Indian Institute of Technology, Delhi

Design Practices in Computer Science
COP290

Starling Simulation

April 18, 2018
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Authors :
Udit Jain
(2016CS10327)
Shashank Goel
(2016CS10332)

Supervisor :
Prof. Subhashish Bannerjee

ABSTRACT

We are going to design and implement a Starling Simulation that will help us understand the Flocking behaviour of birds. Computer Science term for this simulation is Boids.

The simulation software will have the following functionalities:

1. We will be able to interactively input or read from a file :
   - Degree of Seperation, Cohesion, Alignment.
   - Speed of boids .
2. We can interactively add / remove *boids and obstacles* based on combination of inputs from keyboard and mouse.
3. Get the color of boids based by grouping on speed, flock, direction .
4. Calcuate and display average Velocity and Accleration vector of boid groups and global flock dynamically.
5. Calcuate and display average Kinetic Energy of boid groups and global flock dynamically.

Future scope of the problem may include:

1. Interactively control the Gravity vector and see it's effect on the flock .
2. Adding and measuring the effect of Wind speed / direction and Drag on the flock .

In this design project, we shall work as observers, developers and algorithm enthusiasts to understand the ways and finding different means to approach and tackle the objectives in a more well defined mathematical way.

We shall work with full confidence and zeal to achieve the goal or reach to quite an end of the problem so that using our lemmas, proofs and knowledge, someday a perfect model can be implemented using a software by some other Computer Explorer. As a matter of interest, we just wish to argue that these things can be computed by our brain so we do hope to find a solution to this problem using machine learning algorithms. Since, Machine Learning algorithms are more or less based on Mathematical matrices, with the use of computer graphics, we expect to find a start with matrices that we have dealt with further in this report.

Contents

Chapter 1

Introduction

In the last ten years, a significant progress occurred in the area of 3D graphics. Many studies have been conducted in the field of 3D modeling, and a variety of methods that allow us to reconstruct 2D images into 3D were created. Today, 3D graphics industry creates models that can no longer be distinguished from a person in the real world or on photograph. This way of modeling is also the goal of this work; to explore options to create a photo-realistic 3D model shaped from the 2D images. This article has listed and briefly described methods of converting 2D images into 3D models.

The following objectives are aimed to be covered in this paper:

- Working out a mathematical description of the problem.
- Figure out how many views are necessary for reconstruction.
- Figure out how many views are sufficient for reconstruction.
- How to compute the projections from the 3D model?
- How to produce the isometric view using one or more projections?
- What interactions are necessary?

We shall study the direction cosines and direction ratios of a line joining two points and also discuss about the equations of lines and planes in space under different conditions, angle between two lines, two planes, a line and a plane, shortest distance between two skew lines and distance of a point from a plane. Most of the above results are obtained in vector form. Nevertheless, we shall also translate these results in the Cartesian form which, at times, presents a clearer geometric and analytic picture of the situation.

We shall further study the Vector Algebra and the 3D Transformations required to convert a given 3D Object to its projection view on any cross section and also do the same in a reverse manner i.e. converting the given projections back into the 3D Object by using matrices and their properties trying to exploit as many as possible and making them use to determine the number of different possible reconstructions possible (if any). We shall describe the assumptions while formulating the problem and prove to detail all the lemmas and theorems that are being used to define the model.

Chapter 2

Vector Algebra and 3D Transformation

2.1 Homogeneous coordinate System

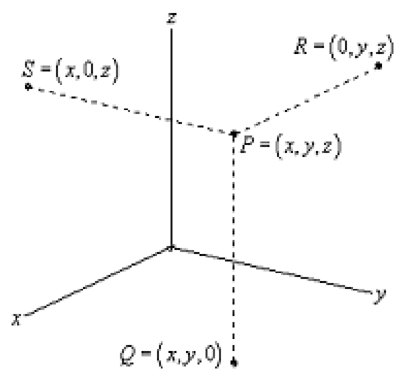
Three-dimensional scene description requires mainly using a 3D Cartesian coordinate system. Points in space are uniquely determined by their three *Cartesian coordinates* (x, y, z) .

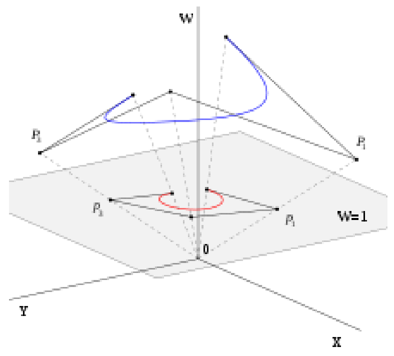
Of greater importance for computer graphics is the usage of *homogeneous* or *projective coordinates* . Ordinary points in space are given four coordinates instead of three:

2.1.1 Introduction

$$(x, y, z) \Leftrightarrow (x, y, z, w)$$

This coordinate system has wide range of applications, including computer graphics and 3D computer vision, where they allow affine transformations and projective transformations to be easily represented by a matrix





This introduces an obvious redundancy, so that the same point in 3D has infinitely many homogeneous coordinates, according to the equivalence

$$(x, y, z, w) \equiv (x_0, y_0, z_0, w_0) \Leftrightarrow \alpha \; (x, y, z, w) = (x_0, y_0, z_0, w_0)$$

2.1.2 Homogeneous Matrix

The concatenation of a translation with a rotation, scaling or shear requires an awkward combination of a matrix addition and a matrix multiplication. The problem can be avoided by using an alternative coordinate system for which computations are performed by 3 × 3 matrix multiplications. Since

$$(x' \;\; y' \;\; l) \;\; = \;\; (x' \;\; y' \;\; l) \begin{pmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{pmatrix}$$

To this end a new coordinate system is defined in which the point with Cartesian coordinates (x, y) is represented by the homogeneous or projective coordinates $(x, y, 1)$, or any multiple (rx, ry, r) with $r \neq 0$.

The set of all homogeneous coordinates (x, y, w) is called the *projective plane* and denoted P^2. In order to carry out transformations using matrix computations the homogeneous coordinates (x, y, w) are represented by the row matrix (x, y, w).

The above equation implies that any planar transformation can be performed by a 3 × 3 matrix multiplication and using homogeneous coordinates. Sometimes homogeneous coordinates will be denoted by capitals (X, Y, W) in order to distinguish them from the affine coordinates (x, y).

2.1.3 Key Aspects

- Any point in the projective plane is represented by a triple (X, Y, Z) called the *homogeneous coordinates* or *projective coordinates* of the point, where X Y and Z are not all 0 .

- The point represented by a given set of homogeneous coordinates is unchanged if *the coordinates are multiplied by a common factor* .
- Conversely, two sets of homogeneous coordinates represent the same point if and only if one is obtained from the other by multiplying all the coordinates by the same *non-zero constant* .
- When Z is not 0 the point represented is the point$(X/Z, Y/Z)$ in the Euclidean plane.
- When Z is 0 the point represented is a point at $infinity$.

2.2 Vector Algebra

In mathematics and linear algebra,vector algebra refers to algebraic operations in vector spaces. Most commonly, it refers to operations on Euclidean vectors.

2.2.1 Vector Operations

Vectors in 3D space are usually given by their Cartesian coordinates, and operations on vector can be defined in terms of them:

Addition : $(x, y, z) + (x_0, y_0, z_0) = (x + x_0, y + y_0, z + z_0)$
Subtraction : $(x, y, z) - (x_0, y_0, z_0) = (x - x_0, y - y_0, z - z_0)$
Scaling : $\lambda(x, y, z) = (\lambda x, \lambda y, \lambda z)$
Dot Product : $(x, y, z) \cdot (x_0, y_0, z_0) = xx_0 + yy_0 + zz_0$
Norm : $\|(x, y, z)\| = \sqrt{|\overrightarrow{v} \cdot \overrightarrow{v}|} = \sqrt{x^2 + y^2 + z^2}$
Cross Product : $(x, y, z) \times (x_0, y_0, z_0) = (yz_0 - zy_0, zx_0 - xz_0, xy_0 - yx_0)$

2.2.2 Geometric Interpretation

The dot product is fundamentally a projection. The dot product of a vector with a unit vector is the projection of that vector in the direction given by the unit vector. This is given by the formula:

$$\overrightarrow{v} \cdot \overrightarrow{w} = |\overrightarrow{v}||\overrightarrow{w}|cos(\theta)$$

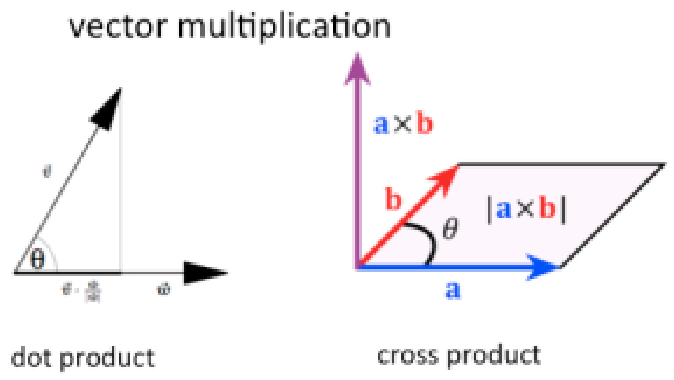
Furthermore, it follows immediately from the geometric definition that two vectors are orthogonal if and only if their dot product vanishes, that is :

$$\overrightarrow{v} \perp \overrightarrow{w} \Leftrightarrow \overrightarrow{v} \cdot \overrightarrow{w} = 0$$

The component form of the dot product now follows from its properties given above. For example,

$$\overrightarrow{v} \cdot \overrightarrow{w} = (v_x\widehat{i} + v_y\widehat{j}) \cdot (w_x\widehat{i} + w_y\widehat{j}) = v_xw_x + v_yw_y$$

Orthogonal : Two vectors are orthogonal if their dot product is zero.
Collinear : Two vector are collinear if their cross product is zero.
Normal : A normal vector to the plane defined by $\overrightarrow{v}$ and $\overrightarrow{w}$ is given by $\overrightarrow{v} \times \overrightarrow{w}$



2.2.3 Direction Cosines

If a directed line $\mathcal{L}$ passing through the origin makes angles α, β and γ with x, y and z-axes, respectively, called direction angles, then cosine of these angles, namely, $cos\alpha, cos\beta$ and $cos\gamma$ are called direction cosines of the directed line $\mathcal{L}$. If we reverse the direction of L, then the direction angles are replaced by their supplements, i.e.,$\pi - \alpha, \pi - \beta$ and $\pi - \gamma$. Thus, the signs of the direction cosines are reversed.

Note that a given line in space can be extended in two opposite directions and so it has two sets of direction cosines. In order to have a unique set of direction cosines for a given line in space, we must take the given line as a directed line. These unique direction cosines are denoted by l, m and n.

Let a,b,c be the direction ratios of a line and let l , m and n be the *direction cosines* of the line. Then,

$$\frac{l}{a} = \frac{m}{b} = \frac{n}{c} = k(constant)$$

Therefore

$$l = ak, m = bk, n = ck$$

But

$$l^2 + m^2 + n^2 = 1$$

Therefore

$$k^2(a^2 + b^2 + c^2) = 1$$

$$k = \pm\frac{1}{\sqrt{a^2 + b^2 + c^2}}$$

2.3 Transformations on 3D Objects

A **transformation** is a function f that maps a set χ to itself, i.e. f: $\chi \to \chi$. Examples include rotations, reflections and translations. These can be carried out in Euclidean space, particularly in $\mathbb{R}^2$and $\mathbb{R}^3$. They are also operations that can be described explicitly using matrices.

2.3.1 Translation

Translation is one of the most basic operations that can be performed on an object. It can be defined as the elementary transformation which can preserve relative positions of any two points in the coordinate space.

A translation is a transformation which maps a point P (x, y) to a point $P'(x, y)$by adding a constant amount to each coordinate so that

$$x' = x + h, y' = y + k$$

for some constants h and k.

In general, for the case of the 3D coordinate system the translation is defined by the following rules:

$$x = x + hy = y + kz = z + l$$

2.3.2 Rotation

A rotation about the origin through an angle θ has the effect that a point P(x, y) is mapped to a point $P(x, y)$ so that the initial point P and its image point P' are the same distance from the origin, and the angle between lines OP and OP is θ .

Since we have been considering the fact that matrices are important in the study of coordinate geometry involving projections, we must not proceed without getting the glimpse of this transformation in terms of matrices. The Rotation of the coordinate system about the conventional orthogonal axis of rotation and along any axis is specified below. It might

be quite intriguing to the reader how the rotation of a point about a given axis can actually be synced or thought about to be multiplication with a matrix.

Lemma 2.3.1. *Multiplication of a matrix, having a unit determinant and transpose same as its inverse, with any point vector in the coordinate space essentially amounts to its rotation around a fixed axis and by a fixed angle.*

1. Anti-clockwise Rotation about x-axis by an angle θ.
2. Anti-clockwise Rotation about y-axis by an angle θ.
3. Anti-clockwise Rotation about z-axis by an angle θ.

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

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

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

2.3.3 General Rotation about any Axis

In order to get the rotation matrix for rotation about any axis with direction cosines u_x, u_y, u_z by an angle θ is given by :

$$R = \begin{bmatrix} cos(\theta) + u_x^2(1 - cos(\theta)) & u_x u_y(1 - cos(\theta)) - u_z sin(\theta) & u_x u_z(1 - cos(\theta)) + u_y sin(\theta) \\ u_y u_z(1 - cos(\theta)) + u_z sin(\theta) & cos(\theta) + u_y^2(1 - cos(\theta)) & u_y u_z(1 - cos(\theta)) - u_x sin(\theta) \\ u_z u_x(1 - cos(\theta)) - u_y sin(\theta) & u_z u_y(1 - cos(\theta)) + u_x sin(\theta) & cos(\theta) + u_z^2(1 - cos(\theta)) \end{bmatrix}$$

2.3.4 A simple derivation for the case of 2D view

First of all, it must been seen that rotating about any of the orthogonal axis is , in practice a rotation in the traditional 2D plane. This is because one of the parameter of the position triple has to be the same after rotation depending upon which axis you choose to rotate about. Thus the general two dimensional matrix can be easily extended to simple three dimensional case.

The rotation about the origin in the x-y pplane corresponds to the following matrix that we wish to derive:

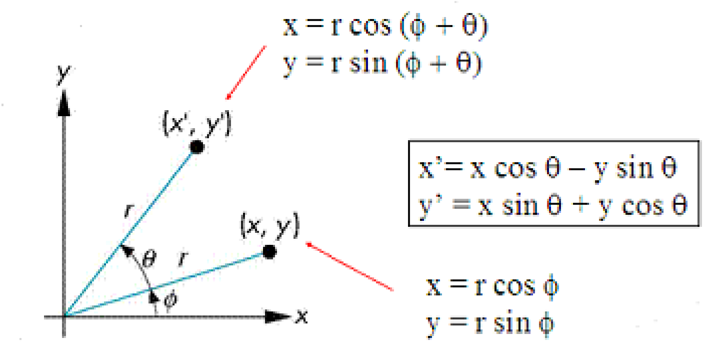
$$\begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

The derivation of the new coordinates of a generic point P can be inferred from the geometry that has been presented in the following figure:

The matrix

$$S(s_x, s_y) = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$$

is called the *scaling transformation matrix.*



The coordinates(x, y)can be obtained from$(x, y)$by the matrix multiplication

$$\mathbf{P} = \begin{pmatrix} x & y \end{pmatrix} \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} = \begin{pmatrix} xcos(\theta) - ysin(\theta) & xsin(\theta) + ycos(\theta) \end{pmatrix}$$

The matrix:

$$Rot(\theta) = \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix}$$

is called the *Rotation Matrix*.
The reader must notice that the rotation matrices may have a difference in sign of . Because of the direction of rotation that we might decide.

2.3.5 Scaling

Scaling is a linear transformation that enlarges or shrinks an object by scale factors that may or may not be equal in all directions. The scaling matrix is any diagonal matrix with positive reals as elements.

A scaling about the origin is a transformation which maps a point$P(x, y)$to a point$P(x, y)$by multiplying the x and y coordinates by non-zero constant scaling factors sx and sy, respectively, to give $x = S_x X and y = S_y Y$.

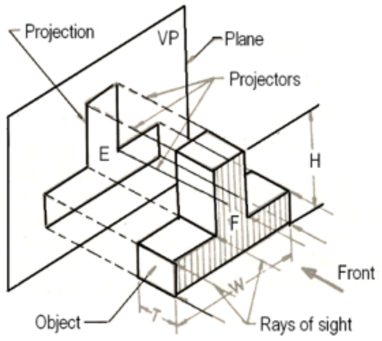
Scaling is not as such a very useful tool to change the view angle or just change the position of the object but enables us to change its size (as seen by us) with a given viewing screen or better say field of view.

Chapter 3

3D to 2D Conversion

3.1 Introduction

In this Chapter, we shall describe a mathematical model to get the projection of a 3D Model onto any plane of projection that the software user wishes to choose. We will exploit the lemmas and the basics of theory that we studied in Chapter 1 and derive new results and then combine them all to lay our model of 3D 2D projection. The whole theory will mostly work around matrices that we have studied but may be in a slightly different way, but understandable.



The idea basically consists of projecting the points that are a part of our object on the plane of interest/projection. Now, it is wise to think that there are an infinite number of points present in the object and it is unrealistic to fin the projection of every point. So, out model must involve some assumptions which have been listed in the next section.

3.2 Assumptions

- The given Object shall consist of *only straight lines* with well defined end points and *no curved surfaces* . The same shall be assumed for the Object whose projections shall be given to us.

- The Views that will be given to us must be *labelled with the coordinates* of their corner points and be aligned with the other necessary views in a perfect orientation for correct interpretation

Basis of Assumptions
make the above heading larger
The first assumption has excluded the possibility of curved surfaces to be present in the object. Excluding these types of surfaces might not essentially rule out the possibility of infinite number of reconstructions from the projection views but surely it would rule out many of them.

Frankly speaking , the mathematical construct used to convert the isometric view to the projection view might get complicated in some of the planes of projection as we will see in some of the examples illustrated further.

The second assumption needs to be taken into account because aligning the views and interpreting the corner points by image reading and then further naming them with some appropriate coordinates has been assumed to be out of the scope of this design project.

3.3 Projection on the Plane

3.3.1 Projection of a point on a plane

Consider a point$P(x_0, y_0, z_0)$ and a general plane in 3D space, represented by the equation $ax + by + cz = d$. This is represented in the following figure:
Image below

Now, let us derive the coordinates of the point P, the projection of the point P on the plane represented by the above equation.

Let the coordinates of the point P be (x, y, z).

Therefore, the direction cosines of the line joining the point$P$and the point$P$ must be the same as the direction cosines of the normal of the plane. Now, the direction ratios of any normal to the plane represented by the above equation is a, b, crespectively along the x, y, zdirection.

Therefore, we must have :

$$\frac{x - x_0}{a} = \frac{y - y_0}{b} = \frac{z - z_0}{c} = \Bbbk$$

Furthermore, the point Plies on the plane and hence must satisfy the equation of the plane and therefore $ax + by + cz = d$. Substituting the value ofx, y, z in terms of $x_0, y_0, z_0, \Bbbk$from the direction equality equation gives us the value of $\Bbbk$ as

$$\Bbbk = \frac{-(ax_0 + by_0 + cz_0 - d)}{a^2 + b^2 + c^2}$$

Hence we can calculate the coordinates of point $P$by substituting the value of $\Bbbk$ in the direction equality equation. [1]

3.3.2 Key Aspects

We have already discussed quite a lot on how we should proceed with the projection onto a given plane and how points can be rotated and transformed in 3D space. Since edges and planes are attached to points that surround them, they also get rotated along with points.

Now, as discussed before we have an infinite number of points on the 3D object, so should we consider all of them? Well . . . of course not. So, before proceeding any further, we must study and understand the following theorem:

Theorem 3.3.1. *The projection of the line on a plane is the line joining the projection of the points that lie on the original line*

[1]Note: We shall use the transformation math to view the object from the different angles i.e isometric views.

Proof. There are two points on a line whose projection is taken on a plane. Now, the projection of a line on a plane must be a line which can be uniquely identified by two points. Also, the projected line must pass through the two points of projection and hence the projection of the line on a plane is the line joining the projection of the points that lie on the original line. □

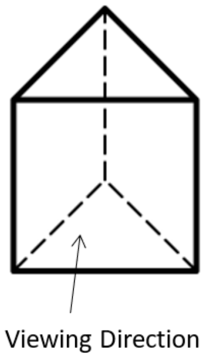
Lemma 3.3.2. *The points that form an edge in the 3D model also form an edge in the projected view provided that both of them are visible in the projection.*

To project a certain orthographic view, we will use the following technique:

1. Firstly, rotate the object so that the plane of projection coincides with the XY plane.
2. Start from the point having Z coordinate farthest from the viewer or the plane of projection depending on the view we choose to pursue.
3. Set the Z coordinate to zero to project points onto the plane. Do this in the order that points farther from the viewer get projected first so that hidden areas can be actually made hidden in the projected view.
4. Use (why not find this ref?-¿) 3.3.2 to create the required edges as solid.
5. Now, for the points that have not been projected, we shall join all of their edges with any other point with a hidden line if the hidden line does not overlap with the solid line.
6. Finish when all points have been projected.

Example

This can be illustrated for the following case of a triangular prism:



The front four points can be seen from the viewing direction and hence we mark them. Then we draw solid edges between those points which also have a solid edge between them in the prism. We therefore get Figure (i). Then the two point at the back are joined by hidden lines to all the other points that they are connected to in such a way that the hidden line does not overlap with a solid line. We therefore get Figure (ii) which is required and correct orthographic projection.

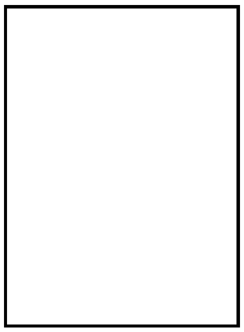


Figure (i)

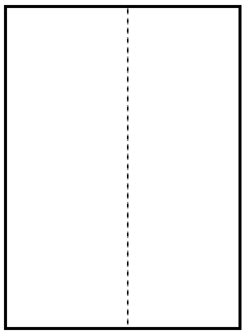


Figure (ii)

Keeping in mind all the assumptions, the analytical and the mathematical model described above quite well portray the method of obtaining the projection along any viewing direction or onto any plane of interest.

We must see that curved surfaces would require an infinite number of applications of the projection formula to get a proper well defined curved edge to be obtained in the projection view.

Chapter 4

2D to 3D Conversion

4.1 Introduction

In the previous chapter, we studied the method of producing a projection of a 3D model onto any plane. In this chapter, we shall, what we say, study the reverse of that process i.e. using two or more orthographic projections to determine the 3D model.
The task of converting multiple 2D images into 3D model consists of a series of processing steps: Since Projection drops a dimension, converting from 2D to 3D is always a complex process.

recon.PNG

Any orthographic projection provides us two views, but this is clearly neither necessary nor sufficient. The key for this process is the relations between multiple views which convey the information that corresponding sets of points must contain some structure and that this structure is related to the poses and the calibration of the camera.

4.2 Overview

First of all, it must be understood that any given set of orthographic projections might not represent a realizable 3D model. So, is there a maximum number for which, that number of given views might always realize a 3D model? Well, let us look at the next example:
Example

Oviews.PNG

The simple orthographic projections shown above do not seem to generate any kind of 3D object. Thus, orthographic views might not always generate a 3D model even if their corresponding dimensions match as shown in the above case.
However, complex figure then these might also generate perfect models and hence one must make sure of all the cases and look for the things very carefully.

4.3 Number of Views

We must take care about the number of views that are available to us in order to produce a given 3D Model. Clearly, two orthographic projections arent sufficient to produce a 3D model. For eg two of the orthographic views in case of a well-chosen cuboid and a triangular prism are same and therefore two views arent sufficient.

4.3.1 Minimum views necessary for reconstruction

4.3.2 Minimum views sufficient for reconstruction

To get the minimum number of views sufficient for reconstruction, we can just prove that it can be done in some n number of views and prove that it cannot be done in less than that.
Now, one view is for sure insufficient because we can just extrude in number of ways to get the same projection all the time.
But consider the two vies given below, the 3D model generated by these views is unique and no other model can generate the same, atleast not with straight lines.
[Something incomprehesible here!]

4.4 Algorithm

Now comes the part of where we will define the procedural structure to carry out the process of reconstructing the 3D object. We have assumed that all the three views are given to us in most of the cases as mentioned in the Assumptions section.

The algorithm we shall use is described as the following:

1. First, we need to satisfy the constraint of point matching in different view including the dimensions.

2. Then we shall project the points in the Front and Top View individually to cover their abscissa and ordinate whatever they might correspond to, without disturbing the constraints.

3. If the side view is available apply above procedure to all other pairs of views.

4. Create the points space as a cross product of coordinate sets that are possible. Create all edges that are present on atleast one view.
5. Since the projections are obtained by only using the points and their corresponding edges, we shall check if the orthographic projections match with the given one.
6. Then we shall start removing edges making the orthographic projection closer to the predicted ones.
7. We shall solidify the face enclosed by four edges.

Depending on the constraints the set of points, that will be generated will be in most of the cases the corner points of the actual 3D model that we wish to generate however in case our algorithm fails, steps 7 and 8 shall take care of it.

Chapter 5

Conclusion

Bibliography

[1] P. Allen, B. Landman and H. Meeks, New Bounds on van der Waerden type numbers for Generalized 3-term Arithmetic Progressions, *arXiv: 1201.3842v2*

[2] S. Burr and S. Loo, On Rado numbers II, unpublished.