# Lecture 11: Group Assignment 1 Review, Procrustes Intro

COMPSCI/MATH 290-04

Chris Tralie, Duke University

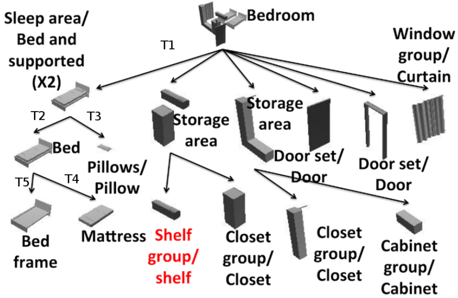
2/18/2016

# Table of Contents

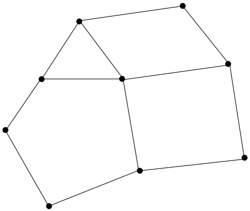
# Code Layout: Recursive Scene Graph Traversal

```
f(node, mvMatrix):
    for c in node.children:
        if ('mesh' in node) {
            //Do some stuff if this isn't a dummy node
        }
        f(c, mat4.mul(mvMatrix, node.transform));
```

Call this function with “scene” to start the recursion



# Code Layout: Meshes (What Is A Mesh??)



```
var mesh = node.mesh;
//Loop through faces
for (var f = 0; f < mesh.faces.length; f++) {
    //"Pointer" to face
    var face = mesh.faces[f];
    //For each face get vertices in CCW order
    var verts = face.getVerticesPos();
    //Do stuff with the vertices...
}
```

# Code Layout: Image Sources

```
scene.imsources = [scene.source];
for (order = 1:k):
    for s in scene.imsources:
        if s.order == order-1 {
            //Reflect (call recursive scene tree
                function)
            //Generate a bunch of images snew
            snew.parent = s
            snew.genFace = face reflected
        }
```

# Reflections / Projections

# Reflections / Projections

Plane: $(\vec{q}, \vec{n})$

Point: $\vec{p}$

Reflection: $\vec{p} - 2((\vec{p} - \vec{q}) \cdot n)\vec{n}$

# Ray Intersect Plane

# Ray Intersect Plane

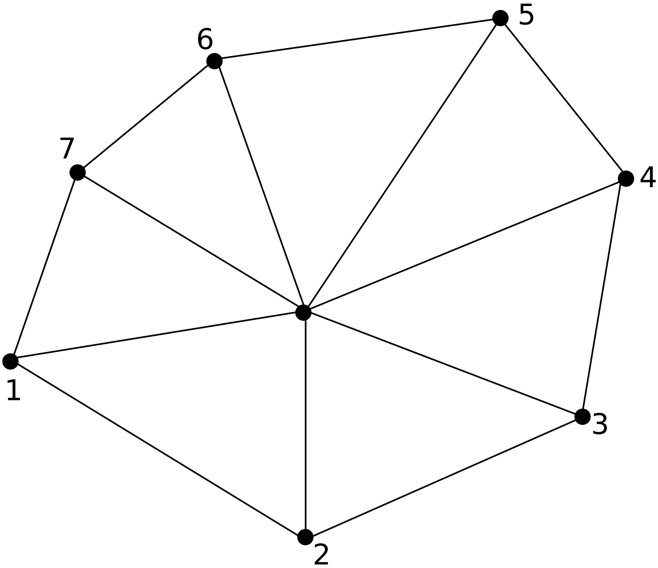
$$(\vec{p_0} + t\vec{v} - \vec{q}) \cdot \vec{n} = 0$$

# Ray Intersect Plane

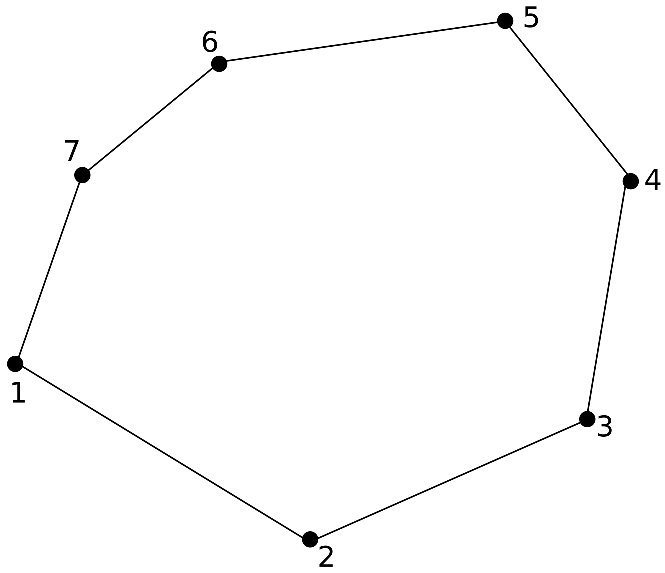
$$(\vec{p_0} + t\vec{v} - \vec{q}) \cdot \vec{n} = 0$$

$$t = \frac{(\vec{q} - \vec{p_0}) \cdot n}{\vec{v} \cdot \vec{n}}$$

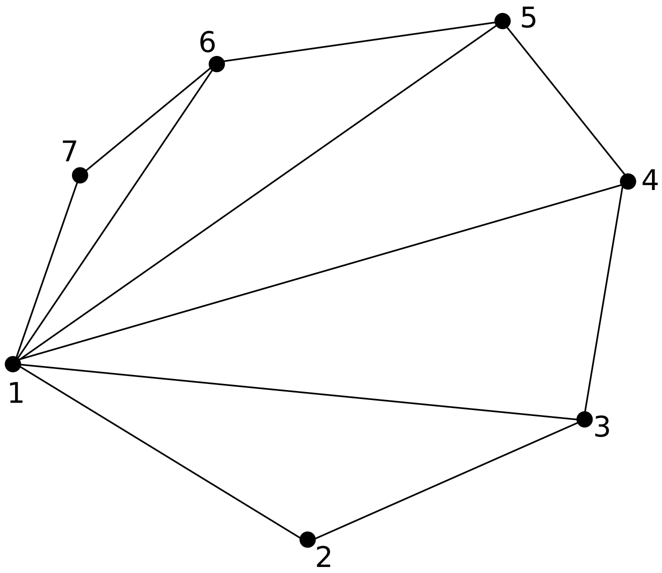
# Point Inside Convex Polygon: Area Test



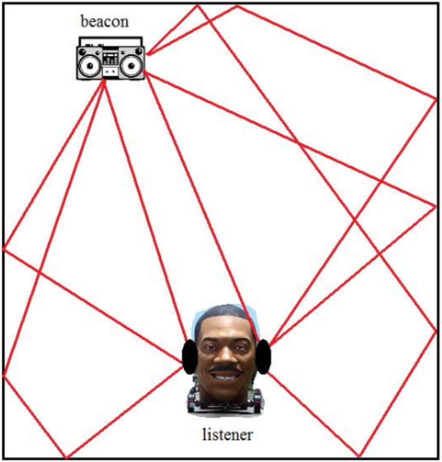
# Convex Polygon Area?



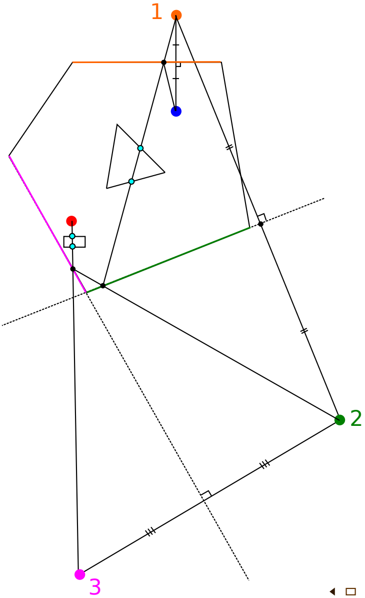
# Convex Polygon Area: Triangle Fan



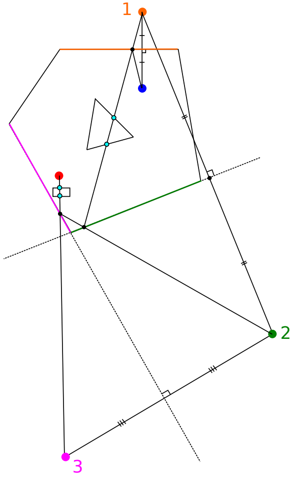
# Extra Stuff: Binaural Sound



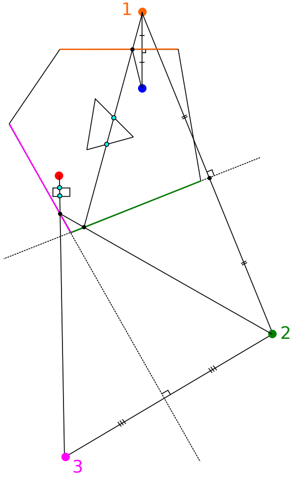
# Extra Stuff: Transmission



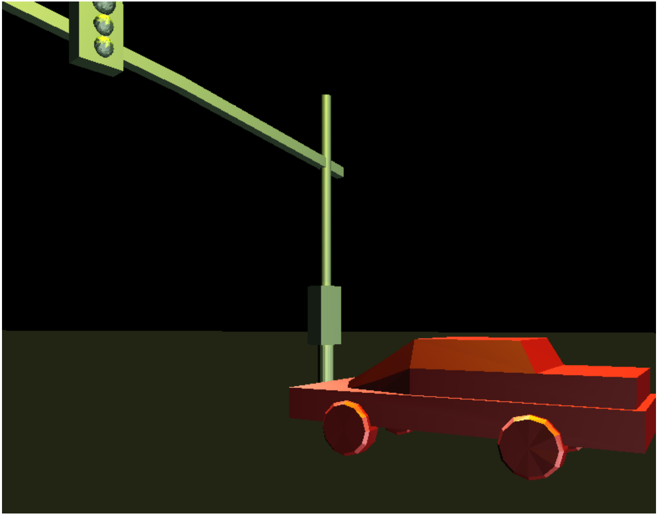
Regular expressions



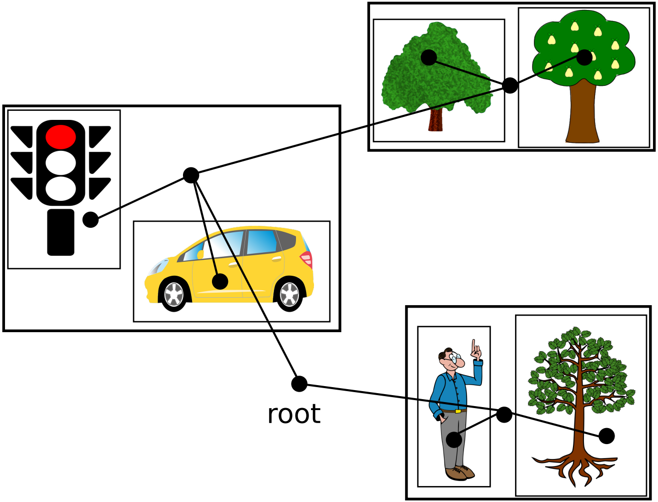
Regular expressions $(r|t)*$



# Extra Stuff: Frequency dependent transmission



# Extra Stuff: Bounding Box Speedup



# Table of Contents

# PCA New Convention

Organize point cloud into $d \times N$ matrix, each point *along a column*

$$X = \begin{bmatrix} | & | & \cdots & | \\ \vec{v}_1 & \vec{v}_2 & \vdots & \vec{v}_N \\ | & | & \cdots & | \end{bmatrix}$$

Choose a unit column vector direction $u \in \mathbb{R}^{d \times 1}$
Then

$$d = u^T X$$

gives projections onto $u$

# PCA New Convention

Organize point cloud into $d \times N$ matrix, each point *along a column*

$$X = \begin{bmatrix} | & | & \dots & | \\ \vec{v}_1 & \vec{v}_2 & \vdots & \vec{v}_N \\ | & | & \dots & | \end{bmatrix}$$

Choose a unit column vector direction $u \in \mathbb{R}^{d \times 1}$
Then

$$d = u^T X$$

gives projections onto $u$

- ▷ More consistent with what we've done; points in columns

# PCA New Convention

$$d = u^T X$$

# PCA New Convention

$$d = u^T X$$

- ▷ How to express the sum of the squares of the dot products?

# PCA New Convention

$$d = u^T X$$

- How to express the sum of the squares of the dot products?

$$dd^T$$

# PCA New Convention

$$d = u^T X$$

- How to express the sum of the squares of the dot products?

$$dd^T$$

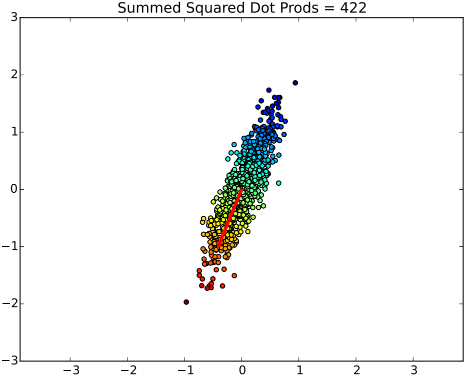
$$dd^T = (u^T X)(u^T X)^T = u^T X X^T u$$

Want to find $u$ that maximizes the above quadratic form

# PCA New Convention

Use eigenvectors of $A = XX^T$ to find principal directions maximizing $u^T A u$

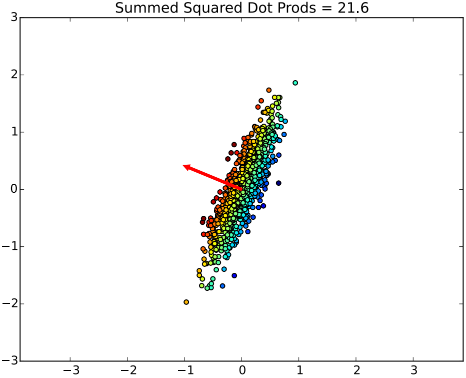
$$\lambda_1 = 422$$



# PCA New Convention

Use eigenvectors of $A = XX^T$ to find principal directions maximizing $u^T Au$

$$\lambda_2 = 21.6$$

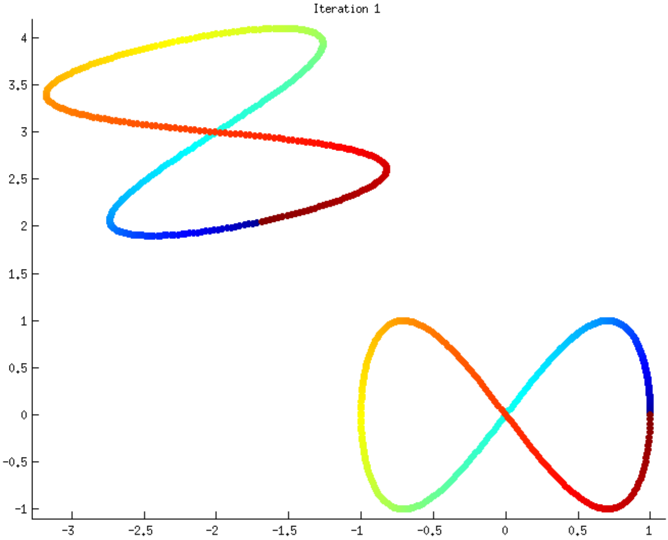


# Table of Contents

# Procrustes Distance

http://www.procrustes.nl/gif/illustr.gif

# Procrustes Alignment



# Procrustes Distance

Given two point clouds $\{\vec{x}_i\}_{i=1}^N$ and $\{\vec{y}_i\}_{i=1}^N$
where $x_i$ and $y_i$ are in correspondence
Seek to minimize

$$\sum_{i=1}^{N} ||R(\vec{x}_i + \vec{t}) - \vec{y}_i||_2^2$$

over all orthogonal matrices $R$ and translation vectors $t$. $||.||^2$ is squared distance