

Agisoft



Metashape

Полярный Николай
polarnick@agisoft.com

Metashape

Основная задача:

По множеству фотографий восстановить трехмерную модель.



DJI_0127



DJI_0128



DJI_0129



DJI_0130



DJI_0131



DJI_0132



DJI_0133



DJI_0134



DJI_0135



DJI_0136



DJI_0137



DJI_0138



DJI_0139



DJI_0140



DJI_0141



DJI_0142



DJI_0143



DJI_0144



DJI_0145



DJI_0146



DJI_0147

Metashape

Основная задача:

По множеству фотографий восстановить трехмерную модель.



Данные предоставил Stéphane Prodent

Предложенные задачи

1. Реконструкция 3D-линий
2. Равномерная нарезка 3D модели на части

1. Реконструкция 3D-линий

Хочется по множеству фотографий сцены при известных калибровочных параметрах камер и их положений найти 3D линии:



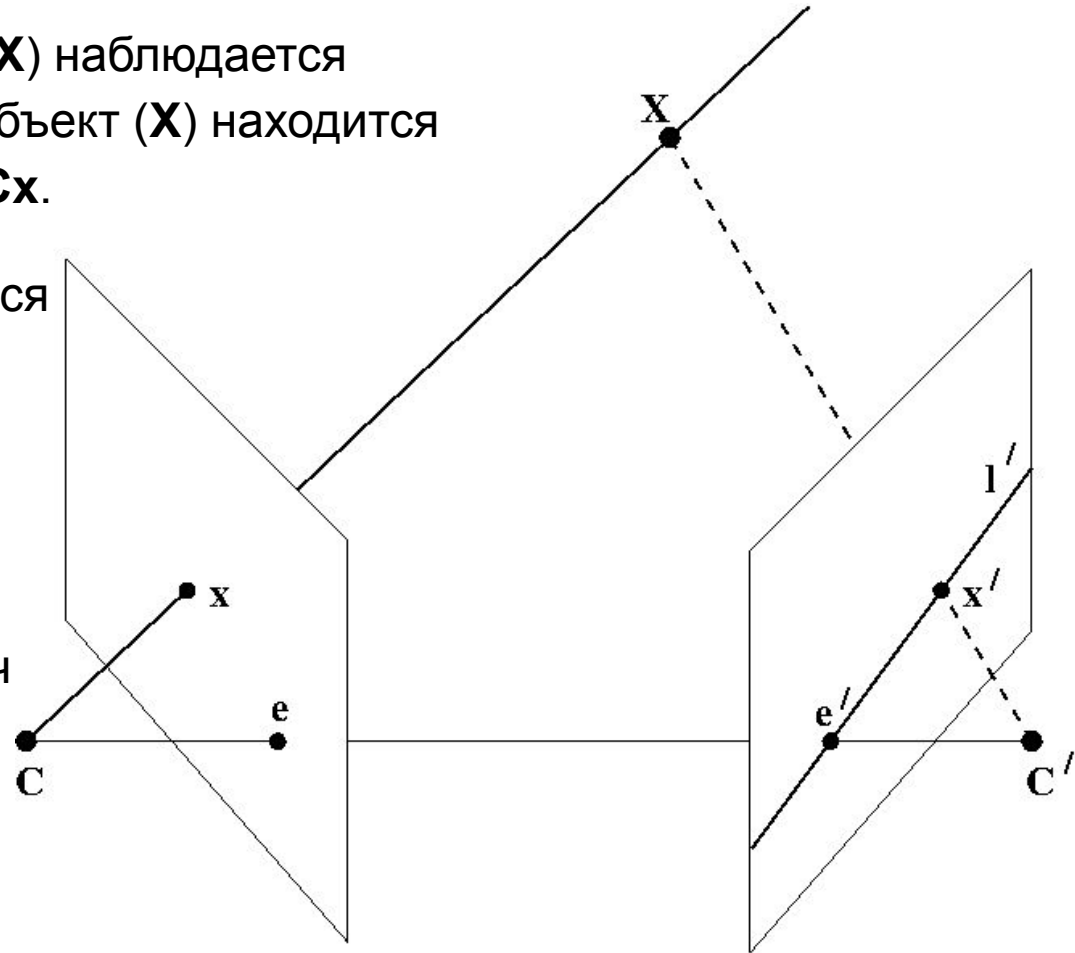
Иллюстрация - результат самой близкой к задаче статьи [1]

1. Введение: эпиполярная геометрия

Если на левой картинке что-то (X) наблюдается в пикселе x , то наблюдаемый объект (X) находится в пространстве где-то на луче Cx .

А значит этот объект наблюдается на правой картинке где-то на прямой-проекции луча Cx (т.н. **эпиполярной линии**).

Но если объект интереса - отрезок, то он порождает не луч а плоскость.



1. Введение: идея про distance transform

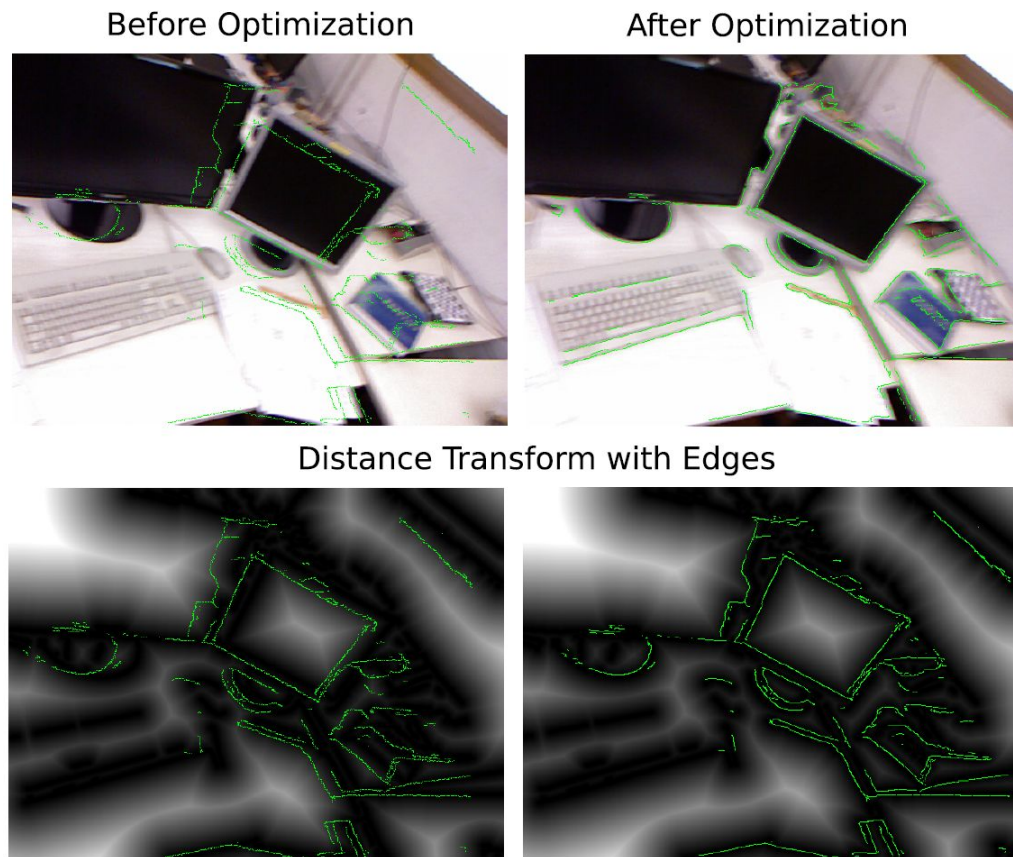
В **RESLAM** сдвиг и поворот от кадра к кадру ищется как задача оптимизации “**границы должны переходить в границы**”.

С отдельными границами на прошлом и новом кадре работать сложно.

Поэтому в новом кадре строится карта минимального расстояния до ближайшей границы.

И т.о. задача сводится к “**границы должны переходить в пиксели данной карты с малыми значениями**”.

А это считается быстро и всегда легко посчитать производные.



1. Реконструкция 3D-линий. Предлагается:

Детектировать на всех фотографиях границы и преподсчитать по ним **Distance Transform** карты.

Затем для каждой из фотографий **C1** определить 3D положение каждой ее границы **E1**:

- 1) Взять три ближайшие фотографии **C2, C3, C4**.
- 2) Выбрать на текущей границе **E1** случайную точку **x1** - она породит на вспомогательной фотографии экиполярную линию.
- 3) Эта экиполярная линия пересекает несколько границ, т.е. проходит через несколько минимумов в **Distance Transform** карте.
- 4) Перебираем каждый из этих минимумов:
- 5) Текущий минимум и изначальная точка **x1** пересекаются в пространстве, и т.о. мы можем посчитав проекцию в камеры **C3, C4** проверить - поддерживают ли они эту границу, и в том же ли направлении.
- 6) Если мы угадали с минимумом и нашу границу поддерживают вся четверка камер - нашли 3D-линию, иначе вновь пробуем с шага 2.

1. Реконструкция 3D-линий. Ссылки:

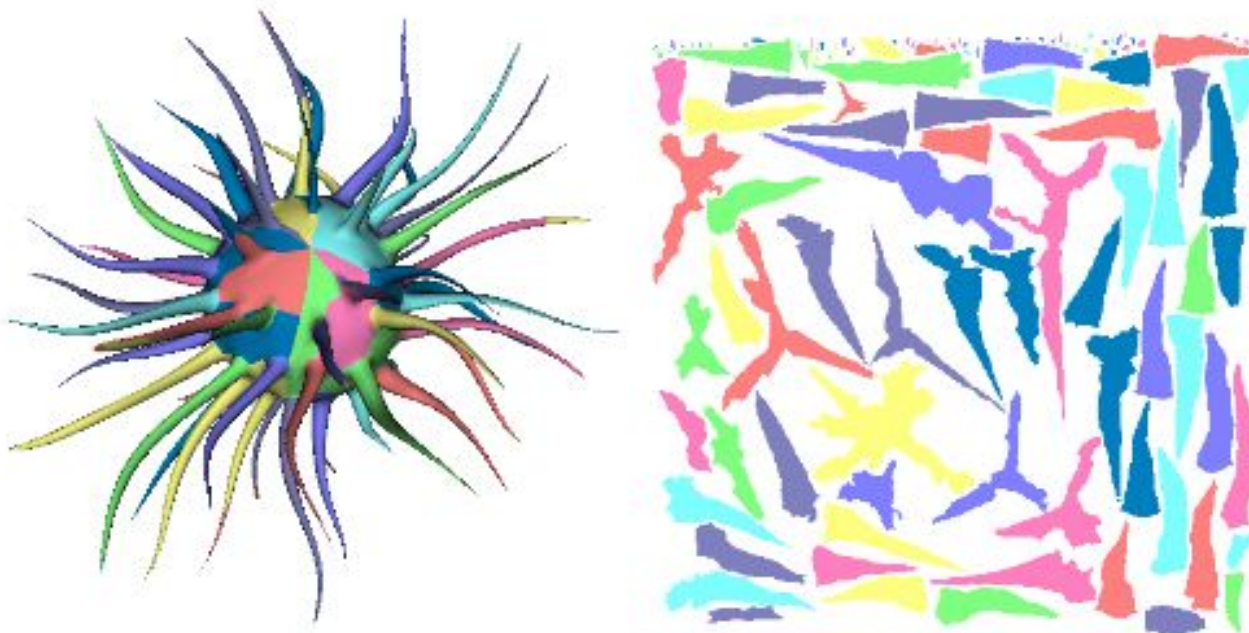
[1] [Semi-Global 3D Line Modeling for Incremental Structure-from-Motion](#), **Hofer, Donoser, Bischof, 2014**

[2] [Robust Edge-based Visual Odometry using Machine-Learned Edges](#), **Schenk, Fraundorfer, 2017**

2. Введение: текстурный атлас

Для 3D-моделей текстура хранится в виде картинки.

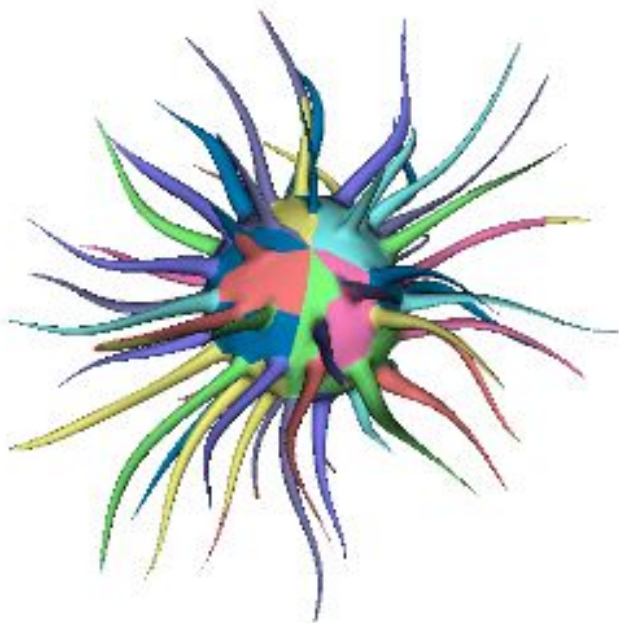
Но как нарезать модель на кусочки (чарты) так чтобы они с одной стороны были достаточно большими ради эффективного использования пространства текстурного атласа (текстурки-картинки), а с другой стороны не были излишне сильно растянуты или наоборот зажаты?



2. Введение: UVAtlas

Существует open-source реализация раскладки 3D-модели в текстурый атлас: [Microsoft/UVAtlas](#)

Но на моделях детальнее ста тысяч треугольников алгоритм работает слишком медленно - десятки минут. И при этом он однопоточный.



2. Равномерная нарезка 3D модели на части

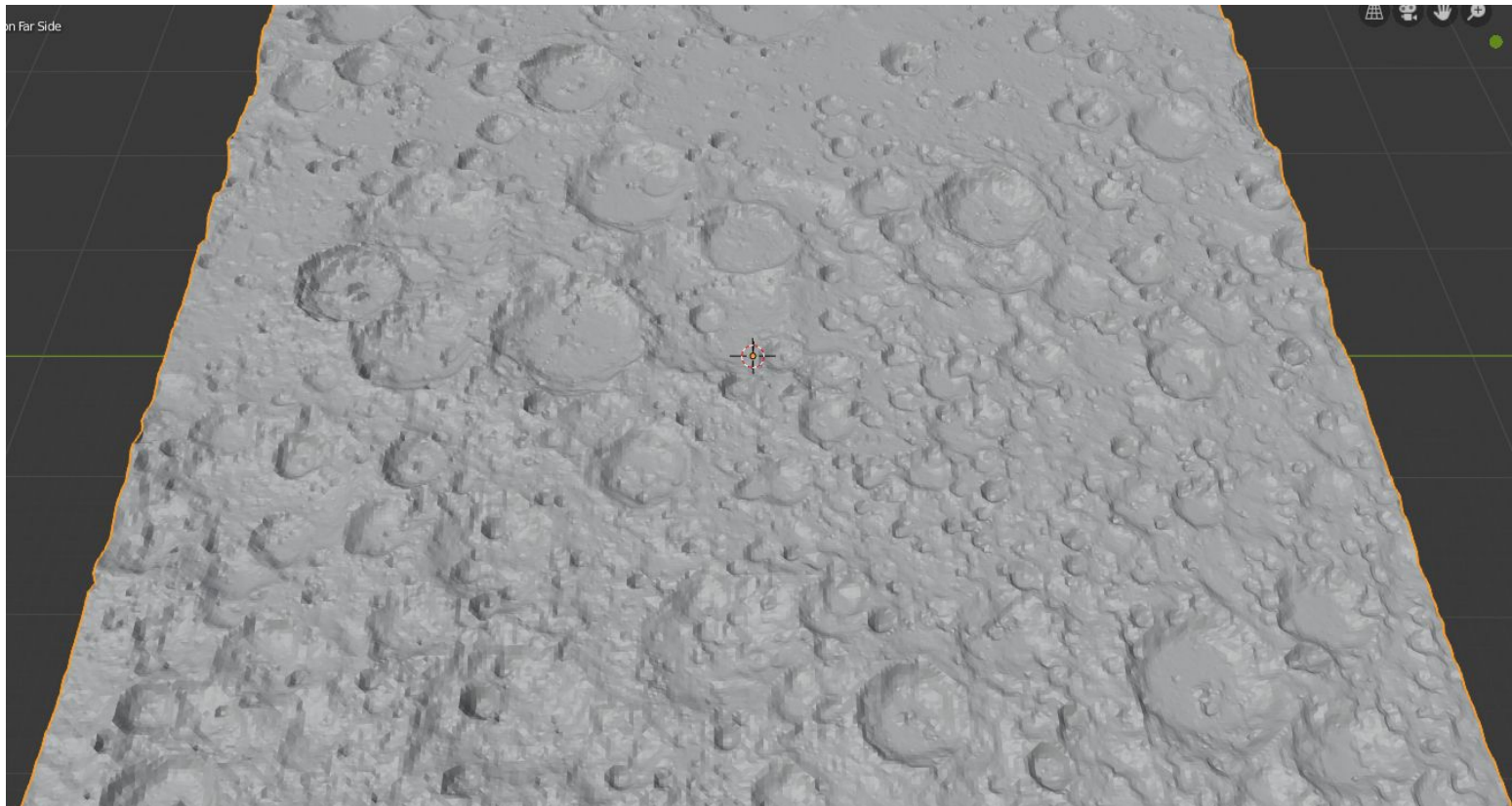
Предлагается внести параллелизм на уровне входных данных:

- 1) Нарезать 3D модель на примерно равные (по числу треугольников) части
- 2) Раскладывать их в разные атласы параллельно и независимо (и соответственно получить идеальное ускорение благодаря многопоточности)

2. Равномерная нарезка 3D модели на части

Если делать “как-то”, то возникнут проблемы:

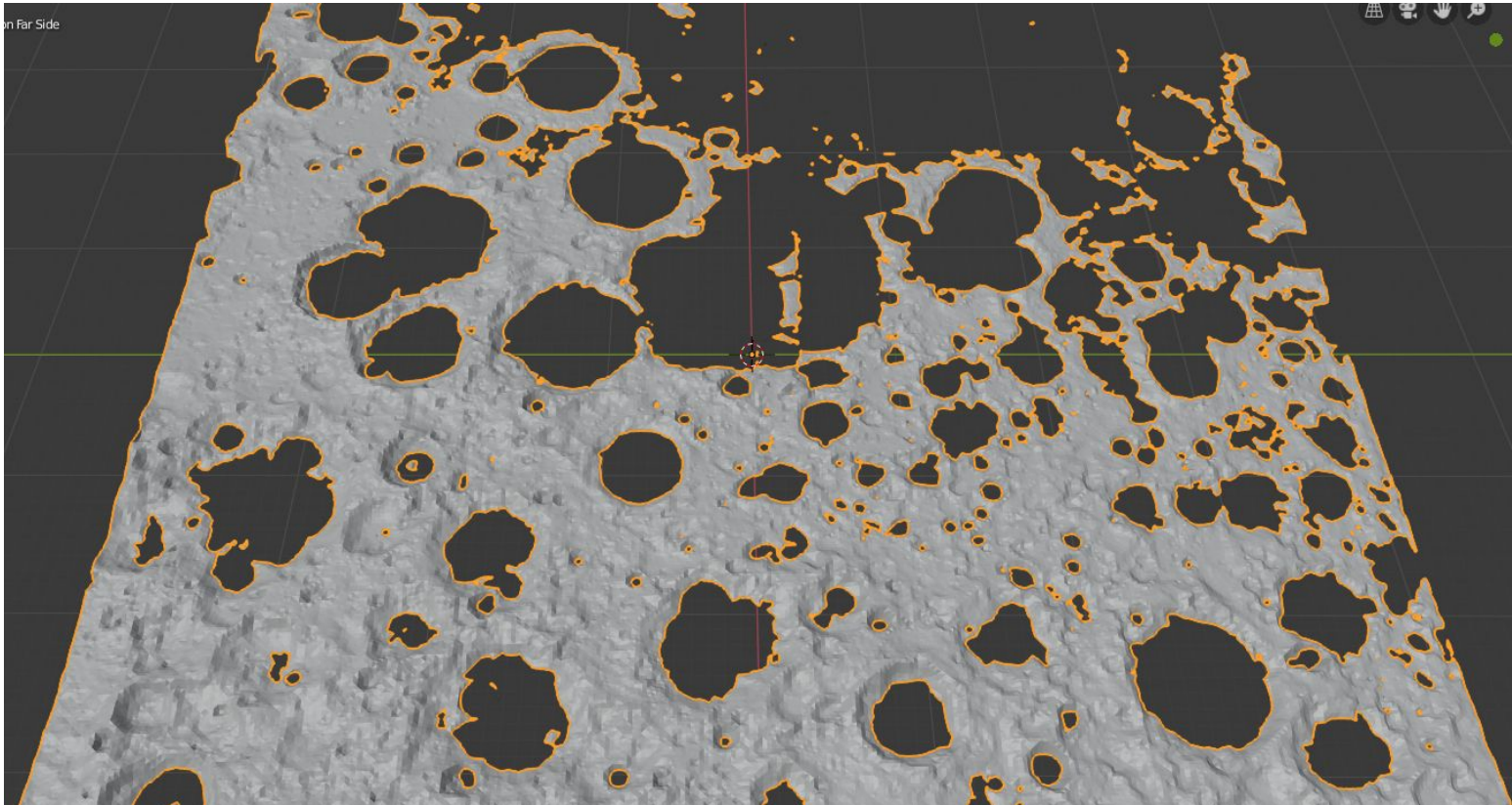
- 1) Почти идеальная для раскладки ровная поверхность может развалиться и попасть в разные части:



2. Равномерная нарезка 3D модели на части

Если делать “как-то”, то возникнут проблемы:

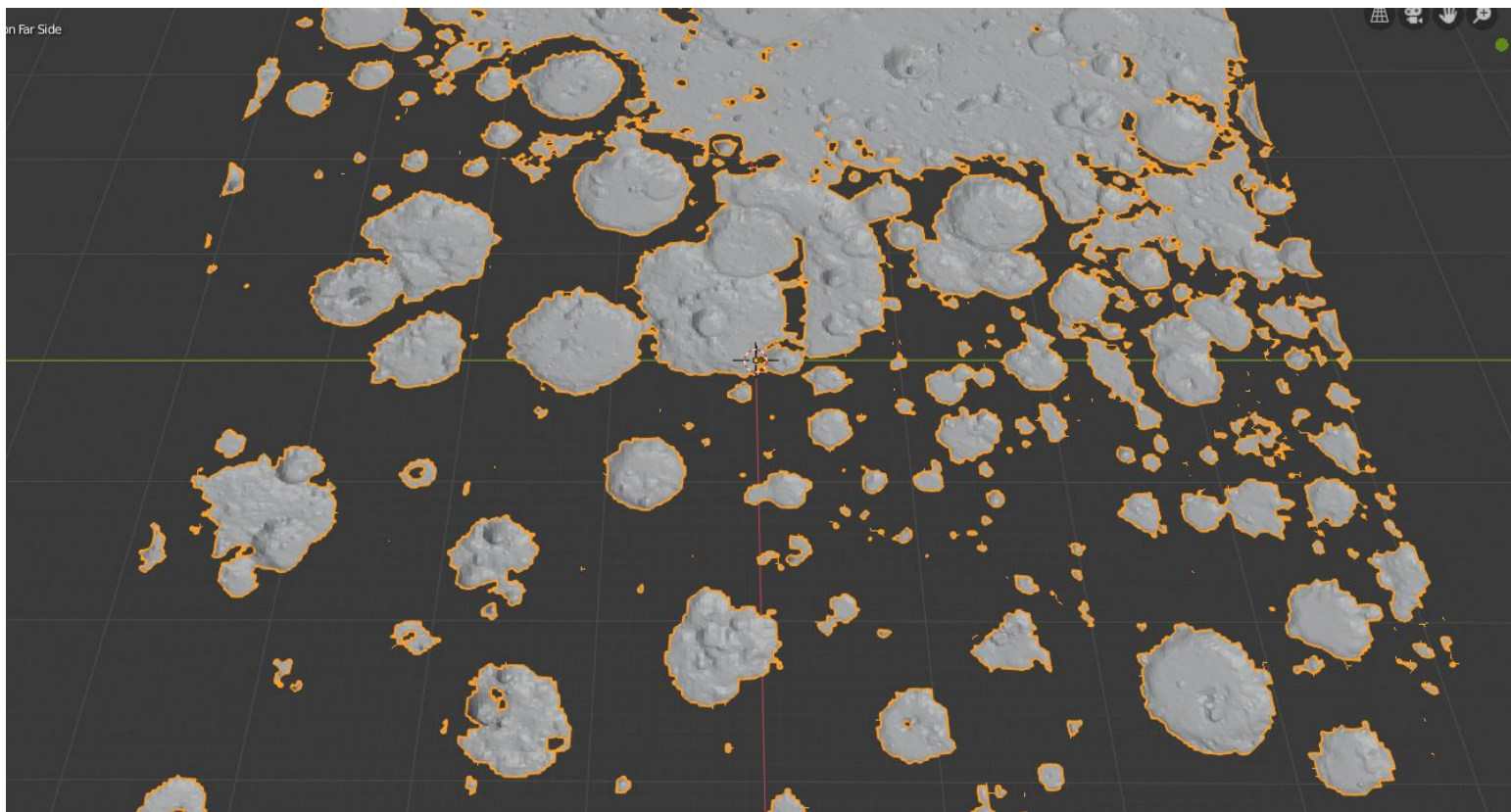
- 1) Почти идеальная для раскладки ровная поверхность может развалиться и попасть в разные части:



2. Равномерная нарезка 3D модели на части

Если делать “как-то”, то возникнут проблемы:

- 1) Почти идеальная для раскладки ровная поверхность может развалиться и попасть в разные части:



2. Равномерная нарезка 3D модели на части

Если делать “как-то”, то возникнут проблемы:

- 1) Почти идеальная для раскладки ровная поверхность может развалиться и попасть в разные части.

Поэтому хочется угадать лучший угол и место рассечения перебором. (чем меньше треугольников разрезалось - тем лучше)

2. Равномерная нарезка 3D модели на части

Если делать “как-то”, то возникнут проблемы:

- 1) Почти идеальная для раскладки ровная поверхность может развалиться и попасть в разные части.

Поэтому хочется угадать лучший угол и место рассеечения перебором. (чем меньше треугольников разрезалось - тем лучше)

- 2) В сложных случаях идеального рассеечения плоскостью не существует. Но найденное рассеечение можно улучшить еще - **приклеив обратно отвалившиеся небольшие “кратеры”** если добавленное т.о. число треугольников не сместит равномерность деления слишком сильно.

2. Равномерная нарезка 3D модели на части

Задача не зависит от самой раскладки на чарты, а является очень удобным инструментом предобработки.

Подход так же делает чарты в объединенном атласе более локальными в 3D пространстве. А это полезно для многих алгоритмов (например для алгоритма построения текстуры).

В самом **UVAtlas** много чего хотелось бы улучшить и в целом задача раскладки в атлас является сложной и на данный момент не решенной до конца.

Организационные детали

- Язык: **C++**
- Адрес офиса: [Дегтярный Переулок, 11 лит. Б](#)
- С любыми вопросами можно писать на polarnick@agisoft.com или <http://t.me/PolarNick239>
- К предложенным темам есть тестовые задания
- Во всех практиках нужно будет читать статьи

Вопросы?



Agisoft

Полярный Николай

polarnick@agisoft.com