

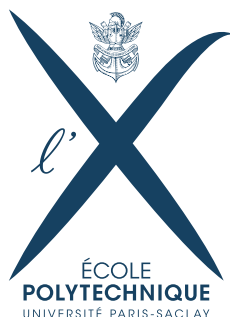


SHORTEST PATHS ON SURFACES GEODESICS IN HEAT

**INF555 Digital Representation
and Analysis of Shapes**

22 novembre 2015

Ruoqi HE & Chia-Man HUNG



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 of Heat Flow based on « Geodesics in Heat : A New Approach to Computing Distance Based on Heat Flow », written 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. The first part is a quick description of the method introduced in the paper. The second part is an algorithm created on our own.

2.1 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 .

2.1.1 • 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.

2.1.2 • 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.

2.1.3 • 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 direction 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 $\Delta\phi = \nabla \cdot X$.

2.2 NAVIGATION

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 applies 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 HEAT FLOW

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

3.1.1 • FIRST STEP

3.1.2 • SECOND STEP

3.1.3 • THIRD STEP

3.2 NAVIGATION

3.3 BOUNDARY CONDITIONS

Dirichlet condition
Neumann condition

4 RESULTS

4.1 MAIN RESULTS

lots of graphs here!

numerical error when $t = 1$ (choice of t), tweak parameter

4.2 COMPARISON BETWEEN DIJSTRA AND HEAT FLOW

4.3 ADDED VALUES

4.3.1 • HOW WE OBTAIN MESHES ?

Below are the four options we used.

1. OFF format (high genus, bague, triceratops, horse) -> mesh -> halfedge mesh
2. FBX format (dragon) -> processing (weld uv seams) -> halfedge mesh
3. 3Dmax handmade mesh (maze) -> halfedge mesh
4. Code generated mesh (sphere) -> halfedge mesh

weld uv seams by Kd tree

Fix vertices (triceratops)

4.3.2 • OPTIMISATION

Our solver ALGLIB solves sparse symmetric positive-definite matrix system a lot faster than only symmetric matrix system. We noticed that $-Lc$ is a symmetric positive-semidefinite matrix. Therefore, $A - tLc$ (for $t > 0$) in the first step is a symmetric positive-definite matrix and $\epsilon I - Lc$ (for $\epsilon > 0$) in the third step is also a symmetric positive-definite matrix. Below is a math proof.

Prove that $-Lc$ is a symmetric positive-semidefinite matrix.

We note $U_{ij} = E_{ii} + E_{jj} - E_{ij} - E_{ji}$ where E_{ij} is an elementary matrix with only one nonnull value 1 on position (i, j) . By definition, $-Lc$ is in form $-Lc = \sum_{i,j} u_{ij} U_{ij}$ with all $u_{ij} > 0$. Since the eigenvalues of U_{ij} are 0 and 2, it is a symmetric positive-semidefinite matrix. We conclude by saying that any positive combination of symmetric positive-semidefinite matrices is also a symmetric positive-semidefinite matrix.

store a Lc matrix, a cot array

4.3.3 • MAPPING

Albedo, Normal, Specular Smoothness, Emission

4.3.4 • NAVIGATION

See above.

5 EXTENSIONS

add weight

applications : video games, robotics