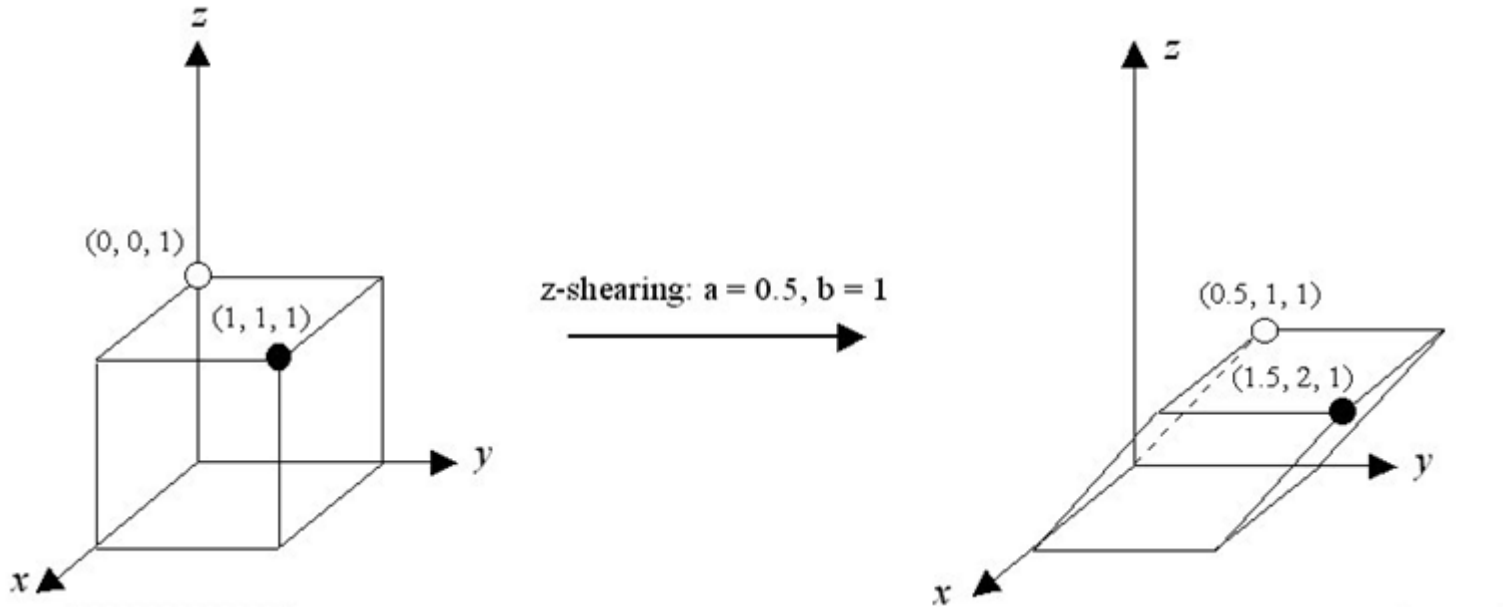
- Scaling

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

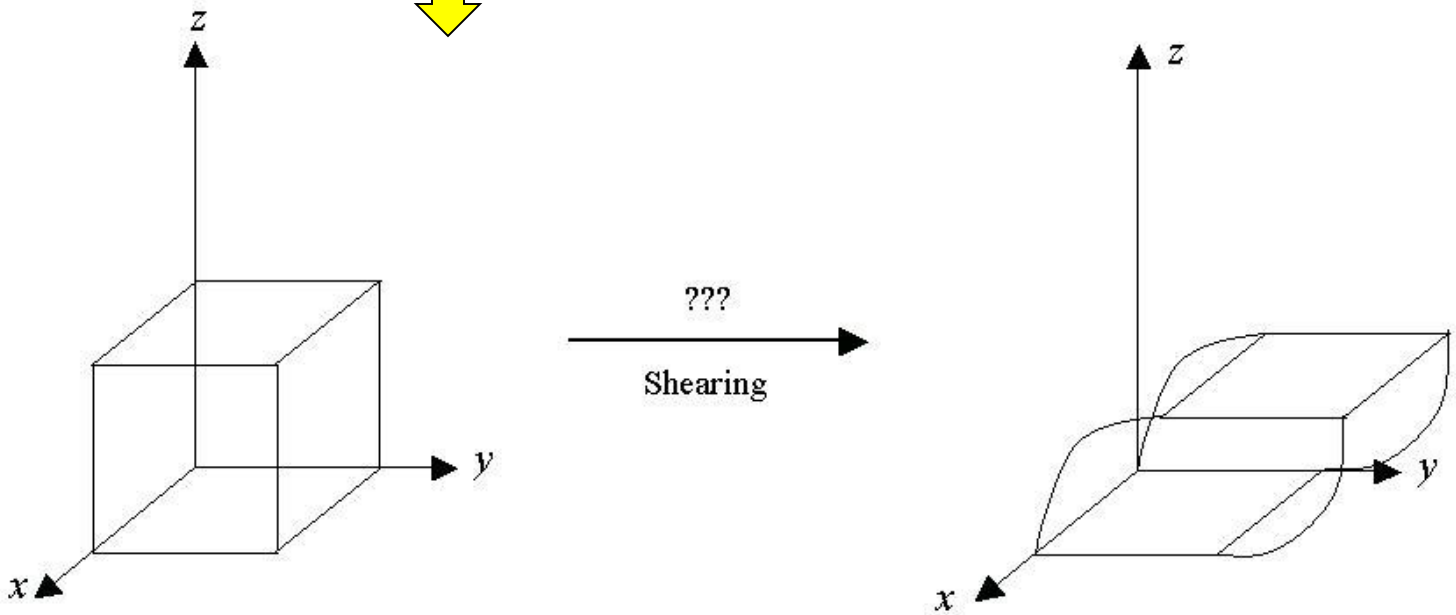
- Shearing (z-axis)

$$[X_h, Y_h, Z_h, 1]^T = [X + a \bullet Z,\ Y + b \bullet Z,\ Z,\ 1]^T$$

$$\begin{bmatrix} X_h \\ Y_h \\ Z_h \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

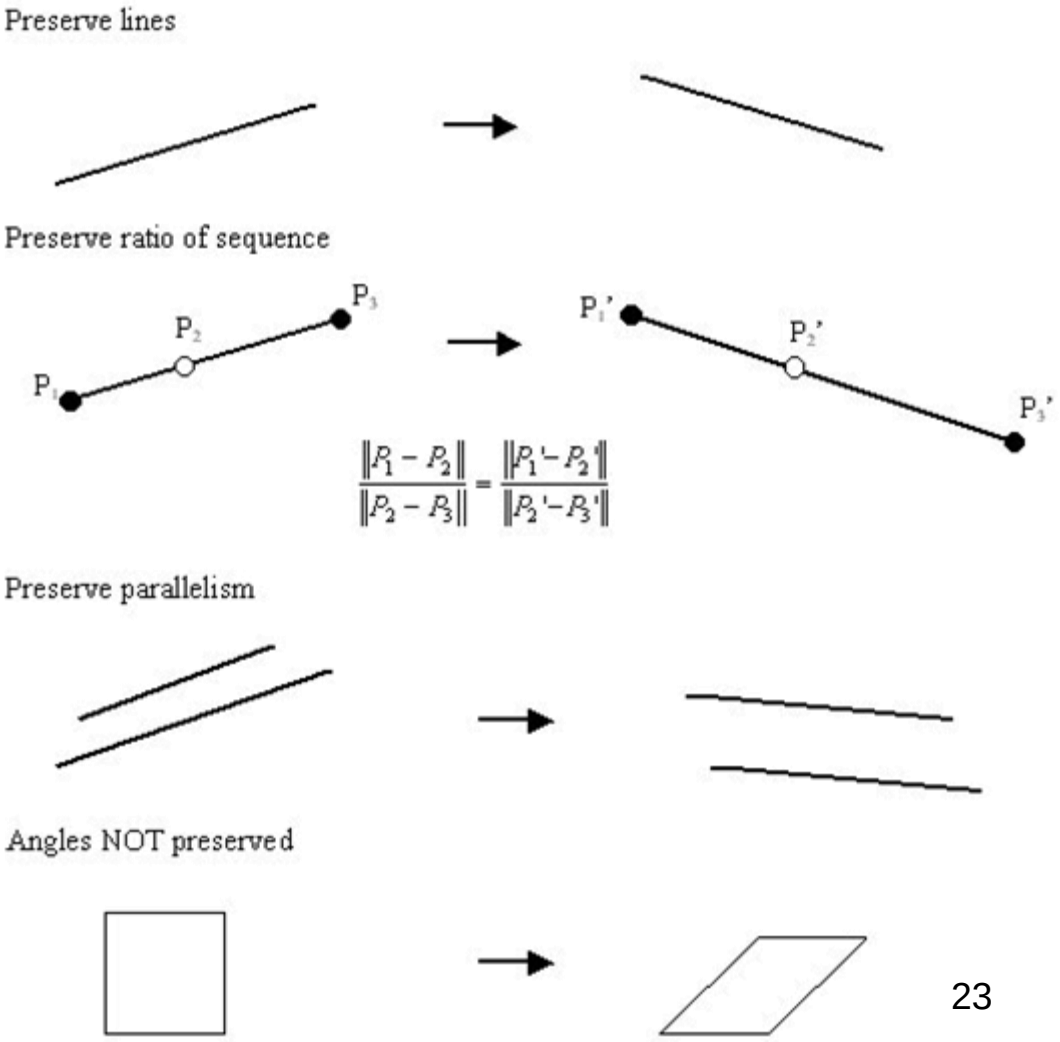


Shearing question: Can a cube become a curved barrel after a shearing transformation?



Affine Transformation

- Transformation made of
  - translation,
  - rotation,
  - scaling, and
  - shearing



Hidden-Line and Hidden-Surface Removal

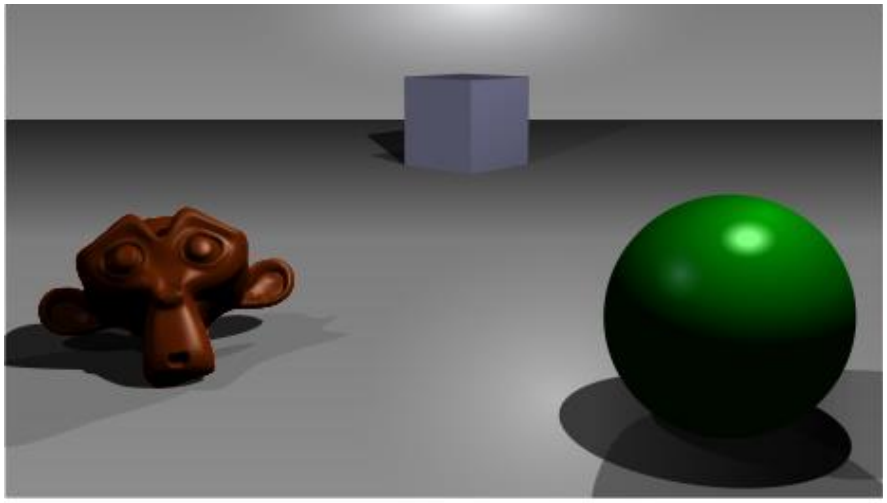
- Types of algorithms
  - *Visible-line determination*: examine edge geometry, and determine edge segments that are visible or are hidden (efficiency is the major problem)
  - *Z-buffer*: device-precision algorithm that records the visible object found at each pixel
  - *Area subdivision algorithms*: divide-and-conquer applied to object areas
  - *Ray tracing*: determine visible object by tracing line from user eye to each pixel in scene

  Useful link:
  http://www.cosc.brocku.ca/Offerings/3P98/course/lectures/hiddenlines/

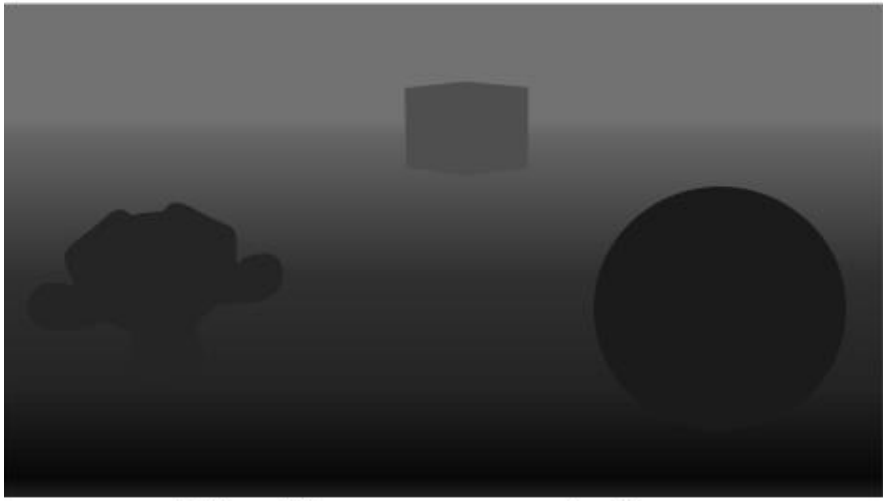
z-Buffering

- The most widely used technique
  - Raster-graphics based
  - Efficient
  - Also known as depth buffering
  - When an object is rendered by a 3D graphics card, the depth of a generated pixel (z coordinate) is stored in a buffer.
  - Is arranged as a 2D array (x-y) with one element for each screen pixel

  When depths conflict: a close object hides a farther one (called *z-culling*).

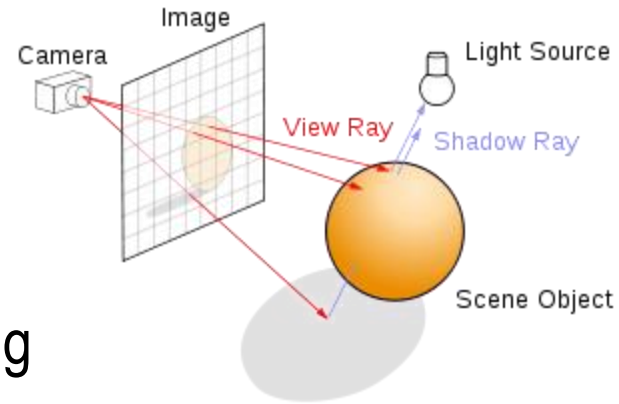


A simple three-dimensional scene

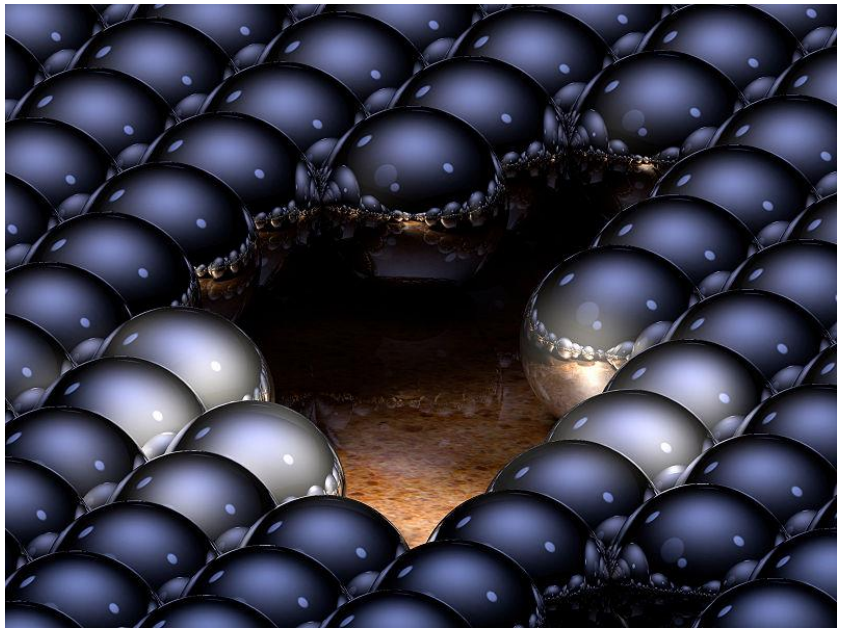


Z-buffer representation

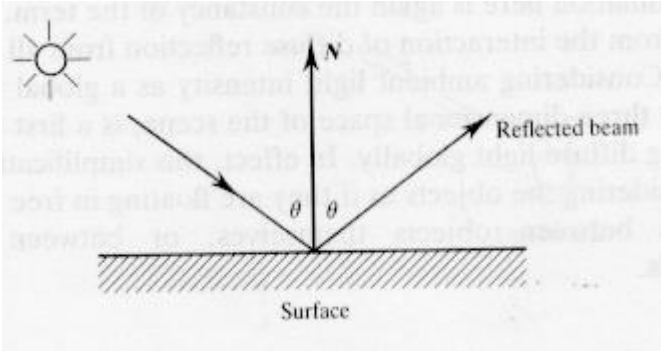
Ray Tracing



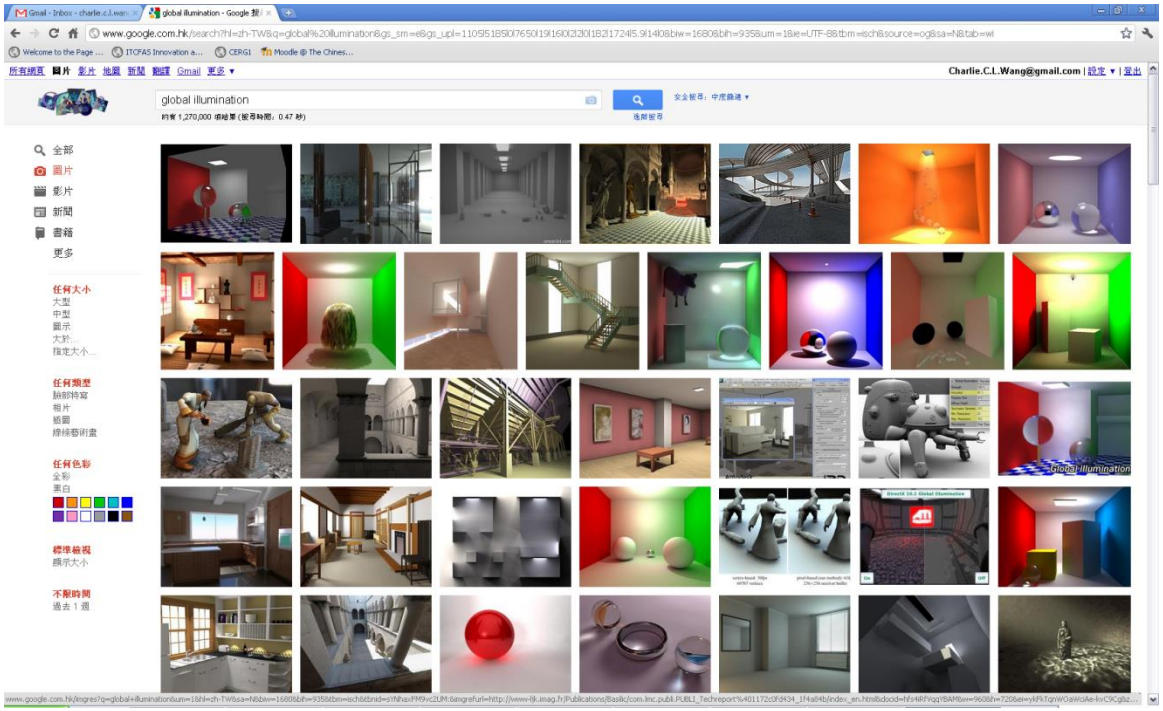
- A technique for generating an image by tracing the path of light through pixels in an image plane and simulating the effects of its encounters with virtual objects



Basic Light Reflection Model



- For a perfect mirror surface the angle of reflection is equal to the angle of incidence
- However, many more factors need to be considered
  - Direct light
  - Ambient light
  - Reflected light
  - Etc.
  - Global illumination

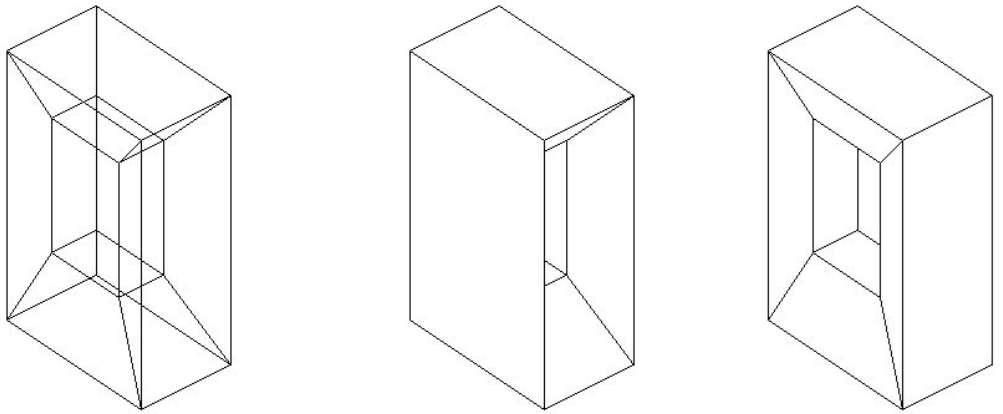


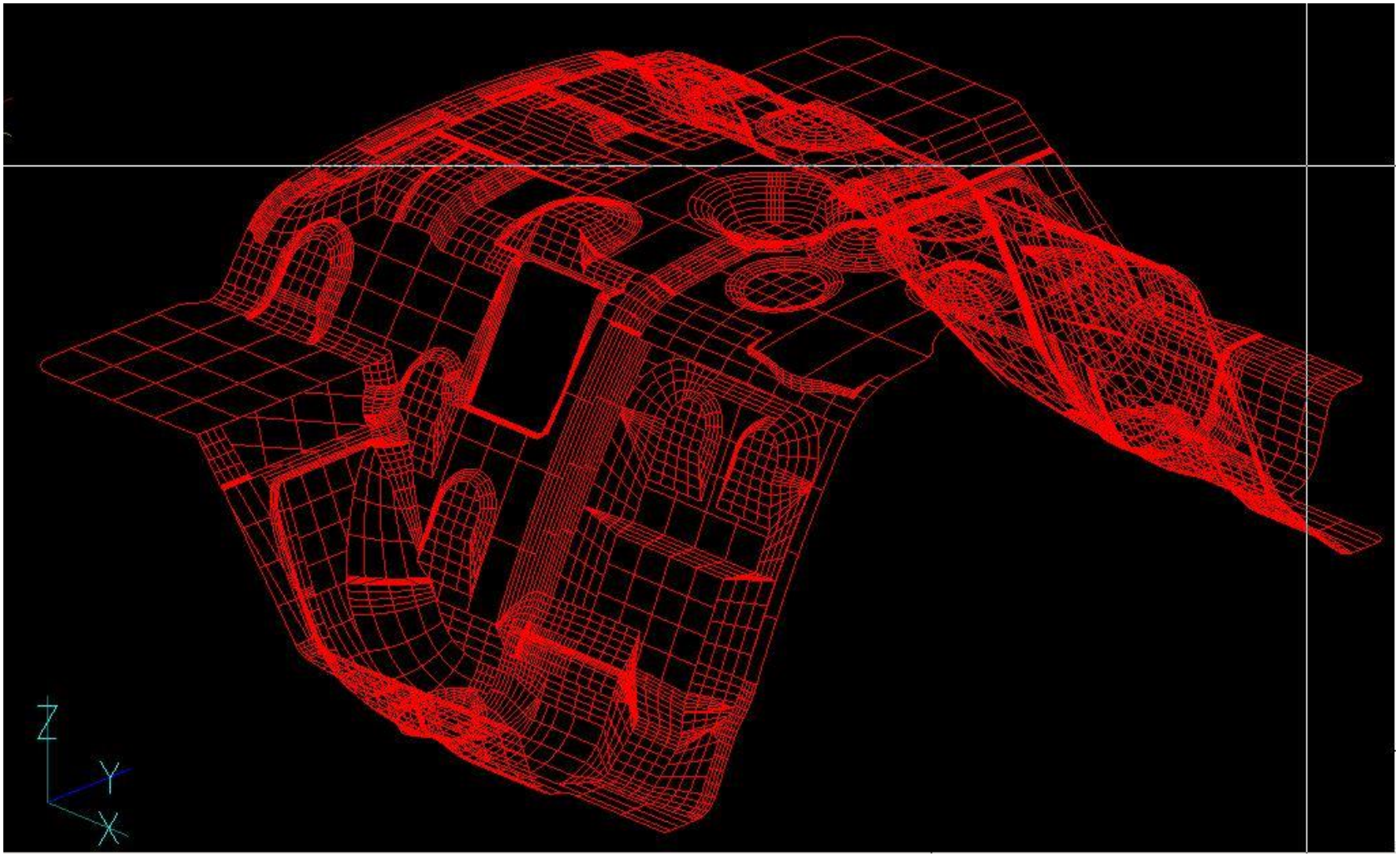
Geometric Modeling Systems

- Contents
  - Wireframe modeling system
  - Surface modeling system
  - Solid modeling system (Most widely used)
  - Non-manifold modeling system (CAE system)
  - Mesh generation for manufacturing and analysis
- What is geometric modeling system?
  - A computer system to model the geometry of a part
  - It provides an environment similar to the one in which the physical model is created and naturally manipulated.
  - The designer deforms, adds, and cuts pieces off the visual model in the process of detailing a shape (virtually)

Wireframe Modeling Systems

- Represents a shape by its characteristic lines and end points (60's and 70's)
- Advantages:
  - Requires less computer speed and memory
  - Does not require sophisticated math
- Disadvantages:
  - Can't handle solid parts with inside/outside info
  - Difficult to visualize (just wires)
  - No volumetric info
  - Ambiguous when displaying a solid model

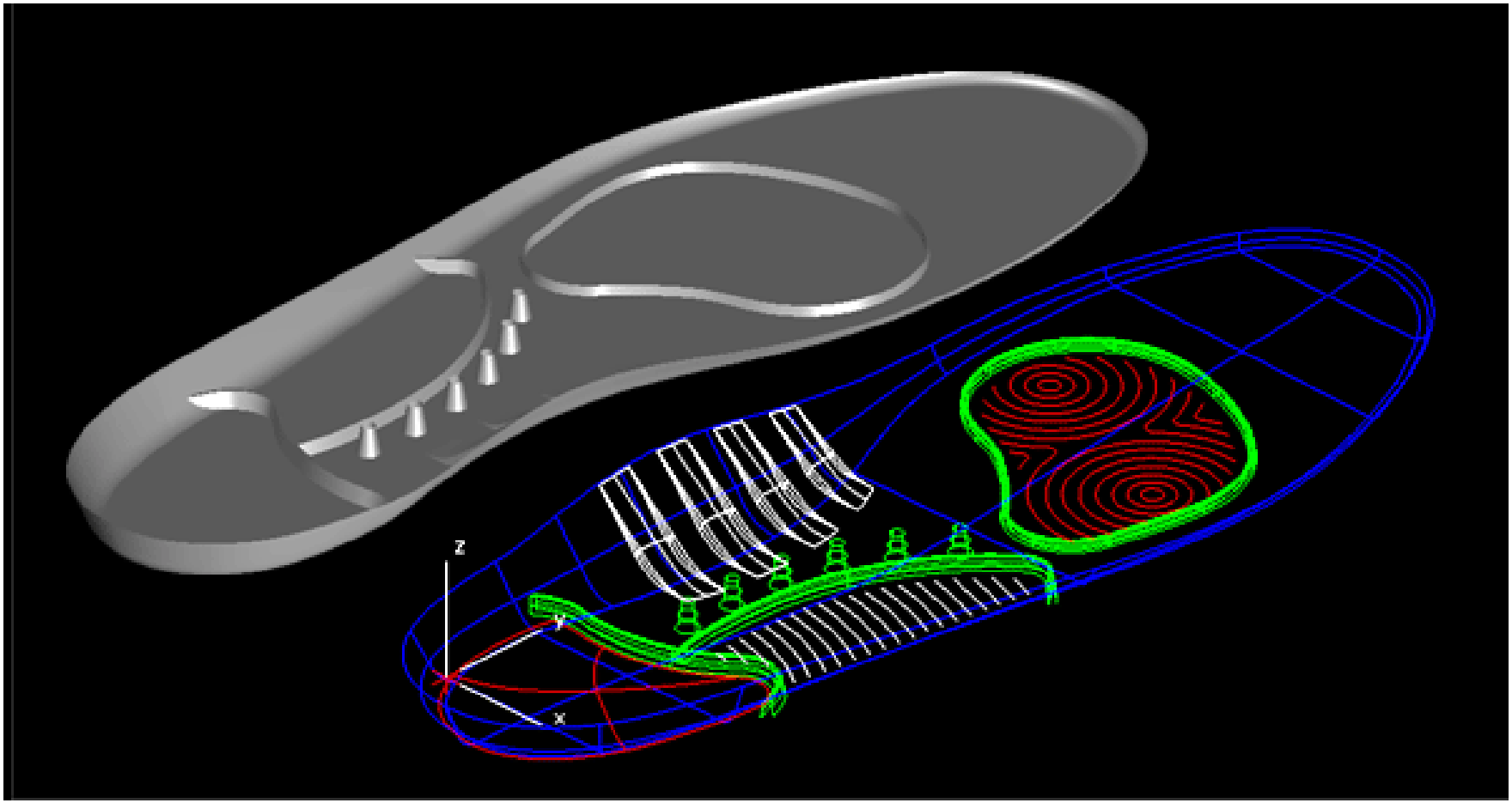




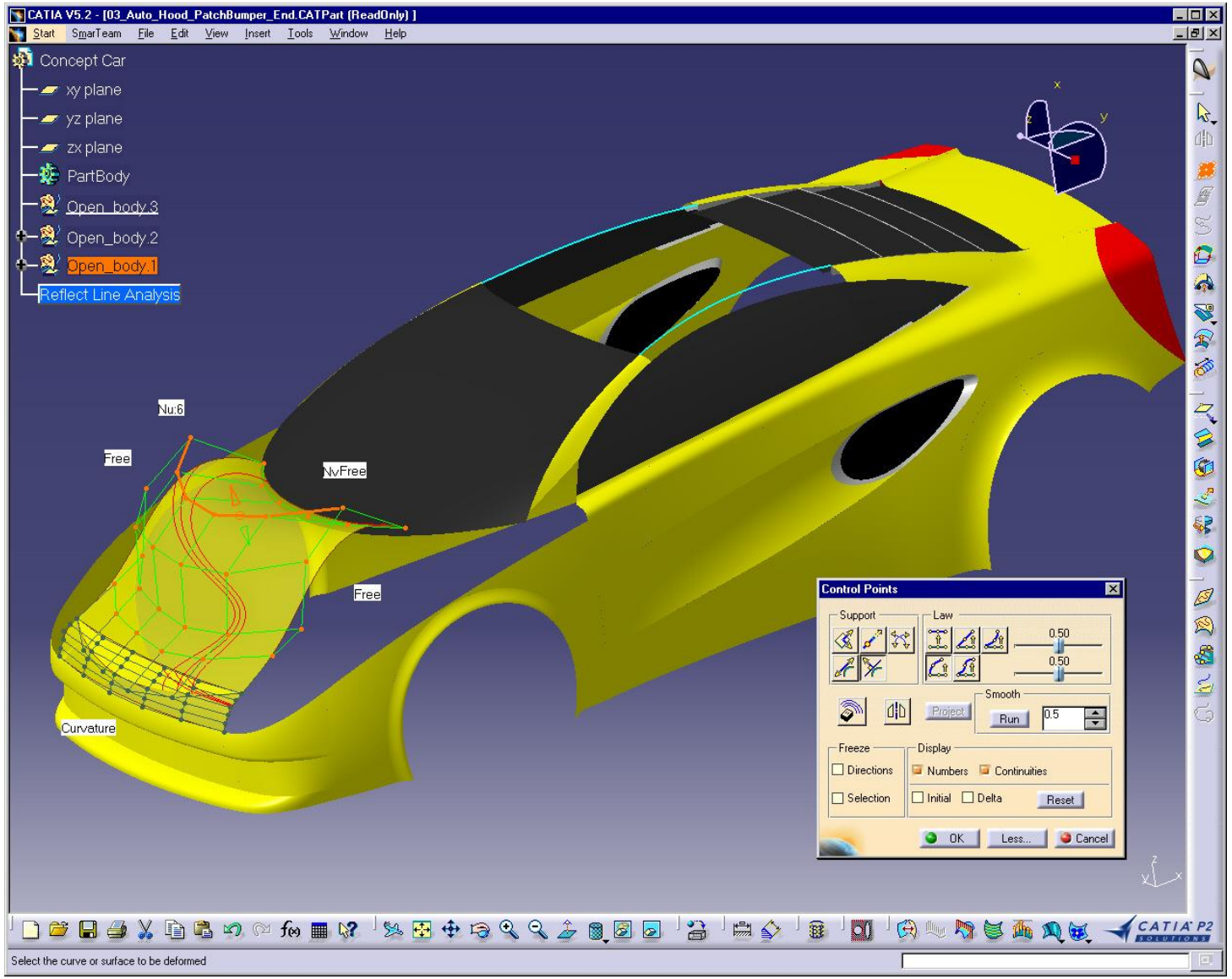
A wireframe model of a car hood

Surface Modeling Systems

- In addition to the characteristic curves and end points and their connectivity information, surface information is also included (70's and 80')
- Two main purposes:
  - visual model used for evaluating the model aesthetically
  - math description is used to generate NC tool path to machine the surfaces
- Major disadvantage:
  - Still can't model a solid part
  - Can't calculate volumetric properties such as mass, weight, etc.



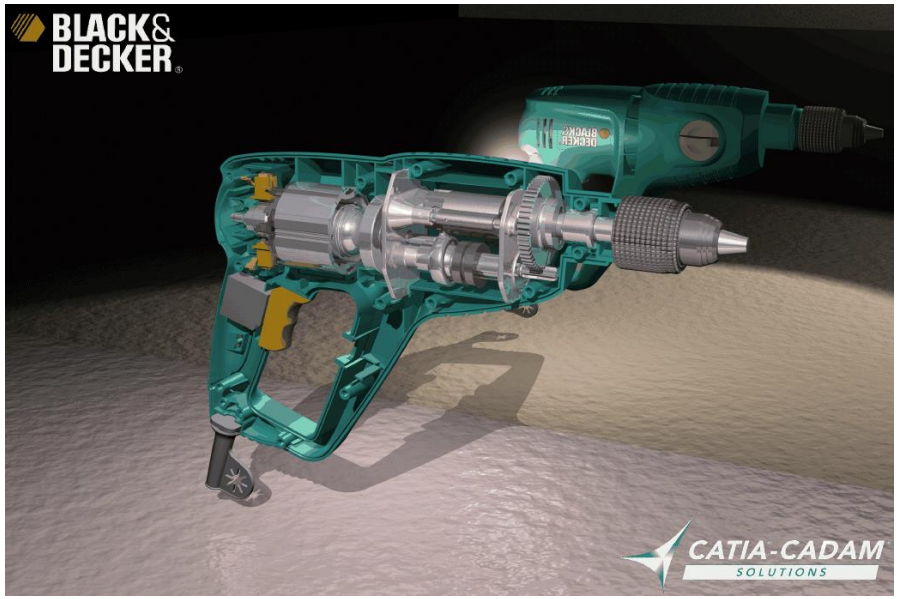
A surface model of a shoe’s bottom

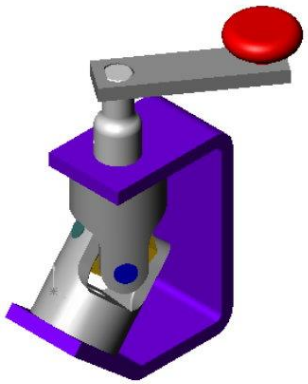


A surface model of a car's body

Solid Modeling Systems

- It models a shape that forms a closed volume, called a solid.
- In addition to all the functions provided by a surface modeling system, it contains the information that determines whether any location is inside, outside, or on the closed volume (80's and 90's)
- Must be a closed model

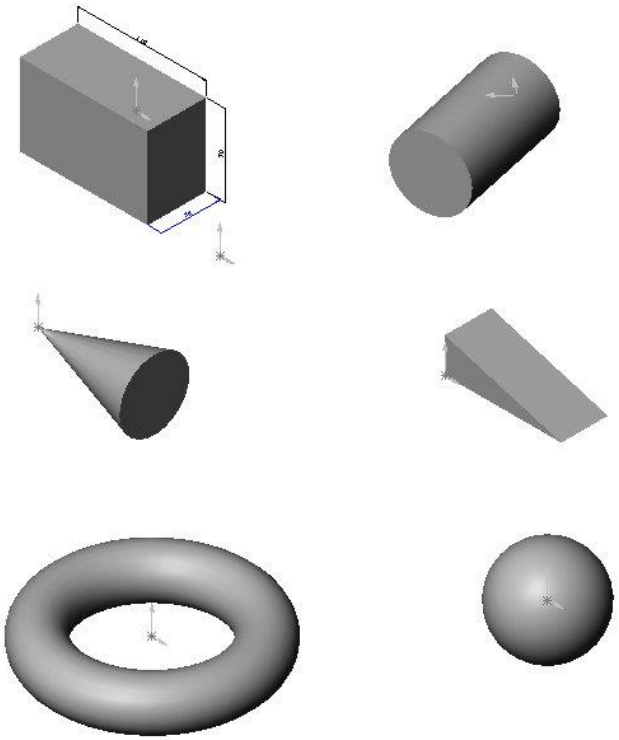




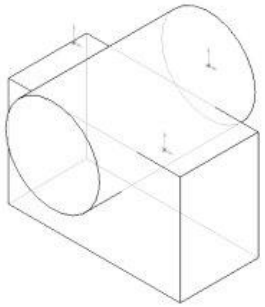
Modeling Functions

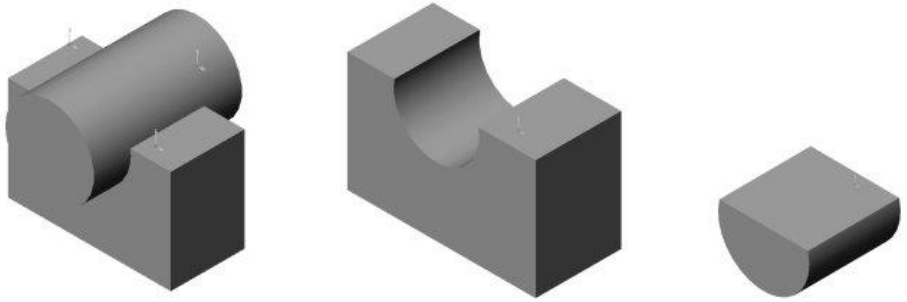
- Primitive creation functions
  - Most used analytical shapes such as blocks, cylinders, cones, wedges, torus, sphere, etc.
- Boolean operations
  - To provide the user a suite of Boolean operations that can combine primitives into other geometric shapes
- Sweeping, Skinning, Rounding (or blending)
  - Sweeping: sweep a planar closed curve (profile curve) along another 3D curve (path curve)
  - Skinning: creating a skin surface over pre-specified cross-sectional planar surfaces (a construction example)
- Feature based modeling
  - Model a solid by pre-set features such as slots, holes, chamfers (via subgraph)

Primitive Creation and Boolean Operations



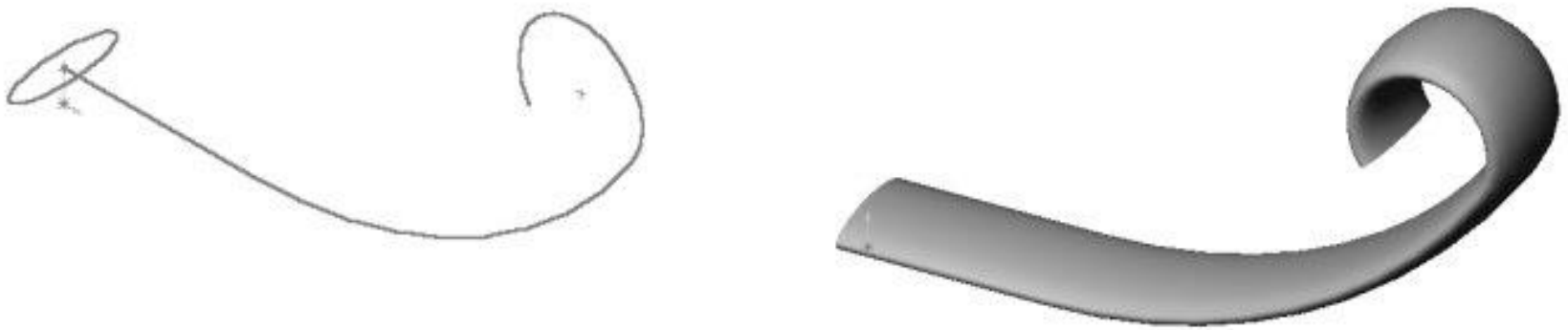
Primitives generally supported





Union Difference Intersection

Sweeping Operation

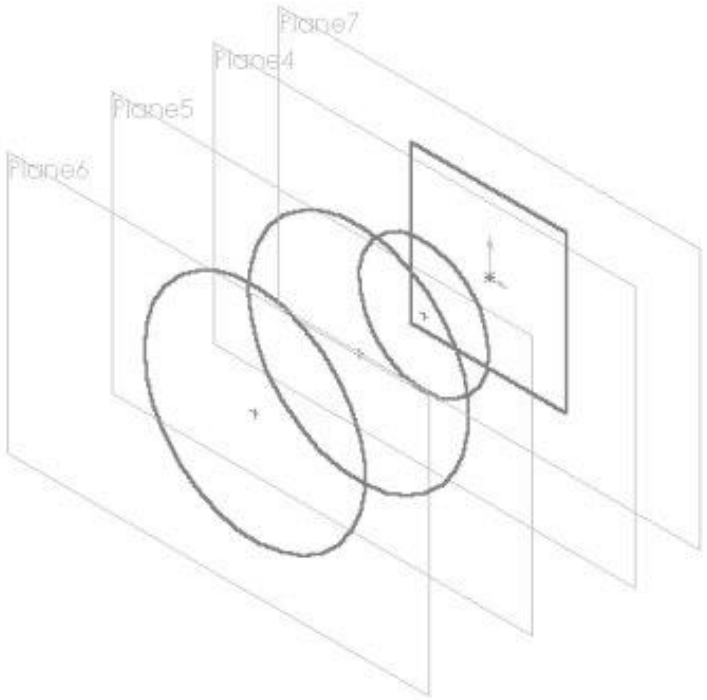


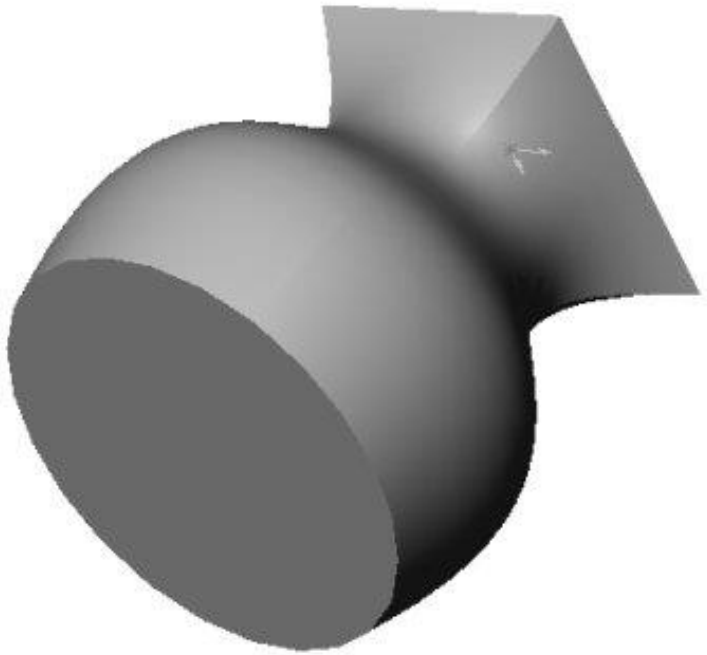
Two types:

1. Orthogonal sweeping (normal of the plane remains parallel to the tangent of the path curve)

2. Non-orthogonal sweeping (otherwise)

Skinning Operation

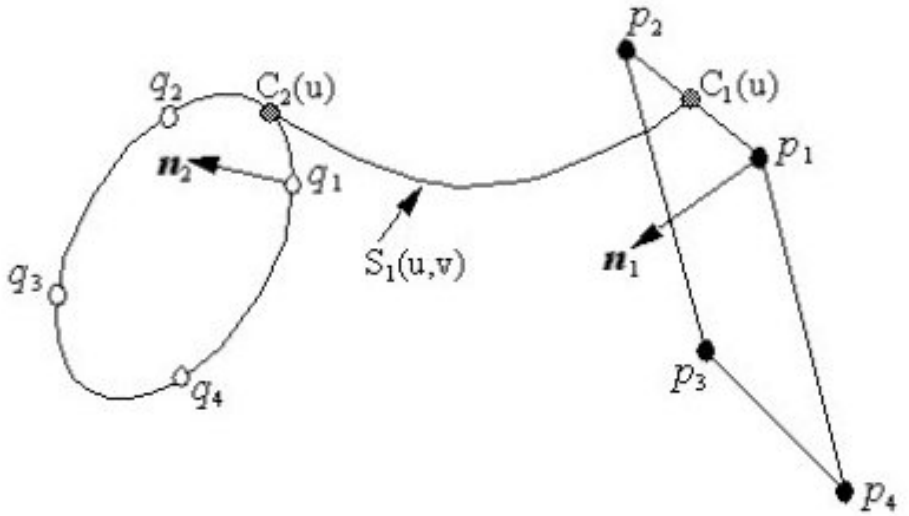




Questions: 1. How to decide compatible points on neighboring cross-sections?

2. What is the relationship between sweeping and skinning?

An Example: Constructing the skin surface over a circle and a square



Step 1. Select four points $\{q_1, q_2, q_3, q_4\}$ on the cirle corresponding to the points $\{p_1, p_2, p_3, p_4\}$. Many ways for selection; but the most widely used one is the closest distance criterion, i.e. q_i is closest to p_i, i=1,2,3,4.

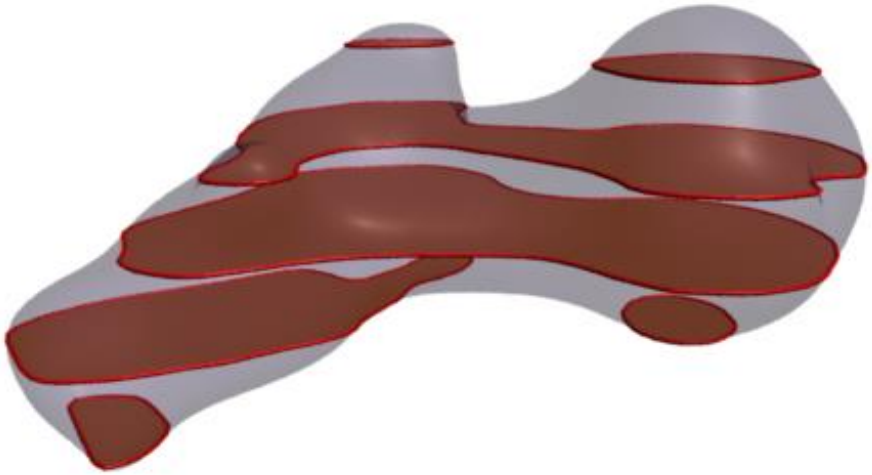
Step 2. Let $C_1(u)$ $(0 \le u \le 1)$ be the curve on the square between p_1 and p_2, and $C_2(u)$ be the curve on the circle between q_1 and q_2. Let $H(p, p', t, t', v)$ $(0 \le v \le 1)$ be the Hermite interpolation of two points p and p' and their tangents t and t'. Between two corresponding points $C_1(u)$ and $C_2(u)$, define the curve

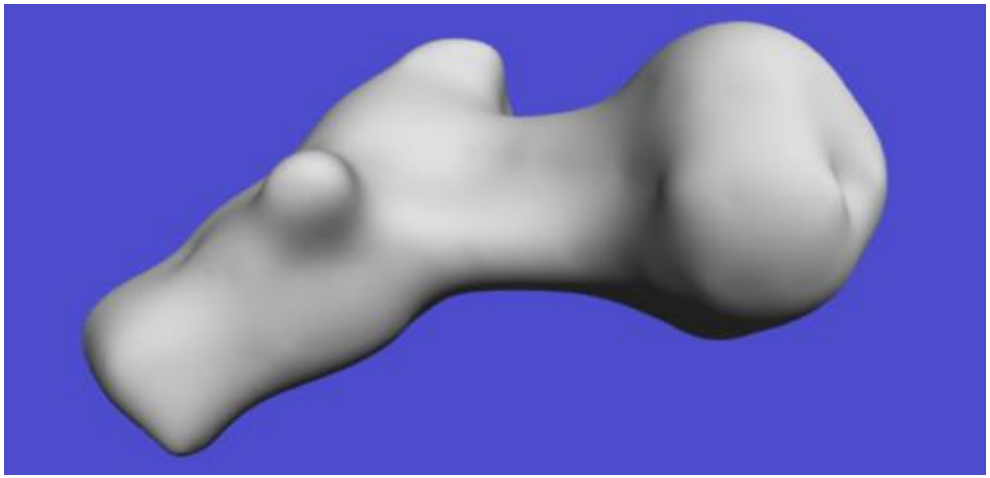
$$S_1(u,v) = H(C_1(u), C_2(u), n_1, n_2, v) \ (0 \le v \le 1).$$

Letting u change from 0 (p_1 and q_1) to 1 (p_2 and q_2), $S_1(u,v)$ $(0 \le u \le 1, 0 \le v \le 1)$ becomes a smooth surface connecting $C_1(u)$ and $C_2(u)$.

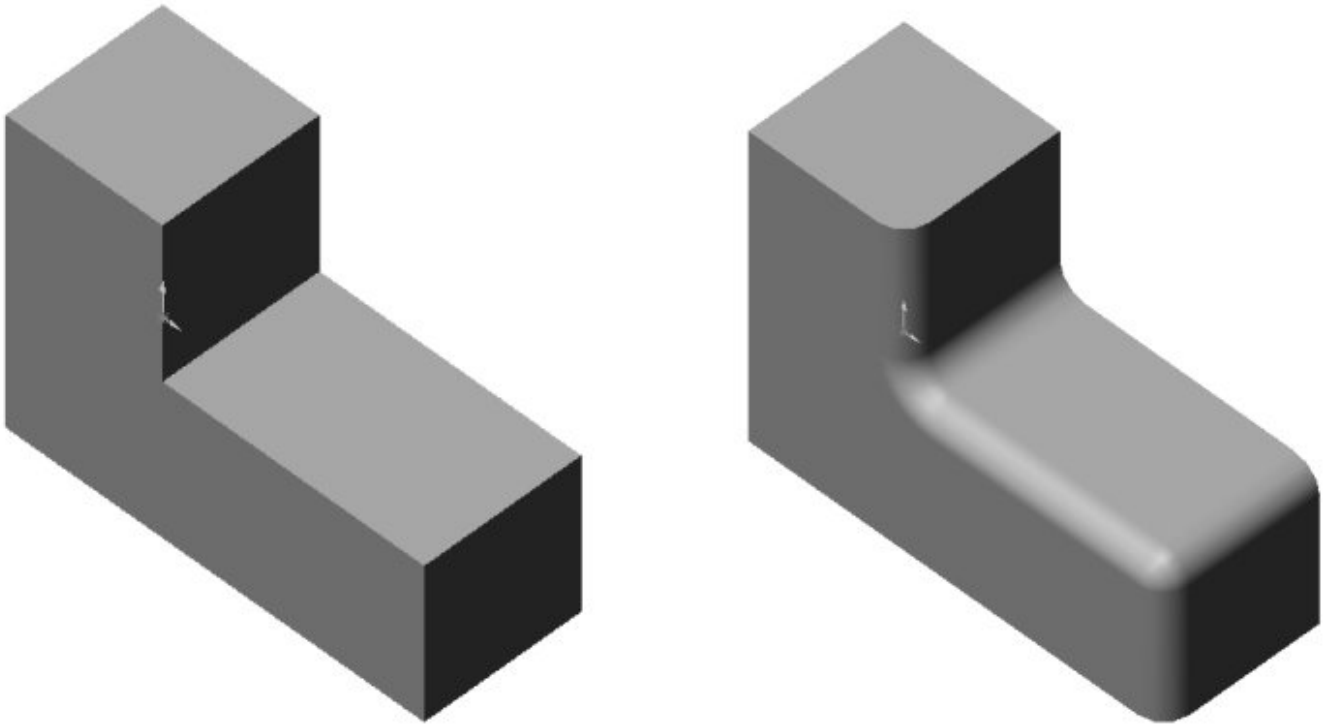
Step 3. Repeat Step 2 for the pairs $(p_2, p_3) \rightarrow (q_2, q_3)$, $(p_3, p_4) \rightarrow (q_3, q_4)$ and $(p_4, p_1) \rightarrow (q_4, q_1)$, we can define another three smooth surfaces $S_2(u,v)$, $S_3(u,v)$, and $S_4(u,v)$. Together, these four surfaces form the skin surface of the circle and the square.

General Skinning Operation Example: Reconstructing Bone Joint from CT Scans





Rounding Operation

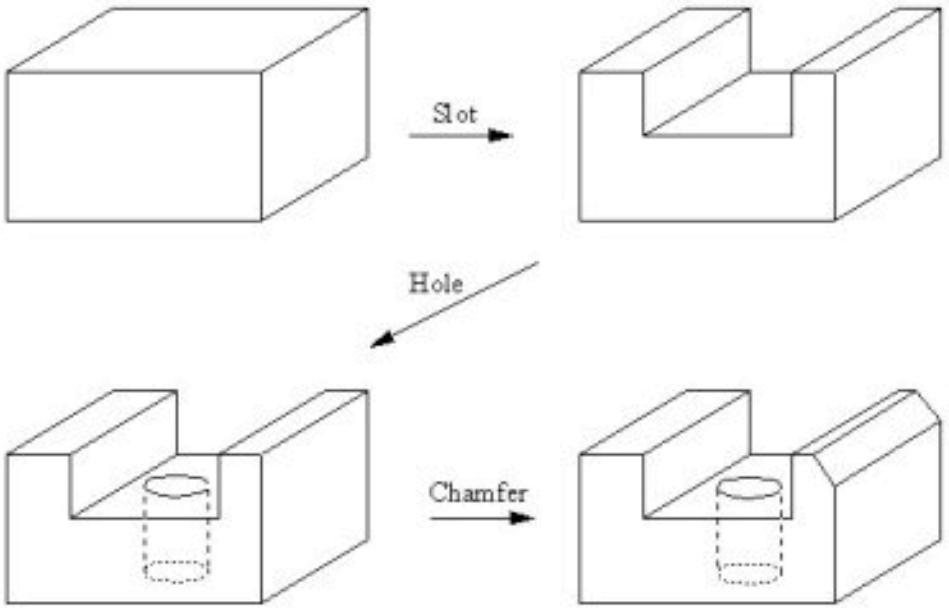


A function that modifies an existing model so that a sharp edge or vertex is replaced with a smooth curved surface whose normal vectors are continuous with those of the surfaces meeting at the original sharp edge or vertex.

Usually this smooth surface is a cylinder or generalized cylinder. Filleting is a special case of rounding in which material is added, not subtracted from the original volume.

Feature Based Modeling

To model a part by using some pre-set features such as slot, hole, chamfer, fillet, pocket, etc.



Major benefits: bridge between CAD and CAM – each feature usually is associated with a manufacturing process; automatic process planning.

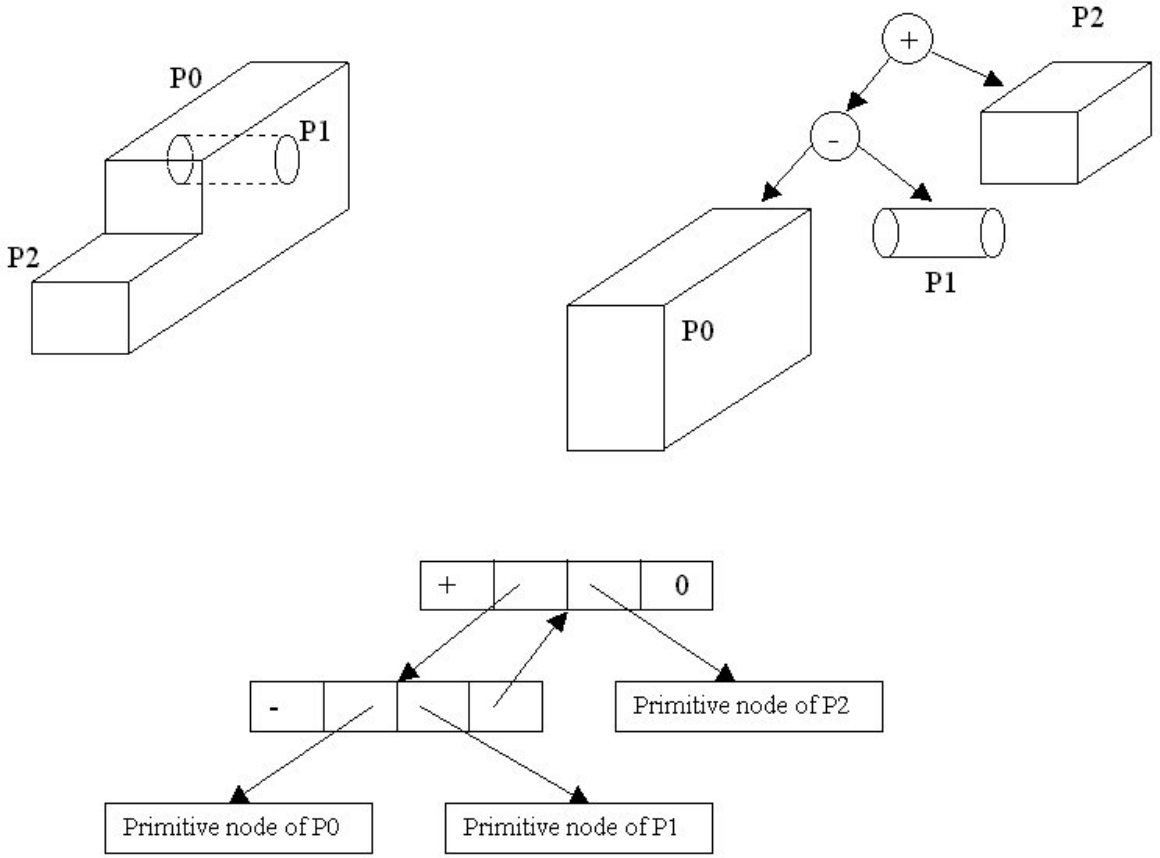
Issues:
1. Can't define and provide all the features necessary for many applications
2. Difficulty and ambiguity when defining a feature.

Data Structure

- Once a model is created, its exact mathematical description is also defined.
- But how is this description stored in a solid modeling system?
- Three types of data structures used in a solid modeling system to describe a solid
  - *Constructive Solid Geometry* (CSG) tree structure
  - *Boundary Representation* (B-rep) data structure
  - Decomposition model structure

CSG Tree Structure

- A data structure that stores the history of applying Boolean operations on the primitives

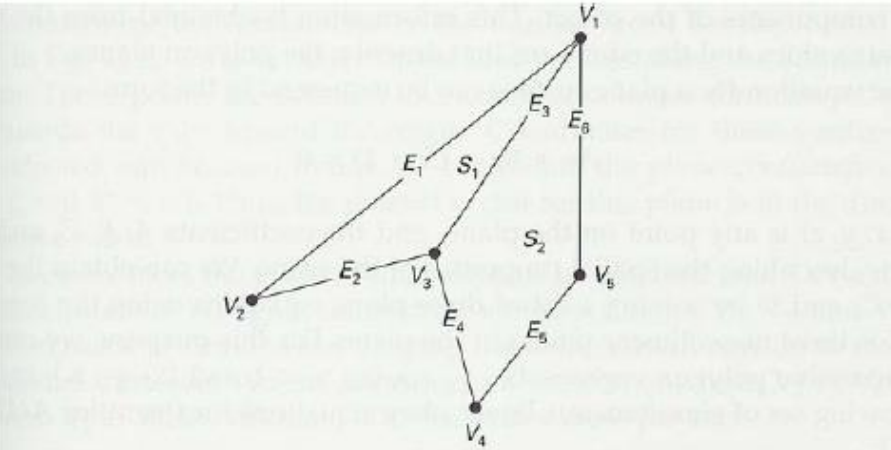


CSG Tree Structure (cont.)

- Advantages:
  - Simple and compact data, management of data is easy
  - The solid stored in a CSG is always a valid solid
  - Always convertible to other types of rep., such as B-rep
  - Parametric modeling can be realized easily by changing the parameter values of the associated primitives
- Disadvantages:
  - Limited modeling ability. Only primitives and their Boolean operations; no sweeps or lofts
  - Computationally expensive and difficult to derive the boundary data (faces, edges, vertices) and their topology, needed for:
    - Display of the model (faces and edges needed)
    - Calculation of volumetric data such as mass and centroid

B-rep Data Structure

- Drawback of CSG-tree => a hybrid representation of CSG and B-rep
- Basic elements composing the boundary of a solid are *faces*, *edges*, and *vertices* (simplest example of B-Rep (already shown earlier))



VERTEX TABLE	
V_1:	x_1, y_1, z_1
V_2:	x_2, y_2, z_2
V_3:	x_3, y_3, z_3
V_4:	x_4, y_4, z_4
V_5:	x_5, y_5, z_5

EDGE TABLE	
E_1:	V_1, V_2
E_2:	V_2, V_3
E_3:	V_3, V_1
E_4:	V_3, V_4
E_5:	V_4, V_5
E_6:	V_5, V_1

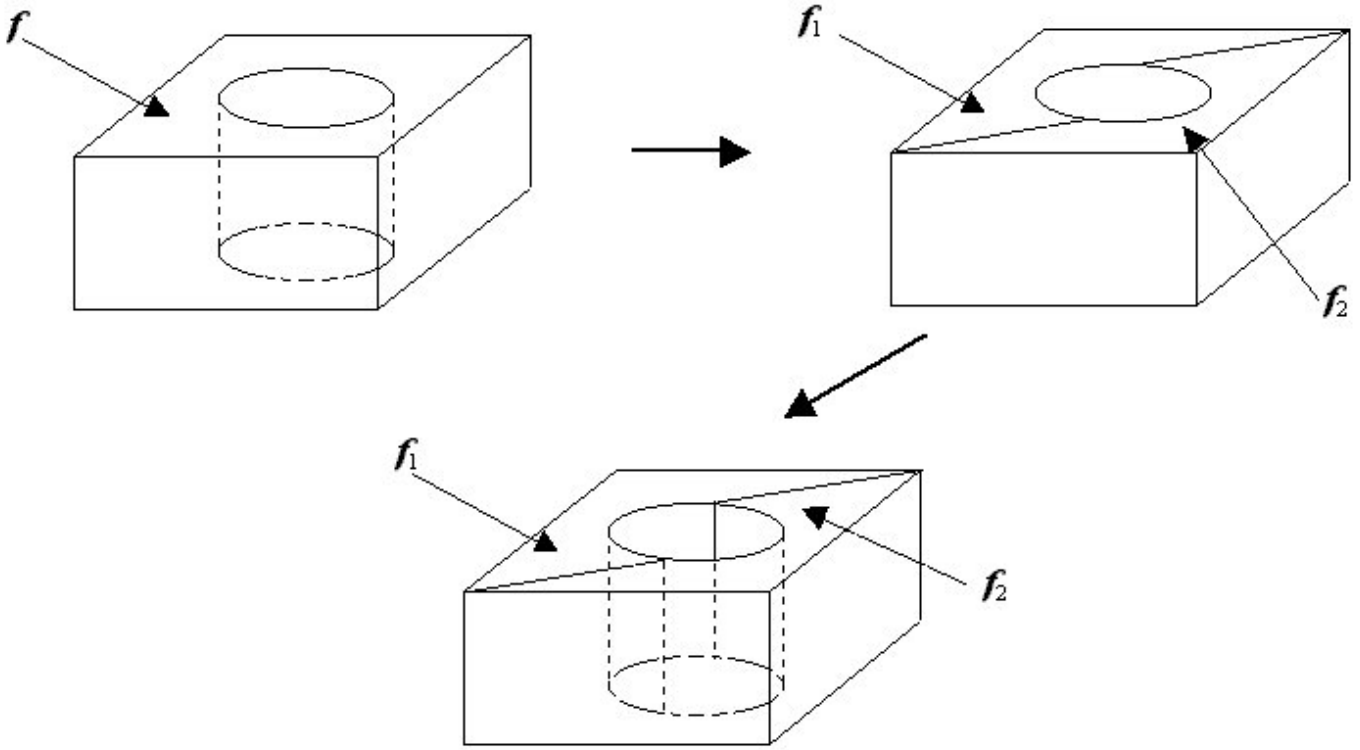
POLYGON-SURFACE TABLE	
S_1:	E_1, E_2, E_3
S_2:	E_3, E_4, E_5, E_6

A convenient organization for storing geometric data is to create three lists:
- A *vertex table*, an *edge table*, and a *polygon table*

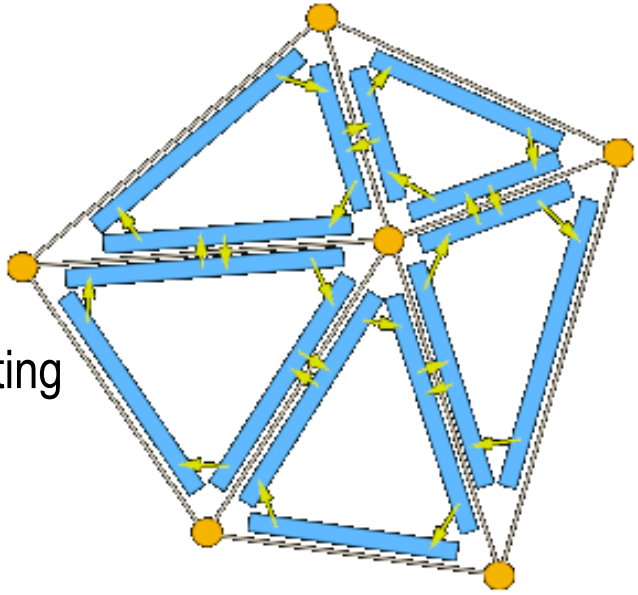
An alternative arrangement is to use just two tables:
- A *vertex table* and a *polygon table* (e.g., OBJ file)

- Problems with this simple B-Rep
  - Unable to represent a face with two loops
  - Hard to answer queries like what is the edge shared by two faces

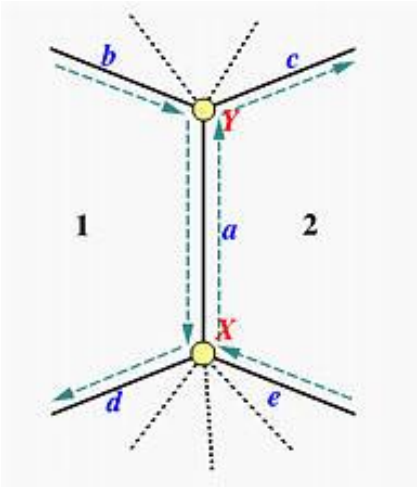
The face f can not be stored in the face table, since it has two boundary loops: the outer one and the inner one. To resolve this, we have to split f into two faces f_1 and f_2. But this requires further to split that single cylinderical face into two, as a vertex can not be in the middle of an edge (the original circular edge).



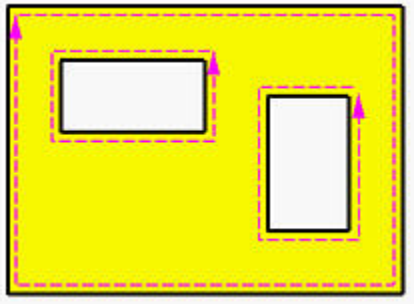
Half-edge and Winged-edge Data Structure



- Half-edge (link)
  - Double linked list as the primitive data structure
  - Better than the table based
  - Still hard to process model with holes without splitting
- Winged-edge (link)
  - Linked list only
  - A edge table with winged edges



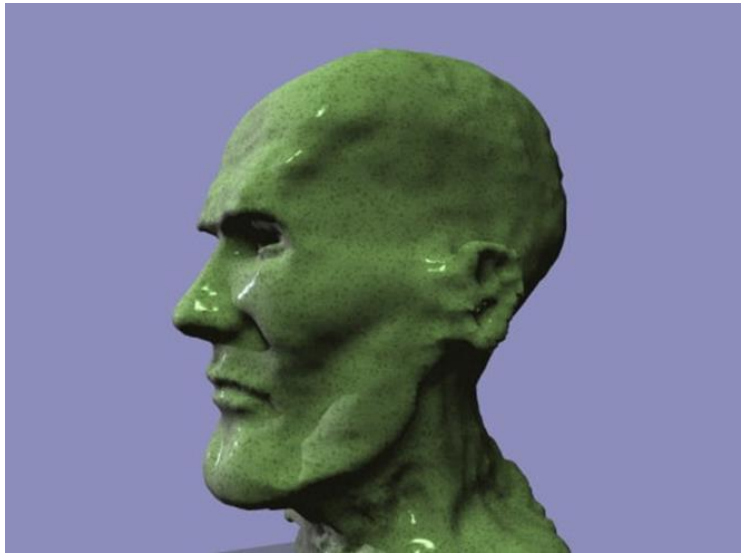
- A face table
- A node table
- Can process faces with multiple holes more easily



(For a face with inner loops, the outer boundary is ordered clockwise, while its inner loops, if any, are ordered counter clockwise)

Decomposition Model Structure

- A solid model is described approximately as an aggregate of simple solids such as cubes.
- Voxel representation:
  - To represent a solid by equal-sized cubes, called voxels
  - Simple and easy to develop
  - Memory intensive data structure
  - Resolution cannot be high
  - It is inherently an approximation

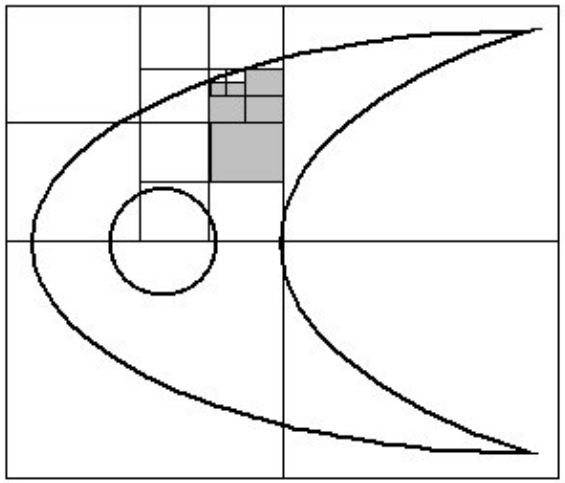


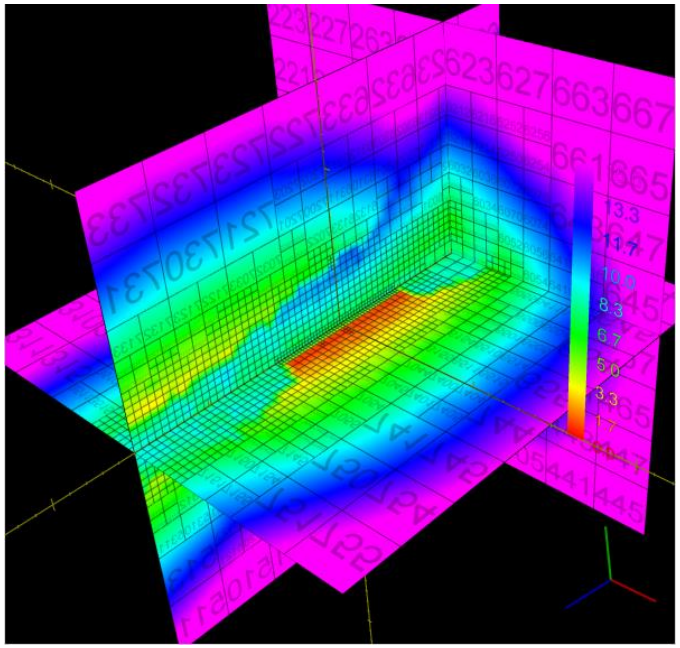
- Oct-tree (Octree) representation:
  - To represent a solid by a number of non-equal-sized hexahedra
  - Adaptive to the shape of model to be presented

Octree Representation

Step 1. Make a hexahedron H that completely encloses the solid.
Step 2. Divide H into 8 octants
Step 3. For each octant obtained in Step 2, mark it in one of three colors:
White: completely outside the solid
Black: completely inside the solid
Gray: partially inside the solid.
Step 4. For each and every "gray" octant obtained in Step 3, do Step 2 through Step 4, until either no "gray" octants are left or the size of the octant reaches a minimum tolerance.

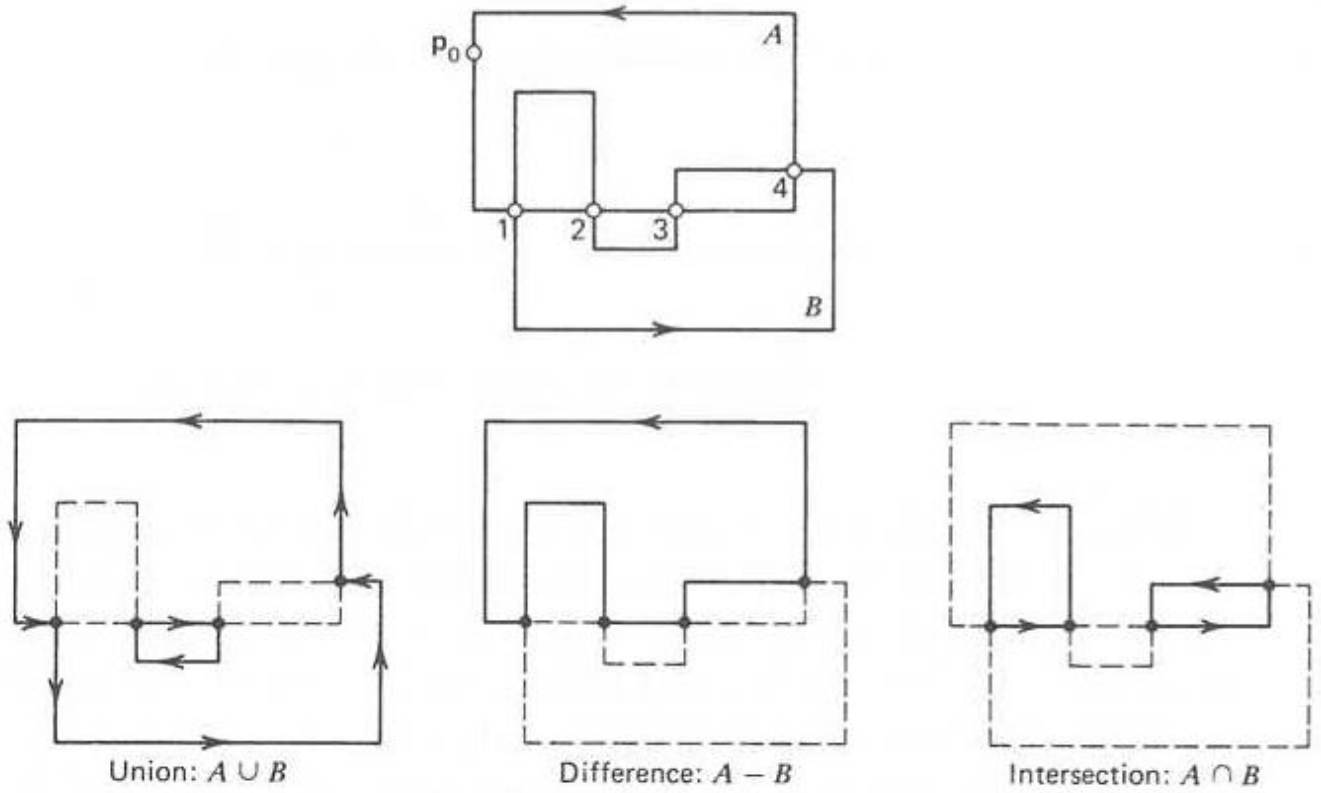
An example in 2D (called quad-tree)





The "black" nodes and their parent nodes (intermediate "gray" nodes") are stored in a tree structure, as illustrated in Fig. 5.38 (c) in the textbook.

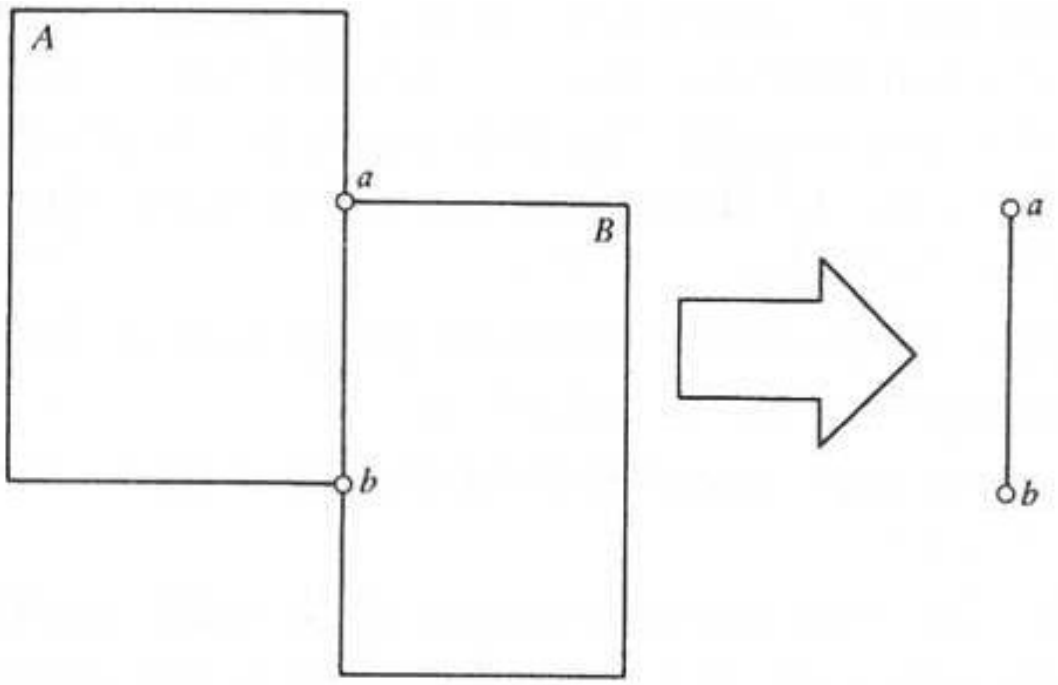
Boolean Operations of Simple Polygons



1. Efficiently find all the intersection points between the edges of A and B.
2. Segment the edges of A and B by the intersection points.
3. Trace the segmented A and B to find the correct Boolean result.

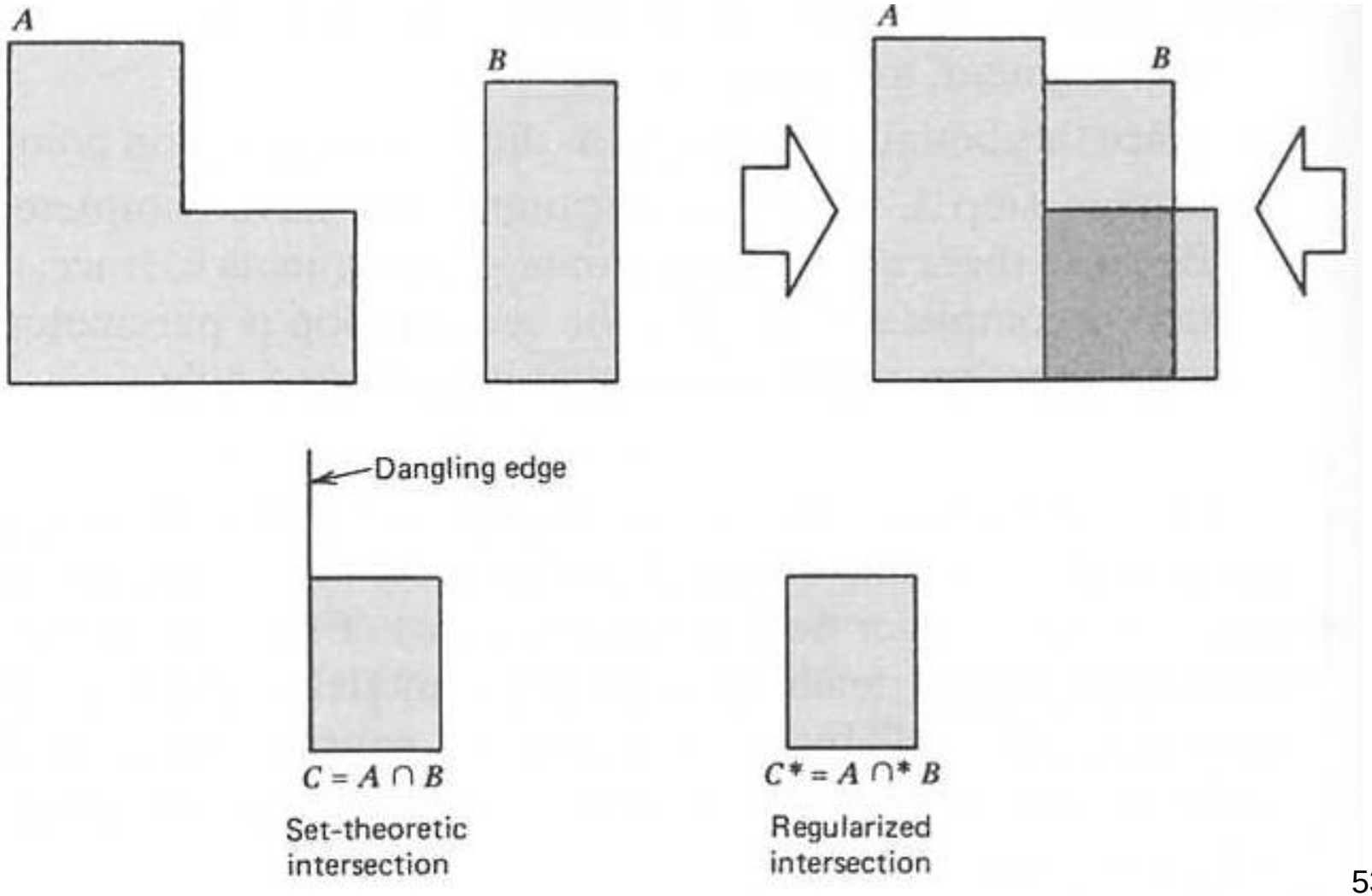
Regularization

- The result of a Boolean operation must preserve the dimensionality and homogeneity of the initial objects

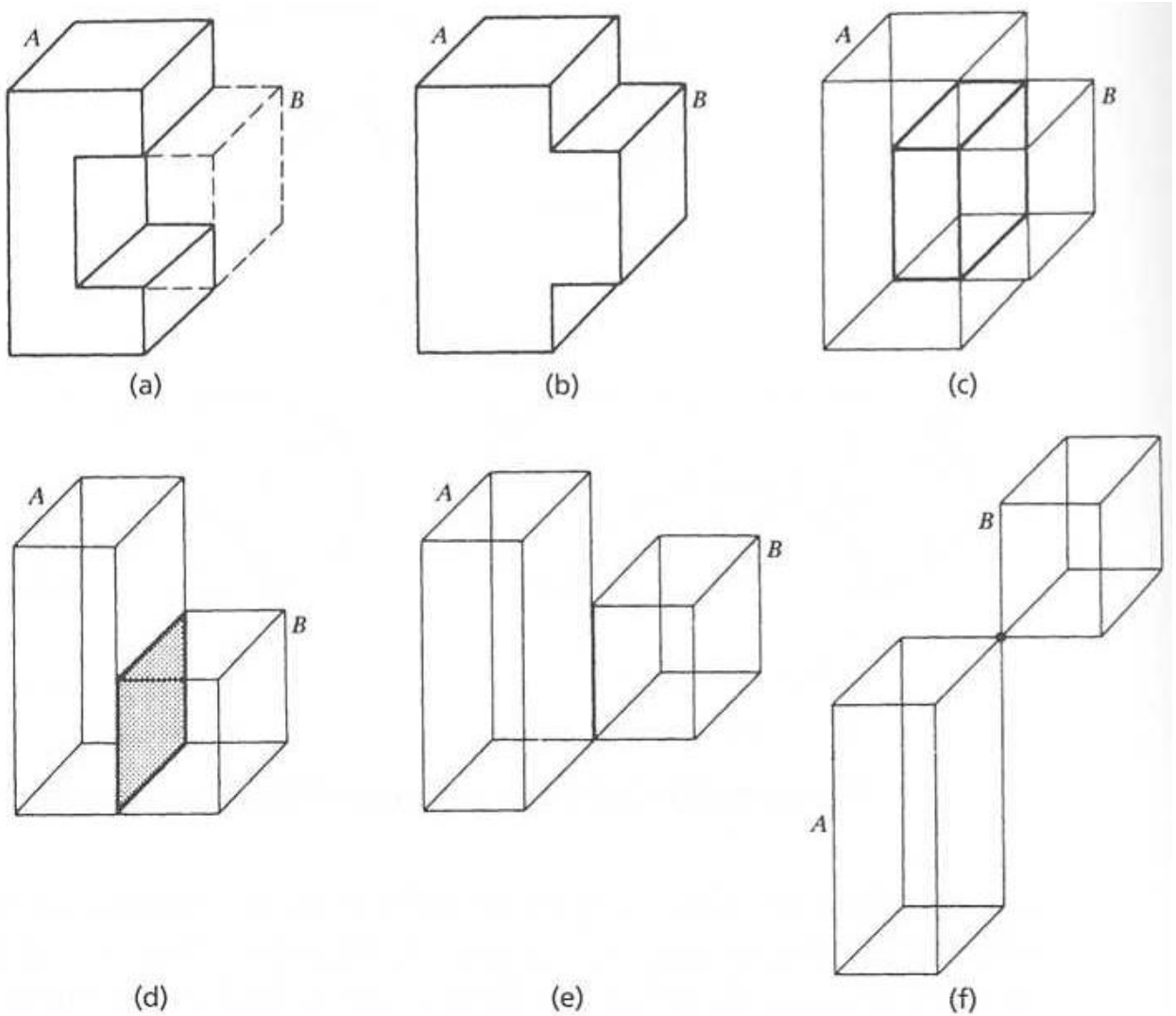


A and B are 2D sets, but the theoretical A$\cup$B is a line which is 1D. By regularization, A$\cup$B = ϕ (empty).

Regularized Boolean Operations

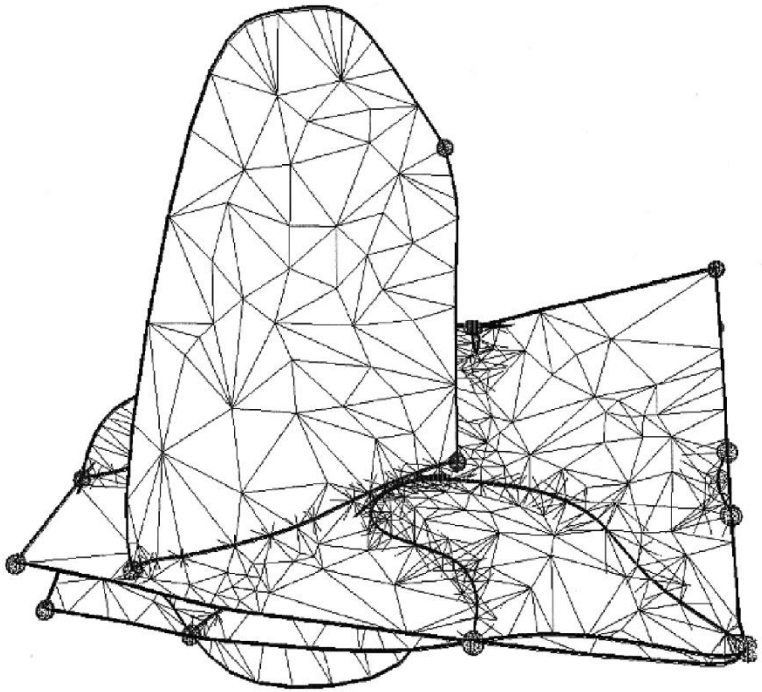


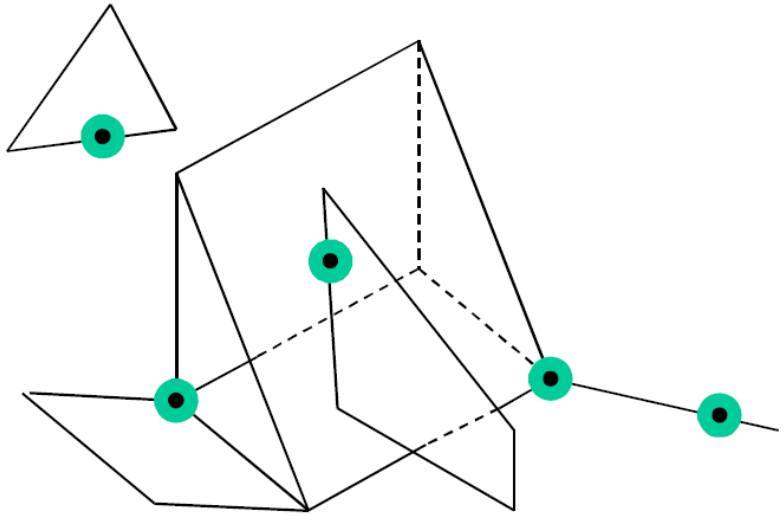
Boolean Operations in 3D

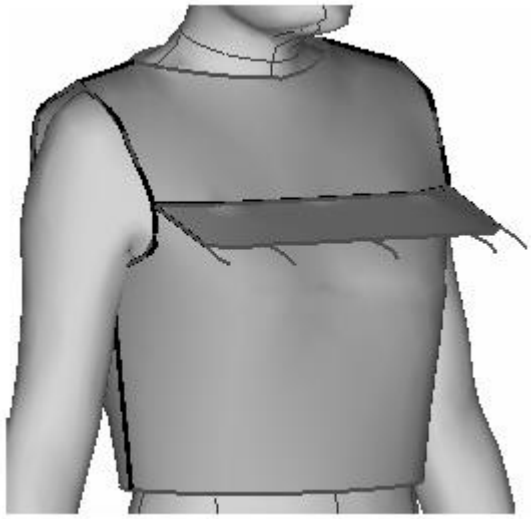


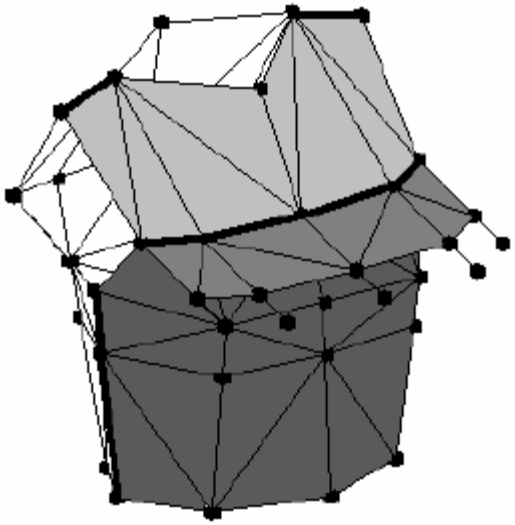
Nonmanifold Modeling Systems

- Further extension of solid modeling



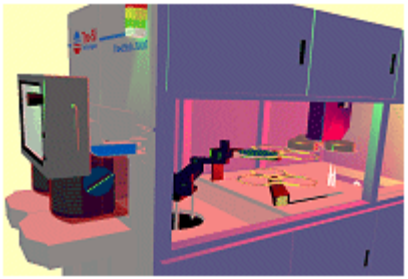






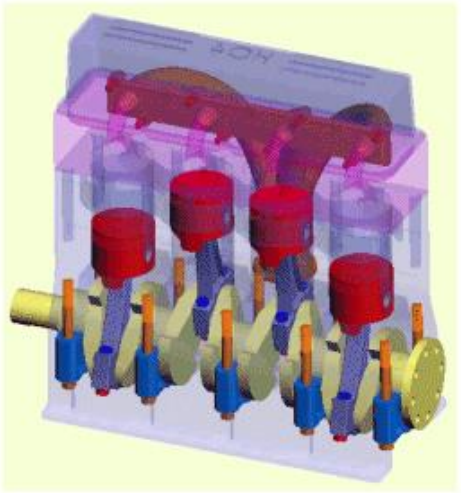
Assembly Modeling

- Provide a logical structure for grouping and organizing parts into assemblies
- Maintain the relationships between the parts and the associated data:
  - Mating relationship
  - Parametric constraint relationship
  - Clearance/non-interference relationship
- Automatic determination of assembly ordering



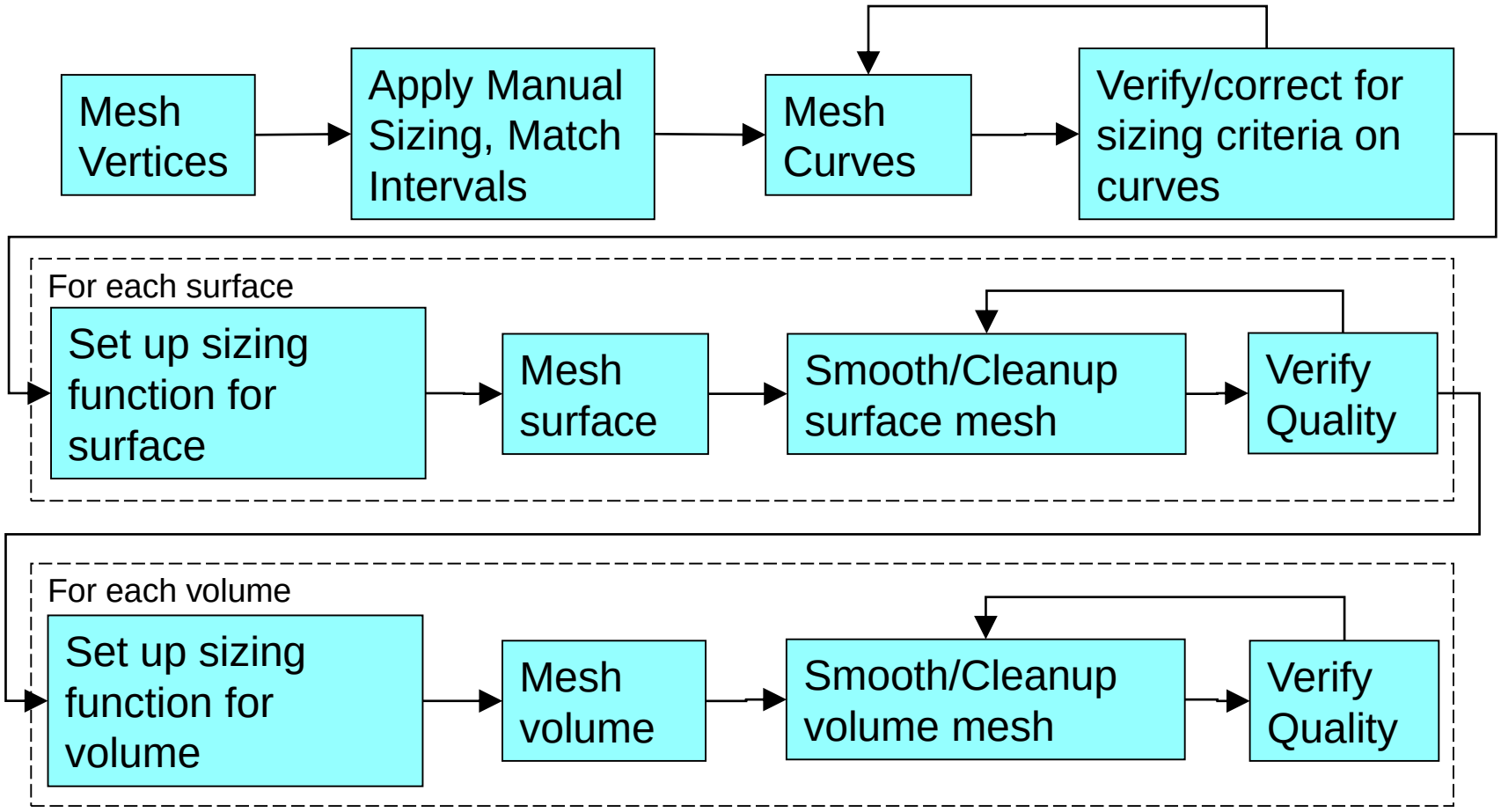
The TE-3000 Semiconductor Tool for Processing 300mm Silicon Wafers is comprised of over 2,000 parts and over 100 assemblies.

Mountain Cycle ShockWave frameset comprised of more than 100 parts



Design of a 4-cylinder engine comprised of 62 parts

Mesh Generation Process



The Mesh Generation Process

Meshing Algorithms