Contents

Project 1

Engineering Drawing Tool

1.1 Objective

To design and implement a software package for Engineering drawing. The package should have the following functionalities:

- We should be able to interactively input or read from a file either i) an isometric drawing and a 3D object model or ii) projections on to any cross section.
- Given the 3D model description we should be able to generate projections on to any cross section or cutting plane.
- Given two or more projections we should be able to interactively recover the 3D description and produce an isometric drawing from any view direction.

1.2 Assumptions and Reasoning

1. The model consists of a wire frame which is equivalent to a solid with only transparent faces. This will eliminate the need to distinguish between hidden lines and visible lines as now every non-coincident line will be visible.
2. The model is represented as a graph hence we will have the coordinates of all the vertices in the polyhedron as well as some data structure to store the pair of vertices that are connected via a line.
3. We will receive direction cosines of the plane on which the user wants the projection of the solid.
4. We will receive three orthogonal views from the user. In some special cases, it may be possible to uniquely identify the three dimensional model

from two orthographic views only but in the most general case, all the three views will be required. Having two views will not be sufficient in all cases and can lead to multiple possible solutions.

5. For the conversion of two dimensional views to three dimensional model, we will assume that we have the views in mutually orthogonal planes. This will enable us to define the coordinate planes of our axis system along the views.

6. We will assume the orthographic views to be labelled with all the points. This means that each vertex in the polyhedron will be labelled in each of the orthogonal views. Hence we will be given a point to point correspondence between the views. In the case where the projections of several points coincide, we will have all these points labelled to that projection in the order of their distance from the plane of projection.

1.3 Mathematical Analysis Techniques

1. **Translation of Origin:** This process involves the shifting of the origin to a convenient position in order to simplify the calculations. As the name suggests, this will only shift our axis system, the orientation of the model with respect to the shifted axis system remains the same even after shifting. The new axis system is parallel to the old axis system.

   Mathematically, the coordinates of a point with respect to the new origin (ie. after translation) are achieved by subtracting the coordinates of the new origin with respect to the old origin from the coordinates of this point with respect to the old origin. Let c_i^1 and c_i^2 denote the coordinates of the point i before and after translation and O denote the coordinates of the new origin with respect to the old origin.

$$c_i^2 = c_i^1 - O = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} - \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}$$

$$c_i^2 = \begin{bmatrix} x_i^2 \\ y_i^2 \\ z_i^2 \end{bmatrix} = \begin{bmatrix} x_i^1 - x_o \\ y_i^1 - y_o \\ z_i^1 - z_o \end{bmatrix}$$

2. **Rotation about Origin:** This process involves redefining the coordinates of all the points in the model with respect to a new axis system.

The origin of the new axis system is same as that of the old axis system. However, the axis of the new system are inclined at an angle to the axis of the old system. The coordinates of a point after rotation about the origin can be achieved by multiplying the coordinate matrix with what is known as the rotation matrix.

The mathematical details regarding rotation have been provided in section 1.3.2 .

3. **Projection on a Cartesian Plane:** The projection of a point on any of the three Cartesian planes can be obtained by taking the coordinate perpendicular to the plane to be zero. This is because this coordinate represents the distance between the plane and the point. Since the projection of the point lies on the plane itself hence its distance from the plane is zero which is why this coordinate will be zero for the projection.

   Mathematically, the coordinates of the a point can be represented by a 3x1 matrix as follows.

$$Matrix\ of\ Coordinates\ of\ point\ i, c_i = \begin{bmatrix} \mathrm{x}_i \\ \mathrm{y}_i \\ \mathrm{z}_i \end{bmatrix}$$

   The projection of a point on any plane would have only 2 dimensions since a plane is a two-dimensional figure. To obtain them, we will multiply the coordinates of the point with the projection matrix. Let

$$P_{xy}, P_{yz}\ and P_{zx}$$

   denote the projection matrix for projection on the XY, YZ and ZX plane respectively.

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

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

$$P_{zx} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$Projection\,on\,XY\ plane,\ p_{xy_i} = P_{xy} * c_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{bmatrix} = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \end{bmatrix}$$

$$Projection\,on\,YZ\ plane,\ p_{yz_i} = P_{yz} * c_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \end{bmatrix} = \begin{bmatrix} \mathrm{y} \\ \mathrm{z} \end{bmatrix}$$

$$Projection\ on\ ZX\ plane,\ p_{zx_i} = P_{zx} * c_i = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \text{x} \\ \text{y} \\ \text{z} \end{bmatrix} = \begin{bmatrix} \text{z} \\ \text{x} \end{bmatrix}$$

1.4 Orthographic Views from Isometric View

The problem at hand involves the task of deducing the projections of a solid figure on any cross-section or cutting plane given the description of its 3-Dimensional model. Different cases arise as follows:

1. **When the user demands the orthographic projections of the solid on a plane parallel to or coinciding with the Cartesian planes:** This is the simplest of all the possible cases. We already know how to obtain the projection of a point on the Cartesian planes. In case of a solid figure, the same holds for all the points. Hence the projection of the solid on a plane will comprise of the projections of all the point visible from that view. The projections of the non-visible points will form the hidden features in the projection.

   However, since our model consists only of a wire frame, hence the projections of all the points will be visible on the plane and there will be no hidden feature.

   First of all, we will translate the origin so as to simplify our calculation. In fact, some calculations may be practically impossible if we omit this step since the size of registers in our computers are limited and hence they have an upper limit to the feasible calculations. We will take the x-coordinate new origin to be the minimum of the x-coordinates of all the points of the model. Similarly, we will take the y and z coordinates of the new origin. This is because then all the points will lie in the first octant and will be as close to origin as possible. With respect to the old origin, let $O^{'}$ represent the coordinates of the new origin and c_i represent the coordinates of the i^{th} point.

   $$O^{'} = \min_{i=1}^{N} c_i$$

   Next, we will project all the points on the plane and connect those projections with a line that are connected in the three dimensional model. This way we can obtain the orthographic projection on any plane parallel to or coinciding with the Cartesian Plane.

2. **When the user demands the projection on a plane inclined at an angle to the Cartesian planes:** This case is slightly more involved than the previous one. In general, the plane of projection is at an angle to all the 3 standard Cartesian planes. Simply setting one of the coordinates to zero will not reap us the projection in this case. The reason for this is that the projection is obtained by moving in a direction perpendicular to the plane starting from the point. Since the perpendicular from the point to the plane cannot be expressed solely in terms of a single unit vector, hence changing only one of the coordinates will not give us the projection.

   First of all, we will translate the origin similar to what we did in the previous case. As done previously, we will take our new origin to be that point whose i^{th} coordinate is minimum of the i^{th} coordinates of all the points. But still we cannot apply projection yet since none of the Cartesian planes are parallel to the projection plane. In order to overcome this problem, we will reorient our axis system such that any one of the new Cartesian planes becomes parallel to the projection plane. This is effectively equal to taking one of the new coordinate axis in a direction perpendicular to the plane of projection. Having done that, the problem reduces to the one that we have dealt with in the previous case. Without loss of generality, let us take our new Z axis to be normal to the plane of projection. Now, the projection on the XY plane is the required projection.

1.4.1 Rotation

Basic Rotation :

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

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

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

General Rotation :

Let α, β, γ represent the (improper) Euler angles of extrinsic rotation about axes x, y, z. Then the overall rotation matrix comes out to be as given below :

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha)$$

$$\begin{bmatrix} \text{x'} \\ \text{y'} \\ \text{z'} \end{bmatrix} = R \begin{bmatrix} \text{x} \\ \text{y} \\ \text{z} \end{bmatrix}$$

`Note : The rotation matrix multiplication is not commutative and hence it's cyclic order must be preserved.`

Finding Proper Euler Angle

Assuming a frame with unit vectors (X, Y, Z) given by their coordinates as in the main diagram, it can be seen that:

$$\cos(\beta) = Z_3$$

And, since

$$\sin^2 x = 1 - \cos^2 x$$

we have

$$\sin(\beta) = \sqrt{1 - Z_3^2}$$

As Z_2 is the double projection of a unitary vector,

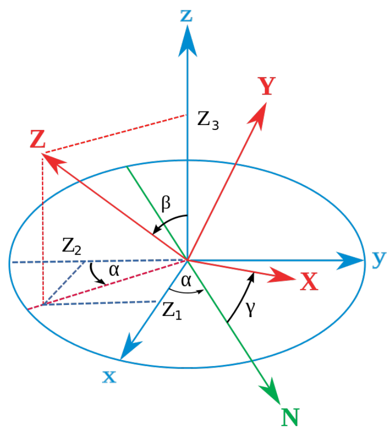
$$\cos(\alpha).\sin(\beta) = Z_2$$
$$\cos(\alpha) = Z_2/\sqrt{1 - Z_3^2}$$

There is a similar construction for Y_3, projecting it first over the plane defined by the axis z and the line of nodes. As the angle between the planes is $\pi/2 - \beta$ and $\cos(\pi/2 - \beta) = \sin(\beta)$, this leads to:

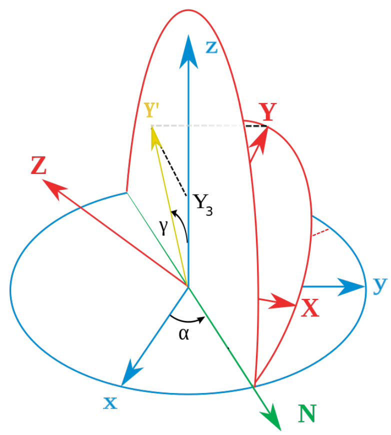
$$\sin(\beta).\cos(\gamma) = Y_3$$
$$\cos(\gamma) = Y_3/\sqrt{1 - Z_3^2}$$

and finally, using the inverse cosine function,

$$\alpha = arccos(Z_2/\sqrt{1 - Z_3^2})$$
$$\beta = arccos(Z_3)$$
$$\gamma = arccos(Y_3/\sqrt{1 - Z_3^2})$$



(a) Projections of Z Vector



(b) Projections of Y Vector

Figure 1.1: Projections of New Co-ordinate Vectors

1.5 Isometric View from Orthographic Views

Our approach consists of the following three stages:

1. Preprocessing the engineering drawings to prepare for 3D reconstruction.

   The input data are checked for validity and each view is separated and identified from the input engineering drawing. Some aided information needed by the reconstruction process is added.

2. Analyzing the relationship between 3D conic edges and their projections using matrix theory to construct a wire-frame model.

   This stage presents details of constructing the wire-frame model from three orthographic views, which includes creating 3D candidate vertices and edges.

   Detailed techniques for generating 3D edges are explained.

3. Reconstructing the 3D object by searching for the faces and solids within the wire-frame model.

   At this stage, all candidate faces are created by searching all fundamental edge loops in the wire-frame.

   Depth information in the engineering drawings is completely utilized to remove pseudo edges and faces.

   At last, all true faces are assembled to form 3D objects.

Determining 3D Vertices from 2D Vertices

A candidate vertex is created from the 2D vertices in three views. Let $N_f = (N_f(x), N_f(z), N_t = (N_t(x), N_t(y), and N_s = (N_s(y), N_s(z)$ be 2D vertices in the front, top, and side view respectively. If

$$|N_f(x) - N_t(x)| < \epsilon,$$
$$|N_t(y) - N_s(y)| < \epsilon, and$$
$$|N_s(z) - N_f(z)| < \epsilon$$

then we know that N_f, N_t, N_s are corresponding in each view of a 3D vertex. Where ϵ is the tolerance.

Determining the edges

As of now we have all the vertices in the three dimensional model but we are yet to determine which vertices are connected by an edge and which are not. This is what we plan to do in this section.

1. First, we will make 3 graphs, one for each of the orthographic view which contain the information on the vertices and edges in that two dimensional view. Each vertex in these graphs is represented by two coordinates. Hence many vertices of the 3-dimensional model will map to the same vertex in these graphs. Points that overlap in a view map to the same vertex in the graph corresponding to that view.
2. Consider a particular pair of vertices be C_1 and C_2. Now take the projections of these two vertices on the X-Y plane. Let the projections be P_1 and P_2.
3. Check if there exists an edge between P_1 and P_2 in graph for X-Y plane.
4. Repeat steps 2 and 3 for the other two planes as well. If a pair of point does not have an edge between their projections in even one of the views then there will not be an edge between them in the three dimensional model. Else, they will have an edge between them.
5. Repeat steps 2,3 and 4 iterating over all possible pairs of vertices in the three dimensional model. Thus all edges in the 3 dimensional model have been obtained and a wire-frame is ready.

Determining the faces

The determination of faces is done in a similar fashion as that of edges. We iterate over the set of all possible faces. For each possible face, the projection of its boundary is considered in all the three views. If the projection of this boundary does not enclose a face in even one of the views then this is not a valid face in the three dimensional model. The face is valid only if all its projections are valid faces in their respective views.

1.6 References

- Rotation Matrix - Wikipedia
- Euler Angles - Wikipedia
- A Matrix-Based Approach to Reconstruction of 3D Objects from Three Orthographic Views - Research Paper