



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

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Разработка алгоритма детектирования и распознавания объектов на основе спектральной панели шумометрии

по направлению 02.03.02 - Фундаментальная информатика и информационные технологии
по образовательной программе 02.03.02_02 - Информатика и компьютерные науки

Студент
Гр. 5130202/10201

Хакимуллина А.М.

Руководитель
Старший преподаватель

Прокофьев О. В.

Актуальность работы

Работа относится к задачам обработки акустических сигналов, поступающих от шумомеров, опускаемых в скважину. По измерениям строится спектральная панель, отражающая распределение шума по глубине и частоте. Эти данные важны для диагностики состояния скважины и дальнейших исследований.

Автоматизация обработки шумометрических данных необходима из-за большого объёма измерений, собираемых в реальных условиях. Разработка алгоритма автоматического детектирования и распознавания объектов с последующей генерацией отчётов позволяет ускорить анализ, снизить нагрузку на специалистов и повысить точность. Визуальный интерфейс делает работу с системой доступной и удобной для инженеров.

Анализ предметной области

Существующие методы обработки детектирования объектов в целом не демонстрируют высокой эффективности при работе с шумометрическими данными, поскольку объекты на таких панелях, как правило, не имеют чётких границ, часто вытянуты по частотной оси, обладают нестабильным уровнем фона и лишены ярко выраженных центров.

В предложенном алгоритме были адаптированы и переосмыслены отдельные подходы, использовавшиеся в других задачах обработки изображений, включая:

- медианную и гауссову фильтрацию,
- пороговую бинаризацию,
- морфологические операции,
- адаптивный подбор параметров в зависимости от структуры сигнала.

Дополнительно был реализован анализ производной частотного профиля контуров. Метод опирается на идею, аналогичную применению оператора Собеля для выделения резких переходов.

Цель и задачи

Цель — автоматизация процесса анализа данных шумометрии для уменьшения трудоемкости и повышения точности результатов.

Задачи, которые решались в ходе исследования:

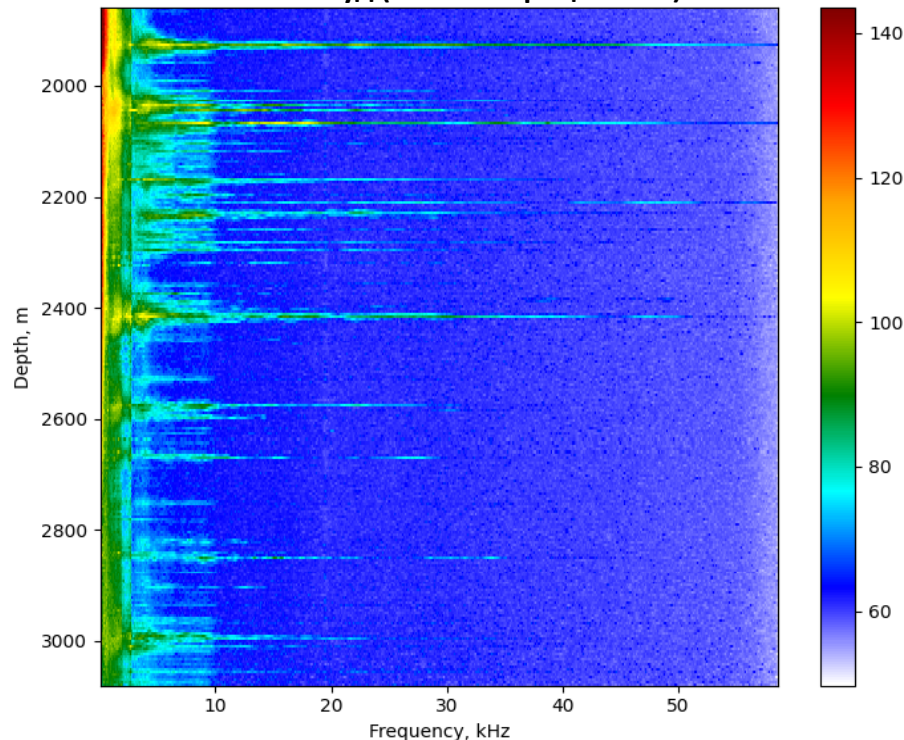
1. Анализ существующих методов детектирования объектов.
2. Разработка и реализация алгоритма детектирования и распознавания объектов по данным шумометрии.
3. Разработка графического интерфейса для визуализации и экспертной корректировки результатов.
4. Апробация программного комплекса на реальных данных.

Описание алгоритма

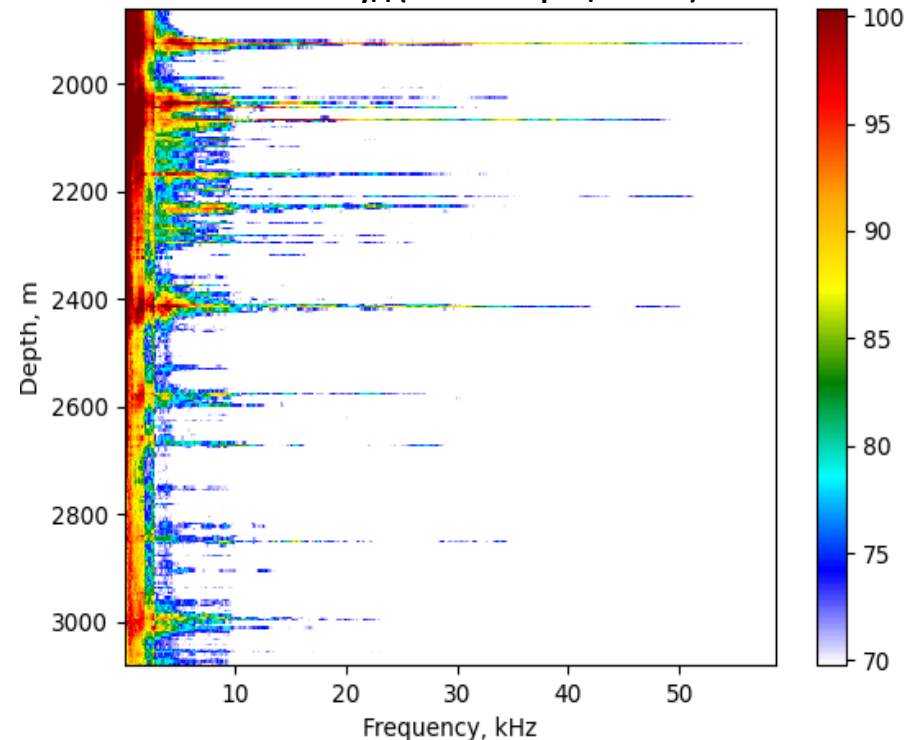
Этап 1. Инициализация параметров

Из входного LAS-файла с шумометрическими данными извлекаются глубинные координаты, значения амплитуд и тип канала (LF или HF). Параметры фильтрации инициализируются на основе характеристик входного сигнала и доступны на всех этапах обработки.

Спектральная панель в полном диапазоне амплитуд (0–100-й процентиль)



Спектральная панель в информативном диапазоне амплитуд (85–99-й процентиль)



Для визуализации используется диапазон амплитуд от 85-го до 99-го процентиля.

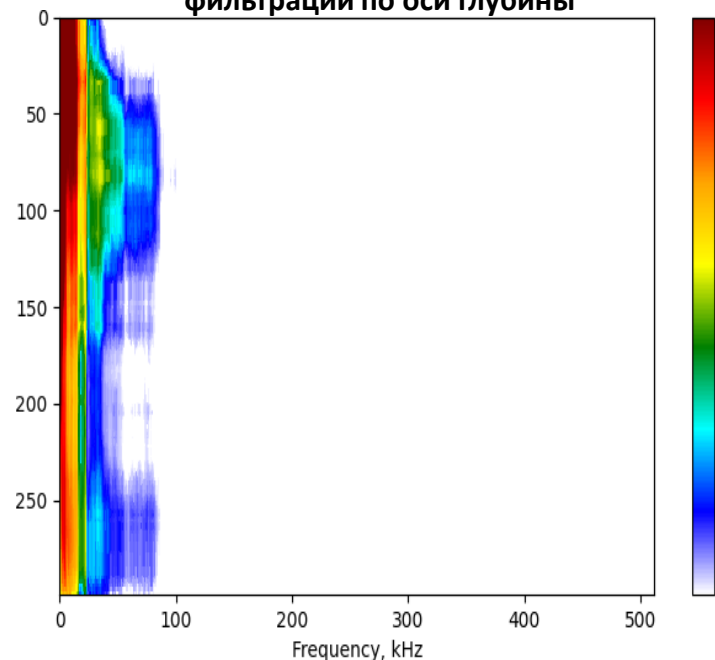
Описание алгоритма

Этап 2. Предварительная обработка данных

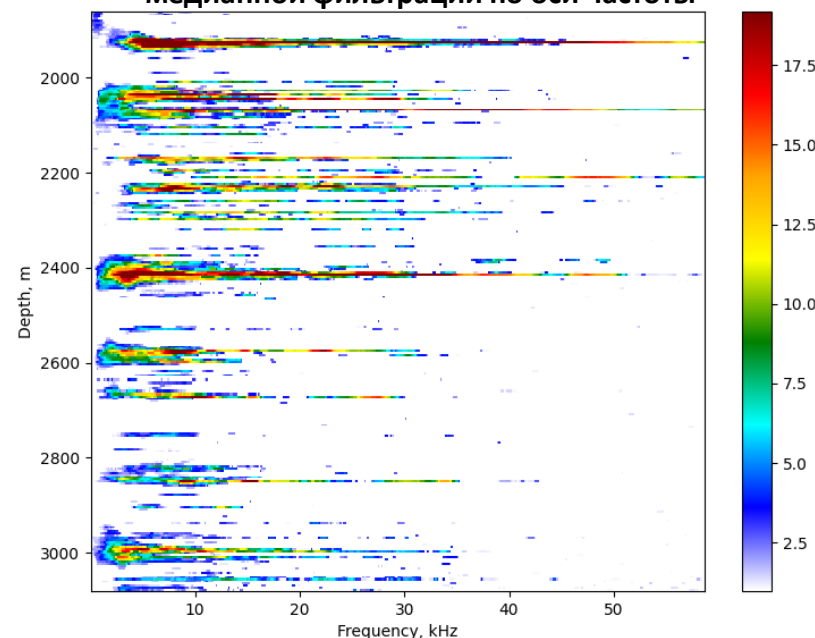
Исходная спектральная панель дополняется сверху и снизу отзеркаленными строками и сглаживается медианным фильтром по глубине. Это позволяет подавить высокочастотные шумы и подготовить данные к удалению фоновой составляющей.

Затем полученный сглаженный сигнал вычитается из исходных данных и сглаживается медианным фильтром по оси частоты. Ранее добавленные строки удаляются. Дополнительно производится сглаживание с использованием фильтра Гаусса.

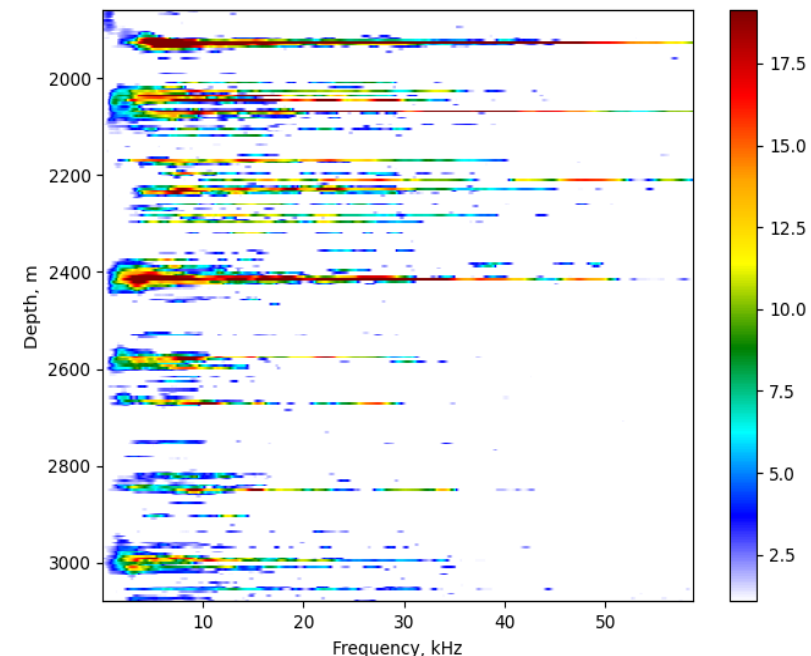
Спектральная панель после медианной фильтрации по оси глубины



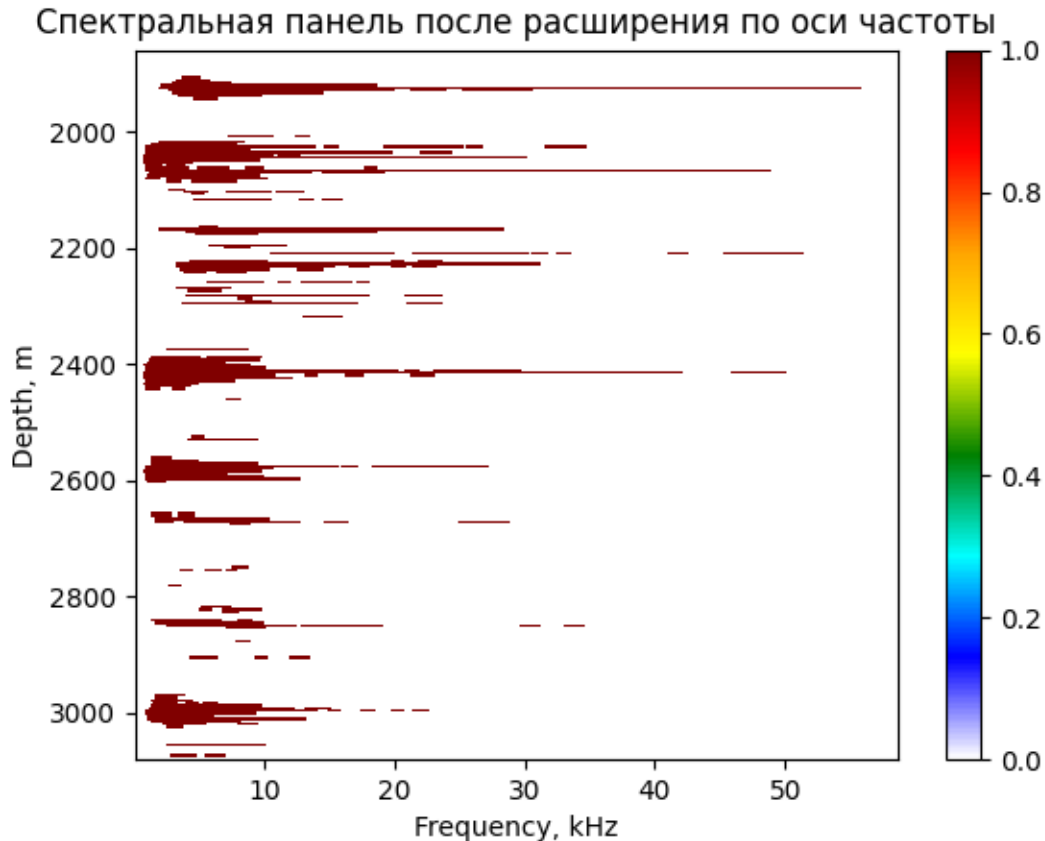
Спектральная панель после вычитания и медианной фильтрации по оси частоты



Спектральная панель после гауссова сглаживания



Описание алгоритма



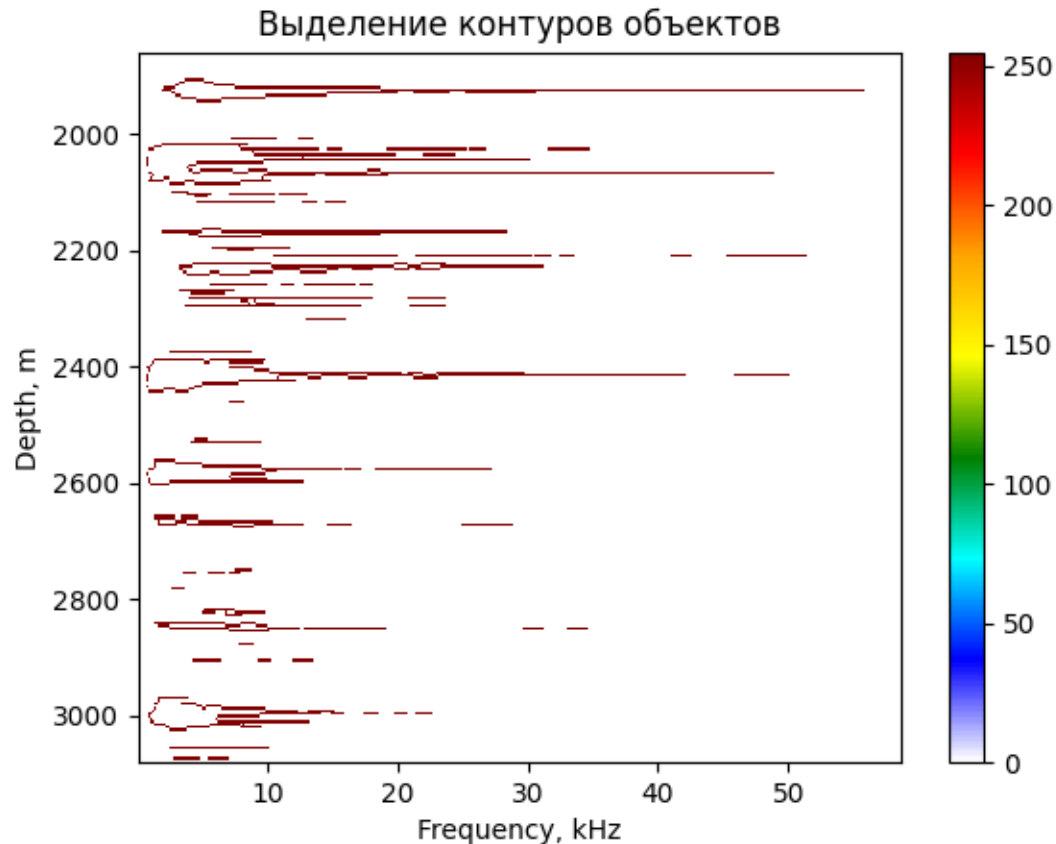
Этап 2. Предварительная обработка данных

Бинаризация осуществляется с адаптивным порогом: на основе исходных данных обнуляются те элементы, амплитуда которых в исходной панели ниже 86-го процентиля. Также обнуляются элементы, амплитуда которых меньше 4.

Затем применяются морфологические операции: эрозия устраняет малые шумы, дилатация по оси частоты помогает объединить фрагменты одного объекта.

На выходе формируется бинарная маска, готовая к извлечению контуров.

Описание алгоритма



Этап 3. Выделение контуров и формирование объектов

Из бинаризованного спектрального изображения извлекаются замкнутые области — контуры потенциальных объектов, с помощью функции `cv2.findContours`.

Анализ частотного профиля каждого контура: извлекаются локальные максимумы и минимумы, позволяющие сегментировать сложные объекты.

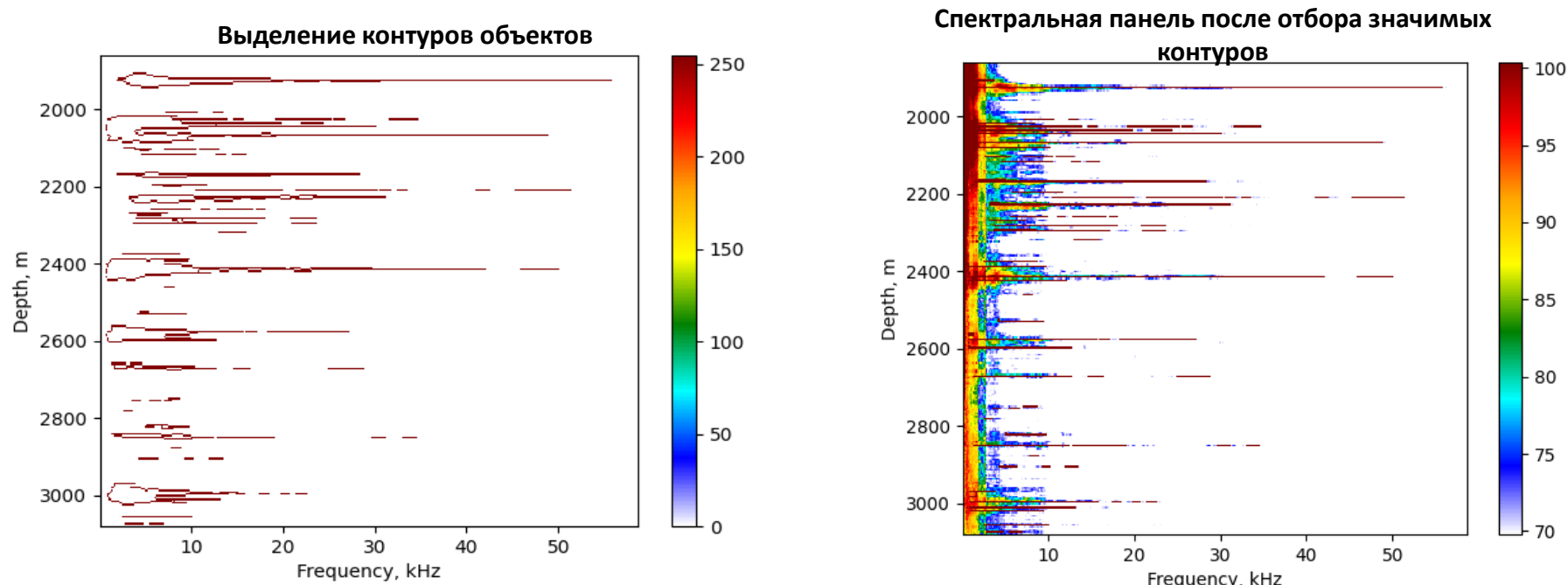
Анализируются участки с нулевой производной — так называемые плато, фиксирующие стабильную частотную характеристику объекта.

Описание алгоритма

Этап 3. Выделение контуров и формирование объектов

Определяется изолированность каждого пика: если расстояние до одного из соседних минимумов оказывается меньше $\frac{1}{4}$ частотной координаты пика, такой пик считается частью более сложной структуры и объединяется с соседними пиками в группу. В противном случае пик считается изолированным и выделяется как отдельный объект.

Каждому отобранному участку сопоставляется ограничивающий прямоугольник, фиксирующий координаты объекта в пространстве "частота – глубина".



Описание алгоритма

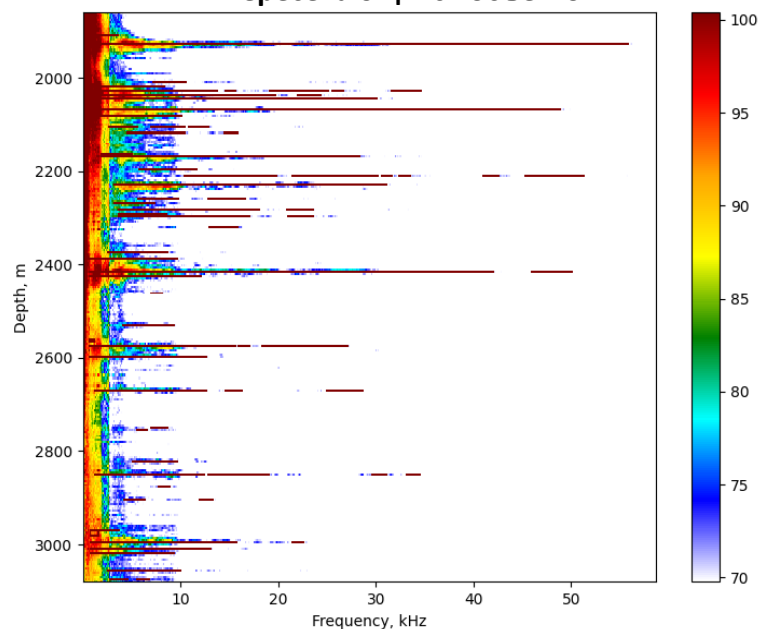
Этап 4. Агрегация объектов на основе анализа структуры и фильтрация

Удаляются объекты с малой площадью или низкой средней амплитудой сигнала, используются пороги 10 пикселей и 40-ой перцентиль соответственно.

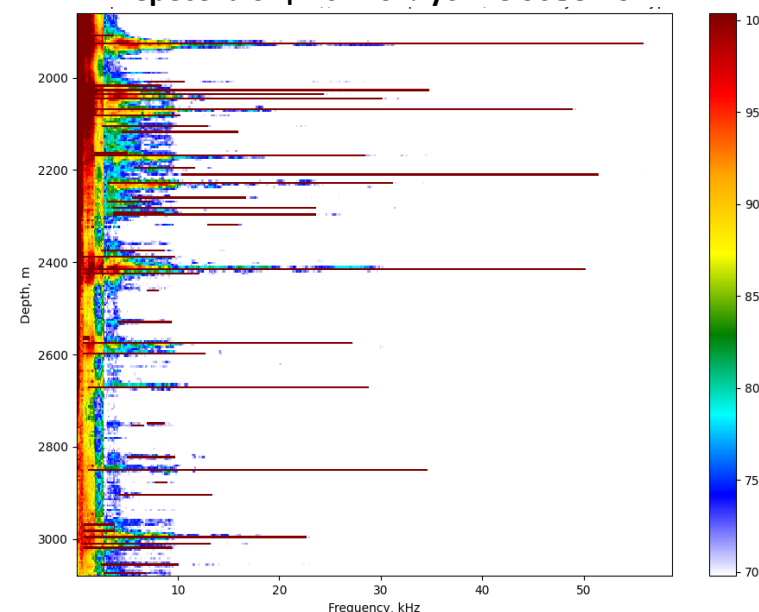
Исключаются пересекающиеся объекты: если два прямоугольника накладываются друг на друга по обеим координатам — частоте и глубине — прямоугольник с меньшей площадью удаляется.

При обнаружении взаимных пересечений по глубине реализуется объединение таких фрагментов в единый объект, охватывающий все входящие участки.

Спектральная панель после удаления
пересекающихся объектов



Спектральная панель после объединения
пересекающихся по глубине объектов



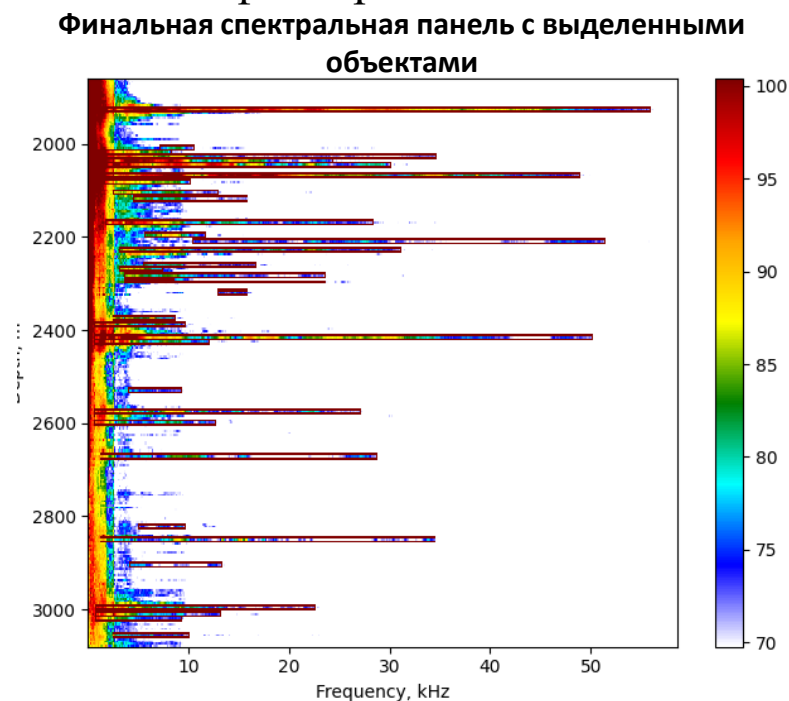
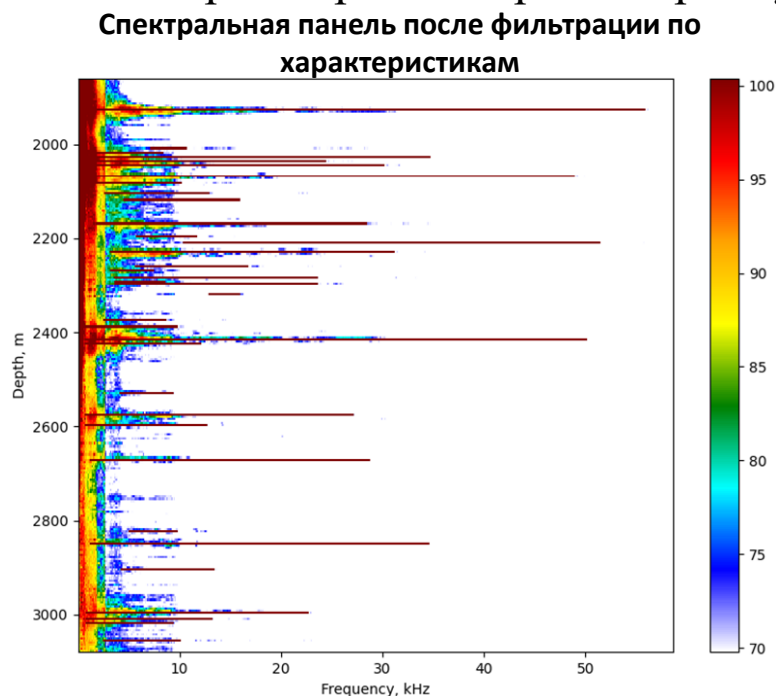
Описание алгоритма

Этап 4. Агрегация объектов на основе анализа структуры и фильтрация

Затем выполняется финальная фильтрация по характеристикам объекта:

- частотный диапазон;
- ширина объекта по частоте;
- площадь;
- средняя амплитуда сигнала.

Завершается этап расширением границ прямоугольников на фиксированное количество пикселей.



Описание алгоритма

Этап 5. Классификация объектов

Для каждого объекта рассчитываются:

- Частотная протяжённость: $\Delta f = f_{\max} - f_{\min}$;
- Глубинная протяжённость: $\Delta d = d_{\max} - d_{\min}$.

Применяются эвристические правила классификации:

1. Если $\Delta f > \Delta d$ и $f_{\min} > 0$, то поток по пласту (reservoir);
2. Если $\Delta f < \Delta d$ и $f_{\min} > 1$, то заколонная циркуляция (chanelling);
3. Иначе:
 - если $f_{\max} > 10$, то поток по пласту (reservoir);
 - если $f_{\max} \leq 10$ то буровая колонна (borhole).

Пример содержания Excel-файла

| | Кровля, м | Подшва, м | Частотный диапазон, кГц | Амплитуда, дБ | Характеристика типа шума |
|---|--------------|--------------|-------------------------------|------------------|-----------------------------|
| 1 | 3601 | 3609 | 0.1-58.5 | 101 | Поток по пласту |
| 2 | 3623 | 3627 | 0.2-58.6 | 111 | Поток по пласту |
| 3 | 3827 | 3831 | 0.3-57.7 | 99 | Поток по пласту |
| 4 | 3841 | 3845 | 0.3-58.6 | 104 | Поток по пласту |
| 5 | 4069 | 4073 | 0.1-58.6 | 107 | Поток по пласту |
| 6 | 4081 | 4085 | 0.1-58.6 | 104 | Поток по пласту |
| 7 | 4195 | 4200 | 0.1-56.2 | 101 | Поток по пласту |
| 8 | 4209 | 4213 | 0.2-57.3 | 99 | Поток по пласту |

Этап 6. Экспорт результатов анализа

Реализованы функции для сохранения результатов обработки и классификации объектов в форматы Excel и LAS.

В результирующую таблицу включаются основные характеристики обнаруженных объектов: глубина залегания, частотный диапазон, амплитуда сигнала, а также классификация типа шума.

В файле формата LAS включается логическая разметка по глубине: каждому уровню сопоставляется значение 1, если на нём обнаружен объект, или 0 в противном случае.

Новизна

Научно-техническая новизна:

- Адаптивная многоступенчатая фильтрация с учётом частотно-глубинной структуры;
- Сегментация и агрегация контуров.
- Анализ производной частотного профиля для локализации границ объектов;

Новизна реализации:

- Модифицированная функция поиска пиков с обработкой граничных значений;
- Выделение участков с постоянной частотной характеристикой;
- Автоматическое определение типа канала (LF/HF) и настройка параметров без участия пользователя;
- Обработка пересекающихся объектов: устранение перекрытий и агрегация по глубине;
- Комплексная фильтрация объектов по геометрическим, спектральным и амплитудным признакам.

Дополнительно реализован графический интерфейс для визуальной верификации и ручной корректировки результатов.

Исходный частотный профиль контура

```
> 00 = {intc: ()} np.int32(423)
> 01 = {intc: ()} np.int32(422)
> 02 = {intc: ()} np.int32(421)
> 03 = {intc: ()} np.int32(420)
> 04 = {intc: ()} np.int32(421)
> 05 = {intc: ()} np.int32(422)
> 06 = {intc: ()} np.int32(423)
> 07 = {intc: ()} np.int32(424)
> 08 = {intc: ()} np.int32(425)
> 09 = {intc: ()} np.int32(426)
> 10 = {intc: ()} np.int32(427)
> 11 = {intc: ()} np.int32(428)
> 12 = {intc: ()} np.int32(429)
> 13 = {intc: ()} np.int32(430)
> 14 = {intc: ()} np.int32(431)
> 15 = {intc: ()} np.int32(432)
> 16 = {intc: ()} np.int32(433)
> 17 = {intc: ()} np.int32(434)
> 18 = {intc: ()} np.int32(435)
> 19 = {intc: ()} np.int32(436)
> 20 = {intc: ()} np.int32(437)
> 21 = {intc: ()} np.int32(438)
> 22 = {intc: ()} np.int32(439)
> 23 = {intc: ()} np.int32(440)
> 24 = {intc: ()} np.int32(441)
> 25 = {intc: ()} np.int32(442)
> 26 = {intc: ()} np.int32(443)
> 27 = {intc: ()} np.int32(444)
> 28 = {intc: ()} np.int32(443)
> 29 = {intc: ()} np.int32(442)
> 30 = {intc: ()} np.int32(441)
> 31 = {intc: ()} np.int32(440)
```

Пики, найденные стандартной функцией

```
peaks = {list: 72} [[27], [29 56], [376]]
> 00 = {ndarray: (1,)} [27] ...View as
```

Пики, найденные модифицированной функцией

```
peaks = {list: 72} [[0 27], [0 29 56]]
> 00 = {ndarray: (2,)} [0 27] ...View as
```

Архитектура программной системы

Система имеет модульную архитектуру и включает:

1. `functions.py` — алгоритмы обработки: фильтрация, контуры, классификация, отчёты;
2. `start_program.py` — точка входа: загрузка, запуск, последовательность вызовов;
3. `constants.py` — динамическая инициализация параметров по данным;
4. `constants_store.py` — хранение параметров;
5. `main.py` — интерфейс пользователя.

Используемые инструменты и библиотеки:

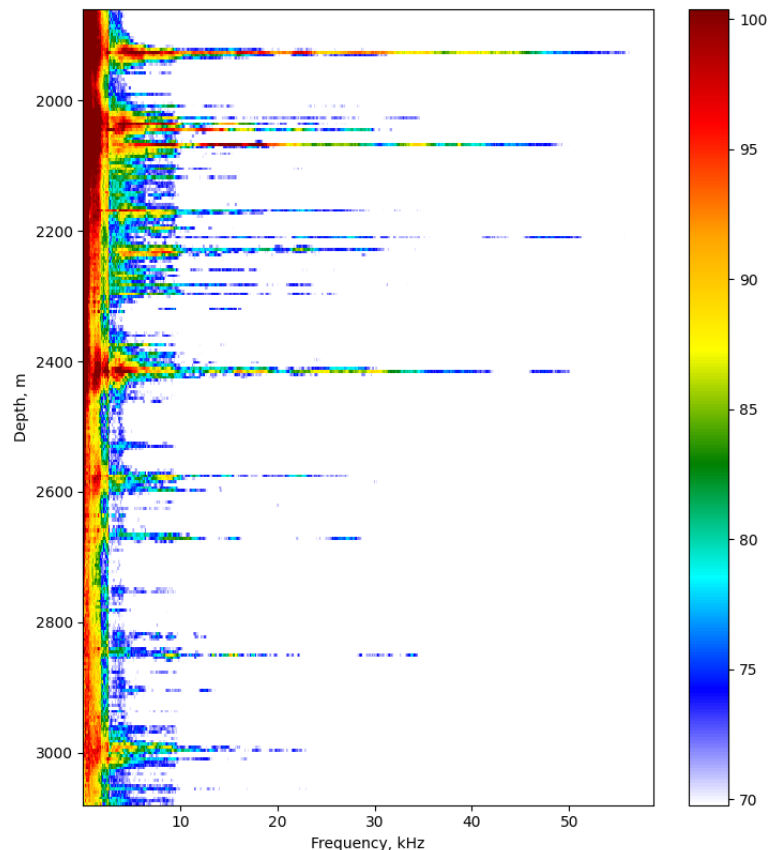
1. Язык программирования – Python.
2. Обработка данных:
 - NumPy — для работы с массивами и выполнения базовых математических операций;
 - SciPy — для медианной фильтрации и поиска локальных максимумов;
 - Pandas — для табличного представления данных и формирования отчётных файлов;
 - Lasio — для работы с LAS-файлами.
3. Обработка изображений:
 - OpenCV — морфологические операции, извлечение контуров;
 - Imutils — надстройка над OpenCV, упрощающая извлечение контуров.
4. Визуализация данных:
 - Matplotlib — визуализация спектральных панелей;
 - Tkinter — графический интерфейс.

Пример работы алгоритма

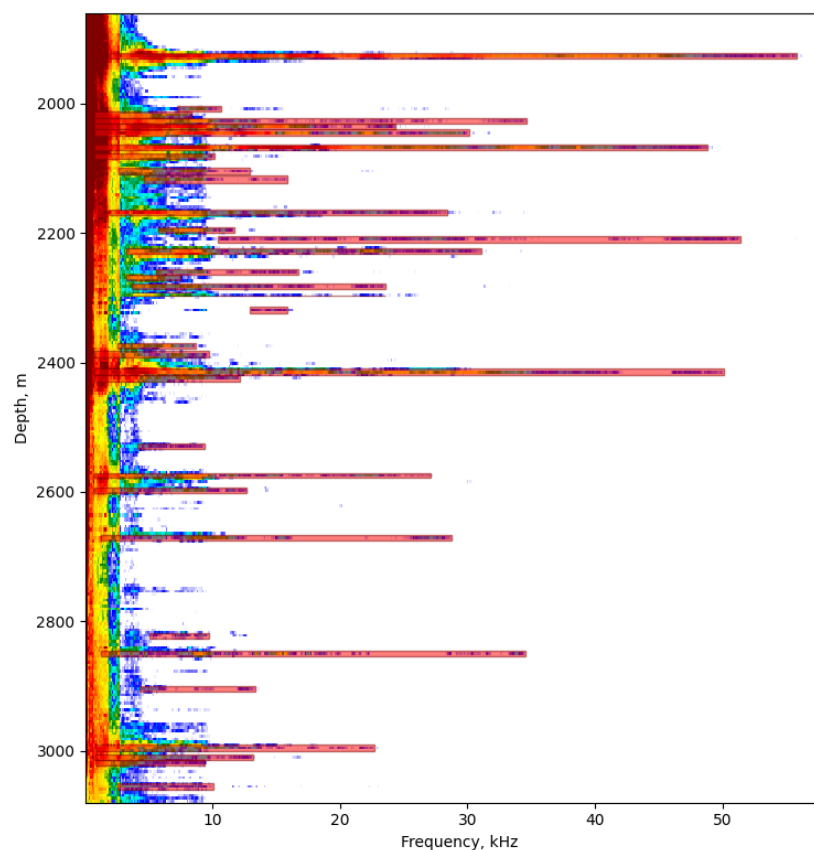
Скважина KHRС-47

Объём входных HF-данных — 265 значений по глубине, 512 по частоте. Обнаружено 35 объектов.

Исходная панель



Панель с выделенными объектами



Результирующая таблица (фрагмент)

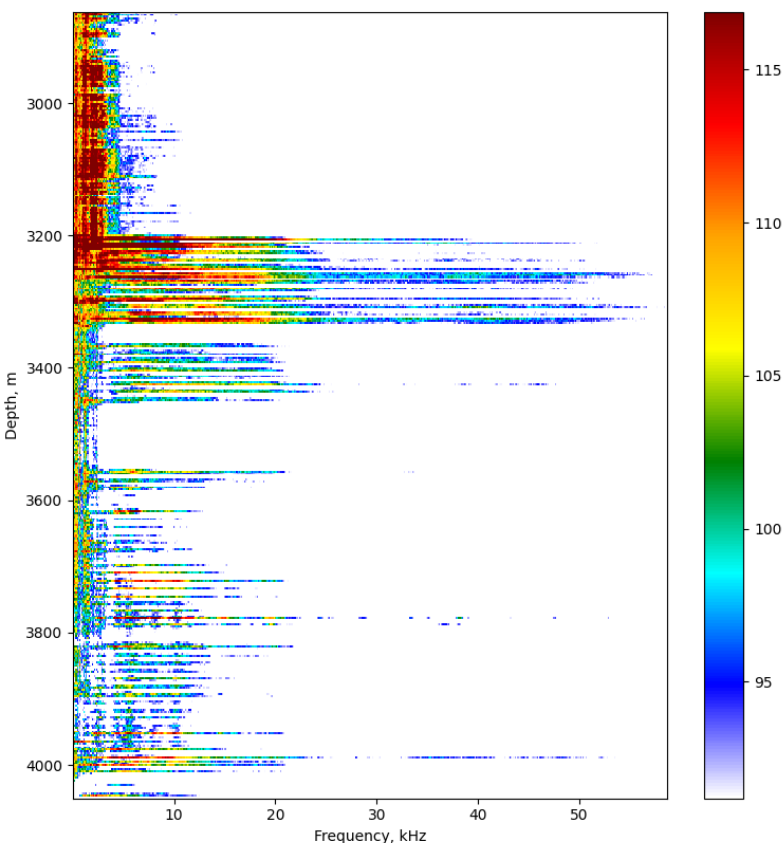
| | Top, m | Bottom, m | Frequency range, kHz | Amplitude, dB | Flow type |
|----|--------|-----------|----------------------|---------------|-----------|
| 1 | 1921 | 1931 | 1.9-55.8 | 100 | Reservoir |
| 2 | 2004 | 2013 | 7.2-10.6 | 82 | Reservoir |
| 3 | 2013 | 2022 | 0.7-8.4 | 108 | Reservoir |
| 4 | 2022 | 2031 | 0.7-34.7 | 108 | Reservoir |
| 5 | 2031 | 2040 | 0.7-24.4 | 109 | Reservoir |
| 6 | 2040 | 2050 | 0.7-30.1 | 111 | Reservoir |
| 7 | 2063 | 2071 | 0.7-48.9 | 106 | Reservoir |
| 8 | 2077 | 2086 | 0.7-10.2 | 106 | Reservoir |
| 9 | 2100 | 2109 | 2.6-12.9 | 90 | Reservoir |
| 10 | 2112 | 2123 | 4.6-15.9 | 85 | Reservoir |
| 11 | 2164 | 2173 | 1.8-28.4 | 98 | Reservoir |
| 12 | 2191 | 2201 | 5.7-11.7 | 90 | Reservoir |
| 13 | 2205 | 2214 | 10.4-51.4 | 78 | Reservoir |
| 14 | 2223 | 2232 | 3.2-31.1 | 95 | Reservoir |
| 15 | 2255 | 2264 | 5.6-16.7 | 82 | Reservoir |
| 16 | 2264 | 2273 | 3.2-7.4 | 88 | Reservoir |
| 17 | 2278 | 2287 | 3.7-23.6 | 85 | Reservoir |
| 18 | 2292 | 2292 | 3.7-8.6 | 82 | Reservoir |
| 19 | 2296 | 2296 | 3.7-23.6 | 89 | Reservoir |
| 20 | 2315 | 2324 | 12.9-15.9 | 77 | Reservoir |
| 21 | 2370 | 2379 | 2.5-8.7 | 90 | Reservoir |

Пример работы алгоритма

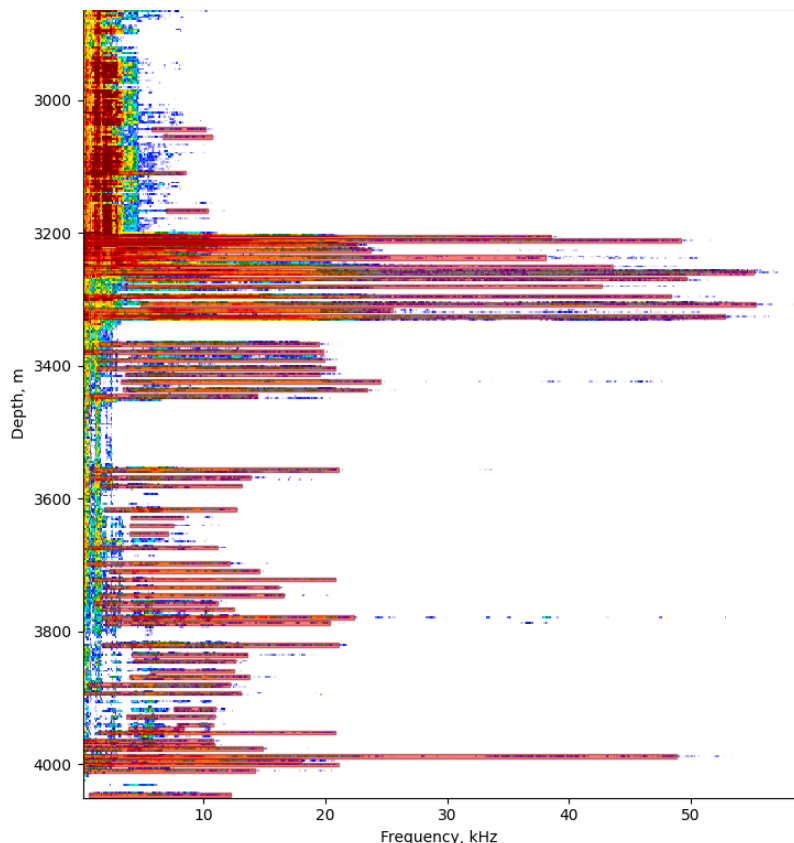
Скважина 8

Объем входных HF-данных — 396 значений по глубине, 512 по частоте. Обнаружено 62 объекта.

Исходная панель



Панель с выделенными объектами



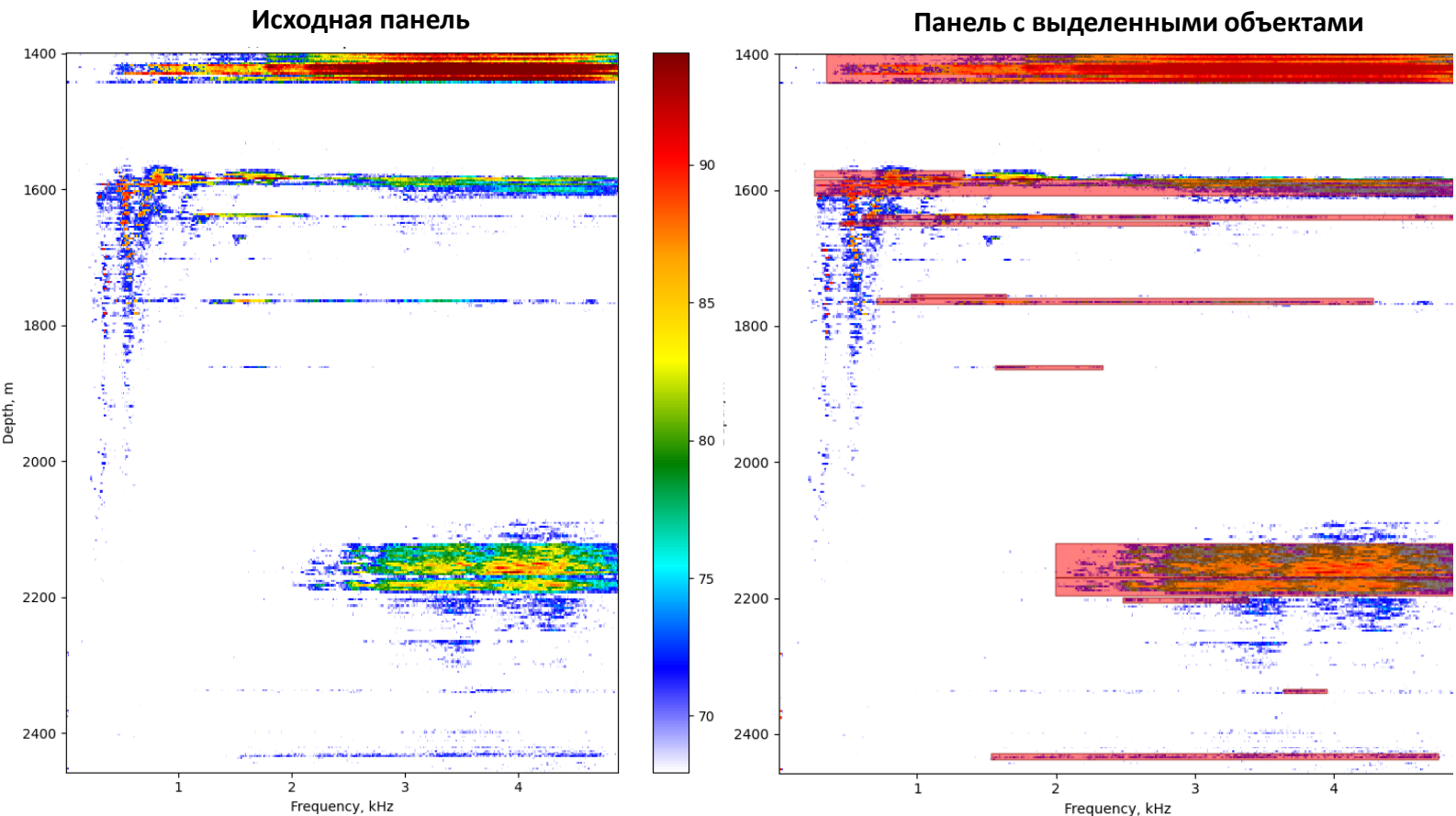
Результирующая таблица (фрагмент)

| | Кровля, м | Подошва, м | Частотный диапазон, кГц | Амплитуда, дБ | Характеристика типа шума |
|----|-----------|------------|-------------------------|---------------|--------------------------|
| 1 | 3040 | 3047 | 5.7-10.2 | 98 | Поток по пласту |
| 2 | 3052 | 3059 | 6.6-10.6 | 98 | Поток по пласту |
| 3 | 3107 | 3113 | 3.0-8.5 | 116 | Поток по пласту |
| 4 | 3164 | 3169 | 6.9-10.3 | 97 | Поток по пласту |
| 5 | 3203 | 3209 | 0.1-38.6 | 132 | Поток по пласту |
| 6 | 3209 | 3215 | 0.1-49.2 | 128 | Поток по пласту |
| 7 | 3215 | 3221 | 0.1-22.7 | 133 | Поток по пласту |
| 8 | 3224 | 3230 | 0.1-23.7 | 126 | Поток по пласту |
| 9 | 3232 | 3241 | 0.1-38.1 | 121 | Поток по пласту |
| 10 | 3248 | 3253 | 0.1-43.6 | 131 | Поток по пласту |
| 11 | 3256 | 3263 | 0.1-55.2 | 120 | Поток по пласту |
| 12 | 3266 | 3272 | 3.4-49.7 | 115 | Поток по пласту |
| 13 | 3278 | 3284 | 3.5-42.7 | 113 | Поток по пласту |
| 14 | 3293 | 3298 | 0.1-48.4 | 124 | Поток по пласту |
| 15 | 3305 | 3311 | 4.8-55.4 | 111 | Поток по пласту |
| 16 | 3313 | 3319 | 0.2-25.5 | 116 | Поток по пласту |
| 17 | 3322 | 3329 | 0.2-52.9 | 124 | Поток по пласту |
| 18 | 3364 | 3371 | 1.4-19.5 | 109 | Поток по пласту |
| 19 | 3377 | 3382 | 0.1-19.8 | 114 | Поток по пласту |

Пример работы алгоритма

Скважина Dafiq-35H1

Объём входных LF-данных — 338 значений по глубине, 512 по частоте. Обнаружено 13 объектов.



Результирующая таблица

| | Top, m | Bottom, m | Frequency range, kHz | Amplitude, dB | Flow type |
|----|--------|-----------|----------------------|---------------|-----------|
| 1 | 1400 | 1443 | 0.3-4.9 | 104 | Reservoir |
| 2 | 1572 | 1581 | 0.3-1.3 | 89 | Reservoir |
| 3 | 1584 | 1609 | 0.3-4.9 | 92 | Reservoir |
| 4 | 1637 | 1643 | 0.6-4.9 | 87 | Reservoir |
| 5 | 1646 | 1652 | 0.4-3.1 | 88 | Reservoir |
| 6 | 1752 | 1758 | 1.0-1.6 | 71 | Reservoir |
| 7 | 1758 | 1767 | 0.7-4.3 | 85 | Reservoir |
| 8 | 1858 | 1864 | 1.6-2.3 | 73 | Reservoir |
| 9 | 2119 | 2169 | 2.0-4.9 | 89 | Reservoir |
| 10 | 2169 | 2196 | 2.0-4.9 | 87 | Reservoir |
| 11 | 2199 | 2206 | 2.5-3.4 | 73 | Reservoir |
| 12 | 2333 | 2340 | 3.6-3.9 | 72 | Reservoir |
| 13 | 2428 | 2436 | 1.5-4.7 | 72 | Reservoir |

Заключение

В результате выполнения работы:

- разработан алгоритм детектирования и распознавания объектов, учитывающий особенности шумометрических данных;
- реализован программный комплекс с интуитивно понятным интерфейсом;
- проведена успешная апробация на реальных акустических данных, подтвердившая эффективность подхода.

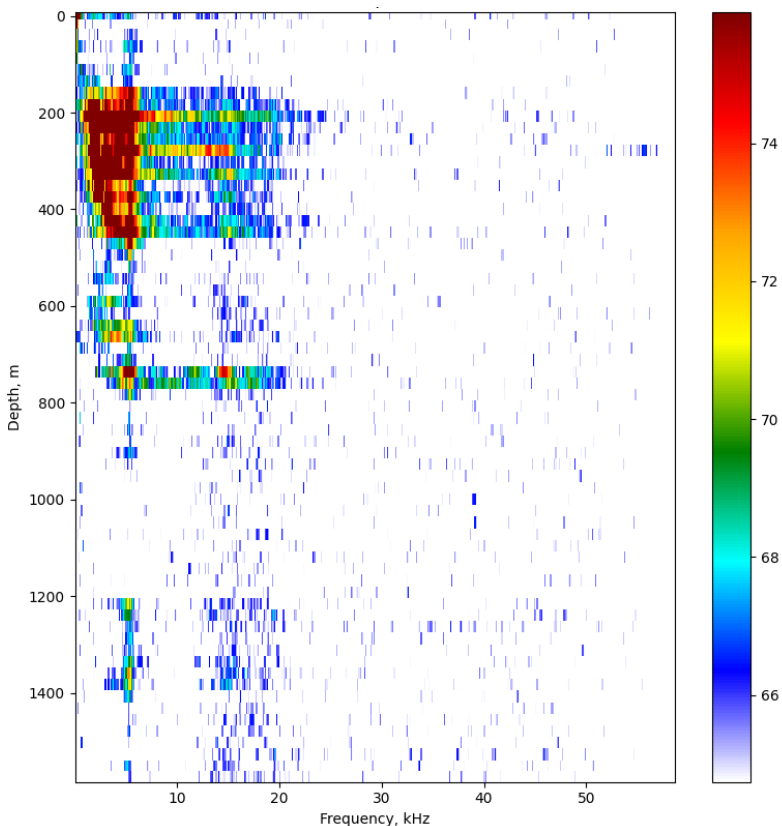
Разработанный алгоритм уже применяется в инженерной практике для ускорения и повышения точности анализа шумометрических данных и интегрирован в текущие производственные процессы.

Пример работы алгоритма

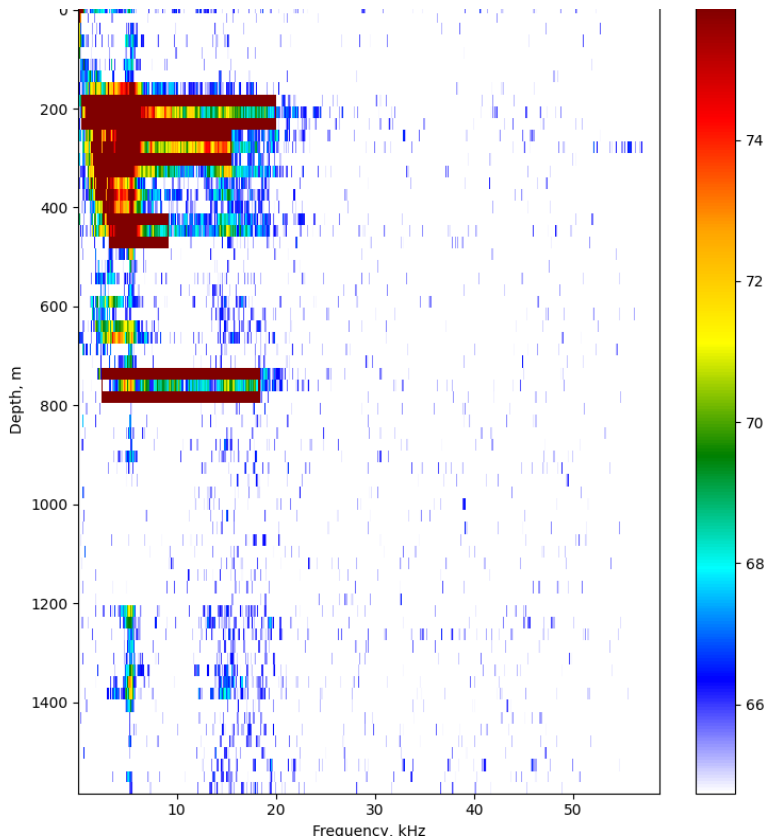
Скважина Aristocrat

Объём входных HF-данных — 67 значений по глубине, 512 по частоте. Обнаружено 4 объекта.

Исходная панель



Панель с выделенными объектами



Результирующая таблица

| | Кровля, м | Подошва, м | Частотный диапазон, кГц | Амплитуда, дБ | Характеристика типа шума |
|---|--------------|---------------|-------------------------------|------------------|-----------------------------|
| 1 | 183 | 231 | 0.5-20.0 | 92 | Поток по пласту |
| 2 | 255 | 303 | 4.2-15.4 | 81 | Поток по пласту |
| 3 | 423 | 471 | 3.2-9.2 | 83 | Поток по пласту |
| 4 | 735 | 783 | 2.5-18.3 | 77 | Поток по пласту |

Низкое разрешение исходных данных, обусловленное редкостью измерений (один замер на каждые 23 метра при рекомендованной частоте — один замер на каждый метр), привело к значительной потере информации и снижению точности алгоритма.