COP290

Pushpam Anand (2016CS10347) , Kulkarni Nikhil Dilip (20116CS10340)

22^{nd} January 2018

1 Objectives

a. To generate projections on to any cutting plane given a 3D model.
b. To reconstruct 3D from given set of 2D views.

2 Scope of the Problem

1) From 3D isometric model to 2D views :

- INPUT= Here we are going to take input as $n \times 3$ matrix as a input ($n \times 3$ = 'n' vertices in (x,y,z)) and a $'n \times n'$ adjacency matrix showing respective edges.

- OUTPUT= format is similar $'m \times 2'$ and $m \times m$ adjacency matrix for both top and front view.

2) From 2D isometric model to 3D views :

- INPUT = We take three 2D views as a input . - OUTPUT = We will take wire-frame model as a output .

3 Transformation of 3D model

1) We represent the vertices of the 3d model using 3*1 matrix . we can do the following transformation of the 3d model .

(i) **TRANSLATION :**

If we want to translate our 3D figure in a any given direction we can do so by translating all the given vertices by using what we call a translation matrix. after transformation the coordinates of the vertices become: $x' = x + T_X$
$y' = y + T_Y$
$z' = z + T_Z$

the translation matrix is: $\begin{bmatrix} 1 & 0 & 0 & T_X \\ 0 & 1 & 0 & T_Y \\ 0 & 0 & 1 & T_Z \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Here the values T_X,T_Y,T_Z are values of respective translations in x,y,z directions.

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x+X \\ y+Y \\ z+Z \\ 1 \end{bmatrix}$$

we're working with 4×1 vectors and subsequently 4×4 transformation matrices, as the last "1" is used in multiplication of the matrices to add the value of transformations. A translation moves a vector a certain distance in a certain direction.

Without the fourth column and the bottom 1 value a translation wouldn't have been possible. We replace 1 by 0 in the rotation matrix .

(ii) **SCALING :**

A scale transformation scales each of a vector's components by a (different) scalar. It is commonly used to shrink or stretch a vector as demonstrated below.

Let say we have to scale a vertex having coordinates (X,Y).

The initial coordinates are : $\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

The final coordiantes are: $\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix}$

and the scaling function does the following: $f(x, y, z, 1) = (x', y', z', 1)$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \times \begin{bmatrix} SX & 0 & 0 & 0 \\ 0 & SY & 0 & 0 \\ 0 & 0 & SZ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} SX.x \\ SY.y \\ SZ.z \\ 1 \end{bmatrix}$$

it is not difficult to come up with a matrix that scales a given vector by (SX,SY,SZ).

For projection onto a general plane

(iii) **ROTATION :**

If we have to take projections on a plane that has direction ratios of normal vector(l,m,n) then we follow the following steps:
1. Rotate the 3D figure by value $R_X(\theta)$, $R_Y(\alpha)$,$R_Z(\beta)$ depending on the values dictated by l,m,n.
$R_X(\theta) : \theta = \frac{m}{n}$

$R_Y(\alpha) : \alpha = \frac{n}{l}$

$R_Z(\beta) : \beta = \frac{l}{m}$

2. Once we have rotated the 3D we can then take projections on former X or Y or Z plane correspondingly to obtain the projection which is equivalent to taking projection on any general plane.

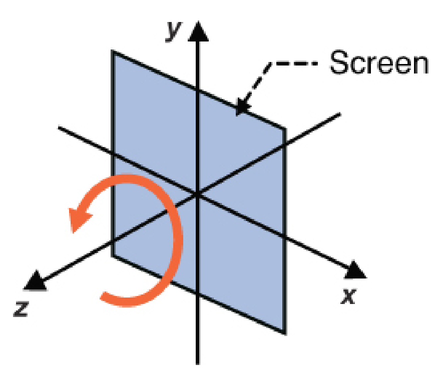

Figure 1: Right hand rotation

A rotation transformation will rotate the edge vectors around the origin using a given axis and angle. After rotation we can take respective projections.

To have a uniform representation we represent a vertex by a 4×1 matrix.

along X-axis

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cosQ & -sinQ & 0 \\ 0 & sinQ & cosQ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ cosQ.y - sinQ.z \\ sinQ.y + cosQ.z \\ 1 \end{bmatrix}$$

along Y-axis

$$\begin{bmatrix} cosQ & 0 & sinQ & 0 \\ 0 & 1 & 0 & 0 \\ -sinQ & 0 & cosQ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} cosQ.x + sinQ.z \\ y \\ -sinQ.x + cosQ.z \\ 1 \end{bmatrix}$$

along Z-axis

$$\begin{bmatrix} cosQ & -sinQ & 0 & 0 \\ sinQ & cosQ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} cosQ.x - sinQ.y \\ sinQ.x + cosQ.y \\ z \\ 1 \end{bmatrix}$$

Any rotation can be given as a composition of rotations about three axes (Euler's rotation theorem), and thus can be represented by a 33 matrix operating on a vector: $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

In a rotation, a vector must keep its original length, so it must be true that:
$x'_i x'_i = x_i x_i$
which rearranges to : $(a_{ij}.x_j)(a_{ik}.x_k) = x_i x_i$
which finally yields: $a_{ij}.a_{ik} = \delta_{jk}$
This condition guarantees that:

$$R^{-1} = R^T$$

and

$$det(R) = 1$$

Proof : **That there is a rotation axis for any given rotation on the body**

since det(R)=1
we get det(R-I)=0
where I is the identity matrix.
i.e., λ=1 is a root of the characteristic equation det(R-λI)=0
In other words, the matrix R I is singular and has a non-zero kernel, that is, there is at least one non-zero vector, say n, for which $(R-I)n = 0$ which implies

$$Rn = n$$

The line n for real is invariant under R, i.e., n is a rotation axis.

4 2D projections from 3D

The basic idea is to project all the vertices of the solid body onto the prospective plane and then according to the given adjacency matrix for the edges we can obtain the solid lines in the orthographic projection.

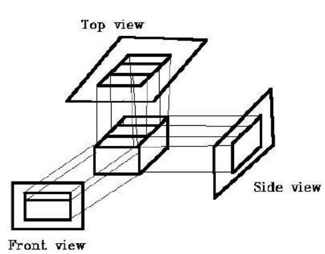

Figure 2: orthographic Projections

For obtaining the dotted lines we have to implement a separate algorithm.

For mapping 3D vertices to 2D points

Here for the Front-view from isometric we can multiply our co-ordinate column matrix with A_{xy} and similarly A_{xz} , A_{yz} for Top , Side view respectively .

$$A_{xy} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Similarly,

$$A_{xz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and $A_{yz} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Now we can plot this (x,y) co-ordinates in Front view .
For showing edges in Front view , We will use adjacency matrix of given isometric view . And we will plot edges in respective enumerated vertices .

How to make hidden lines in 2D views :

While making Top/Front view from 3D isometric figure , there will be some 'Hidden Lines' which needs to be represented as dotted line in 2D views .
For this we will first sort the co-ordinates ((x,y,z,) in 3D view) according their z-coordinates .
And then those edges who have highest z-coordinates as their end points will be represented as solid lines .
But for those edges who have vertices with less z-coordinate will be represented by dotted line –
—For the parts where some points exist in the figure with higher z-coordinate exist for the same x-y-coordinates.
$(i.e.(x, y, z')wherez' > z)$.

5 Reconstruction of 3D polyhedrons from 2D views

For reconstruction of 3D the **input specifications are:**

- We are given 3 $(n \times 2)$ matrices representing the 2-D points of the three orthographic views respectively.

- **Input Assumption :** All the vertices of the 3D is specified in all the orthographic projections.

- Three graphs in the form of adjacency matrix is given to show the respective edges between the points.

Output specifications are:

- We form a $(n \times 3)$ matrix to represent all the 3D vertices.

- We form a adjacency list of enumerated vertices to show the respective edges.

- Hence, we get a wire-frame model of the 3D structure.

We have three adjacency matrix to represent the edges.
We take the logical OR operation of the three adjacency matrices to obtain the list of all the edges.
Once we get all the edges we can construct a wireframe model of the 3D solid.