



Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа программной инженерии

Сравнение методов сжатия атрибутов облаков точек

Выполнил студент: **Поздняков Артемий Анатольевич**, гр. 5130904/00104

Научный руководитель: старший преподаватель ВШПИ ИКНК **Фёдоров Станислав Алексеевич**

Направление подготовки: 09.03.04, **Программная инженерия**

Санкт-Петербург

2024

Разработка программы для оценки работы РСС-кодеков - актуальная задача.

- Популярность технологий компьютерного зрения и расширенной реальности растёт;
- Появляется большое количество кодеков, предназначенных для сжатия облаков точек и их атрибутов (РСС-кодеков);

Использование

Подобная программа может быть использована исследователями для подсчёта метрик разрабатываемых ими кодеков.

Цель работы - разработка подхода к сравнению методов сжатия атрибутов облаков точек.

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 РСС-кодеков;
- Проанализировать результаты работы;

Определение

Реконструированное облако - облако точек, полученное в результате компрессии и декомпрессии оригинального облака точек.

- Показателем качества сжатия являются значения отобранных метрик для пары оригинальное и реконструированное облако.

Сравнение альтернативных решений

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

Таблица 1: Характеристики различных рассмотренных систем

В результате анализа определены следующие требования к разрабатываемому решению:

- Возможность вычисления стандартных метрик искажения геом. структуры (MSE и PSNR, метрика Хаусдорфа);
- 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;

Архитектура разрабатываемого решения

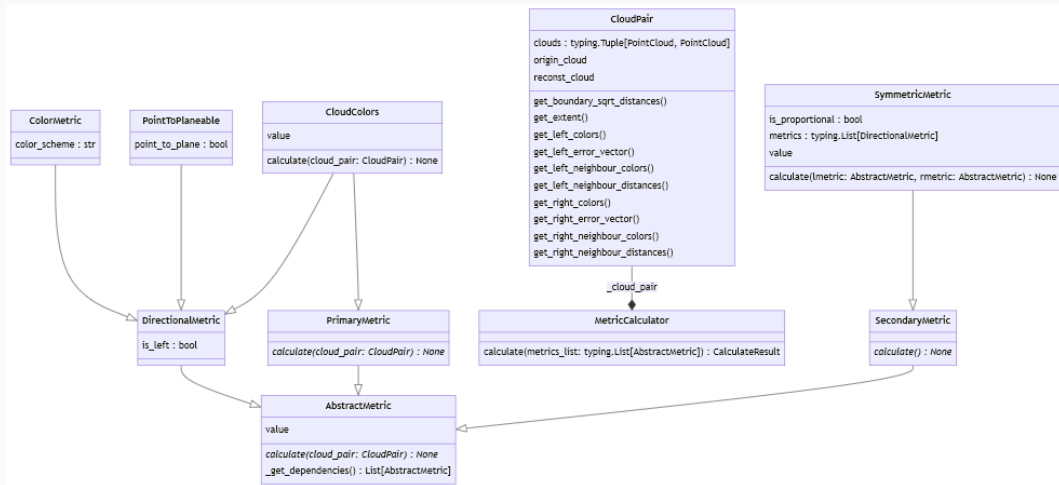


Рис. 1: Диаграмма классов разработанного приложения

Алгоритм внедрения зависимостей

```
1  if metric._key() in self._calculated_metrics:
2      return self._calculated_metrics[metric._key()]
3
4  if isinstance(metric, PrimaryMetric):
5      metric = typing.cast(PrimaryMetric, metric)
6      metric.calculate(self._cloud_pair)
7      self._calculated_metrics[metric._key()] = metric
8      return metric
9
10 calculated_deps = {}
11 for dep_key, dep_metric in metric._get_dependencies().items():
12     calculated_dep_metric = self._metric_recursive_calculate(
13         metric=dep_metric,
14     )
15     calculated_deps[dep_key] = calculated_dep_metric
16
17 metric.calculate(**calculated_deps)
18 self._calculated_metrics[metric._key()] = metric
```

Листинг 1: Алгоритм внедрения зависимостей


```
● 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

```
● 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

Детали реализации

Стек:

- Python 3.8.1
- Open3D 0.18.0
- NumPy 1.24
- click 8.1.7

Линтеры:

- pylint
- flake8

Тестирование:

- pytest

CI/CD:

- GitHub Actions

Объем реализации:

- 1206 строк

```
----- coverage: platform linux, python 3.10.14-final-0 -----
Name                                                    Stmts  Miss  Cover
-----
open_pcc_metric/__init__.py                             0      0   100%
open_pcc_metric/__main__.py                             3      3     0%
open_pcc_metric/calculator.py                          64      7    89%
open_pcc_metric/cloud_pair.py                          63      3    95%
open_pcc_metric/handler.py                             22     22     0%
open_pcc_metric/logger.py                              11      0   100%
open_pcc_metric/metric.py                             213     24    89%
open_pcc_metric/options.py                             21      0   100%
-----
TOTAL                                                    397     59    85%

===== 50 passed in 16.02s =====
```

Рис. 4: Покрытие тестами

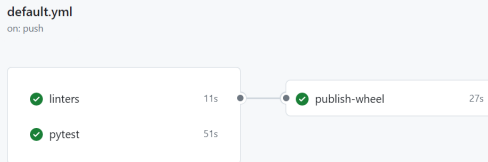


Рис. 5: Статус CI/CD

PCCArena - система
бенчмаркинга PCC-кодеков.
Данная система использует
mpeg_pcc_dmetric для
вычисления стандартных
метрик (MSE, PSNR, и т.д.).

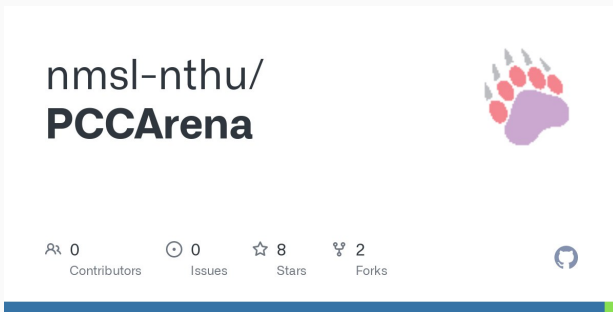


Рис. 6: Репозиторий PCCArena на GitHub

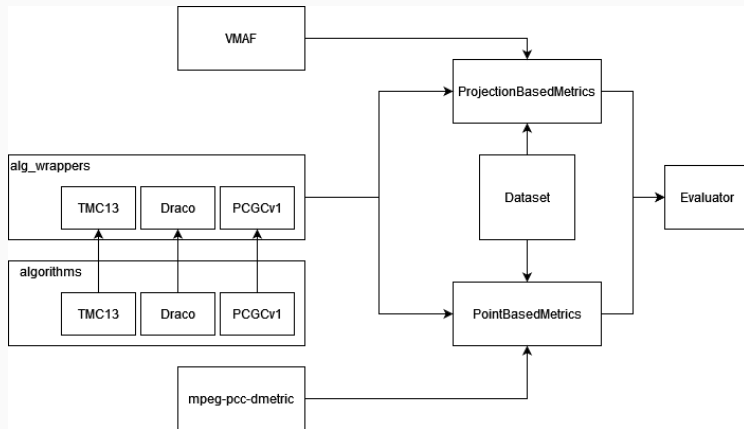


Рис. 7: Архитектура PCCArena

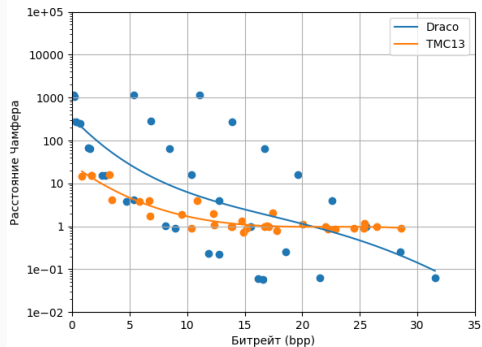
С помощью модифицированной системы PCCArena был произведен набор измерений.

- Произведена оценка кодеков TMC13 и Draco;
- Использовался датасет ShapeNet;
- Для каждой метрики была построена зависимость от битрейта;

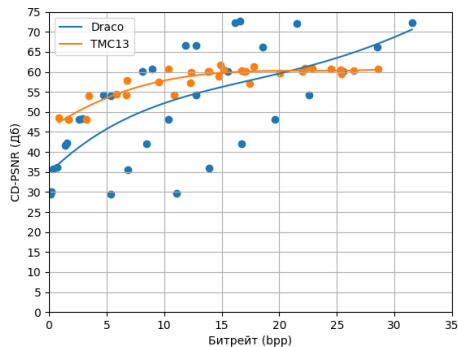
Определение

Битрейт - количество бит, затраченных на кодирование одной точки.

Результаты для расстояния Чамфера



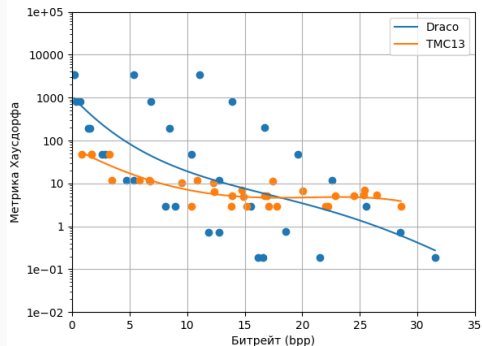
(a)



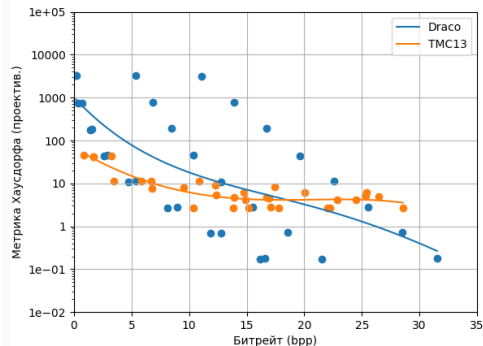
(b)

Рис. 8: (a) Зависимость расстояния Чамфера от битрейта. (b) Зависимость CD-PSNR от битрейта

Результаты для метрики Хаусдорфа и нормалей



(a)



(b)

Рис. 9: (a) Зависимость метрики Хаусдорфа от битрейта. (b) Зависимость проецированной метрики Хаусдорфа от битрейта

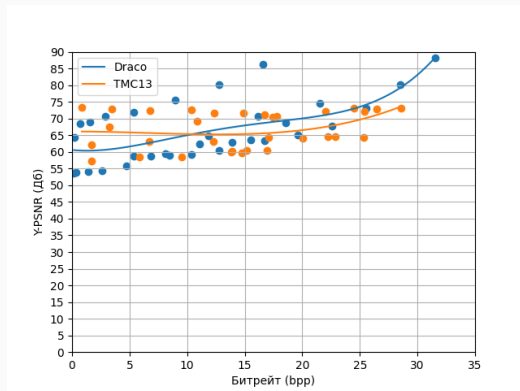


Рис. 10: Зависимость Y'-PSNR от битрейта

Выводы и дальнейшие шаги

- 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;

Разработанное решение упростит оценку методов сжатия атрибутов облаков точек и может быть полезно исследователям, ведущим разработки в данной области.

Дальнейшие шаги:

- Могут быть добавлены метрики, учитывающие более высокоуровневые признаки облаков точек и дающие более подробную оценку качества их сжатия;