МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ Н. Э. БАУМАНА
Факультет информатики и систем управления
Кафедра теоретической информатики и компьютерных технологий

Лабораторная работа №4
по курсу «Моделирование»

«Построение разрезов поверхностей»

Выполнил:
студент ИУ9-111
Выборнов А. И.

Руководитель:
Домрачева А. Б.

Москва 2015

1. Постановка задачи

Одной из базовых задач анализа триангуляционных поверхностей является построение разрезов — вертикальных (профилей) и горизонтальных (изолиний).

Изолиниями уровня h называют геометрическое место точек на поверхности, имеющих высоту h и имеющих в любой своей окрестности другие точки с меньшей высотой: $I_h = \{(x,y)|z(x,y) = h, \forall\epsilon > 0\colon \exists(x',y')\colon |(x',y'),(x,y)| < \epsilon, z(x',y') < h\}$.

Изоконтурами между уровнями h_1 и h_2 называют замыкание геометрического места точек на поверхности, имеющих высоту $h \in [h_1, h_2)$, т.е. множество точек $I_h = \overline{\{(x,y)|h_1 <= z(x,y) < h_2\}}$.

В задаче построения изоконтуров требуется построить множество непересекающихся регионов, каждый из которых представляет область, высоты точек внутри которой лежат в определенном диапазоне. Обычно задаётся система диапазонов с помощью начального значения самого первого диапазона, конечного значения последнего диапазона и шага построения диапазонов.

По трёхмерной модели поверхности нужно построить множество изоконтуров, лежащих в диапазоне $[h_1, h_2)$ с шагом Δh.

2. Теоретическая часть

2.1. Построение изолиний

Для построения изолиний высотой h используется следующий алгоритм:

1. Помечаем каждый треугольник триангуляции, по которому проходят изолинии (т.е. выполняется условие $min(z_1, z_2, z_3) < h << max(z_1, z_2, z_3)$, где z_i — высоты трех его вершин), флагом $C_i := 1$, а все остальные треугольники – $C_i := 0$. Если обнаружен хотя бы один треугольник, у которого хотя бы одно ребро лежит в плоскости изолинии, то h уменьшается на некоторое малое Δh и алгоритм повторяется заново.

2. Для каждого треугольника с $C_i = 1$ выполняем отслеживание очередной изолинии в обе стороны от данного треугольника, пока один конец не выйдет на другой или на границу триангуляции. Каждый пройденный при отслеживании треугольник помечается $C_i := 0$. Конец алгоритма.

Трудоемкость такого алгоритма, очевидно, является линейной относительно размера триангуляции.

2.2. Построение изоконтуров

Для построения множества изоконтуров, лежащих в диапазоне $[h_1, h_M)$ с шагом Δh используется следующий алгоритм:

1. Пусть заданы уровни $h_1, ..., h_M$, Обнуляем множества ломаных, входящих в изоконтуры: $C_i = \varnothing, i = \overline{0, M}$.

2. Для каждого уровня h_i строим изолинии. Каждую замкнутую изолинию добавляем во множество C_i .

3. Определяем все кусочки границы триангуляции между точками выхода изолиний на границу. Формируем граф, в котором в качестве узлов выступают точки выхода на границу, а в качестве рёбер —кусочки границы между этими точками и рассчитанные изолинии. Каждая изолиния должна войти в граф дважды в виде одинаковых ориентированных рёбер, но направленных в разные стороны. Для рёбер —кусочков границы — устанавливаем такую ориентацию, чтобы внутренности триангуляции находились справа по ходу движения. В результате в каждом узле графа должны сходиться четыре ребра.

4. По полученному графу строим контуры. Начинаем движение с любой вершины графа и двигаемся вперед в соответствии с ориентацией рёбер до тех пор, пока не вернемся в начальную вершину. Повторный проход по одному и тому же ребру запрещен, для чего делаются специальные пометки на рёбрах. При попадании в узел графа из граничной цепочки далее надо двигаться по ребру, соответствующему изолинии, иначе — по граничному ребру. Обратим внимание, что каждая изолиния войдет в два контура, соответствующих разным диапазонам высот.

5. Для каждого полученного на предыдущем шаге контура определяем, какому диапазону высот он соответствует. Для этого нужно взять и проверить любое ребро триангуляции, входящее в составе граничной цепочки, использованной в каждом контуре. На основании этого помещаем цепочку в соответствующее множество C_i. Конец алгоритма.

Трудоемкость данного алгоритма линейно зависит от размера триангуляции и количества изолиний.

3. Реализация

В рамках лабораторной работы была написана программа на языке python, Описание всех этапов, которые я выслал Домрачевой

4. Тестирование

один результат и визуализация всех этапов.

Результаты при разных параметрах для разных моделей

5. Выводы