

**ПОДСИСТЕМА МАШИННОГО ЗРЕНИЯ ДЛЯ БУТОБОВ С ФУНКЦИОНАЛОМ
СЕГМЕНТАЦИИ ЭЛЕМЕНТОВ ГОРНОРУДНОЙ МАССЫ НА ГРОХОТНОЙ
РЕШЕТКЕ, ИНТЕРПОЛЯЦИИ ОБЛАКА ТОЧЕК, А ТАКЖЕ РАСЧЕТА
КЛЮЧЕВЫХ МАССОГАБАРИТНЫХ ПАРАМЕТРОВ.**

Описание программы

АННОТАЦИЯ

В данном программном документе приведено описание программного модуля, предназначенного для управления работой программно-аппаратного комплекса «Подсистема машинного зрения для бутобоев с функционалом сегментации элементов горнорудной массы на грохотной решетке, интерполяции облака точек, а также расчета ключевых массогабаритных параметров». Модуль может быть использован в горнодобывающей промышленности для определения направления удара бутобоя с целью повышения эффективности дробления породы, а также для оценки качества дробления после выполнения очередной команды от системы принятия решений.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	4
2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ	5
3. ОПИСАНИЕ СТРУКТУРЫ, ЛОГИЧЕСКОЙ СХЕМЫ И ИНТЕРФЕЙСА ПРОГРАММЫ	6
4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА.....	17
5. ВЫЗОВ, ЗАГРУЗКА, ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ.....	18

1. ОБЩИЕ СВЕДЕНИЯ

1.1. ОБОЗНАЧЕНИЕ И НАИМЕНОВАНИЕ ПРОГРАММЫ

Программный модуль подсистемы машинного зрения для бутобоев с функционалом сегментации элементов горнорудной массы на грохотной решетке, интерполяции облака точек и расчета ключевых массогабаритных параметров крупных элементов горнорудной массы. Программа может быть использована в горнодобывающей промышленности для определения направления удара бутобоя с целью повышения эффективности дробления, а также для оценки качества дробления после выполнения очередной команды от системы принятия решений.

1.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, НЕОБХОДИМОЕ ДЛЯ ФУНКЦИОНИРОВАНИЯ ПРОГРАММЫ

- Необходим предустановленный интерпретатор Python3.8. в папку по умолчанию «/usr/bin/python3.8»
- Для выполнения вычислений на GPU рекомендовано использование архитектуры Nvidia CUDA. В противном случае дополнительное ПО не требуется, все необходимые зависимости поставляются в рамках дистрибутива.

1.3. ЯЗЫКИ ПРОГРАММИРОВАНИЯ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

- Для реализации основных структур данных, алгоритмов и элементов пользовательского интерфейса использовался язык Python 3.8.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программный модуль подсистемы машинного зрения предназначен для определения целевой точки удара бутобоя таким образом, чтобы повысить эффективность и качество процесса дробления породы. Модуль обладает следующим функционалом:

- сегментация элементов горнорудной массы на грохотной решетке;
- интерполяция облака точек для повышения точности дальнейшего анализа;
- расчет ключевых массогабаритных параметров крупных элементов горнорудной массы.

Программа может быть использована в горнодобывающей промышленности для определения направления удара бутобоя с целью измельчения наиболее крупных фрагментов породы, а также для оценки качества дробления после выполнения очередной команды от системы принятия решений.

3. ОПИСАНИЕ СТРУКТУРЫ, ЛОГИЧЕСКОЙ СХЕМЫ И ИНТЕРФЕЙСА ПРОГРАММЫ

3.1 СТРУКТУРА ПРОГРАММЫ

Программный модуль подсистемы машинного зрения имеет следующую структуру (рисунок 1).



Рисунок 1— Структура программного модуля

Модуль интерполяции решает задачу повышения плотности и регулярности поданного на вход облака точек таким образом, чтобы сделать их достаточными для дальнейшего анализа. Увеличение плотности происходит за счет интерполяции исходного облака точек.

Модуль сегментации строит карту высот на основе интерполированного облака, выполняет ее предобработку (фильтрация шумов), а затем на основе полученных данных выполняет поиск и сегментацию крупных обломков породы на фоне горнорудной массы.

Модуль расчета массогабаритных характеристик позволяет определять важные для системы принятия решений целевые состояния каждого фрагмента — центр масс, наивысшую точку на поверхности камня, площадь, занимаемую им на грохотной решетке, и объем камня. Эти данные используются для оценки качества дробления породы бутобоем после выполнения очередной команды от системы принятия решений.

3.2. АЛГОРИТМ ПРОГРАММЫ

Предлагаемый программный модуль подсистемы машинного зрения (ПМЗ) реализует следующий алгоритм (рисунок 2).



Рисунок 2 — Алгоритм работы модуля ПМЗ

Ниже приведено более подробное описание каждого этапа алгоритма.

Этап 1. Интерполяция исходного облака точек

На начальном этапе на вход программного модуля подается облако точек, описывающее сцену с грохотной решеткой и расположенной на ней массой руды. Как правило, плотность точек в таком облаке нерегулярна и недостаточна для дальнейшего статистического анализа, построения карты высот и сегментирования отдельных фрагментов горнорудной массы.

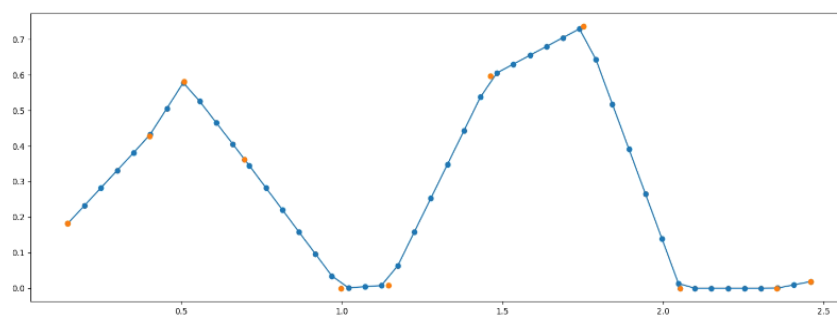


Рисунок 3 — Пример интерполяции кубическим сплайном в одном из срезов XZ

Для повышения плотности точек выполняется интерполяция кубическим сплайном по срезам облака в плоскостях XZ и YZ. На рисунке 3 показан пример

интерполяции в одном из срезов XZ — оранжевым показаны точки исходного облака, синим — путем интерполяции.

Этап 2. Построение карты высот и предобработка

На основе интерполированных данных далее строится карта высот, которая является источником данных для системы принятия решений и дальнейшего анализа облака точек.

Карта высот представляет собой изображение, значения пикселей которого интерпретируются как высота поверхности в некоторой точке: обычно более темный цвет соответствует минимальной высоте, а более светлый — максимальной.

На первом шаге задается двумерная матрица, индекс элемента (i, j) которой, умноженный на шаг сетки $step$ (по умолчанию принят равным 5 см), совпадает с фактическими координатами (x, y) в системе координат гродотной решетки.

Далее определяются два массива, отвечающих за направление исходных осей OX и OY . Каждый массив содержит координаты точек, заданных вдоль оси с шагом $step$.

На вход алгоритма подается интерполированное облако точек. Алгоритм перебирает точки облака, выбирая удовлетворяющие следующим условиям:

$$x_{i-1} < p_x < x_{i+1}$$

$$y_{i-1} < p_y < y_{i+1},$$

где p_x и p_y — координаты точек интерполированного облака.

Для выбранных точек вычисляется среднее арифметическое их координат по оси OZ , результат записывается как элемент (i, j) матрицы (карты высот).

Полученная карта содержит ряд визуальных шумов, поэтому далее к ней применяется фильтр Гаусса, а также морфологические операции открытия (эрозия, затем дилатация) и закрытия (дилатация, затем эрозия), позволяющие сгладить имеющиеся выбросы и удалить шумы.

Здесь же происходит отсечение элементов карты высот, значение которых после преобразований матрицы является близким к нулю. Таким образом, определяются основные выступающие элементы на фоне горнорудной массы.

Далее к полученной карте высот применяется метод водораздела (*watershed*), который позволяет выделить контуры наиболее выраженных объектов.

Этап 3. Сегментация

К каждому главному контуру, обнаруженному методом водораздела, применяется следующий алгоритм.

Начиная с максимального значения высоты z_{max} , определенного для каждого контура, берутся срезы карты высот с шагом z_{step} . На каждом срезе выполняется поиск контуров. Найденные контуры запоминаются, а к соответствующим им срезам в трехмерном облаке применяется алгоритм агломеративной кластеризации.

Каждый последующий срез карты высот включает в себя срез карты высот и точки трехмерной области из предыдущего среза. Это необходимо для контроля поведения контуров и кластеров в следующих возможных ситуациях:

- Контур на текущем срезе не совпадает ни с одним из контуров с предыдущего среза и не содержит ни один из них — в этом случае новый контур добавляется к списку найденных контуров, и для этого контура также осуществляется кластеризация.
- Контур с предыдущего среза лежит в контуре текущего среза — в этом случае новый контур не добавляется к найденным; к точкам из предыдущего среза добавляются точки из текущего среза и уже по ним выполняется кластеризация.
- Два и более контура из предыдущего среза сливаются в контур с текущего среза — в этом случае для контура с текущего среза сохраняется условие наличия в нем нескольких объектов; при этом в алгоритм кластеризации передается количество искомым кластеров (количество контуров с предыдущего среза, которые были объединены в текущем контуре).

Алгоритм сегментации работает до тех пор, пока не достигнет значения z_{min} (минимальное значение координаты z в трехмерном облаке). После этого в результирующих контурах и кластеризованных облаках происходит запись основных требуемых параметров объектов.

Этап 4. Расчет массогабаритных параметров

Разработанный алгоритм сегментации позволяет определять важные для системы принятия решений целевые состояния каждого фрагмента — центр масс, наивысшую точку на поверхности камня, площадь, занимаемую им на грохотной решетке, и объем камня. Эти данные используются для дальнейшей оценки качества дробления породы бутобоем после выполнения очередной команды от системы принятия решений.

Координаты центра масс объекта (камня, крупного обломка породы) служат целевой точкой для следующего удара бутобоя, поэтому необходимо определить их с высокой точностью, чтобы избежать коллизий.

На этом этапе сначала вычисляется центр масс контура объекта в плоскости XU . Расчет происходит с помощью моментов контура — определённых средневзвешенных значений интенсивности пикселей изображения или функции таких моментов. Название «моменты» является аналогией с понятием момента в физике и механике, так как с их помощью можно вычислить другие характеристики изображения.

Моменты контура рассчитываются по следующей формуле:

$$m(p, q) = \sum_{i=1}^n I(x, y) x^p y^q$$

где $I(x, y)$ — значение яркости пикселя с координатами (x, y) ; p, q — параметры функции.

На основе моментов $m(0,0)$, $m(1,0)$ $m(0,1)$ вычисляются координаты центра масс контура (X_{cm}, Y_{cm}) :

$$X_{\text{цм}} = \frac{m(1,0)}{m(0,0)}; \quad Y_{\text{цм}} = \frac{m(0,1)}{m(0,0)}.$$

Поиск контура происходит на интерполированном облаке — оно более плотное, но менее точное по сравнению с исходным, поэтому вычисленный центр масс может не соответствовать реальной точке на поверхности камня в физическом пространстве (так как он изначально принадлежит неточному облаку). Для коррекции этой погрешности применяется следующий метод.

Сначала на основе исходного (неинтерполированного) облака точек строится *KD*-дерево — структура данных с разбиением пространства для упорядочивания точек в *k*-мерном пространстве. Пример такого дерева показан на рисунке 4. В построенном дереве выбирается точка на поверхности камня, ближайшая к центру масс, вычисленному ранее на основе характеристик контура и интерполированного облака. Именно на поверхности — так как таким образом легче задать направление удара бутобоя. Найденная таким образом точка и считается уточненным центром масс.

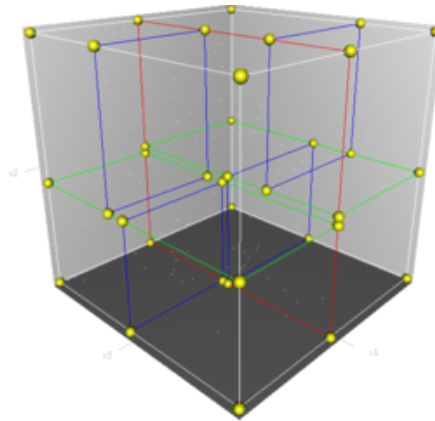


Рисунок 4 — Пример *KD*-дерева

Наивысшая точка объекта определяется как точка с наибольшим значением координаты *z* в исходном облаке в пределах контура, ограничивающего целевой объект.

Пример результата работы алгоритма показан на рисунке 5: черные точки обозначают центр масс объекта, розовые — его наивысшую точку.

Площадь целевого объекта определяется как площадь его контура по формуле Грина — как криволинейный интеграл по замкнутому контуру.

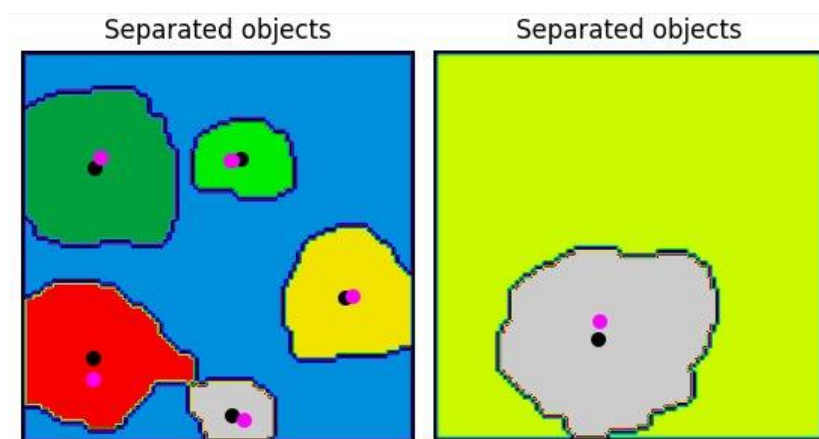


Рисунок 5 — Пример определения центра масс и наивысшей точки объекта

Объем объекта определяется как объем пространства, ограниченного поверхностью этого объекта (рис. 6). Построение поверхности происходит по следующему алгоритму. На первом этапе облако точек, описывающее целевой объект, необходимо ориентировать таким образом, чтобы основание объекта совпало с плоскостью XU . Далее требуется перейти от облака точек к сетке. Для этого ориентированное облако проецируется на плоскость XU путем обнуления координаты z . К полученным точкам на плоскости применяется триангуляция Делоне.

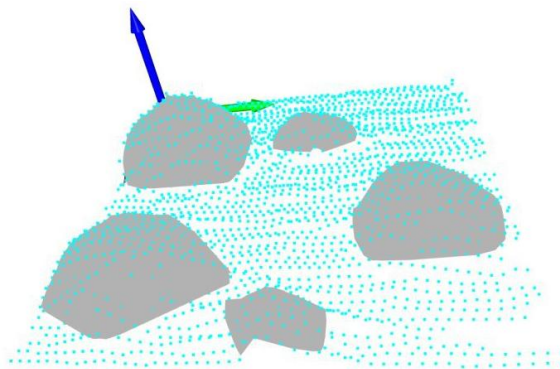


Рисунок 6 — Пример построения поверхности для расчета объема целевого объекта

Триангуляция (рис. 7а) подразумевает разбиение геометрического объекта на симплексы (в данном случае, треугольники). Говорят, что на множестве точек на плоскости задана триангуляция, если некоторые пары точек соединены ребром, и в полученном графе любая конечная грань образует треугольник, ребра не пересекаются, и граф максимален по количеству ребер.

Триангуляцией Делоне (рис. 7б) называется такая триангуляция, в которой для любого треугольника верно, что внутри описанной около него окружности не находится точек из исходного множества (правило Делоне).

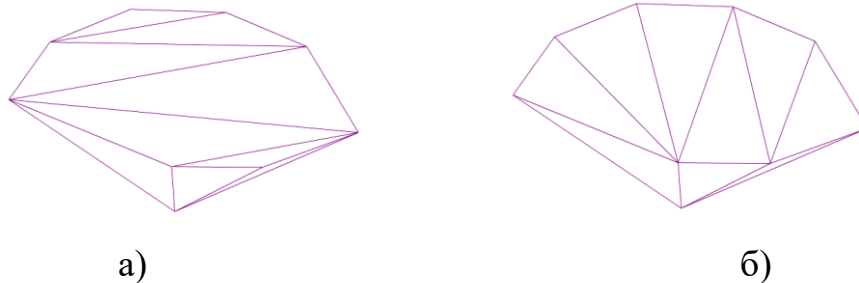


Рисунок 7 — Пример триангуляции: а) по классическому алгоритму, б) по алгоритму Делоне

Каждый четырехугольник в триангуляции представляет собой 2 треугольника, имеющих общее ребро. У каждого ребра есть ровно 2 треугольника, прилегающих к нему. Таким образом, каждый четырехугольник в триангуляции описывается ребром и двумя вершинами, находящимися напротив ребра в прилегающих треугольниках.

В данном случае для построения триангуляции используется алгоритм заметающей прямой, идея которого заключается в том, что все точки сортируются по координате x , а затем по очереди обрабатываются.

Для дальнейшего объяснения метода важно уточнить несколько терминов. Минимальная выпуклая оболочка множества точек — ребра, соединяющие некоторые из точек так, чтобы они образовывали многоугольник, содержащий все точки множества. Точка считается видимой для точки A , если отрезок, соединяющий ее с точкой A , не пересекает минимальную выпуклую оболочку множества точек. Ребро считается видимым для точки A , если его концы видимы для A .

Таким образом, триангуляция Делоне строится по следующему алгоритму:

- все точки сортируются по координате x ;
- на первых трех точках строится треугольник;
- далее для каждой следующей точки добавляются треугольники, образованные самой точкой и видимыми ребрами;

- четырехугольники, порожденные видимыми ребрами, проверяются на выполнение условия Делоне;
- если условие не выполняется, триангуляция в четырехугольнике перестраивается (для каждого четырехугольника всегда существует только два варианта разбиения), и запускается рекурсивная проверка для четырехугольников, порожденных ребрами текущего четырехугольника.

В результате выполнения описанного алгоритма получаем триангуляцию на множестве точек, представляющем собой проекцию облака. На основе 2D-координат вершин полученных треугольников и значений координаты z этих вершин (берутся из облака точек), строится трехмерная поверхность.

Объем объекта определяется как объем пространства, ограниченного полученной поверхностью и плоскостью XY .

3.3. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС ПРОГРАММЫ

Программный модуль подсистемы машинного зрения не имеет как такового графического пользовательского интерфейса и управляется через консоль.

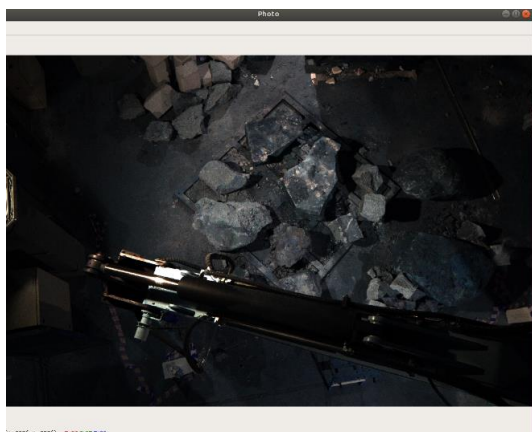
Запустить работу модуля возможно с помощью команды «./Environment/bin/python main_loop_stones.py» (на *Ubuntu*) или с помощью команды «.\Environment\bin\python main_loop_stones.py» (на *Windows*).

Модуль по умолчанию включает в себя 4 демонстрационных примера (находятся в директории *Data*). Примеры 1 и 2 содержат полные оцифрованные сцены лаборатории и демонстрируют процесс обрезки и анализа горнорудной массы. Примеры 3 и 4 содержат уже обрезанные сцены для дальнейшего анализа. Чтобы выполнить анализ соответствующей сцены, необходимо ввести номер сцены («1», «2», «3» или «4») и нажать *Enter*.

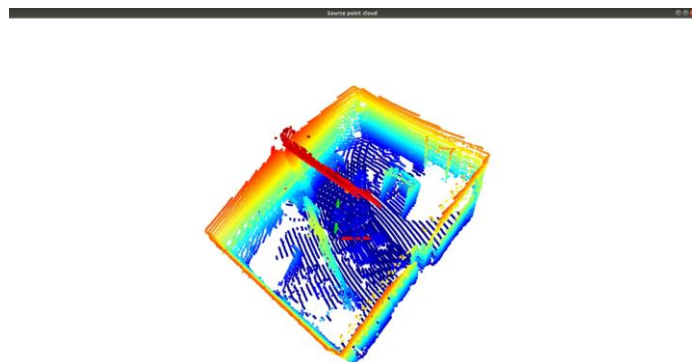
После запуска автоматически открываются следующие окна:

- окно с изображением исходной сцены для визуальной оценки (рисунок 8а);
- окно, содержащее исходное облако (рисунок 8б);
- окно, содержащее вырезанное облако, соответствующее области интереса (рисунок 8в);

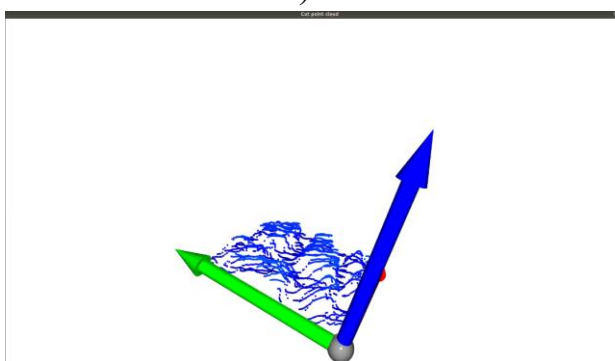
- окно, содержащее сегментированное облако с найденными отдельными элементами горнорудной массы (каждый цвет – отдельный камень) (рис. 8г);
- окно, содержащее интерполированное облако, которое используется для анализа элементов породы (рисунок 8д);
- окно, содержащее реконструированную поверхность, построенную по интерполированному облаку (рисунок 8е).



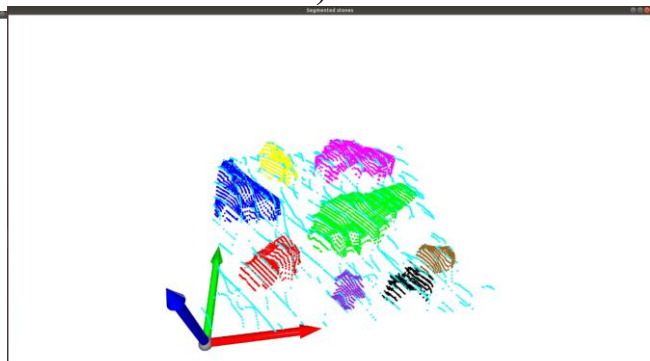
а)



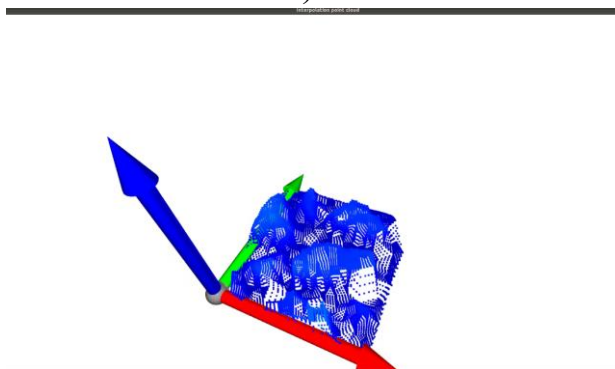
б)



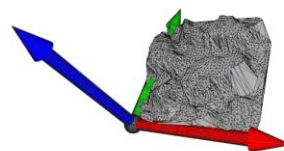
в)



г)



д)



е)

Рисунок 8 — Окна программного модуля

Все окна являются интерактивными, пользователь имеет возможность подробно рассмотреть сцену, изменяя ракурс и масштаб сцены.

Далее в консоль выводятся рассчитанные значения характеристик для отдельных фрагментов горнорудной массы (рисунок 9):

- *Square* – площадь камня, м^2 ;
- *Center of gravity* – координаты $[x, y, z]$ центра масс камня на поверхности камня, м ;
- *Highest point* – координата z наивысшей точки на камне, м ;
- *Volume* – объем камня, м^3 ;
- *Flat areas found* – количество найденных плоских площадок на камне, соответствующих заданным параметрам

```

ns@ns: ~/PycharmProjects/Software_Registration_Stone_Analizer
File Edit View Search Terminal Help
1
RESULT:
Stones Found 6
-----
Stone 1:
  Square (m2): 0.655
  Center of gravity: [1.5079627963824376, 1.0219362454157161, 0.3596644951650067]
  Highest point: [1.1437920346948183, 0.645136917953683, 0.4733112809367723]
  Volume (m3): 0.2433
  Flat areas found: 12
-----
Stone 2:
  Square (m2): 0.488625
  Center of gravity: [1.6885663365760306, 1.9987951387872638, 0.3878618654351387]
  Highest point: [1.801033125649498, 2.038688666294257, 0.49223575723412175]
  Volume (m3): 0.1679
  Flat areas found: 9
-----
Stone 3:
  Square (m2): 0.40406250000000005
  Center of gravity: [0.22247452638070197, 1.4261650686945328, 0.4875324973100694]
  Highest point: [0.64125554017310806, 1.453581944199401, 0.5001946581707236]
  Volume (m3): 0.1626
  Flat areas found: 6
-----
Stone 4:
  Square (m2): 0.22312500000000002
  Center of gravity: [0.5411181710824784, 0.5787640858432852, 0.3441594656726883]
  Highest point: [0.694667883453395, 0.6038753537293926, 0.4491764334056585]
  Volume (m3): 0.0643
  Flat areas found: 0
-----
Stone 5:
  Square (m2): 0.19624999999999998
  Center of gravity: [0.6908395018305478, 2.0466079759475875, 0.33430510820752257]
  Highest point: [0.6159650736422987, 2.2194219954121466, 0.41098556933319585]
  Volume (m3): 0.0517
  Flat areas found: 1
-----
Stone 6:
  Square (m2): 0.091875
  Center of gravity: [2.1795718695983384, 0.4368048397607317, 0.2363138742582107]
  Highest point: [2.325512399827706, 0.5811696510147332, 0.31192869402398293]
  Volume (m3): 0.0285
  
```

Рисунок 9 — Вывод рассчитанных значений в консоль

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для функционирования программы МВЦУБ требуется ПЭВМ со следующей конфигурацией:

- ОС *Ubuntu* 18.04;
- процессор *i5* и выше, со следующими характеристиками:
 - частота от 2,5 ГГц и выше;
 - 4 и больше ядер;
- оперативная память — 8 ГБ;
- *SSD* — 1 ТБ;
- видеокарта *Nvidia* с *CUDA* 4 ГБ;
- дисплей 21 дюйм.

5. ВЫЗОВ, ЗАГРУЗКА, ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ.

5.1 Вызов и загрузка

Вызов программы осуществляется через консоль с помощью команды «./Environment/bin/python main_loop_stones.py».

5.2 Входные данные

Входными данными программы является облако точек, описывающее сцену с грохотной решеткой и расположенной на ней массой руды;

5.3 Выходные данные

Выходными данными программы являются значения следующих характеристик для отдельных фрагментов горнорудной массы:

- *Square* – площадь камня, m^2 ;
- *Center of gravity* – координаты $[x, y, z]$ центра масс камня на поверхности камня, м;
- *Highest point* – координата z наивысшей точки на камне, м;
- *Volume* – объем камня, m^3 ;
- *Flat areas found* – количество найденных плоских площадок на камне,