

Санкт-Петербургский государственный электротехнический университет им.  
В.И. Ульянова (Ленина)

# Разработка платформы для фотограмметрического контроля

Выполнила:

Руководитель:

Дегтярева Алиса Андреевна, гр. 7303  
Заславский Марк Маркович, к.т.н., доцент

Санкт-Петербург, 2021

# Цель и задачи

## Актуальность:

- Строительный контроль включает в себя задачу оценки соблюдения технологии постройки объекта, от точности получаемых данных зависит безопасность людей
- Высокая погрешность при ручном расчете, дополнительные денежные и временные ресурсы при использовании LIDAR

**Цель:** разработка платформы для контроля строительства с использованием технологий фотограмметрии

## Задачи:

1. Изучить существующие способы вычисления объема и линейных размеров трехмерных моделей
2. Разработать алгоритмы, позволяющие вычислить по трехмерной модели объем объекта, линейные размеры объекта, оценить погрешность вычислений
3. Разработать платформу для строительного контроля
4. Провести оценку точности алгоритмов из п.2

# Задача 1. Способы вычисления геометрических характеристик моделей

**Вычисление объема:** тетраэдры через начало координат и полигоны модели

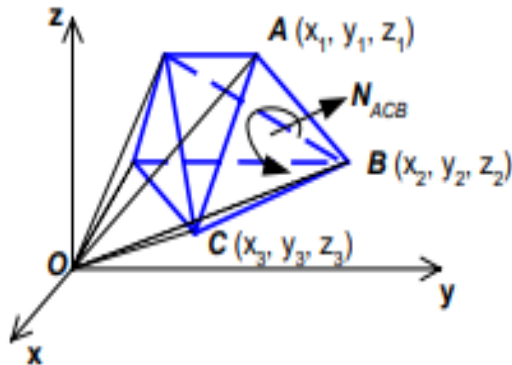


Рисунок 1 – Тетраэдры модели

$$V_i = \frac{1}{6}(-x_{i3}y_{i2}z_{i1} + x_{i2}y_{i3}z_{i1} + x_{i3}y_{i1}z_{i2} - x_{i1}y_{i3}z_{i2} - x_{i2}y_{i1}z_{i3} + x_{i1}y_{i2}z_{i3})$$

$$V_{total} = \sum_i V_i$$

$(x_{i1}y_{i1}z_{i1}), (x_{i2}y_{i2}z_{i2}), (x_{i3}y_{i3}z_{i3})$  – точки полигона в порядке их расположения

**Вычисление линейных размеров:** расстояния между крайними точками в продольном, поперечном и вертикальном направлении

## Задача 2. Разработка алгоритмов. Подготовка облака точек

1. Нахождение плоскостей горизонта
2. Удаление шумов над и под плоскостями горизонта
3. Построение ограничивающих плоскостей
4. Поворот ограничивающих плоскостей
5. Сдвиг ограничивающих плоскостей

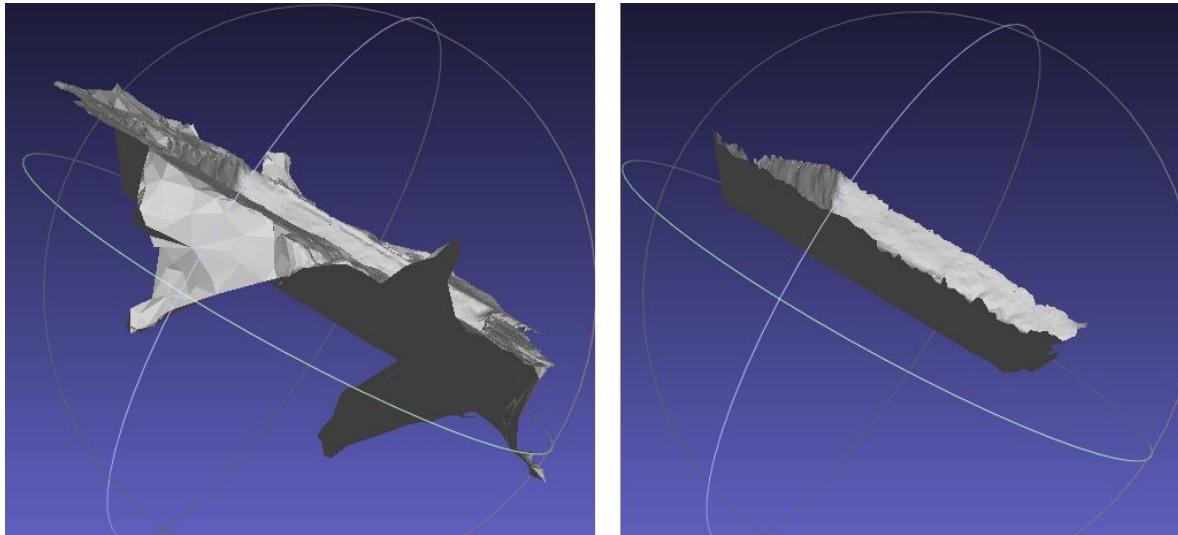


Рисунок 2 – Подготовка трехмерной модели  
1 – исходная модель, 2 – подготовленная модель

# Задача 2. Разработка алгоритмов. Вычисление объемов

1. Через плоскость тегов
2. С использованием триангуляции Делоне:
  - Через выпуклую оболочку
  - Через тетраэдры триангуляции

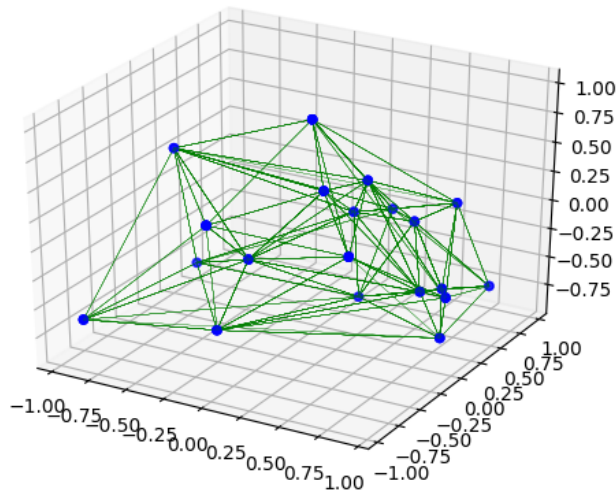


Рисунок 3 – Триангуляция Делоне  
в трехмерном пространстве

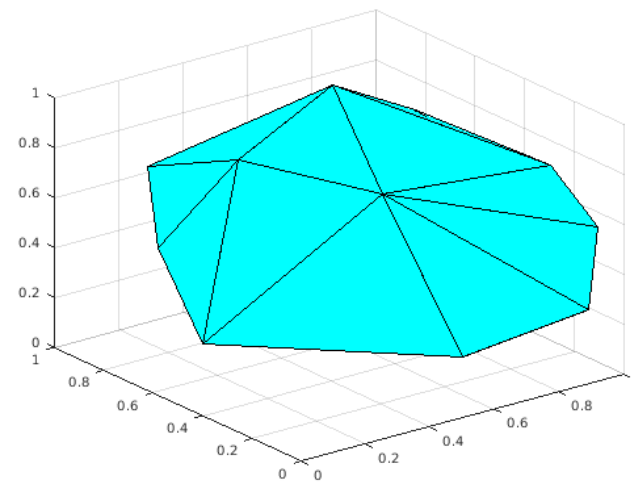


Рисунок 4 – Выпуклая оболочка  
триангуляции Делоне

## Задача 2. Разработка алгоритмов. Вычисление линейных размеров. Погрешность

**Линейные размеры:** размеры внешнего ограничивающего тела

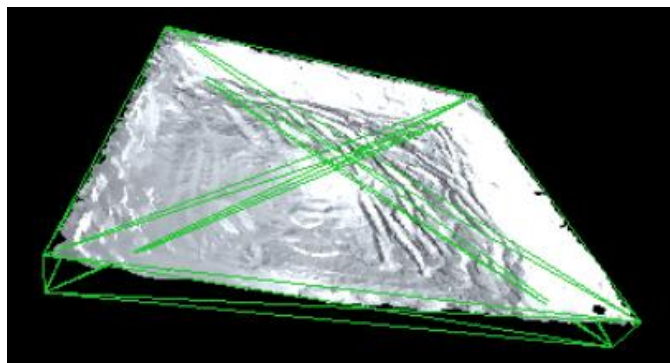


Рисунок 5 – Внешнее ограничивающее тело

**Погрешность:** разница между фактическими и вычисленными данными (расстояния между тегами, высота объекта)

$$\delta = \frac{d - d_{\text{Д}}}{d_{\text{Д}}} * 100\%$$

$d$  – расстояние между парой тегов в модели,  $d_{\text{Д}}$  – фактическое расстояние между парой тегов

## Задача 3. Разработка платформы

- Консольное приложение
- Вычисление объема, линейных размеров, оценка погрешности
- Генерация моделей (Meshroom)/работа с готовыми моделями
- Настройка параметров обрезки модели

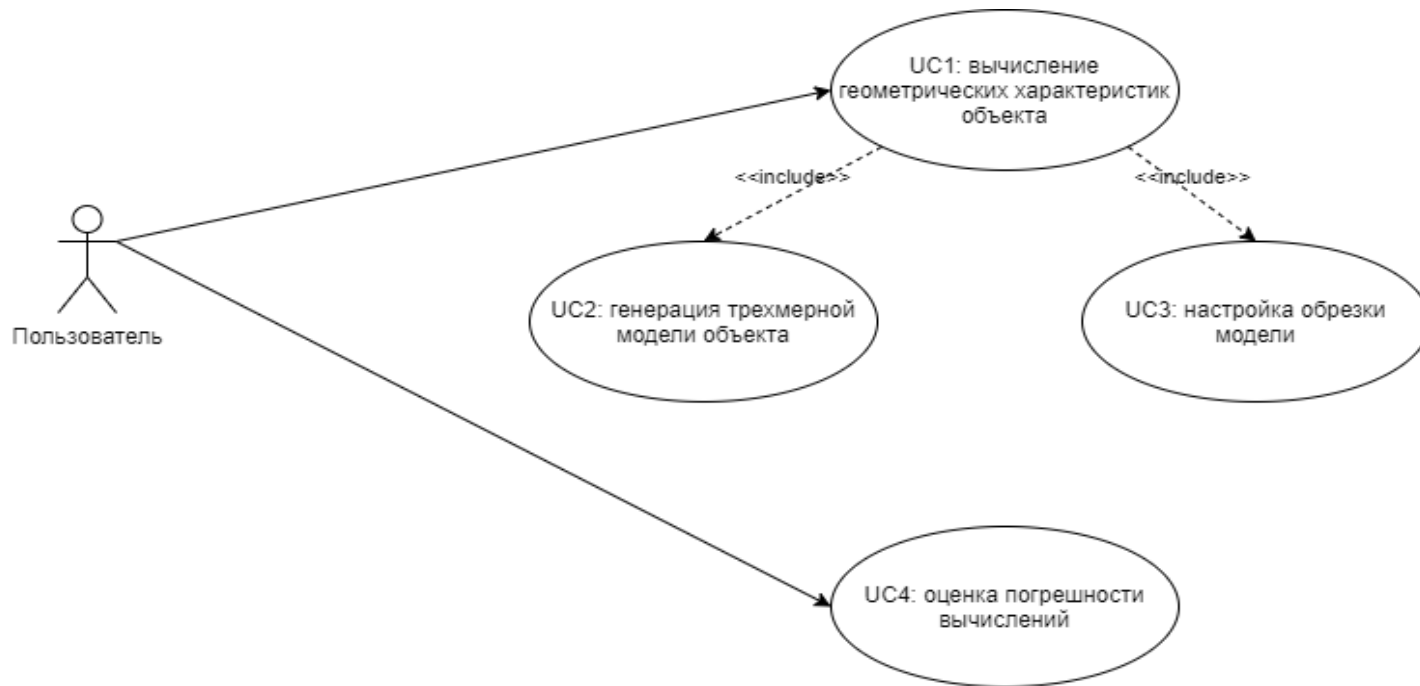


Рисунок 6 – Сценарии использования

## Задача 4. Оценка точности вычисления объема, линейных размеров, погрешности

### **1. Тестирование с построением модели (объем + размеры)**

- Эксперимент 1 (8 фотографий): 7 из 10 запусков
- Эксперимент 2 (6 фотографий): 7 из 10 запусков

### **2. Тестирование без построения модели**

- Эксперимент 3 (объемы + размеры): 8 из 10 запусков
- Эксперимент 4 (погрешность): вычисленная погрешность и фактическая погрешность отличаются на 3,7%



# Заключение

Выводы по задачам:

1. Изучены существующие способы вычисления характеристик моделей: объем – суммарный объем тетраэдров через полигоны и начало координат, линейные размеры – расстояния между крайними точками модели в разных направлениях
2. Разработаны алгоритмы для вычисления объема и линейных размеров модели и оценки погрешности, а также для подготовки модели к данным вычислениям (удаление шумов, поиск положения ограничивающих плоскостей)
3. Разработана платформа для контроля строительства на основе алгоритмов из п.2.
4. Проведен эксперимент с построением модели (7 успешных запусков из 10), эксперимент без построения модели (8 успешных запусков из 10), эксперимент с оценкой погрешности (вычисленная погрешность и фактическая погрешность отличаются на 3,7%.)

Направление дальнейших исследований – разработка платформы для мониторинга процесса строительства, увеличение количества вычисляемых характеристик

# Апробация работы

- «Обзор инструментов фотограмметрии с точки зрения пригодности для решения задач строительного контроля» // Научно-технический семинар кафедры МОЭВМ СПбГЭТУ «ЛЭТИ», 2021
- Репозиторий проекта  
[https://github.com/moevm/bsc\\_degtyareva](https://github.com/moevm/bsc_degtyareva)

Санкт-Петербургский государственный электротехнический университет им.  
В.И. Ульянова (Ленина)

# Разработка платформы для фотограмметрического контроля

Выполнила:

Руководитель:

Дегтярева Алиса Андреевна, гр. 7303  
Заславский Марк Маркович, к.т.н., доцент

Санкт-Петербург, 2021

Запасные слайды

# Подготовка объекта к фотограмметрии

- 4-8 камер
- Использование CS Tags: насыщение снимков, вычисление коэффициента размера

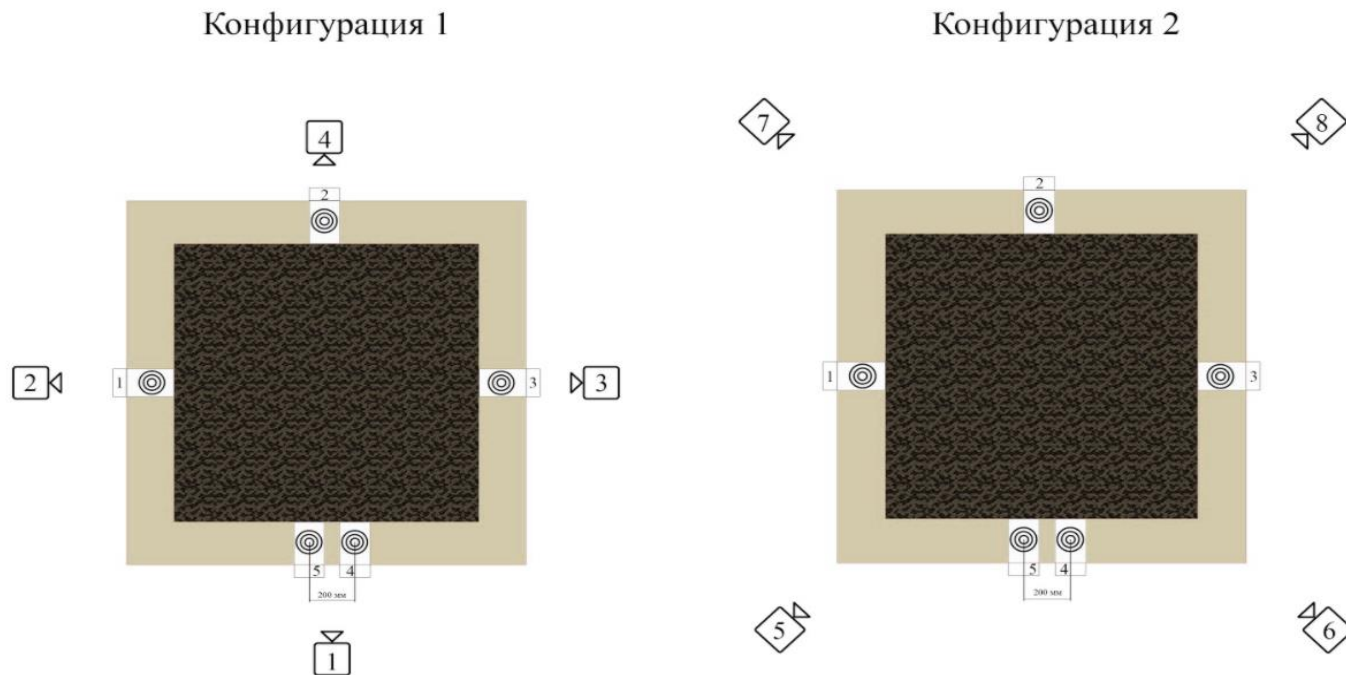


Рисунок 7 – Конфигурации камер и тегов

# Алгоритм поиска плоскости горизонта

**вычислить\_плоскости\_горизонта():**

    точка\_плоскости=**вычислить\_среднее**(координаты\_тегов)

    нормаль\_плоскости=**вычислить\_нормаль\_методом\_наименьших\_квадратов**(координаты\_тегов)

    отдаленные\_точки=**отсортировать\_точки\_модели\_в\_порядке\_отдаления\_от\_плоскости\_горизонта**(точки\_модели)

    самые\_дальние\_точки=**выбрать\_самые\_дальние\_точки**(отдаленные\_точки, процент\_отдаленных\_точек)

    точка\_противоположной\_плоскости=**вычислить\_среднее**(самые\_дальние\_точки)

    точка\_плоскости=**ориентировать\_от\_цента**(точка\_плоскости)

    точка\_противоположной\_плоскости=**ориентировать\_от\_центра**(точка\_противоположной\_плоскости)

**вернуть** нормаль, точка\_плоскости, точка\_противоположной\_плоскости

# Алгоритм удаления шумов

**убрать\_шумы():**

**точки\_в\_плоскости=[]**

**пока** плоскость **не достигла** точки остановки:

**точки\_в\_плоскости.добавить**( **вычислить\_точки\_близкие\_к\_плоскости**(  
плоскость\_противоположная\_горизонту, точки\_модели))

**новая\_плоскость=сдвинуть\_плоскость**(  
плоскость\_противоположная\_горизонту, **единичный\_сдвиг** \* номер\_итерации)

        номер\_итерации=**индекс\_максимального\_значения**(точки\_в\_плоскости)

        плоскость\_противоположная\_горизонту = **сдвинуть\_плоскость**(  
плоскость\_противоположная\_горизонту, **единичный\_сдвиг** \* номер\_итерации)

**точки\_модели=отбросить\_точки\_вектора\_которых\_сонаправленны\_нормали\_плоскости**(плоскость\_горизонта, точки\_модели)

**точки\_модели=отбросить\_точки\_вектора\_которых\_сонаправленны\_нормали\_плоскости**(плоскость\_противоположная\_горизонту, точки\_модели)

## Пример модели с удалением шумов

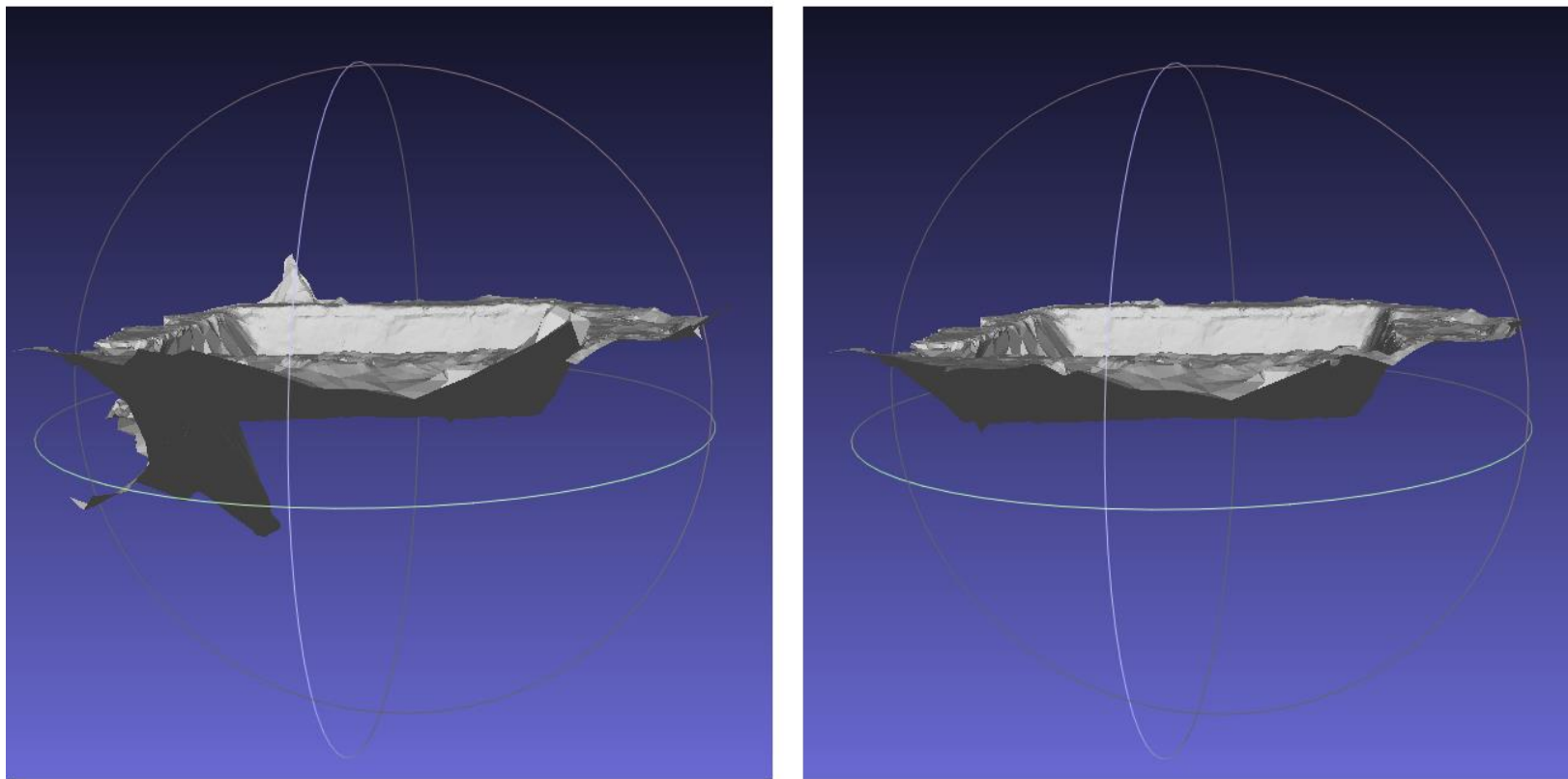


Рисунок 8 – Удаление шумов



# Построение ограничивающих плоскостей

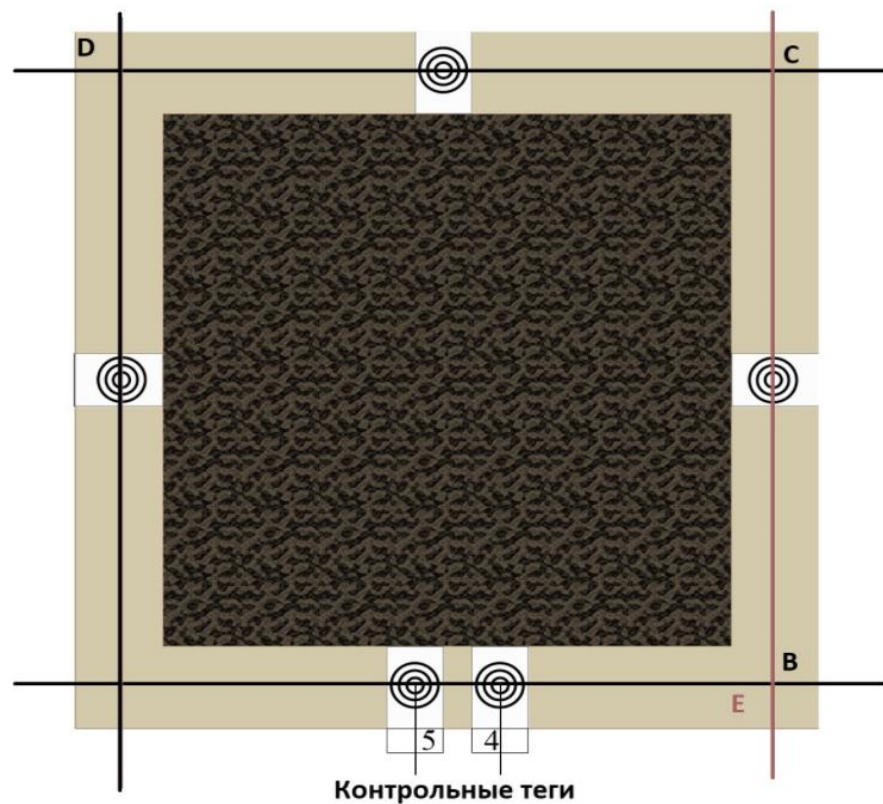


Рисунок 9 – Построение ограничивающих плоскостей

# Алгоритм построения ограничивающих плоскостей

**построить\_ограничивающие\_плоскости():**

вектор\_А=**вектор\_через\_точки**(контрольные\_теги)

нормаль\_контрольной\_плоскости=**перпендикулярный\_вектор**(вектор\_А,  
нормаль\_плоскости\_горизонта)

точка\_контрольной\_плоскости=**вычислить\_среднее**(контрольные\_теги)

контрольная\_плоскость=[нормаль\_контрольной\_плоскости, точка\_контрольной\_плоскости]

точка\_противоположной\_контрольной\_плоскости=**самая\_дальняя\_точка\_от\_плоскости**(контроль  
ная\_плоскость, теги)

противоположная\_контрольная\_плоскость=[нормаль\_контрольной\_плоскости,  
точка\_противоположной\_контрольной\_плоскости]

плоскость\_Р=[вектор\_А, точка\_контрольной\_плоскости]

точка\_боковой\_плоскости=**самая\_дальняя\_точка\_от\_плоскости**(плоскость\_Р, теги)

боковая\_плоскость=[вектор\_А, плоскости]

точка\_противоположной\_боковой\_плоскости=**самая\_дальняя\_точка\_от\_плоскости**(боковая\_плоскость, теги)

противоположная\_боковая\_плоскость=[вектор\_А, точка\_противоположной\_боковой\_плоскости]

**для каждой** плоскости:

**корректировать\_нормаль**(плоскость)

**вернуть** плоскости

# Пример модели, обрезанной по ограничивающим плоскостям

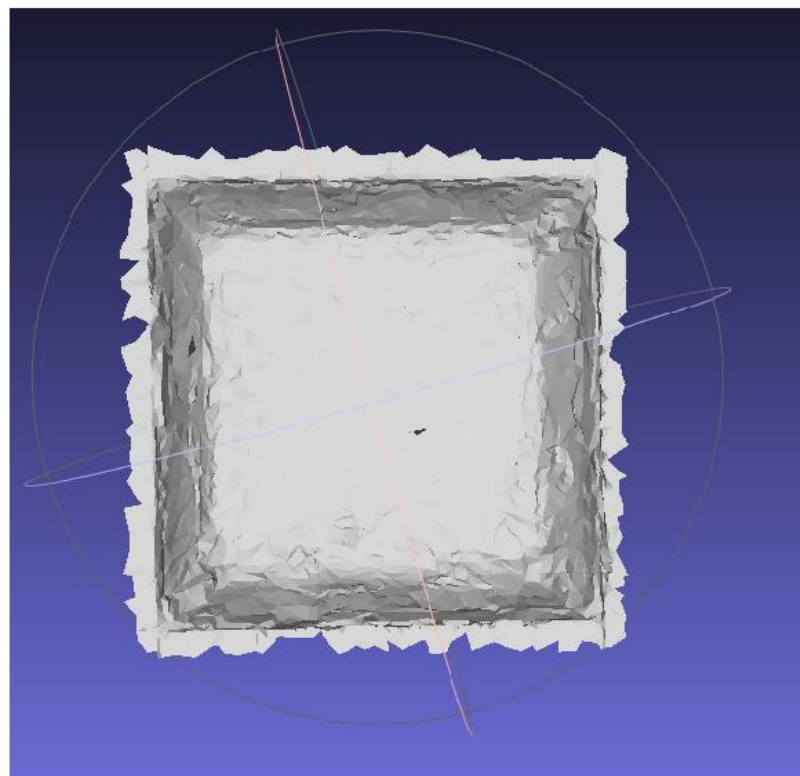
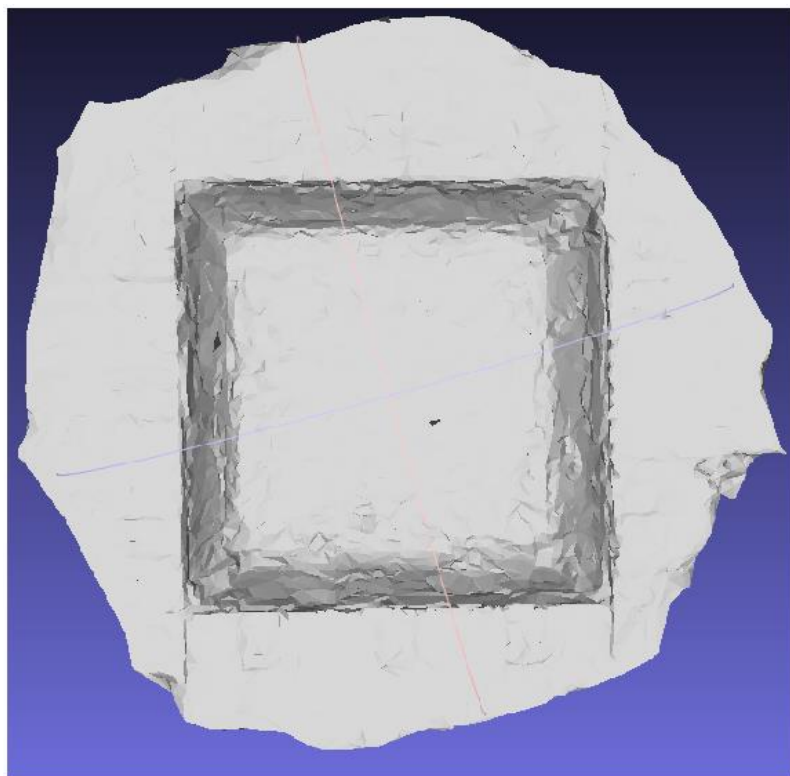


Рисунок 10 – Модель, обрезанная по ограничивающим плоскостям

# Алгоритм поворота ограничивающих плоскостей

**повернуть\_все\_ограничивающие\_плоскости():**

    центр\_объекта=**вычислить\_центр**(точки\_модели)

    точки\_модели=[]

**для** угла **от 0 до 365:**

        плоскости=**повернуть\_плоскости**(ограничивающие\_плоскости, угол \*  
единичный\_поворот, центр\_объекта)

        точки\_модели.**добавить**(количество\_точек\_внутри\_ограничивающей\_фигуры(  
плоскости, точки\_модели))

        угол=**индекс\_минимального\_значения**(точки\_модели)

        плоскости=**повернуть\_плоскости**(ограничивающие\_плоскости, угол \*  
единичный\_поворот, центр\_объекта)

**вернуть** плоскости

# Алгоритм поворота каждой ограничивающей плоскости

**повернуть\_каждую\_ограничивающую\_плоскость():**

**для каждой** плоскости:

        центр\_объекта=**вычислить\_центр**(

        точки\_пересечения\_плоскости\_с\_перпендикулярными)

        точки\_модели=[]

**для** угла **от 0 до 45:**

            плоскость=**повернуть\_плоскость**(плоскость, угол \* единичный\_поворот,  
            центр\_объекта)

            точки\_модели.**добавить**(

**количество\_точек\_внутри\_ограничивающей\_фигуры**(плоскость, точки\_модели)

            угол=**индекс\_минимального\_значения**(точки\_модели)

            плоскость=**повернуть\_плоскость**(плоскость, угол \* единичный\_поворот,  
            центр\_объекта)

**вернуть** плоскости

# Примеры моделей, где требуется повернуть ограничивающие плоскости

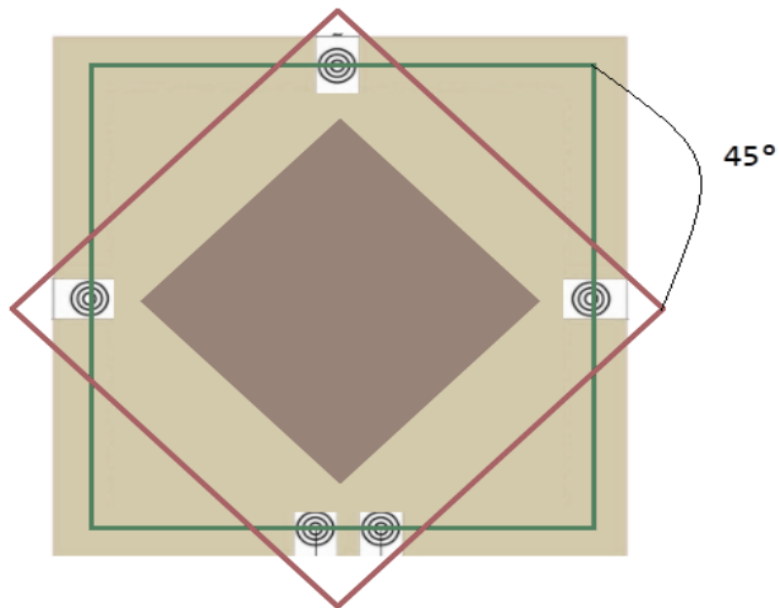


Рисунок 11 – Модель, где требуется повернуть все ограничивающие плоскости

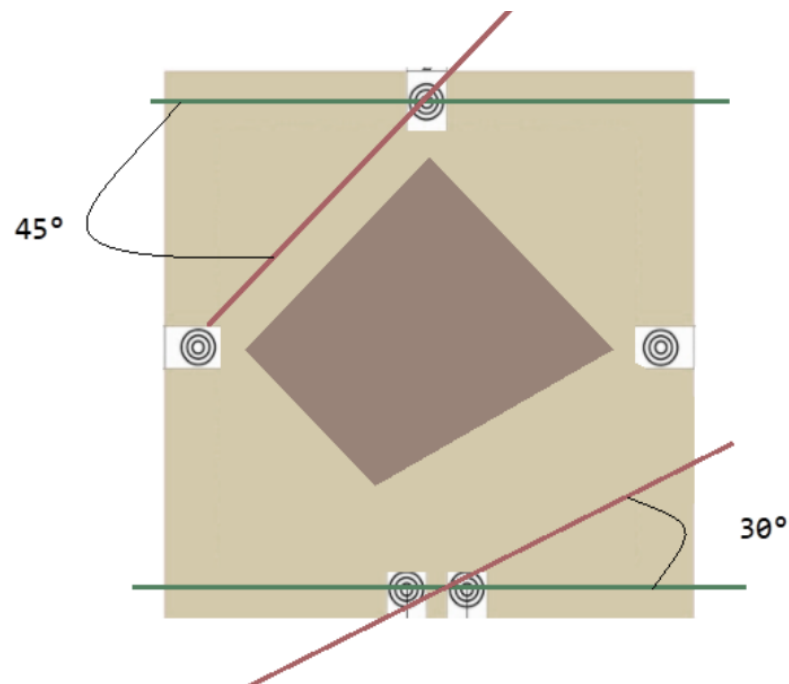


Рисунок 12 – Модель, где требуется повернуть некоторые ограничивающие плоскости

## Пример модели с повернутыми ограничивающими плоскостями

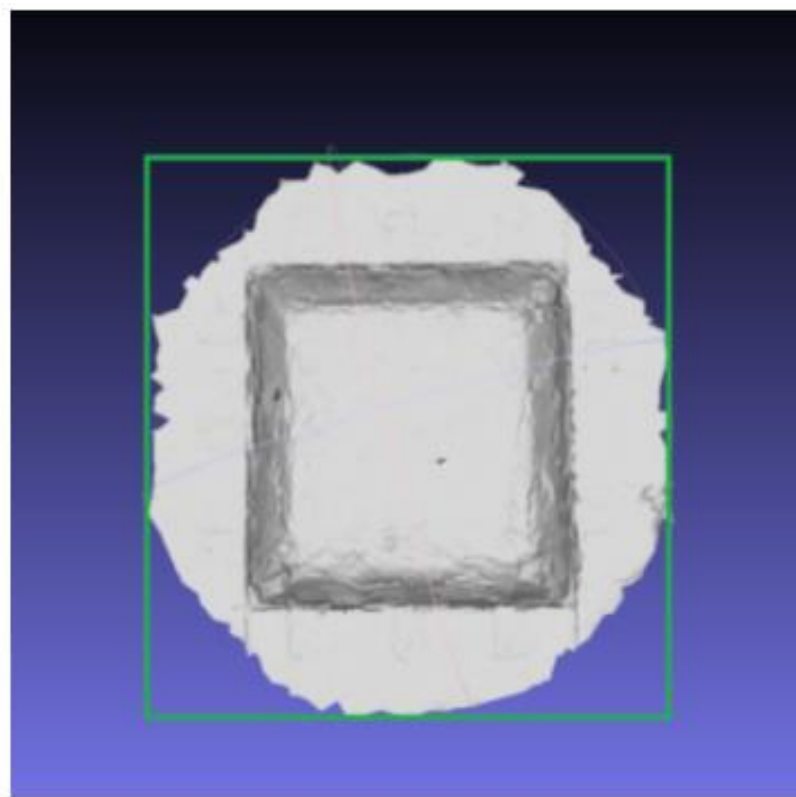
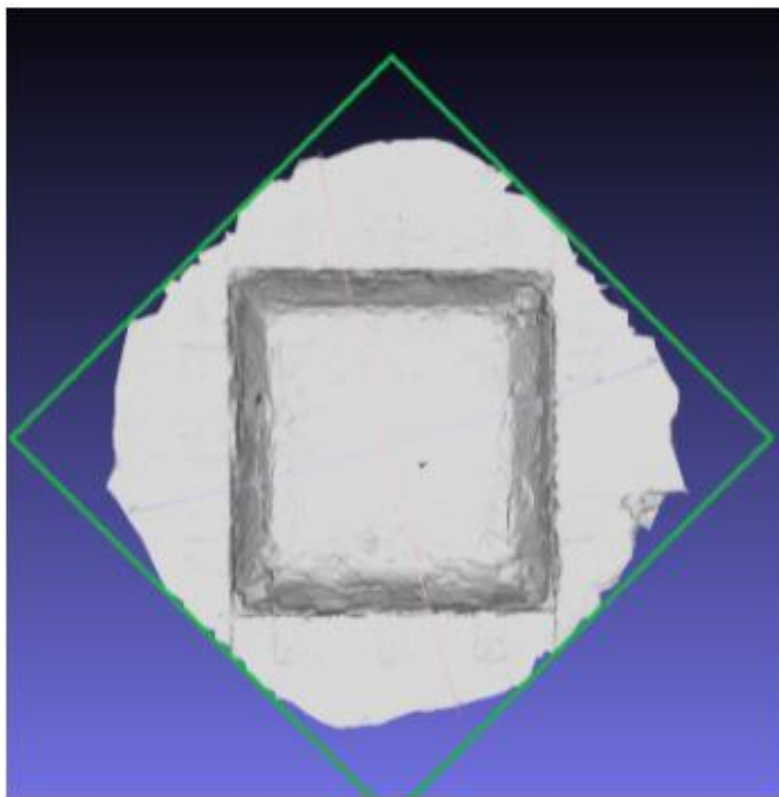


Рисунок 13 – Поворот ограничивающих плоскостей модели

# Алгоритм сдвига ограничивающих плоскостей с расчетом количества точек в плоскости горизонта

**сдвинуть\_ограничивающие\_плоскости\_через\_процент\_точек():**

**для каждой** плоскости:

индекс=0

предыдущее\_количество\_точек = 0

**пока** плоскость **не достигла** точки остановки:

новая\_плоскость=**сдвинуть\_плоскость**(плоскость, единичный\_сдвиг \* номер\_итерации)

внешние\_точки=**выделить\_внешние\_точки**(плоскость, точки\_модели)

внешние\_точки=**выделить\_точки\_лежащие\_в\_плоскости**( плоскость\_горизонта, внешние\_точки)

текущее\_количество\_точек=**количество**(внешние\_точки)

**если** предыдущее\_количество\_точек **!=** 0:

процент\_разницы=(предыдущее\_количество\_точек-текущее\_количество\_точек)\*100/предыдущее\_количество\_точек

**если** процент\_разницы **<=** заданный\_процент:

индекс = номер\_итерации

**прервать\_цикл**

предыдущее\_количество\_точек=текущее\_количество\_точек

новая\_плоскость=**сдвинуть\_плоскость**(плоскость, единичный\_сдвиг \* индекс)

**вернуть** новые\_плоскости



# Алгоритм сдвига ограничивающих плоскостей с вычислением расстояний

**сдвинуть\_ограничивающие\_плоскости\_через\_вычисление\_расстояний():**

**для каждой** плоскости:

расстояния = []

**пока** плоскость **не достигла** точки остановки:

новая\_плоскость=**сдвинуть\_плоскость**(плоскость, единичный\_сдвиг \*  
номер\_итерации)

внешние\_точки=**выделить\_внешние\_точки**(плоскость, точки\_модели)

дальняя\_точка=**найти\_дальнюю\_точку**(плоскость\_горизонта, внешние\_точки)

расстояния.**добавить**(расстояние(плоскость\_горизонта, дальняя\_точка))

ненулевые\_расстояния=**отфильтровать\_массив**(расстояния, расстояние!=0)

наименьшее\_расстояние=**наименьшее\_значение**(ненулевые\_расстояния)

индекс=0

**для каждого** расстояния из расстояний:

**если** расстояние = 0:

**следующая\_итерация**

процент\_разницы = (расстояние-  
наименьшее\_расстояние)\*100/наименьшее\_расстояние

**если** процент\_разницы >= заданный\_процент:

индекс = номер\_итерации

**прервать\_цикл**

новая\_плоскость=**сдвинуть\_плоскость**(плоскость, единичный\_сдвиг \* индекс)

**вернуть** новые\_плоскости

# Пример модели со сдвинутыми ограничивающими плоскостями

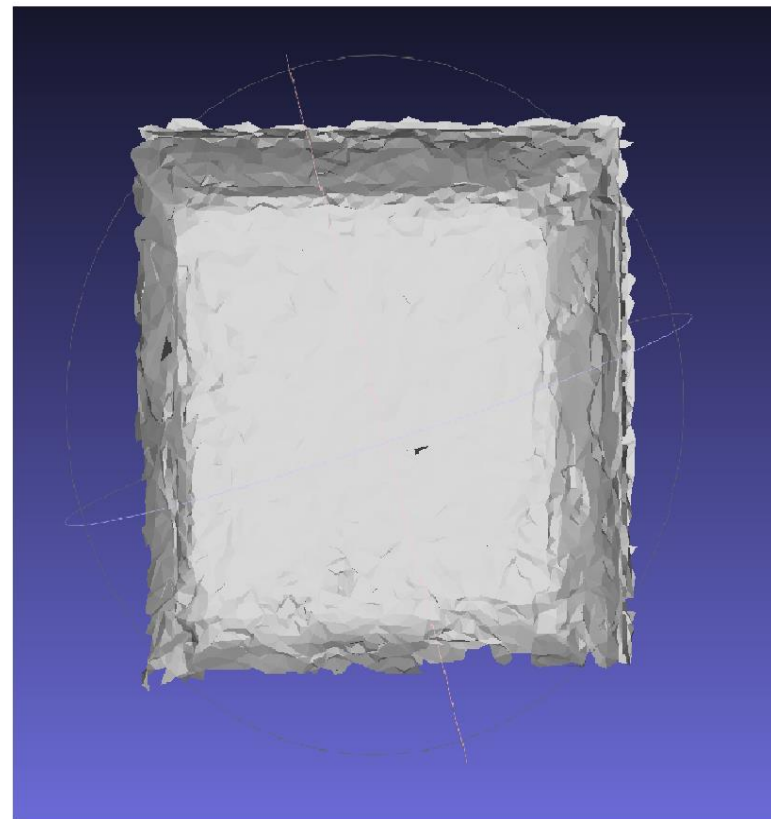
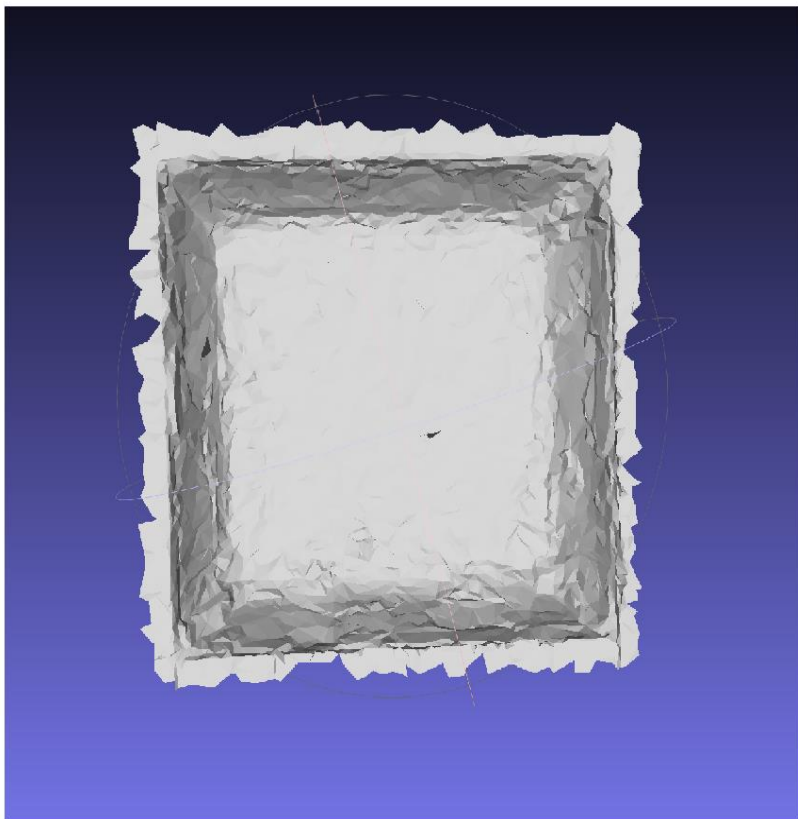


Рисунок 14 – Сдвиг ограничивающих плоскостей модели

# Алгоритмы вычисления объемов

**объем\_через\_ограничивающие\_плоскости():**

крайние\_точки=**вычислить\_точки\_пересечения**(ограничивающие\_плоскости)  
крайние\_точки=**спроецировать\_на\_плоскость**(крайние\_точки, плоскость\_горизонта)  
новые\_полигоны=**вычислить\_полигоны**(крайние\_точки)  
точки\_модели.**добавить**(крайние\_точки)  
полигоны\_модели.**добавить**(новые\_полигоны)  
объем=0

**для каждого** полигона:

объем=объем+**вычислить\_объем\_тетраэдра**(полигон)

**вернуть** объем

**объем\_через\_выпуклую\_оболочку():**

тетраэдры\_триангуляции=**триангуляция\_делоне**(точки\_модели)  
выпуклая\_оболочка\_триангуляции=**выпуклая\_оболочка**(тетраэдры\_триангуляции)  
объем=0

**для каждого** полигона **в** выпуклой\_оболочке:

объем=объем+**вычислить\_объем\_тетраэдра**(полигон)

**вернуть** объем

**объем\_через\_тетраэдры\_триангуляции():**

тетраэдры\_триангуляции=**триангуляция\_делоне**(точки\_модели)  
объем=0

**для каждого** тетраэдра **в** выпуклой\_оболочке:

объем=объем+**вычислить\_объем\_тетраэдра**(тетраэдр)

**вернуть** объем

# Алгоритмы построения ограничивающих тел и вычисления линейных размеров

**внутреннее\_ограничивающее\_тело():**

точки=**вычислить\_точки\_пересечения**(ограничивающие\_плоскости)

точки=**спроецировать\_на\_плоскость**(точки, плоскость\_горизонта)

нижние\_точки=**спроецировать\_на\_плоскость**(точки,плоскость\_противоположная\_плоскости\_горизонта)

**вернуть** точки, нижние\_точки

Внешнее ограничивающее тело строится следующим образом: выбираются точки внутреннего ограничивающего тела, лежащие в плоскости горизонта. Далее построения происходят также, как в случае с построением ограничивающих плоскостей, с той разницей, что вместо тегов вычисления выполняются над этими точками, а в качестве контрольных тегов выбираются две соседние точки.

**линейные размеры():**

длина=**вычислить\_расстояние**(боковая\_плоскость\_внешнего\_тела\_1, противоположная\_боковая\_плоскость\_тела\_1)

ширина=**вычислить\_расстояние**(боковая\_плоскость\_внешнего\_тела\_2, противоположная\_боковая\_плоскость\_тела\_2)

высота=**вычислить\_расстояние**(верхняя\_плоскость\_внешнего\_тела,нижняя\_плоскость\_внешнего\_тела)

**вернуть** длина, ширина, высота

# Эксперимент 1

№ запуска	TagsPlane	Convex Hull	Triangulation	Parallelepiped	Длина	Глубина	Ширина	Расстановка маркеров	Оценка модели		
	Фактические объемы, м^3				Фактические размеры, м						
	0,441	0,441	0,441	0,506	1,500	0,225	1,500				
	Вычисленные объемы, м^3				Вычисленные размеры, м						
1	0,448	0,439	0,439	0,536	1,470	0,250	1,460	успешно	есть замечания		
2	0,448	0,424	0,424	0,528	1,450	0,250	1,460	успешно	есть замечания		
3	0,462	0,466	0,466	0,558	1,480	0,260	1,470	успешно	без замечаний		
4	0,450	0,418	0,418	0,531	1,450	0,250	1,450	успешно	есть замечания		
5	0,443	0,400	0,400	0,512	1,440	0,250	1,430	успешно	есть замечания		
6	0,459	0,421	0,421	0,530	1,450	0,250	1,450	успешно	без замечаний		
7	0,454	0,418	0,418	0,510	1,460	0,250	1,420	успешно	без замечаний		
8	0,456	0,432	0,432	0,538	1,470	0,250	1,450	успешно	без замечаний		
9	0,454	0,404	0,404	0,515	1,430	0,250	1,460	успешно	без замечаний		
10	0,456	0,415	0,415	0,531	1,450	0,250	1,450	успешно	без замечаний		
№ запуска	Относительная погрешность каждого эксперимента, %							Качество вычисления объема			
1	1,497	0,544	0,544	5,889	2,000	11,111	2,667%	Фактический объем, м^3			
2	1,655	3,855	3,855	4,407	3,333	11,111	2,667%	Минимум	Среднее	Максимум	
3	4,739	5,760	5,760	10,198	1,333	15,556	2,000%	0,426	0,441	0,456	
4	2,132	5,193	5,193	4,862	3,333	11,111	3,333%	Допустимая погрешность, %			
5	0,363	9,365	9,365	1,107	4,000	11,111	4,667%	-3,401%	- x -	3,401%	
6	3,991	4,626	4,626	4,743	3,333	11,111	3,333%	Кол-во входящих в допустимую погрешность			
7	2,834	5,329	5,329	0,751	2,667	11,111	5,333%	7	из	10	
8	3,492	2,041	2,041	6,304	2,000	11,111	3,333%				
9	2,902	8,458	8,458	1,739	4,667	11,111	2,667%				
10	3,356	5,828	5,828	4,881	3,333	11,111	3,333%				
Минимальная	0,363	0,544	0,544	0,751	1,333	11,111	2,000%				
Средняя	2,696	5,100	5,100	4,488	3,000	11,556	3,333%				
Максимальная	4,739	9,365	9,365	10,198	4,667	15,556	5,333%				

# Эксперимент 2

№ запуска	TagsPlane	Convex Hull	Triangulation	Parallelepiped	Длина	Глубина	Ширина	Расстановка маркеров	Оценка модели	
	Фактические объемы, м^3				Фактические размеры, м					
	0,441	0,441	0,441	0,506	1,500	0,225	1,500			
	Вычисленные объемы, м^3				Вычисленные размеры, м					
1	0,449	0,524	0,524	0,695	1,690	0,280	1,490	успешно	есть замечания	
2	0,455	0,438	0,438	0,537	1,450	0,250	1,460	успешно	есть замечания	
3	0,434	0,508	0,508	0,574	1,500	0,250	1,500	успешно	есть замечания	
4	0,470	0,458	0,458	0,539	1,440	0,250	1,480	успешно	без замечаний	
5	0,452	0,445	0,455	0,544	1,470	0,250	1,470	успешно	есть замечания	
6	0,450	0,461	0,461	0,547	1,470	0,250	1,480	успешно	есть замечания	
7	0,446	0,457	0,457	0,544	1,460	0,250	1,480	успешно	есть замечания	
8	0,448	0,464	0,464	0,544	1,470	0,250	1,450	успешно	есть замечания	
9	0,467	0,456	0,456	0,544	1,470	0,250	1,470	успешно	есть замечания	
10	ошибка	ошибка	ошибка	ошибка	ошибка	ошибка	ошибка	ошибка	есть замечания	
№ запуска	Относительная погрешность каждого эксперимента, %							Качество вычисления объема		
1	1,882	18,707	18,707	37,273	12,667	24,444	0,667%	Фактический объем, м^3		
2	3,107	0,680	0,680	6,146	3,333	11,111	2,667%	Минимум	Среднее	Максимум
3	1,565	15,170	15,170	13,439	0,000	11,111	0,000%	0,426	0,441	0,456
4	6,508	3,923	3,923	6,561	4,000	11,111	1,333%	Допустимая погрешность, %		
5	2,540	0,930	3,197	7,411	2,000	11,111	2,000%	-3,401%	- x -	3,401%
6	1,950	4,580	4,580	8,162	2,000	11,111	1,333%	Кол-во входящих в допустимую погрешность		
7	1,179	3,515	3,515	7,431	2,667	11,111	1,333%	7	из	10
8	1,587	5,147	5,147	7,569	2,000	11,111	3,333%			
9	5,896	3,333	3,333	7,569	2,000	11,111	2,000%			
10	ошибка	ошибка	ошибка	ошибка	ошибка	ошибка	ошибка			
Минимальная	1,179	0,680	0,680	6,146	0,000	11,111	0,000			
Средняя	2,913	6,221	6,473	11,285	3,407	12,593	1,630			
Максимальная	6,508	18,707	18,707	37,273	12,667	24,444	3,333			

# Эксперимент 3

№ запуска	Объем, м³				Линейные размеры, м			Фактический объем, м³		Фактические линейные размеры, м		
	TagsPlane	Convex Hull	Triangulation	Параллелепипед	Длина	Глубина	Ширина	Объект	Параллелепипед	Фактическая длина	Фактическая глубина	Фактическая ширина
1	0,283	0,363	0,363	0,426	1,670	0,150	1,670	0,158	0,191	1,500	0,085	1,500
2	0,286	0,315	0,315	0,358	1,500	0,160	1,540	0,222	0,248	1,500	0,110	1,500
3	0,257	0,291	0,291	0,317	1,500	0,140	1,560	0,252	0,304	1,500	0,135	1,500
4	0,434	0,508	0,508	0,574	1,500	0,250	1,500	0,515	0,615	1,500	0,274	1,500
5	0,454	0,410	0,410	0,516	1,440	0,250	1,450	0,441	0,506	1,500	0,225	1,500
6	0,452	0,445	0,455	0,544	1,470	0,250	1,470	0,441	0,506	1,500	0,225	1,500
7	0,450	0,461	0,461	0,547	1,470	0,250	1,480	0,441	0,506	1,500	0,225	1,500
8	0,446	0,457	0,457	0,544	1,460	0,250	1,480	0,441	0,506	1,500	0,225	1,500
9	0,012	0,026	0,024	0,030	0,380	0,300	0,260	0,028	0,031	0,385	0,300	0,265
10	0,363	-	-	0,484	1,450	0,230	1,440	0,358	0,506	1,500	0,225	1,500
№ запуска	Относительная погрешность каждого эксперимента, %							Допустимая погрешность, %		Входит в допустимую погрешность		
1	79,114	129,747	129,747	123,037	11,333	76,471	11,333	3,165		нет		
2	28,829	41,892	41,892	44,355	0,000	45,455	2,667	11,306		нет		
3	2,167	15,683	15,683	4,276	0,000	3,704	4,000	13,536		да		
4	15,646	1,263	1,263	6,667	0,000	8,592	0,000	2,041		да		
5	2,948	7,029	7,029	1,996	4,000	11,111	3,333	3,401		да		
6	2,540	0,930	3,197	7,411	2,000	11,111	2,000	3,401		да		
7	1,950	4,580	4,580	8,162	2,000	11,111	1,333	3,401		да		
8	1,179	3,515	3,515	7,431	2,667	11,111	1,333	3,401		да		
9	57,143	7,143	14,286	3,226	1,299	0,000	1,887	7,857		да		
10	1,397	-	-	4,348	3,333	2,222	4,000	5,028		да		

# Эксперимент 4

Фактический объем, м <sup>3</sup>	Вычисленный объем, м <sup>3</sup>	Фактическая погрешность, %	Вычисленная погрешность, %
0,153..0,164	0,213	39..40	41