KAIST CS492 - Homework 3

XU YIN - 20205445

November 8, 2021

Question 4

- **We need to first calculate T_{ij} and K_{ij} (see Fig. 2).**

```
Matrix3d app_M = Matrix3d::Zero();


for (Mesh::VertexOHalfedgeIter vf_it = mesh.voh_iter(vi); vf_it.is_valid(); ++vf_it)// find the neighbohood point J
{

    Vec3f point_j = mesh.point(mesh.to_vertex_handle(vf_it.handle()));
    Vector3d coor_j(point_j[0], point_j[1], point_j[2]);
    //cout << coor_i << endl;
    //cout << coor_j << endl;
    Vector3d vji = coor_i - coor_j;

    Vector3d T_ij = (Matrix3d::Identity() - n * n.transpose())*vji / ((Matrix3d::Identity() - n * n.transpose())*vji).

    //cout << T_ij << endl;
    double K_ij = 2 * n.transpose().dot((vji)) / vji.dot(vji);
```

Figure 1: Calculate T and K.

- **Calculate the surface area and then get the weight.**

```
//calculate the weight w_{ij}

    double wij = mesh.calc_sector_area(vf_it.handle()) + mesh.calc_sector_area(mesh.opposite_halfedge_handle(vf_it.handle()));
    area += wij;

    app_M += wij * K_ij*T_ij*T_ij.transpose();
```

Figure 2: Calculate the weight of the triangle.

- **Calculate the eigen vectors and eigen values.**

```
SelfAdjointEigenSolver<Matrix3d> solver(app_M);
double eigen1 = solver.eigenvalues()(0);
double eigen2 = solver.eigenvalues()(1);

Vector3d T11 = solver.eigenvectors().col(0);
Vector3d T22 = solver.eigenvectors().col(1);


double m11 = T11.transpose()*app_M*T11;
double m22 = T22.transpose()*app_M*T22;
```

Figure 3: Get the eigen vectors and eigen values.

• **Set the property.**

```
// In the end you need to fill in this struct

CurvatureInfo info;
info.curvatures[0] = 3 * m11 - m22;
info.curvatures[1] = 3 * m22 - m11;
info.directions[0] = Vec3f(T11(0), T11(1), T11(2));
info.directions[1] = Vec3f(T22(0), T22(1), T22(2));

if (fabs(info.curvatures[0]) > fabs(info.curvatures[1])) {
    double exchange = info.curvatures[0];
    info.curvatures[0] = info.curvatures[1];
    info.curvatures[1] = exchange;

    Vec3f exchange2 = info.directions[0];
    info.directions[0] = info.directions[1];
    info.directions[1] = exchange2;
}
```

Figure 4: Set the curvature with the property.

• **Demonstration of different models**

   – Example result of the "hand" model

   – Example result of the "horse" model

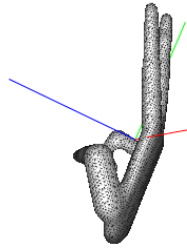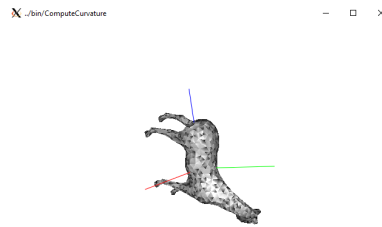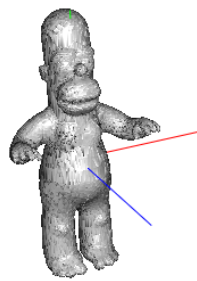   – Example result of the "homer" model

Figure 5: Example result 1.

Figure 6: Example result 2.

Figure 7: Example result 3.