SHORTEST PATHS ON SURFACES GEODESICS IN HEAT

INF555 Digital Representation and Analysis of Shapes

20 novembre 2015

Ruoqi He & Chia-Man Hung

ÉCOLE POLYTECHNIQUE
UNIVERSITÉ PARIS-SACLAY

1 INTRODUCTION

In this project we present practical methods for computing approximate shortest paths (geodesics) on a triangle mesh. We implemented a naive method based on Dijkstra's algorithm and a method based on heat flow, proposed by Crane [2013]. We created a walking man on surface to represent the shortest path taken. The implementation is done in C# on the Unity platform.

2 ALGORITHMS

In this section we explain the main algorithms used in this project.

2.1 Dijkstra

2.2 Heat flow

This algorithm determines the geodesic distance to a specified subset of a given domain.

Notations : heat flow u, vector field X, distance function ϕ.

2.2.1 • First Step

Integrate the heat flow $\dot{u} = \Delta u$.

u of a point on a considered surface is a value between 0 and 1. Source have 1 as its value.

2.2.2 • Second Step

Evaluate the vector field $X = -\nabla u / |\nabla u|$.

We are only interested in the direction of ∇u and not in its value. X points to the opposite direction of the source.

2.2.3 • Third Step

Solve the Poisson equation $\Delta \phi = \nabla \cdot X$.

If a distance function ϕ exists, $\nabla\phi$ should give us a unit vector on every point, pointing to the opposite directin of the source. We approximate such a distance function ϕ by minimizing $\int |\nabla\phi - X|^2$, which is equivalent to solving the Poisson equation $\Delta\phi = \nabla \cdot X$.

2.3 Walking man

We also implemented a navigation system, represented by a walking man. He is positioned on the surface of the mesh and moves toward a given source point.

A walking man's position is defined by barycentric coordinates of a triangle of the mesh. A walk function takes a distance as its argument and first applys on the triangle on which he is standing and then recursively calls itself at the next triangle he comes across, until walking the given distance or reaching the source.

3
Implementation

3.1 Dijkstra

3.2 Heat Flow

We used the half-edge data structure to present triangle meshes.

3.2.1 • First Step

3.2.2 • Second Step

3.2.3 • Third Step

3.3 Mapping

Il faut chercher où se trouve le répertoire de ton installation LaTeX. Pour cela (et quelque soit ton système d'exploitation), ouvre une ligne de commandes et tape :

```
kpsewhich -var-value TEXMFMAIN
```

Rends-toi alors dans le répertoire indiqué que l'on nommera par la suite *INSTALL*. Dans une autre fenêtre de ton explorateur de fichiers, extrais l'archive téléchargée et ouvre le dossier *source*. Il suffit ensuite de copier tous les fichiers `.pdf`, `.eps` et `polytechnique.sty` vers le dossier *INSTALL/tex/latex/polytechnique*.

Il te faut enfin mettre à jour la la liste de package de ta distribution. Pour Mac ou Linux, ouvre un terminal et entre `sudo texhash`. Pour Windows, ouvre dans tous les programmes l'utilitaire *Settings (Admin)* et clique sur *Refresh FNDB*.

De cette manière, l'installation n'est pas complète mais le package marchera très bien en utilisation. Il te manquera par contre les fichiers source commentés si tu veux le modifier.

3.4 Walking Man

La documentation du package ets le fichier `polytechnique.pdf` qui se trouve entre autres. dans le dossier *source* de l'archive extraite.

4
DIFFICULTIES

choice of t (numerical error)
boundary conditions
mesh : triceratops, sphere

5
RESULTS

5.1 Comparison between Dijstra and heat flow

5.2 Added values

OFF parser
mesh : sphere + maze
math proof (matrix definite negative)
mapping