Développement d'applications interactives

Compte-rendu TP3

Déformation de surfaces

Louis Jean

Master 2 IMAGINE
Université de Montpellier
N° étudiant : `21914083`

25 septembre 2024

Table des matières

1 Introduction

L'objectif de ce TP était d'étudier les manières de déformer les maillages surfaciques, notamment en explorant la méthode As-Rigid-As-Possible (ARAP) vue en cours, grâce à une base de code fournie.

2 Systèmes linéaires

Dans un premier temps, nous avons fait quelques rappels d'algèbre linéaire.

2.1 Exercice 1

Nous avons commencé doucement, en résolvant un système d'équations à 3 inconnues.

$$\begin{cases} x_0 + x_1 = 1 \\ x_1 + x_2 = 0 \\ x_0 + x_2 = 0 \end{cases} \implies x_1 - x_2 = 1 \implies x_2 = x_1 - 1 \implies \begin{cases} x_0 + x_1 = 1 \\ 2x_1 = 1 \\ x_0 + x_1 = 1 \end{cases} \implies \begin{cases} x_0 = \frac{1}{2} \\ x_1 = \frac{1}{2} \\ x_2 = -\frac{1}{2} \end{cases}$$

2.2 Exercice 2

Puis, nous avons réécrit le système ci-dessus sous forme de produit matrice-vecteur.

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

2.3 Exercice 3

Enfin, nous avons traduit ce système linéaire en `C++` pour se familiariser avec la base de code fournie.

```
void testlinearSystem() {
    {
        linearSystem mySystem;
        mySystem.setDimensions(3 , 3);

        mySystem.A(0,0) = 1.0;  mySystem.A(0,1) = 1.0;  mySystem.A(0,2) = 0.0;
        mySystem.A(1,0) = 0.0;  mySystem.A(1,1) = 1.0;  mySystem.A(1,2) = 1.0;
        mySystem.A(2,0) = 1.0;  mySystem.A(2,1) = 0.0;  mySystem.A(2,2) = 1.0;

        mySystem.b(0) = 1.0;
        mySystem.b(1) = 0.0;
        mySystem.b(2) = 0.0;

        mySystem.preprocess();
        Eigen::VectorXd X;
        mySystem.solve(X);
        std::cout << X[0] << "  " << X[1] << "  " << X[2] << std::endl;
    }
}
```

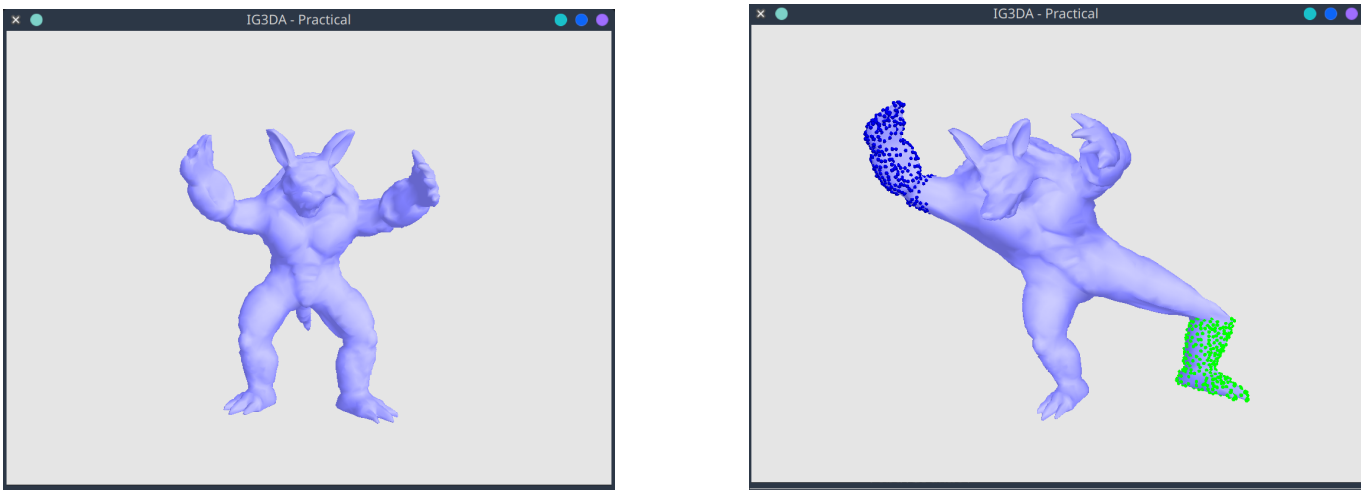
3 Déformation aussi rigide que possible (ARAP)

Dans cette section, nous avons implémenté la méthode ARAP pour déformer les maillages surfaciques. Cette méthode vise à minimiser l'énergie de déformation en conservant autant que possible les structures locales du maillage. Nous avons d'abord mis à jour la matrice **A**, qui dépend des indices des sommets sélectionnés, puis le vecteur **b**, qui dépend des positions des handles et des rotations locales des sommets, le tout en se basant sur les équations du cours.

```
// Mise a jour des dimensions
unsigned int ncolumns = mesh.V.size() * 3;
unsigned int nrows = 0;
for( unsigned int v = 0 ; v < mesh.V.size() ; ++v ) {
    unsigned int numberOfNeighbors = edgeAndVertexWeights.get_n_adjacent_edges(v);
    nrows += numberOfNeighbors * 3;
}
for( unsigned int v = 0 ; v < mesh.V.size() ; ++v ) {
    if(verticesHandles[v] != -1) {
        nrows += 3;
    }
}
...
// Mise a jour de A
unsigned int equationIndex = 0;
for( unsigned int v = 0 ; v < mesh.V.size() ; ++v ) {
    for( std::map< unsigned int , double >::const_iterator it = edgeAndVertexWeights.get_weight_of_adjacent_edges_it_begin(v) ;
    it != edgeAndVertexWeights.get_weight_of_adjacent_edges_it_end(v) ; ++it) {

        unsigned int vNeighbor = it->first;
        for(unsigned int i = 0; i < 3; ++i) {
            arapLinearSystem.A(equationIndex,v*3 + i) = -1;
            arapLinearSystem.A(equationIndex++,vNeighbor*3 + i) = 1;
        }
    }
}
for( unsigned int v = 0 ; v < mesh.V.size() ; ++v ) {
    if(verticesHandles[v] != -1) {
        for(unsigned int i = 0; i < 3; ++i) {
            arapLinearSystem.A(equationIndex++, v*3 + i) = 1;
        }
    }
}
...
// Mise a jour de b
for(unsigned int i = 0; i < 3; ++i) {
    arapLinearSystem.b(equationIndex++) = rotatedEdge[i]; // x, y et z
}
for(unsigned int v = 0 ; v < mesh.V.size() ; ++v ) {
    if(verticesHandles[v] != -1) {
        for(unsigned int i = 0; i < 3; ++i) {
            arapLinearSystem.b(equationIndex++) = mesh.V[v].p[i];
        }
    }
}
...
// Mise a jour de la matrice de rotation
tensorMatrix += it->second * rotatedEdge * initialEdge.transpose();
```

Ensuite, nous avons pu sélectionner des handles et les utiliser dans le cadre de ARAP pour déformer le maillage.



(a) Avant la déformation

(b) Après la déformation

Figure 1: Illustration de la déformation ARAP

4 Conclusion

Ce TP nous a permis de mettre en pratique les concepts vus en cours sur la déformation de surfaces, notamment via la méthode ARAP. En implémentant cette technique, nous avons appris à gérer des systèmes linéaires complexes et à appliquer des transformations locales rigides pour conserver les propriétés géométriques des maillages. La semaine prochaine, nous verrons comment sélectionner des handles sur une sphère, avec une approche basée sur la distance euclidienne puis la distance géodésique.

Merci pour le temps et l'attention que vous avez consacrés à la lecture de ce compte-rendu.