**(Слайд 1) Титульник**

Добрый день, уважаемые члены комиссии, меня зовут Поздняков Артемий Анатольевич и я хочу представить выпускную квалификационную работу бакалавра на тему ”Сравнение методов сжатия атрибутов облаков точек”. Научным руководителем данной работы является Федоров Станислав Алексеевич.

**(Слайд 2) Актуальность**

Растущая популярность технологий компьютерного зрения и расширенной реальности влечёт за собой потребность в способах компактного хранения и передачи облаков точек. В настоящее время появляется большое количество кодеков, предназначенных для сжатия облаков точек и их атрибутов (PCC-кодеков), что делает актуальной задачей разработку программы для оценки работы PCC-кодеков. Подобная программа может быть использована исследователями для подсчёта метрик разрабатываемых ими кодеков.

**(Слайд 3) Цели и задачи**

**Цель работы** - разработка подхода к сравнению методов сжатия атрибутов облаков точек. В рамках данной работы необходимо решить следующие задачи:

- Проанализировать системы оценки качества сжатия облаков точек
- Изучить релевантные метрики, отображающие эффективность и качество сжатия атрибутов облаков точек
- Разработать программу подсчёта метрик
- Получить метрики для отобранных PCC-кодеков
- Проанализировать результаты работы

**(Слайд 4) Более расширенное введение, классификация метрик**

Сравнение кодеков будет осуществляться следующим образом: возьмем некоторое контрольное облако точек, осуществим его компрессию и декомпрессию с помощью некоторого кодека $C$, полученное в результате декомпрессии облако точек назовём реконструированным облаком точек. Показателем качества сжатия данного облака точек, то есть качественной оценкой работы кодека, будут являться значения отобранных метрик для пары оригинальное и реконструированное облако.

**(Слайд 5) Сравнение указанных альтернативных решений**

Среди существующих решений, предназначенных для данной задачи, можно отметить системы оценки mpeg_pcc_dmetric от MPEG и geo_dist от авторов кодека GeoCNNv1.

| | mpeg_pcc_dmetric | geo_dist |
|---|---|---|
| Полнота метрик | +- | +- |
| Оценка искажения атрибутов | + | - |
| Поддерживаемость | + | - |
| Возможность расширения | - | - |
| Открытый исх. код | - | - |

Таблица 1: Характеристики различных рассмотренных систем.

На таблице 1 приведены характеристики данных систем оценки. Здесь, под полнотой метрик подразумевается возможность подсчёта среднеквадратичной ошибки, отношения пикового сигнала к шуму, а также значения данных метрик, проецированные вдоль нормалей точек (нормаль точки - нормаль к плоскости, на которой лежит точка). Что касается открытости исходного кода, решение от MPEG предоставляется лишь исследователям по специальному запросу, а программа geo_dist

опубликована на Github, но не обладает лицензией, что формально не даёт возможности данный код использовать.

**(Слайд 6) Требования, выведенные в рез-те проведения анализа**

- Возможность вычисления стандартных метрик искажения геом. структуры (MSE и PSNR, метрика Хаусдорфа) геометрической структуры;
- Возможность вычисления проецированных значений отклонения;
- Возможность вычисления искажения цветов в цветовых схемах RGB и Y'CbCr;
- Использование архитектуры, допускающей дальнейшее расширение приложения;
- Наличие тестов;
- Использование лицензии MIT;

**(Слайд 7) Архитектура разрабатываемого решения**

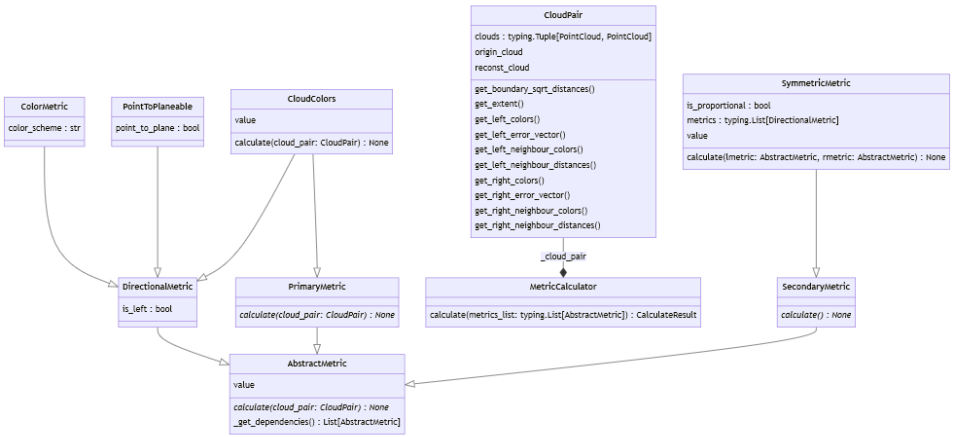



Рис. 1: Диаграмма классов разработанного приложения

Для реализации данного проекта использовался язык Python. Диаграмма классов разработанного приложения приведена на рисунке 1. Для добавления новой метрики достаточно реализовать класс, унаследованный от класса AbstractMetric, и реализовать в нём методы `calculate` и `_get_dependencies`. `calculate` реализует вычисление, а `_get_dependencies` сообщает о зависимостях, необходимых для вычисления. Например, для PSNR зависимостью будет MSE.

**(Слайд 8) Алгоритм внедрения зависимостей**

Листинг 1: Алгоритм подсчёта метрик.

```
if metric._key() in self._calculated_metrics:
    return self._calculated_metrics[metric._key()]

if isinstance(metric, PrimaryMetric):
    metric = typing.cast(PrimaryMetric, metric)
    metric.calculate(self._cloud_pair)
```

```
    self._calculated_metrics[metric._key()] = metric
    return metric

calculated_deps = {}
for dep_key, dep_metric in metric._get_dependencies().items():
    calculated_dep_metric = self._metric_recursive_calculate(
        metric=dep_metric,
    )
    calculated_deps[dep_key] = calculated_dep_metric

metric.calculate(**calculated_deps)
self._calculated_metrics[metric._key()] = metric
```

На листинге 1 приведен алгоритм внедрения зависимостей, позволяющий избежать повторного вычисления уже вычисленных метрик. Данный алгоритм реализован в методе `recursive_calculate` класса MetricCalculator.

**(Слайд 9) Консольное приложение (help)**

```
● open-pcc-metric-py3.10vscode ➜/workspaces/open-pcc-metric (main) $ python3 \
  > -m open_pcc_metric --help
  Usage: python -m open_pcc_metric [OPTIONS]

  Options:
    --ocloud TEXT      Original point cloud.  [required]
    --pcloud TEXT      Processed point cloud.  [required]
    --color [rgb|ycc]  Report color distortions as well.
    --hausdorff        Report hausdorff metric as well. If --point-to-plane is
                       provided, then hausdorff point-to-plane would be reported
                       too
    --point-to-plane   Report point-to-plane distance as well.
    --csv              Print output in csv format.
    --help             Show this message and exit.
```

Рис. 2: Help-сообщение программы

Разработанное решение представляет собой консольное приложение. Help-сообщение программы и входные параметры приведены на рисунке 2. По умолчанию программа выводит MSE и PSNR для координат точек, а также для цветов в цветовой схеме RGB, клиент дополнительно может указать программе вычислить метрику Хаусдорфа, значения метрик, проецированные вдоль нормалей (для MSE, PSNR координат и метрики Хаусдорфа), а также указать цветовую схему, в которой должно вычисляться искажение цветов. Дополнительно поддерживается вывод в формате CSV, что может быть использовано при машинной обработке результатов.

**(Слайд 10) Консольное приложение (вывод)**

```
● open-pcc-metric-py3.10vscode ➜/workspaces/open-pcc-metric (unittests) $ python3 -m open_pcc_metric \
  > --ocloud="./files/oskull_reduced.ply" \
  > --pcloud="./files/pskull_reduced.ply" \
  > --color="ycc"
                 label is_left point-to-plane                                    value
0        MinSqrtDistance                                          0.0005394594300728226
1        MaxSqrtDistance                                             3.3819776943374595
2                 GeoMSE    True          False                   0.008200834632373888
3                 GeoMSE   False          False                   0.008181548987474831
4      GeoMSE(symmetric)                                          0.008200834632373888
5                GeoPSNR    True          False                      72.4728937253675
6                GeoPSNR   False          False                     72.48311891997054
7      GeoPSNR(symmetric)                                           72.4728937253675
8                ColorMSE    True  [1.26237366e-05 2.97011418e-07 2.35043354e-07]
9                ColorMSE   False  [1.86551090e-05 4.75979959e-07 3.83567251e-07]
10   ColorMSE(symmetric)            [1.86551090e-05 4.75979959e-07 3.83567251e-07]
11               ColorPSNR    True          [48.98812076 65.27226855 66.28852024]
12               ColorPSNR   False          [47.29202209 63.22411332 64.1615848 ]
13   ColorPSNR(symmetric)           [47.29202209 63.22411332 64.1615848 ]
```

Рис. 3: Пример вывода программы

Пример работы программы приведен на рисунке 3, здесь отображены значения MSE и PSNR для координат и цветов, а также минимальное и минимальное расстояние между парами точек в оригинальном и реконструированном облаке.

**(Слайд 11) Метрики разработанного ПО**

Тут про объем реализации (1к строк кода), тесты, CI/CD, много картинок, пару красивых слов.

**(Слайд 12) Введение про PCCArena**

Разработанное решение было внедрено в платформу PCCArena. PCCArena представляет собой систему бенчмаркинга PCC-кодеков. Данная система использует mpeg_pcc_dmetric для вычисления математически-обоснованные метрик (MSE, PSNR, и т.д.).

**(Слайд 13) Архитектура PCCArena**

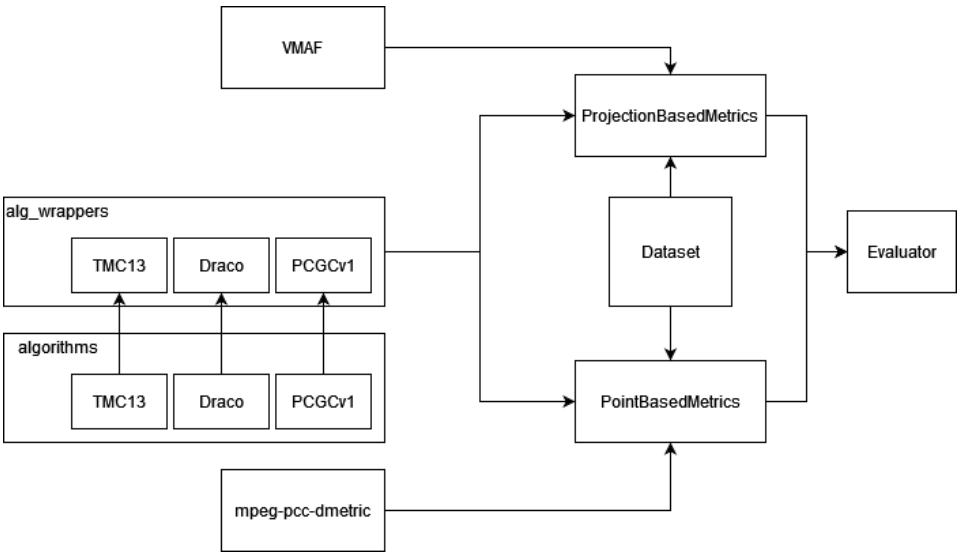


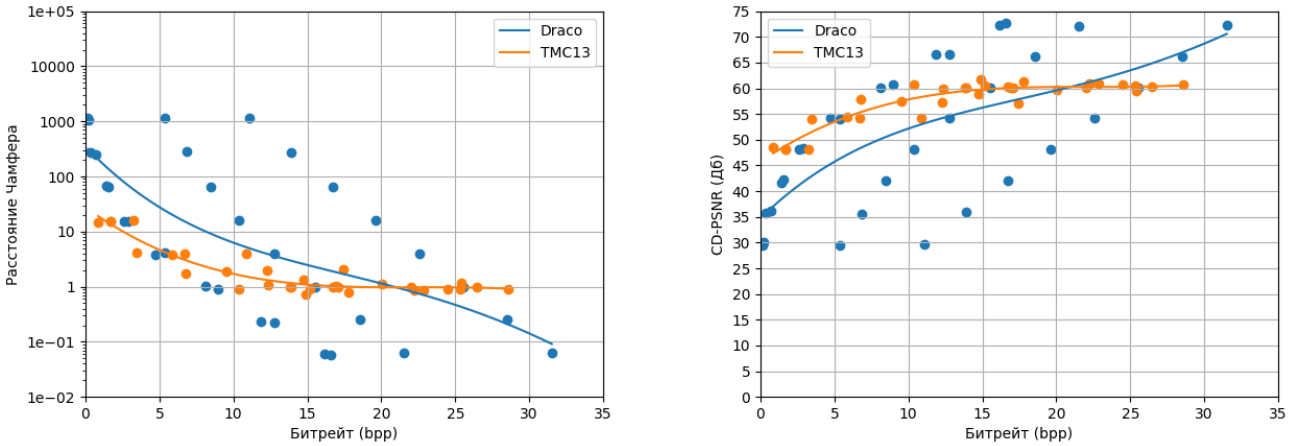

Рис. 4: Архитектура PCCArena

Архитектура PCCArena приведена на рисунке 4. Для внедрения разработанного решения в данную систему были внесены изменения в класс PointBasedMetrics.

**(Слайд 14) Описание проведенных экспериментов**

С помощью модифицированной системы PCCArena была произведена оценка кодеков TMC13 и Draco, использовался датасет ShapeNet. Для каждой метрики строится зависимость от битрейта - количества бит, затраченных на кодирование одной точки.

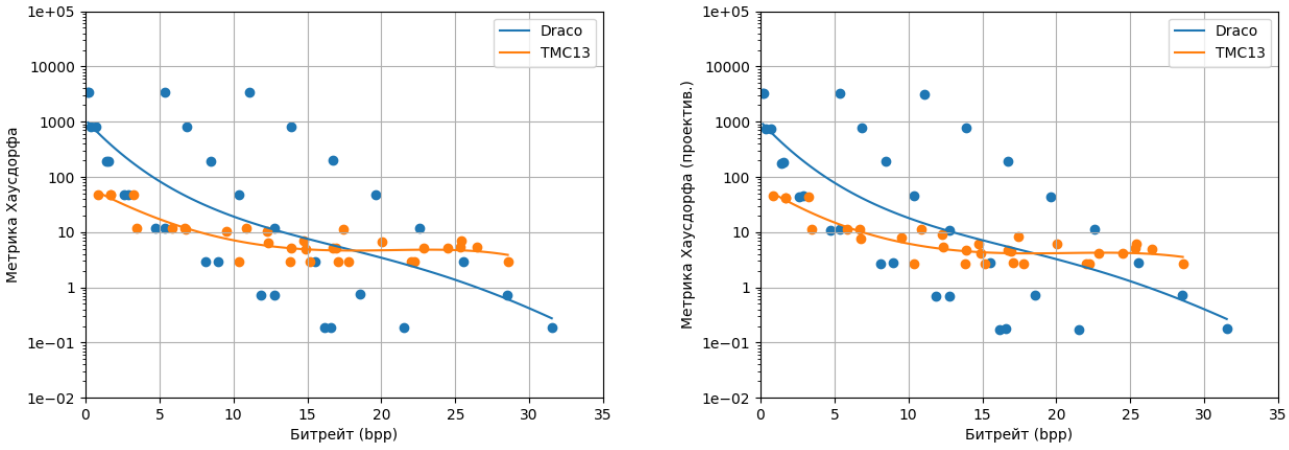
**(Слайд 15) Результаты для расстояния Чамфера**




(a) (b)

Рис. 5: (a) Зависимость расстояния Чамфера от битрейта. (b) Зависимость CD-PSNR от битрейта.

**(Слайд 16) Результаты для метрики Хаусдорфа и нормалей**




(a) (b)

Рис. 6: (a) Зависимость метрики Хаусдорфа от битрейта. (b) Зависимость проецированной метрики Хаусдорфа от битрейта.

**(Слайд 17) Результаты для цветов**

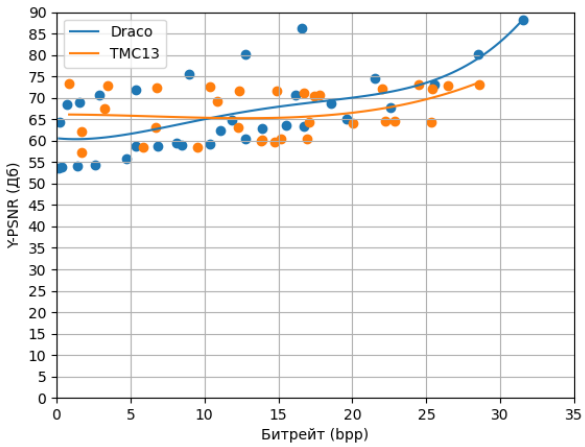



Рис. 7: Зависимость Y’-PSNR от битрейта.

**(Слайд 18) Выводы и дальнейшие шаги**

В данной работе был проведён анализ существующих систем оценки методов сжатия облаков точек и их атрибутов. Разработана программа для оценки качества облака точек при наличии оригинального облака точек. Произведён сравнительный анализ кодеков Draco и TMC13. Полученные данные позволяют судить о качестве сжатия облаков точек данными кодеками при различной степени сжатия. Разработанное решение упростит оценку методов сжатия атрибутов облаков точек и может быть полезным исследователям, ведущим разработки в данной области.

В рамках дальнейшей работы в программу могут быть добавлены метрики, учитывающие более высокоуровневые признаки облаков точек и дающие более подробную оценку качества их сжатия.

## Extra. Метрики

Метрика - мера, значение, полученное в результате измерения.
Метрика - ф-я, удовл. аксиомам тождества, симметричности и нер-ву треугольника.