

Computer Graphics: Geometry and Simulation

Coursework 3: Debugging Geometry and Simulation

Oliver Jones (s2153980)

Introduction

Throughout this report are a series of code snippets, highlighting the changes made to the original code to fix the bugs. The pink sections have been removed from the original code, while the green sections have been added.

1 As-Rigid-as-Possible Deformation

1.1

```
1 g.row(j) = (origV.row(E(j, 0 1)) - origV.row(E(j, 1 0))) * REdge;
```

Listing 1: The 0 and 1 have been swapped.

1.2

```
1 MatrixXd rhs = d0I.transpose() * W * (g - d0B * constPositions);
```

Listing 2: The * W was missing.

1.3

```
1 for (int k = 0; k < oneRings[j].size(); k++) {
2     P.row(k) = origV.row(oneRings[j][k]) - origV.row(j);
3     P.row(k) = origV.row(oneRings[j][k]) - origV.row(j);
4     Q.row(k) = currV.row(oneRings[j][k]) - currV.row(j);
5     Q.row(k) = currV.row(oneRings[j][k]) - currV.row(j);
6 }
7 S = Q.transpose() * P;
8 Eigen::JacobiSVD<Eigen::Matrix3d> svd(S, Eigen::ComputeFullU
  | Eigen::ComputeFullV);
9 Eigen::Matrix3d U = svd.matrixU();
10 Eigen::Vector3d Sigma = svd.singularValues();
11 Eigen::Matrix3d Vt = svd.matrixV().transpose();
12 Matrix3d currR = U * Vt;
13 if (currR.determinant() < 0.0) {
14     // check where the smallest singular values falls
```

```

15 int minValue, minIndex;
16 minValue = Sigma.minCoeff(&minIndex);
17 Matrix3d newSigma = Matrix3d::Identity();
18 newSigma(minIndex, minIndex) = -1;
19 currR = U * newSigma * Vt;
20 }
21 R[j] = currR;
22 }

```

Listing 3: Lines 3 and 5 have been removed. The `for` loop is now closed on line 6 instead of 22. Lines 7 to 21 are no longer indented.

2 Multi-body Rigid Simulation

2.1

Pairwise gravity was reversed. Two possible fixes:

```

1 forces.row(i).array() = forces.row(i) - + forceDirection *
  gravityConstant / sqrDistance;
2 forces.row(j).array() = forces.row(j) + - forceDirection *
  gravityConstant / sqrDistance;

```

Listing 4: The `+` and `-` have been swapped.

```

1 RowVector3d forceDirection = spherePoses.row(j,i) - spherePoses.
  row(i,j);

```

Listing 5: The `i` and `j` have been swapped.

2.2

```

1 if (sqrDistance < 2 4 * sqrRadius) {

```

Listing 6: The `2` has been replaced with `4`.

2.3

```

1 if (sphereVelocities(i, 1) > < 0.0)

```

Listing 7: The `>` has been swapped with `<`.