

УТВЕРЖДАЮ

Должность

_____ ФИО

“ ” _____ 2018 г.

Пояснительная записка № 003

Этап 3. Математическая модель и алгоритм решения

НИР

**Реконструкция 3D модели поверхности микроскопического объекта по
серии изображений**

(«Get3DModel»)

Н.Новгород

2018

Реферат

Пояснительная записка 003, страниц 17.

КЛЮЧЕВЫЕ СЛОВА: математическая модель, структура решения, ограничения на решение, эталон, оценка решения, алгоритм решения, схема оценки работы алгоритма, Get3DModel.

В пояснительной записке в рамках проекта НИР «Get3DModel» представлены:

- Содержательная постановка задачи;
- Математическая модель;
- Алгоритмы решения;
- Схема оценки работы алгоритма.

Оглавление

Термины и определения	4
1 Содержательное описание	5
2 Математическая модель	5
3 Пример	7
4 Алгоритмы решения	9
4.1 Алгоритм приложения «Get3DModel»	9
4.2 Алгоритм оценки градиента	9
4.3 Алгоритм валидации точек	10
4.4 Алгоритм динамического подбора ядра	10
4.5 Алгоритм вычисления высоты	11
4.6 Алгоритм работы программы	11
5 Схема оценки работы алгоритма	11
5.1 Суммарная ошибка	12
5.2 Максимальная ошибка	12
5.3 Средняя ошибка	13
5.4 Глубина уровня	13
5.5 Заполняемость	14
5.6 Равномерность распределения точек	14
5.7 Время выполнения	15
6 Список используемых источников	17

Термины и определения

- *Get3DModel* – разрабатываемый в рамках текущей НИР ([1]) программный модуль реконструкции 3D модели поверхности микроскопического объекта по серии изображений
- *3D изображение* – изображение, полученное путем моделирования объемных объектов в трехмерном пространстве.
- *Градиент* - вектор, своим направлением указывающий направление наибольшего возрастания некоторой величины, значение которой меняется от одной точки пространства к другой (скалярного поля), а по величине (модулю) равный скорости роста этой величины в этом направлении.

1 Содержательное описание

СО: Есть серия изображений поверхности объекта с малой глубиной резкости. Изображения получены микросъемкой одного и того же объекта на разной высоте. Также имеется информация об оптической системе (фокусное расстояние, наблюдаемая ширина в фокусе, коэффициент для вычисления абсолютной высоты фокуса).

Допущение: Рассматриваются только непрозрачные объекты, а их изображения имеют одинаковый размер.

Задача: Необходимо определить координаты точек принадлежащих поверхности восстанавливаемых объектов, равномерно распределенных по исследуемой области, а также восстановить изображение объекта с высокой глубиной резкости.

2 Математическая модель

Исходные параметры:

n - количество изображений поверхности объекта

m, s – размер полученных изображений: ($m*s$ пикселей)

r_i – относительная высота оптической системы, на которой получено i -ое изображение, $i = \overline{1, n}$

F – фокусное расстояние

W – наблюдаемая ширина в фокусе

coef- коэффициент для вычисления абсолютной высоты фокуса

Z^* - матрица z^* координат эталонной модели.

Z^*	1	2	...	m
1				
2				
...				
s				

z^* - высота точки (z-координата) эталонной модели, соответствующая пикселям i, j ; $i = \overline{1, m}, j = \overline{1, s}$

$$z^*_{ij} \geq 0$$

R – параметр равномерности, выраженный в процентах

Структура решения:

Z	1	2	...	m
1				
2				
...				
s				

Решение представляет собой матрицу Z размером $m \times s$. Элемент матрицы:

$$z_{ij} \begin{cases} \geq 0, \text{ если алгоритм нашел высоту (z – координату) точки 3D изображения,} \\ \text{соответствующую пикселям } i, j; i = \overline{1, m}, j = \overline{1, s} \\ = -1, \text{ если алгоритм не нашел высоту (z – координату) точки 3D изображения.} \end{cases}$$

Ограничения на решение:

Точки восстанавливаемых объектов, высоты которых найдены алгоритмом, должны быть равномерно распределены по исследуемой области.

Для этого необходимо вычислить вектор:

$$V = (v_1, v_2, \dots, v_l), \text{ где}$$

l – количество уровней равномерного распределения.

Координата вектора:

$$v_i = \frac{fact_i}{real_i}, i = \overline{1, l}, \text{ где}$$

$fact_i$ – количество областей i -ого уровня, содержащих хотя бы одну точку с найденной высотой.

$real_i$ – количество областей i -ого уровня, на которые делим изображение.

Поэтому исходя из параметра равномерности необходимо выполнение следующего условия:

$$\frac{\sum_{i=1}^l v_i}{l} * 100 \geq R$$

Оценка решения:

Рассмотрим вектора:

$$H^* = (z_{11}^*, z_{12}^*, \dots, z_{1m}^*, \dots, z_{s1}^*, \dots, z_{sm}^*)$$

$$H = (z_{11}, z_{12}, \dots, z_{1m}, \dots, z_{s1}, \dots, z_{sm})$$

Пусть $k = m * s$, тогда с точностью до обозначения:

$$H^* = (h_1^*, h_2^*, \dots, h_k^*)$$

$$H = (h_1, h_2, \dots, h_k)$$

Пусть $I = \{i | h_i \geq 0, i = \overline{1, k}\}$ - множество номеров компонент вектора H , при которых алгоритм нашел высоты точек 3D изображения ($h_i \neq -1$). Оценка решения происходит на основе сравнения найденных высот 3D изображения с высотами эталонной модели.

Необходимо минимизировать модуль среднего отклонения решения от эталонной модели (при расчете не учитываются точки, чьи высоты не найдены):

$$\frac{\sum_{i \in I} |h_i^* - h_i|}{|I|} \rightarrow \min$$

3 Пример

Есть эталонная 3D модель изображения 4*4 пикселя:

Параметр равномерности $R=50$.

Эталонная модель:

Z^*	1	2	3	4
1	3	2	1	0
2	3	2	1	0
3	0	1	1	0
4	0	0	2	3

Решение:

Z	1	2	3	4
1	-1	-1	2	-1
2	-1	-1	2	0
3	0	1	2	0
4	-1	0	3	3

Пусть Z – решение задачи. Проверим, удовлетворяет ли оно ограничению равномерности.

Будем делить область сначала на 4, затем на 16 подобластей.

Таким образом, будет 2 уровня.

На 1-ом уровне только три области содержат точки, у которых найдены высоты.

Z	3	4
1	2	-1
2	2	0

Z	1	2
3	0	1
4	-1	0

Z	3	4
3	2	0
4	3	3

Рис.1 Области, содержащие хотя бы одну точку с найденной высотой

На 2-ом уровне только 6 из 16 областей не имеют достоверных точек. Это области:

Z	1
1	-1

Z	2
1	-1

Z	1
2	-1

Z	2
2	-1

Z	4
1	-1

Z	1
4	-1

Рис.2 Области, не содержащие ни одной точки с найденной высотой

Вычисляем координаты вектора $V = (v_1, v_2)$:

$$v_1 = \frac{3}{4}; \quad v_2 = \frac{10}{16}$$

Проверяем ограничение: $\frac{\frac{3}{4} + \frac{10}{16}}{2} * 100 \approx 68,75 \geq R$. Таким образом, ограничение выполняется.

Вектора имеют вид:

$$H^* = (h_1^*, h_2^*, \dots, h_k^*) = (3, 2, 1, 0, 3, 2, 1, 0, 0, 1, 1, 0, 0, 0, 2, 3)$$

$$H = (h_1, h_2, \dots, h_k) = (-1, -1, 2, -1, -1, -1, 2, 0, 0, 1, 2, 0, -1, 0, 3, 3)$$

Множество $I = \{3,7,8,9,10,11,12,14,15,16\}$, $|I| = 10$

А среднее отклонение решения от эталонной модели составляет:

$$\frac{\sum_{i \in I} |h_i^* - h_i|}{|I|} = \frac{4}{10} = 0,4$$

4 Алгоритмы решения

4.1 Алгоритм приложения «Get3DModel»

Разработанный алгоритм является итерационным, где количество итераций равно количеству входных снимков, по которым нужно построить карту глубин.

$\|C\|_{M \times N}$ - оптимальное значение градиента для каждого пикселя на конкретной итерации;

$\|X\|_{M \times N}$ - значение пикселя в изображении, переведенного в монохром;

$\|H\|_{M \times N}$ - матрица высот. Изначально все элементы в ней равны -1;

4.2 Алгоритм оценки градиента

Шаг 1: Принимаем входные данные: снимок, сделанный на определенной высоте.

Шаг 2: Преобразуем снимок в монохромное изображение по формуле:

$$X = 0.3 * RED + 0.59 * GREEN + 0.11 * BLUE$$

Шаг 3:

Вычисляем «градиент» каждого пикселя, используя определенное ядро свертки G (список ядер в приложении1):

$$D_{ij} = \begin{pmatrix} X_{i-1\ j-1} & X_{i-1\ j} & X_{i-1\ j+1} \\ X_{i\ j-1} & X_{ij} & X_{i\ j+1} \\ X_{i+1\ j-1} & X_{i+1\ j} & X_{i+1\ j+1} \end{pmatrix} * G$$

* — операция свертки

Если матрица «окружения» не может быть полностью определена для какого-либо пикселя в силу того, что это граничный пиксель изображения

(пример: X_{00}), тогда заполняем нулями те позиции в матрице, которые не могут быть определены.

Пример граничного пикселя и его матрицы «окружения»:

$$X_{00} = \begin{matrix} & 0 & 0 & 0 \\ 0 & X_{00} & X_{01} \\ 0 & X_{10} & X_{11} \end{matrix}$$

Шаг 4:

Нормируем «градиент» каждого пикселя, то есть значение, полученное на шаге 3, делим на t , где $t = \sum_{v \in V} v * 255$, где

$$V = \{g_{ij} | g_{ij} \in G, g_{ij} > 0, i = \overline{1, n}, j = \overline{1, n}\}$$

G – ядро свертки размерности n .

4.3 Алгоритм валидации точек

В каждом ядре есть граничная Δ , $0 \leq \Delta \leq 1$ – процент порогового значения (пока равна 0), которая получена в результате работы дополнительной программы, где был найден такой процент порогового значения Δ , что сохраняя ограничение на равномерность средняя ошибка стремится к минимуму.

Порог th вычисляется так:

$$th = \min(grad) + (\max(grad) - \min(grad)) * \Delta$$

Все градиенты, полученные алгоритмом поиска градиента на зафиксированном ядре, которые меньше полученного порога th , отбрасываются.

4.4 Алгоритм динамического подбора ядра

Выбор стоит между ядрами с размерностями: 3×3 ; 5×5 ; и 7×7 .

Для каждого изображения в точке (x, y) вычисляются три значения градиента с помощью трех ядер разной размерности.

Обработав все изображения, для точки (x, y) мы будем иметь три списка градиентов. Каждый список соответствует одному из ядер: 3×3 , 5×5 , 7×7 .

Чтобы определить оптимальное ядро для точки (x, y) необходимо:

1. Отсортировать каждый список по возрастанию.
2. Отбросить в каждом списке первые 2/3 значений.
3. Вычислить для каждого списка дисперсию.

Ядром для точки (x, y) будет являться ядро, соответствующее списку с минимальной дисперсией.

Дисперсия вычисляется по формуле: $D = \sum_{i=1}^n x_i^2 * p_i - (\sum_{i=1}^n x_i * p_i)^2$

где x_i – значения элементов списка;

$p_i = 1/n$, где n – количество элементов в списке.

4.5 Алгоритм вычисления высоты

Последовательно находим матрицу градиентов для каждого изображения.

Для подсчета градиента в определенном пикселе используется ядро, найденное алгоритмом динамического подбора ядра.

После расчета градиентов текущего изображения проводим операцию сравнения найденных значений матрицы градиентов с оптимальными значениями матрицы градиентов на данном этапе:

Если $D_{ij} > C_{ij}$, то $C_{ij} = D_{ij}$ и $H_{ij} = h$

4.6 Алгоритм работы программы.

1. Расчет матрицы градиентов на зафиксированном (стартовом) ядре.
2. Получаем достоверные точки алгоритмом отсева «плохих» точек.
3. Динамически подбираем ядра для достоверных точек.
4. Используя найденные ядра алгоритмом динамического подбора вычисляем высоту.

5 Схема оценки работы алгоритма

Для анализа полученных выходных данных была разработана схема их оценки[2]. При условии, что ограничения по времени и по заполнению точек выполняются, оценка производится по следующим параметрам (по убыванию значимости для Заказчика):

5.1 Суммарная ошибка

Ошибка – модуль разности высот соответствующих точек из эталонной и полученной моделей.

Математическое описание параметра:

Пусть $R_{S_t \times N_t}$ и $M_{S_t \times N_t}$ – матрицы t – ого тестового набора, содержащие высоты точек эталонной и полученной моделей соответственно, тогда r_{ij_t} и m_{ij_t} – элементы эталонной и полученной матриц соответственно, где r и m – высоты точек с координатами i, j . ($r, m \in \mathbb{Z}$; $(i, j) \in \{G_t\}$, где G_t – множество точек t -ого тестового набора, для которых $r_{ij_t} > 0$ (высота точки положительна), $t = \overline{1, p}$, где p – количество тестовых наборов, $S_t \times N_t$ – размеры картинки t – ого тестового набора).

Индивидуальная оценка теста:

$$\sum_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|$$

Средняя оценка тестового набора:

$$\frac{\sum_{t=1}^p (\sum_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|)}{p}$$

5.2 Максимальная ошибка

Индивидуальная оценка теста:

$$\max_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|$$

Нормированная оценка:

$$\frac{\max_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|}{\max r_{sn_t} - \min r_{sn_t}},$$

где $s=\overline{1, S}$, $n=\overline{1, N}$, знаменатель дроби - разность между максимальной и минимальной высотами эталона соответствующего t - ого теста.

Максимальная ошибка тестового набора:

$$\frac{\sum_{t=1}^p \frac{\max_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|}{\max r_{sn_t} - \min r_{sn_t}}}{p},$$

где $s=\overline{1, S}$, $n=\overline{1, N}$, и p – количество тестовых наборов.

5.3 Средняя ошибка

Описание параметра:

Равна отношению суммарной ошибки теста к количеству точек, с высотой

$$r_{ij_t} > 0$$

Индивидуальная оценка теста:

$$\frac{\sum_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|}{G_t},$$

где G_t – множество точек t -ого тестового набора, для которых $r_{ij_t} > 0$ (высота точки положительна), r_{ij_t} и m_{ij_t} – элементы эталонной и полученной матриц соответственно, где $ги m$ – высоты точек с координатами i, j .

Нормированная оценка:

$$\frac{\sum_{t=1}^p \frac{\sum_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|}{G_t}}{p}, \text{ где}$$

p – количество тестовых наборов.

5.4 Глубина уровня

Описание параметра:

Равна отношению средней ошибки теста к количеству его (теста) картинок.

Индивидуальная оценка теста:

$$\frac{\frac{\sum_{(i,j) \in G_t} |r_{ij_t} - m_{ij_t}|}{G_t}}{W_t},$$

где числитель дроби – средняя ошибка, W_t – количество картинок t -ого, $t = \overline{1, p}$ теста.

Нормированная глубина уровня:

$$\frac{\sum_{t=1}^p \sum_{(i,j) \in G_t} \frac{|r_{ij_t} - m_{ij_t}|}{\frac{G_t}{W_t}}}{p},$$

p – количество тестовых наборов.

5.5 Заполняемость

Математическое описание параметра:

Пусть D – множество точек полученной модели, для которых соответствующие высоты больше нуля ($m_{ij} > 0$), $D \in \{S \times N\}$.

Тогда заполняемость вычисляется по формуле:

$$\frac{D}{S \cdot N}$$

5.6 Равномерность распределения точек

Математическое описание параметра:

Точки восстанавливаемых объектов, высоты которых найдены алгоритмом, должны быть равномерно распределены по исследуемой области. Для этого необходимо вычислить вектор:

$$V = (v_{1_t}, v_{2_t}, \dots, v_{l_t}), \text{ где}$$

l – количество уровней равномерного распределения;

t – количество тестовых наборов;

Координата вектора:

$$v_{it} = \frac{fact_{it}}{real_{it}}, i = \overline{1, l}, \text{ где}$$

$fact_{it}$ – количество областей i -ого уровня, содержащих хотя бы одну точку с найденной высотой.

$real_{it}$ – количество областей i -ого уровня, на которые делим изображение.

Поэтому исходя из параметра равномерности необходимо выполнение следующего условия:

Индивидуальная оценка теста:

$$\frac{\sum_{i=1}^l v_{it}}{l} * 100 \geq R$$

Среднее распределение для всех тестовых наборов высчитывается по формуле:

$$\frac{\sum_{t=1}^p \frac{\sum_{i=1}^l v_{it}}{l} * 100}{p}$$

5.7 Время выполнения

Высчитывается время выполнения одного теста. Для оценки алгоритма необходима общая оценка времени выполнения всех тестов. В связи с этим, применяется нормировка времени.

Математическое описание параметра:

Пусть t_i – время выполнения i -го теста ($i = \overline{1, p}$, где p – количество тестовых наборов);

n_i – число картинок i -го теста;

$m_i * s_i$ – количество точек картинки i – го теста.

Индивидуальная оценка теста:

$$\frac{t_i}{m_i \cdot s_i \cdot n_i}$$

Средняя оценка тестового набора:

$$\frac{\sum_{i=1}^p \frac{t_i}{m_i \cdot s_i \cdot n_i}}{p}$$

Результаты будут представлены в виде итоговой таблицы, в которой будут отражены сведения по каждому тесту отдельно, а также нормированные сведения по всем тестам.

6 Список используемых источников

1. Техническое задание на научно-исследовательскую работу «Реконструкция 3D модели поверхности микроскопического объекта по серии изображений), Нижний Новгород, 2018.
2. Пояснительная записка № 001. Тестовый базис для тестирования ПО «Get3DModel». НИР «Get3DModel» (Тестовый базис). Н.Новгород, 2018