**УТВЕРЖДЕНО**

**НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА**

**Reconstructor3D**

**Отчёт**

Л**ИСТОВ** 11

**2018**

Оглавление

[1. Введение 3](#_Toc515329196)

[1.1. Описание предметной области 3](#_Toc515329197)

[1.2. Формальная постановка 3](#_Toc515329198)

[1.2.1 Математическая модель 4](#_Toc515329199)

[2. Алгоритм решения 5](#_Toc515329200)

[2.1.1 Общий подход 5](#_Toc515329201)

[2.1.2 Линейная интерполяция 6](#_Toc515329202)

[2.1.3 Трёхстороння интерполяция 6](#_Toc515329203)

[2.1.4 Итоговая интерполяция 7](#_Toc515329204)

[2.2 Аппроксимация 7](#_Toc515329205)

[2.3. Критерий качества 7](#_Toc515329206)

[3. Эксперимент 8](#_Toc515329207)

[3.1.2 Структура тестового базиса 9](#_Toc515329208)

[3.1.3 Формат выходных данных 10](#_Toc515329209)

[3.2 Критерий успеха эксперимента 10](#_Toc515329210)

[3.3 Результаты 11](#_Toc515329211)

[4. Заключение 11](#_Toc515329212)

# 1. Введение

## 1.1. Описание предметной области

Современные достижения в области измерительной технологии, например, такие как лазерное сканирование, предоставляют возможность цифрового описания исследуемых объектов с достаточно высокой степенью точности. Это обстоятельство обусловило значительное повышение спроса на создание трехмерных моделей, сгенерированных по данным, полученным в результате измерений, в таких областях как медицина, геология, археология, компьютерная графика и автоматизированное проектирование.

В случае, когда вместо специализированной измерительной техники для получения информации о поверхности объекта используют программный комплекс извлечения точек по серии фото, важность минимизации погрешности при восстановлении трехмерной модели объекта возрастает в разы, из-за уменьшения объема информации, по которой производится реконструкция.

Существуют два основных подхода к реконструкции поверхности на основе облака точек, представляющего модель объекта, - интерполяция и аппроксимация. Интерполяционный тип подгонки поверхности использует каждую точку исходных данных в реконструируемой поверхности, тогда как при аппроксимации поверхности отыскивается наилучшая подгонка гладкой поверхности к оцифрованным данным с минимальным отклонением.

Таким образом, основная задача реконструкции поверхности формулируется следующим образом - необходимо найти оптимальную подгонку для неизвестной поверхности так, чтобы минимизировать погрешность измерения, возникающую из-за ограничения по точности в измерительных приборах, или из-за недостаточного качества поверхности физической модели.

## 1.2. Формальная постановка

С помощью макросъемки с малой глубиной резкости получена серия изображений поверхности. Каждое изображение снято на разной высоте от поверхности. Полученная серия изображений с информацией о высоте съемки анализируется и формируется множество геометрических точек, принадлежащих 3D модели поверхности. Данное множество точек отвечает требованию равномерности.

Кроме множества точек, описанных в файле формата .obj, на вход алгоритму обязательно подается файл формата .png с текстурой реконструируемой поверхности. Так же опционально третьим параметром подается количество точек решения.

Задачей НИР является создание 3D модели поверхности, натянутой на множество точек, полученных по серии изображений.

В качестве решения алгоритм должен генерировать файл формата .obj, содержащий точки реконструированной поверхности, и файл .mtl, описывающий материал модели и использующий текстуру из файла .png.

### 1.2.1 Математическая модель

*Математическая модель интерполятора*

Исходные данные:

а, b ∈ ℕ,

М ⊂ – множество исходных точек такое, что ∃ M’ ⊇ M: M’ – регулярная прямоугольная сетка размера а\*b точек с одинаковым расстоянием между узлами.

∀ : - (x,y)-координаты разных точек не могут совпадать, потому что иначе по этим точкам поверхность построить невозможно.

Структура решения:

F (x, y) – функция интерполяции, (x, y) ∈ D

D = {(x, y)| Xmin ⩽ x ⩽ Xmax, Ymin ⩽ y ⩽ Ymax}

Xmin = min{x| (x, y, z) ∈ M}

Xmax = max{ x| (x, y, z) ∈ M }

Ymin = min{y| (x, y, z) ∈ M}

Ymax = max{ y| (x, y, z) ∈ M }

F(x, y) – дифференцируема на D

F(x, y) = z, ∀(x, y, z) ∈ M

*Математическая модель фасеточной аппроксимации*

Исходные данные:

n – количество точек искомого решения

9 ⩽ n ⩽ 8млн

D – область определения

F(x, y) – функция интерполяции, (x, y) ∈ D

Структура решения:

S ⊂ { (x, y, F(x,y)) ∈ }

|S| = n

A(x, y) – фасеточная аппроксимация, построенная по (x, y, z) ∈ S

A(x, y) = z, ∀ (x, y, z) ∈ S

#### 1.2.2 Оценка алгоритма

Оценка алгоритма по одной эталонной модели:

Ə = {(x,y,z)} – множество точек эталона

– критерий

Оценка алгоритма по тестовому базису:

m – количество задач в тестовом базисе.

– множество эталонов тестового базиса.

– множество построенных моделей.

– критерий.

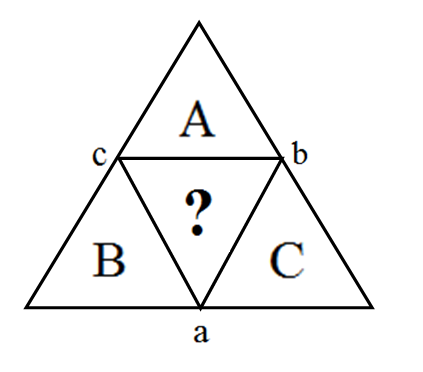
2. Алгоритм решения2.1 Интерполяция

### 2.1.1 Общий подход

Интерполяция строится по триангуляции Делоне исходного множества точек. Каждому треугольнику ставится в соответствие своя собственная функция интерполяции, которые по построению обладают свойствами непрерывности и дифференцируемости на всей области определения кроме конечного множества точек, включая рёбра треугольников, где проходит граница между разными функциями.

Такой подход позволяет использовать для вычисления значения в конкретной точке только небольшую часть модели – один треугольник, что существенно улучшает производительность.

### 2.1.2 Линейная интерполяция

Простейший способ задать интерполяцию в треугольнике – линейно. Если вершины треугольника – точки A(), B(), C(), уравнение плоскости записывается как

Из этого матричного уравнения можно выразить z, получив уравнение плоскости вида z = ax + by + c. Далее записывается как L(x,y) = ax + by + c.

### 2.1.3 Трёхстороння интерполяция

Представим, что для всех соседей нашего треугольника уже заданы интерполяции , ,.

Определим также функции , , – расстояния точки от соответствующих рёбер нашего треугольника. Если мы позволим этому расстоянию быть положительным по одну сторону ребра и отрицательным – по другую, то эти функции запишутся линейно. Зададим искомую интерполяцию T(x,y) таким образом:

Очевидно, что эта функция будет обладать производной во всех точках, кроме вершин треугольника; что на границах с другими треугольниками значение T(x,y) и производные по всем направлениям будут равны значению и производным соответствующего интерполятора; что при равенстве , , функция T также равна им.

### 2.1.4 Итоговая интерполяция

Для построения гладкой поверхности можно было бы задать треугольникам линейную и трёхстороннюю интерполяцию в шахматном порядке, но это возможно только если все вершины граничат с чётным числом треугольников. Вместо этого для каждого треугольника строятся обе интерполяции, а потом берётся их среднее арифметическое. Очевидно, взятие среднего сохраняет дифференцируемость и непрерывность этих функций:

F(x,y) = (T(x,y) + L(x,y))/2

## 2.2 Аппроксимация

Первоначально планировалось написать стратегию разброса точек на плоскости, и для построения фасеточной аппроксимации взять разбросанные этой стратегией точки, вычислить значение интерполятора в каждой точке, и построить поверхность триангуляцией этих точек, но такой подход оказался неэффективным: самый быстрый алгоритм триангуляции Делоне требует O(n\*log(n)) времени (n – количество точек в итоговой модели). Вместо этого был написан линейный алгоритм построения правильной треугольной сетки по интерполятору. Так как все треугольники в этой сетке близки к равносторонним, то такую сетку тоже можно считать триангуляцией Делоне.

## 2.3. Критерий качества

Оценка реконструируемой модели проводится с помощью тестов с эталонной поверхностью. Эталон представляет собой набор трёхмерных точек, находящихся на искомой поверхности, которые могут частично совпадать с набором точек в исходном .obj файле. В каждой из этих точек измеряется разность высот между эталоном и смоделированной поверхностью, затем находится среднее арифметическое абсолютных значений этих разностей, нормированное по величине разности высот в эталоне.

Ə = {(x,y,z)} – множество эталонных точек. Эталонных точек больше, чем точек во входных данных.

F(x,y) – модель поверхности; высота смоделированной поверхности в точке (x,y). Точки из эталона не приближаются к границам области определения F(x,y) (прямоугольник) ближе чем на 0,5% от ширины или длины области определения соответственно.

– критерий качества одной модели

Оценка качества работы алгоритма производится с помощью тестового базиса: набора входных данных и эталонов к ним. Итоговый критерий – среднее арифметическое значений критерия K для каждой отдельной поверхности.

m– количество задач в тестовом базисе.

– множество эталонов тестового базиса.

– множество построенных моделей.

– итоговый критерий.

Значения критериев K и L всегда будут не меньше нуля. Значение больше единицы будут означать, что построенная поверхность(-ти) хуже полностью случайной(-ых). Под случайной поверхностью понимается функция высоты, возвращающая случайное значение в интервале высот эталона. Значение ровно ноль – что модель(-и)построена(-ы) идеально, но алгоритма, который гарантированно строит поверхность с значением критерия ноль, не существует, так как на вход подаётся неполная информация о поверхности. Чем ближе критерии к нулю, тем ближе построенные модели к искомым поверхностям.

Другие распространённые критерии – минимаксный, квадратичный – рассматривались, но не были приняты в качестве определяющих, так как они дают слишком большой вес большим отклонениям аппроксимации от эталона на небольших участках, а такие отклонения практически неизбежны в рамках данной задачи.

3. Эксперимент3.1 Входные и выходные данные3.1.1 Формат входных данных

На вход алгоритму должны подаваться два файла и параметр (опционально):

1) Один файл формата .obj согласно стандарту Wavefront[[1]](#footnote-1). Этот файл должен содержать только координаты точек (геометрических вершин), по которым будет строиться аппроксимация поверхности: предполагается, что искомая поверхность проходит через все эти точки. Пара первых двух координат (x и y)–координаты точки на плоскости, а третья (z)–высота поверхности над этой точкой. Пара координат (x, y) не должна повторяться с разными координатами z, потому что поверхность не может иметь разную высоту в одной и той же точке.

Пример файла:

gbody

v -10.500000 -37.500000 0.000000

v -25.500000 -37.500000 1.000000

v -23.500000 37.500000 0.000000

v -39.500000 37.500000 1.000000

v -31.000000 -35.000000 1.000000

v -32.000000 -15.000000 1.000000

v -35.000000 -10.000000 1.000000

v -39.000000 -10.000000 2.000000

v -40.000000 10.000000 1.000000

v -41.000000 10.000000 2.000000

2) Один файл формата .png. Изображение, содержащееся в этом файле, будет использовано как текстура для наложения на полученную аппроксимацию поверхности. На геометрию построенной поверхности это изображение никак не влияет.

3) Количество точек, которые должны содержаться в итоговой фасеточной аппроксимации – целое число. Этот параметр опционален – если его не указать, количество точек в аппроксимации будет равно количеству точек в исходном .obj файле.

### 3.1.2 Структура тестового базиса

Один тест представляет собой три файла: test.obj и test.png – входные файлы для алгоритма, и test.dat – файл эталонной поверхности, с которым будет сравниваться результат работы алгоритма.

Поверхности, используемые в тестовом базисе, генерируются как комбинация примитивов: эллипсоид, парабола, наклон и ступень, использующиеся как фигуры движения и вращения.

Тестовый базис содержит простые и сложные тесты. Простые тесты содержат каждый по одному из восьми примитивов. Сложные тесты в среднем содержат 200 случайных примитивов, объединённые как сумма высот.

Тестовый базис состоит из 108 папок: 100 папок имеют названия от complex\_001 до complex\_100 и содержат каждая по одному сложному тесту, и 8 папок с названиями simple\_conus, simple\_sphere, simple\_groove, simple\_cylinder, simple\_step, simple\_slope, simple\_parabola и simple\_parabolic\_groove, содержащие соответствующие простые тесты.

В каждом .obj-файле содержится 512 точек и в каждом .dat-файле содержится 40000 точек.

### 3.1.3 Формат выходных данных

В качестве решения алгоритм генерирует:

1) Один файл формата .obj согласно стандарту Wavefront. Этот файл будет содержать связную фасеточную аппроксимацию искомой поверхности. Число вершин поверхности будет совпадать с входным параметром, если он был введён, и с количеством вершин входного файла формата .obj - в противном случае.

2) Один файл формата .mtl (material library), также по стандарту Wavefront. Этот файл необходим для визуального отображения построенной поверхности: многие программы для просмотра и редактирования 3D-моделей, такие как 3D Builder и Blender, автоматически используют соответствующий .mtl файл при просмотре .obj модели.

## 3.2 Критерий успеха эксперимента

Для каждого из тестов будет рассчитано и записано в .csv-файл значение критерия для каждого из трёх интерполяторов:

* LinearInterpolator: поверхность строится как двухмерная триангуляция Делоне, натянутая на исходные точки. Не соответствует критерию гладкости.
* ByClosestInterpolator: находит высоту поверхности как высоту ближайшей исходной точки. Не соответствует ни требованию гладкости, ни требованию непрерывности.
* SmoothInterpolator: интерполятор, разработанный нашей командой. Соответствует критерию гладкости и непрерывности.

Также в конце рассчитывается среднее значение критерия для каждого из интерполяторов. Тест считается пройденным успешно, если среднее значение для SmoothInterpolator окажется лучше (меньше) чем значение для ByClosestInterpolator.

## 3.3 Результаты

Среднее значение критерия для гладкого интерполятора на тестовом базисе оказалось значительно лучше, чем для интерполятора по ближайшей точке. На машине, подходящей по характеристикам, программа укладывается в отведённое время.

# 4. Заключение

Нашей командой был разработан программный продукт, отвечающий всем заявленным требованиям. Поставленные задачи были выполнены, а цели – достигнуты.

1. [http://www.cs.utah.edu/~boulos/cs3505/obj\_spec.pdf](https://vk.com/away.php?to=http%3A%2F%2Fwww.cs.utah.edu%2F~boulos%2Fcs3505%2Fobj_spec.pdf&cc_key=#_blank) [↑](#footnote-ref-1)