

Моделирование изгибов бумаги

Позднякова Алиса Б05-924

12 декабря 2022 г.

1 Описание проекта

Цель проекта: Написать библиотеку, с помощью которой возможно получить 3d mesh файл, содержащий представление изогнутой бумаги.

Получение представления:

(1) Построение геодезической кривой

Реализованный подход к моделированию изгиба бумаги основан на том, что разворачивающаяся поверхность однозначно определяется геодезической кривой на ней.

Def. Геодезическая кривая - кривая на поверхности, такая, что на разворачивающейся поверхности она представляет собой прямую линию.

Геодезическую кривую можно представить с помощью кривой Безье. В проекте использовались кривые Безье 3 степени.

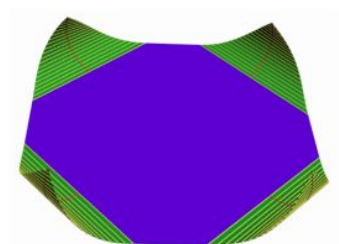
Для регулирования изгибов бумаги, в проекте используются 4 геодезические кривые. Каждая из них начинается в углу бумаги, и заканчивается в точке пересечения плоского участка бумаги и искривленного региона, соответствующего ей.

Точки на кривой добавляются до тех пор, пока кривая не станет гладкой (все углы не меньше некоторого ϵ).

(2) Вычисление вершин и граней 3d mesh

По точкам геодезической кривой построим прямые. Назовем их правилами. По свойствам бумаги при изгибе сохраняются углы и расстояния. Значит по соответствующим им прямым на развертке можно вычислить длины сторон граней-четыреугольников.

Таким образом для каждой геодезической кривой получаем изогнутый регион. Ограничением в принимаемых данных является условие, что искривленные регионы не пересекаются (функциональность расчета 3d mesh в точках сингулярности не реализована)



Красные линии - геодезические кривые
Зеленые линии - правила
Синяя часть - плоский регион

2 Зависимости и сборка проекта

Код: https://github.com/alicepozd/crumpled_paper

В этом же репозитории есть doxygen документация

Чтобы начать использовать модуль:

(1) Зависимости:

Проект написан с использованием библиотеки 3d визуализации libgl [4].

Для запуска проекта необходимо установить её, а так же ее зависимости (библиотека Eigen [5] для работы с векторами и матрицами).

(2) Сборка проекта:

Соберите библиотеку rapper_mesh:

Замените в CMakeLists путь к библиотекам libgl и Eigen.

Подставьте вместо path_to_project путь до директории и выполните следующий код:

```
1 doxygen
2 mkdir build
3 cd build
4 cmake ./path_to_project
5 make
```

3 API библиотеки:

(1) Структура Point:

Является представлением 3d точки.

Имеет поля:

```
1 double x;
2 double y;
3 double z;
```

И следующие функции:

```
1 Point(double x, double y, double z)
```

Конструктор точки по набору координат.

```
1 Point(const Point& another)
```

Конструктор точки по другой.

```
1 Point get_mid(const Point& another, double t = 0.5) const;
```

Функция, возвращающая точку, делящую отрезок в отношении t.

```
1 double get_dist(const Point& another) const;
```

Функция, находящая расстояние между точками.

```
1 double get_angle(const Point& first, const Point& second) const;
```

Функция находит угол, между лучами, проходящими через точки first и second.

(2) Структура Line:

Структура, реализующая прямую $ax + by + c = 0$.

Имеет поля:

```
1 double a;  
2 double b;  
3 double c;
```

И следующие функции:

```
1 Line(std::pair<double, double> p, std::pair<double, double> q)
```

Конструктор прямой по двум точкам.

```
1 Line(double a, double b, double c)
```

Конструктор прямой по ее коэффициентам.

```
1 std::pair<double, double> intersection(Line another) const
```

Возвращает точку пересечение прямых.

```
1 Line perpendicular(std::pair<double, double> p) const
```

Возвращает прямую, проходящую через точку p и являющуюся перпендикуляром.

```
1 std::pair<double, double> get_point_on_dist(const std::pair<double, double>& from,  
2      const std::pair<double, double>& to, double dist)
```

Параметры:

Point from - точка начала отсчета

Point to - точка к которой направлен вектор

double dist - расстояние

Находит точку на прямой, отстоящую от точки from на расстояние dist по направлению к точке to.

(3) Класс Geodesic_line:

Класс реализует геодезическую кривую, как кривую Безье.

Содержит следующие публичные функции:

```
1 Geodesic_line(Point vertex, Point begin, Point end, double theta = 175)
```

Параметры:

Point vertex - угол бумаги в изначальном состоянии

Point begin - точка, в которую переходит угол бумаги после изгиба

Point end - точка, на границе изогнутого и плоского регионов
double theta - угол, достаточный для того, чтобы считать бумагу гладкой

Конструктор геодезической кривой.

```
1 std::vector<Point> get_geodesic_curve() const
```

Функция возвращает геодезическую кривую.

(4) Класс Paper_Mesh:

Класс, реализующий представление бумаги в виде 3d mesh.

Содержит следующие публичные функции:

```
1 Paper_Mesh(const std::vector<Geodesic_line>& curves, const std::vector<Point>& corners)
```

Параметры:

curves вектор геодезических кривых

corners вектор соответствующих углов бумаги

```
1 void get_file(std::string file_name)
```

Получает на вход file_name строку с названием, создает файл типа .obj с 3d mesh представлением бумаги.

4 Использование библиотеки:

Для создания 3d mesh файла необходимо выполнить следующие шаги.

(1) Создание необходимых для задания кривых точек:

В качестве входных параметров необходимо задать искривленные регионы для каждой вершины прямоугольного листа бумаги.

Необходимы точка вершины угла в плоском состоянии бумаги, точка в которую перейдет угол при изгибе бумаги и точка на границе изогнутого и плоского регионов.

(2) Создание геодезических кривых:

Построение по соответствующим точкам для каждого угла бумаги геодезические кривые.

Сбор вектора геодезических кривых и вектора углов бумаги на плоскости.

(3) Создание 3d mesh:

Из векторов, полученных в предыдущем пункте конструируем Paper Mesh.

(3) Интерпритация результатов:

Из Paper Mesh получаем obj файл с 3d mesh. Полученный файл можно открыть в стороннем редакторе. Например <https://3dviewer.net/>.

5 Примеры использования

В директории `crumpled_paper/use_examples` лежат файлы, с примерами работы алгоритма с разными параметрами. При выполнении сборки проекта в папке `$your_path/source` появляются исполняемые файлы. Запустив их, можно получить директорию `results` с файлами.

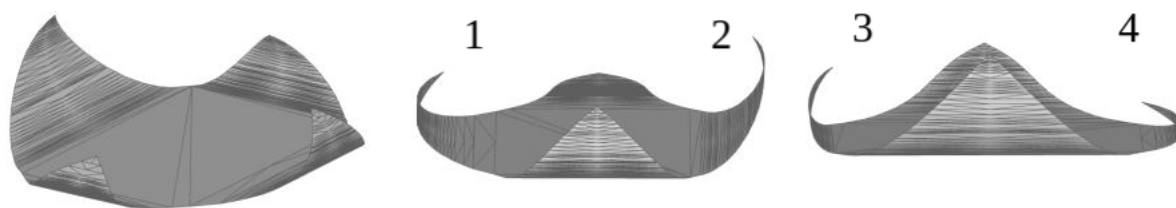
Первый пример использования:

different_parameter_example - демонстрирует влияние таких параметров, как высота угла и длина искривленного региона при фиксированной проекции.

2 - угол находится выше, длина искривленного региона больше, чем у 1

3 - угол находится на той же высоте, длина искривленного региона меньше, чем у 1

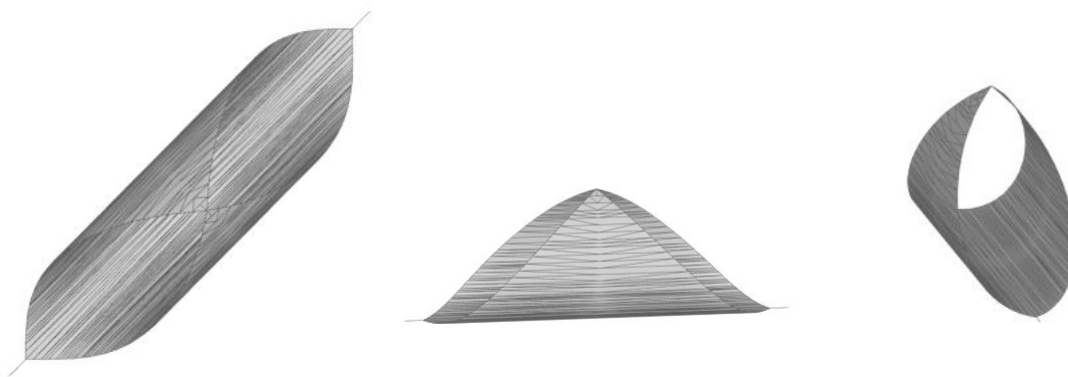
3 - угол находится ниже, длина искривленного региона меньше, чем у 1



Второй пример использования:

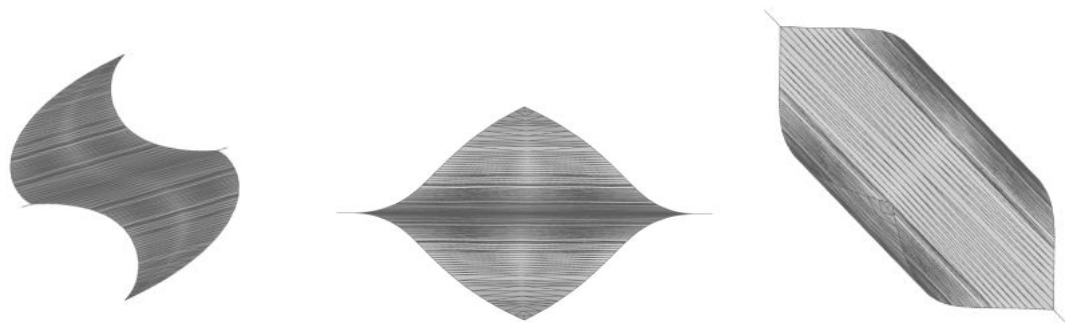
Объект *ring_like* - сомкнутые противоположные углы

Разные ракурсы:



Третий пример использования:

negative_values - случай сгибания угла ниже плоскости бумаги.



6 Ссылки

[1] Camille Schreck. Interactive deformation of virtual paper. Modeling and Simulation. Université Grenoble Alpes, 2016

[2] D. Cohen-Or and P. Slavík. Geodesic-Controlled Developable Paper Bending. Surfaces for Modeling. EUROGRAPHICS 2007

[3] ZHANG Xing-wang, WANG Guo-jin. A new algorithm for designing developable Bézier surfaces. SCIENCE A 2006

[4] Библиотека libgl: <https://libigl.github.io/>

[5] Библиотека Eigen: https://eigen.tuxfamily.org/index.php?title=Main_Page