Thursday, September 21, 2006
3:45 PM

Assignment 4:

- very similar to assignment 2, just in 3D
- steps 3b & 3c of phase I (all implementation work)

1. read in pts

```
Point *ptP;
vector<Point*> pts
while (std::cin >> (ptP = new Point(0.0, 0.0, 0.0)))
    pts.push_back(ptP);
```

3D! make sure your point class supports this

2. calculate mean - same as 2D except

```
Point mean(0.0, 0.0, 0.0)
```

make sure you sum over x, y, z

3. Calculate covar. matrix

```
matrix<double> cov(3, 3)
```

3x3 matrix

4. get eigenvals, eigenvects — jacobi() steps too soon

3 floats — 3 1x3 vectors → 3x3 matrix

Thursday, September 21, 2006
3:45 PM

Phase I

1. read in list of 3D points, $x_i = (x, y, z)$
2. create kd-tree on points (find min, max bounding box)
   - before hand
   - Point min(0,0,0) max(20,0,0)
   - min: smallest x, y, z values
   - max: largest x, y, z values
3. for each point x_i:
   a. obtain Nbhd(x_i) (Knn query with x_i as query point; use $K = 5$) — a vector of Point 3D
   b. calculate mean of Nbhd(x_i), call it o_i
   c. calculate & sort eigenvalues + eigenvectors of Nbhd(x_i)
      3 evals: eval$_0$, eval$_1$, eval$_2$
      3 vects: evec$_0$, evec$_1$, evec$_2$
      set n_i = evec$_2$ — eigenvector associated with smallest eval
   d. store Nbhd(x_i), o_i, n_i in TangentPlane class, add new TangentPlane(Nbhd, o, n) to list of tangent planes

Vector<Point3D> Point vector<double>