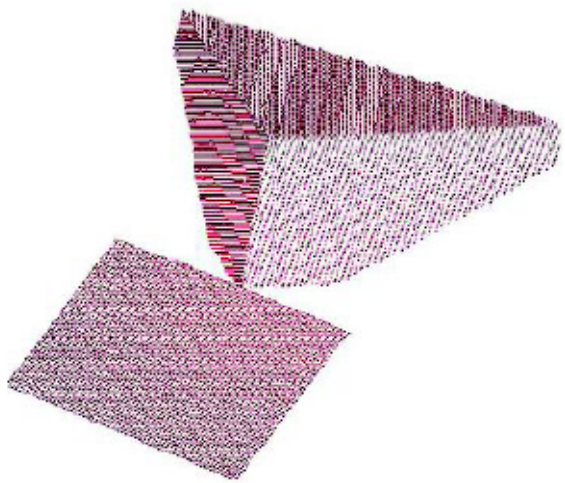
3D Wireframe Modelling

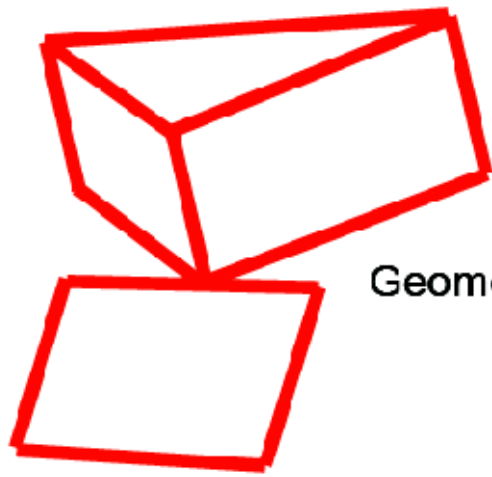
Robert B. Fisher
School of Informatics
University of Edinburgh

3D Geometric Modelling

Goal: model 3D objects for recognition



Data (from scanner)



Geometric Model

Recognition requires some sort of model
Easier matching if data and model use same representations

3D Coordinate Systems

Like 2D systems: for modelling and object pose

Need rotation and translation specification
Translation easy - 3D vector $\vec{t} = (t_x, t_y, t_z)'$

Rotation needs 3 values. Many different coding systems.

Altogether, 6 degrees of freedom = 6 position parameters.

Typical Rotation Specification

Arbitrary angle order, so specify rotation as:

$$\mathrm{R} = \mathrm{R}_x(\theta_x)\mathrm{R}_y(\theta_y)\mathrm{R}_z(\theta_z)$$

Where

$$\mathrm{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta_x) & -sin(\theta_x) \\ 0 & sin(\theta_x) & cos(\theta_x) \end{bmatrix}$$

$$\mathrm{R}_y(\theta_y) = \begin{bmatrix} cos(\theta_y) & 0 & sin(\theta_y) \\ 0 & 1 & 0 \\ -sin(\theta_y) & 0 & cos(\theta_y) \end{bmatrix}$$

$$\mathrm{R}_z(\theta_z) = \begin{bmatrix} cos(\theta_z) & -sin(\theta_z) & 0 \\ sin(\theta_z) & cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation parameters are: $\{\theta_x, \theta_y, \theta_z\}$
Here, we used a right-hand coordinate system. A positive rotation is defined as clockwise when you look along the rotation axis starting from the origin.

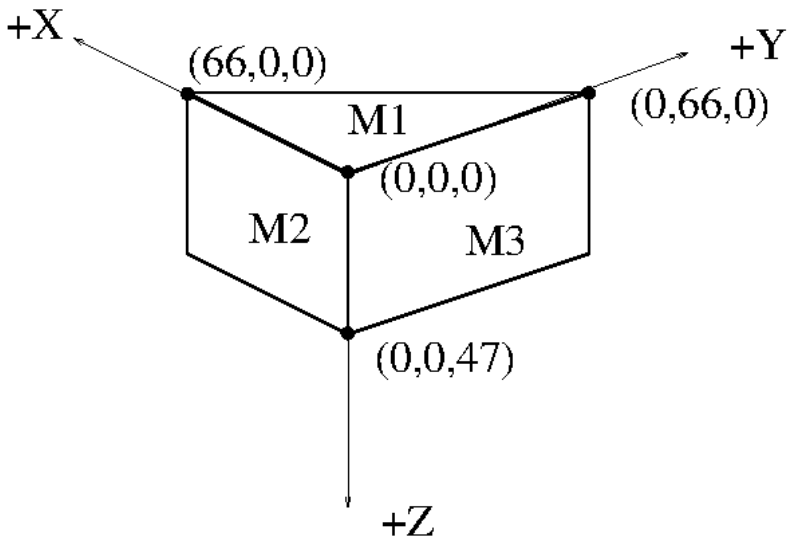
Other systems possible: yaw/pitch/roll, azimuth/elevation/twist

Different parameter values, but always the same rotation, when encoded in matrix R

Object position/translation: vector in R^3.

3D Shape Modelling

Similar to 2D Modelling
Needs 3D coordinate system + 3D shape primitives



Our primitives: polyhedra, defined by polygonal patches, defined by lists of edges

Wireframe modelling

Representation Scheme

Model: set of polygons (object faces)

Polygons: set of edges (polyhedron edges)

Edge: 2 points in R^3 (edge endpoints)

Wedge Model

```
planenorm(1,:) = [0,0,-1];        % tri face 1 surf normal
facelines(1) = 3;                 % # of boundary lines
model(1,1,:) = [0,0,0,66,0,0];    % Edge 1
model(1,2,:) = [0,0,0,0,66,0];    % edge 2
model(1,3,:) = [0,66,0,66,0,0];   % edge 3
planenorm(2,:) = [0, -1, 0];      % rect face 2 surf normal
facelines(2) = 4;
model(2,1,:) = [0,0,0,0,0,47];
model(2,2,:) = [0,0,0,66,0,0];
model(2,3,:) = [66,0,0,66,0,47];
model(2,4,:) = [0,0,47,66,0,47];
planenorm(3,:) = [-1, 0, 0];      % rect face 3 surf normal
facelines(3) = 4;
model(3,1,:) = [0,0,0,0,0,47];
...
```

What We Have Learned

- A simple 3D shape modelling scheme
- A review of 3D coordinate systems