

УДК 519.688

DOI: 10.15827/0236-235X.139.428-437

Дата подачи статьи: 29.04.22

2022. Т. 35. № 3. С. 428–437

Автоматическое детектирование аудиodefектов с применением параллельных вычислений

А.А. Рыбаков¹, к.ф.-м.н., ведущий научный сотрудник, rybakov@jsgcc.ru

С.А. Фрейлехман¹, к.т.н., научный сотрудник, freysa@jsgcc.ru

¹ Межведомственный суперкомпьютерный центр (МСЦ) РАН – филиал
ФНЦ Научно-исследовательский институт системных исследований (НИИСИ) РАН,
г. Москва, 117218, Россия

Работа посвящена исследованиям, направленным на автоматическое детектирование дефектов и аномалий, возникающих в цифровом сигнале аудиозаписи. Методы определения дефектов применяются, главным образом, при оцифровке аналоговых аудиозаписей и восстановлении поврежденных сигналов. Методы поиска аномалий имеют широкий спектр применения, включая разработку охранных систем и систем мониторинга окружающей обстановки, определение искусственно смонтированных записей, реставрацию и восстановление архивных аудиозаписей, имеющих культурную ценность для определенного временного промежутка развития и становления социального общества, борьбу с так называемыми дип-фейками, шифровку и расшифровку закодированной в аудиоданных секретной информации и многое другое.

Современные технологии и методики позволяют с высокой эффективностью устранять найденные дефекты путем математических манипуляций аудиоинженера с сигналом или с применением умных и адаптивных инструментов редактирования цифрового сигнала, однако для этого дефект должен быть точно детектирован и локализован, а также должны быть определены его тип и возможная природа возникновения.

Для решения задачи автоматического детектирования дефектов в цифровом сигнале аудиозаписи в рамках данной работы разработано специальное ПО, верифицированное на оцифрованных аудиозаписях различного качества. Так как цифровые медиаданные, к которым относятся аудиозаписи, характеризуются большим размером, при проведении анализа особую важность приобретает параллельная распределенная обработка. Разработанный программный код определения дефектов был модернизирован с учетом необходимости запуска на массивно-параллельных микропроцессорах Intel Xeon Phi Knights Landing и продемонстрировал высокую эффективность масштабирования.

Ключевые слова: аудиозапись, дефект, аномалия, спектрограмма, Intel Xeon Phi.

Большинство аудиозаписей используются не на аналоговых носителях, а в цифровом виде, представляя собой дискретные последовательности амплитуд звука, зафиксированные в отдельные моменты времени. Существует большое количество моделей и форматов кодирования звука в цифровом виде, отличающихся друг от друга точностью воспроизведения и размером аудиозаписи, среди них наиболее известны такие форматы, как MP3, WMA, FLAC и другие [1]. Основным методом обработки аудиосигналов является применение различных фильтров (фильтры ограничения полосы частот, присутствия, высоких и низких частот, плавного подъема/спада амплитудно-частотной характеристики и другие), которые используются для изменения частотных и фазовых характеристик, сужения или расширения динамического диапазона, применения амплитудной, частотной или фазовой модуляции, удаления шумов и многих других действий [2]. Использование цифровых форматов открыло ши-

рокие возможности по защите аудиозаписей, например, с помощью стеганографического метода, заключающегося в добавлении специальных цифровых зашифрованных меток в аудиозапись, которые невозможно обнаружить без специального ПО [3]. Также широко используется обработка звука в художественных целях – в современной музыке активно применяется наложение различных эффектов: искусственное эхо, компрессия звука, специальные искажения, дребезжание, замедление или ускорение звучания и многие другие [4]. Отдельной широкой областью обработки звука является распознавание звуковой информации, в основном человеческой речи [5], однако данная задача может быть применена к определению любых звуковых событий (звуки животного, разбитого стекла, оружейные выстрелы и прочее).

Кроме специально смоделированных и наложенных эффектов в аудиозаписях могут встречаться случайно появившиеся помехи и

нежелательные искажения. Наиболее частым источником искажений звука и аномалий являются оцифровка аналоговых записей либо дефекты записывающей аппаратуры (если запись ведется в изначально цифровом формате). Для борьбы с такими дефектами и аномалиями используется специальное ПО. Реализующее функционал по работе со звуком ПО широко распространено во всем мире, ярким примером являются продукты Adobe Audition, Audacity и многие другие. Данные инструменты позволяют редактировать и восстанавливать аудио-записи, устранять повреждения. С их помощью профессиональный звукоинженер может качественно восстанавливать исходный цифровой сигнал [6] или изменять его объективные характеристики. Например, метод удаления шума из аудиосигнала [7] эффективно справляется с поставленной задачей, но лишь в том случае, когда заведомо известно о наличии данного дефекта, при этом проверка методом грубой силы каждого файла из архива может привести к значительным временным и ресурсным затратам. Для достижения положительного результата инженеру необходимо четко детектировать местонахождение дефекта и примерный вид повреждения сигнала, после чего он сможет приступить к работе по реставрации. При поиске аномалий, например, следов вмешательства в цифровой аудиосигнал [8] или необычных событий при работе систем мониторинга окружающей обстановки, также необходимо выполнять автоматическую проверку большого количества входных данных.

Данные потребности специалистов в области работы с цифровыми аудиофайлами демонстрируют актуальность исследований и разработок, направленных на автоматическое детектирование дефектов и аномалий в массиве данных большого объема (либо в режиме обработки аудиопотоков) за приемлемое время.

В рамках настоящего исследования было создано ПО, позволяющее автоматически определять наличие дефектов в цифровом сигнале (место их нахождения и тип) при анализе большого количества данных за разумное время и с допустимой точностью.

Выполнялись следующие задачи: анализ существующих исследований в области обработки цифровых сигналов, разработка программного кода по существующим и проверенным алгоритмам, разработка и тестирование собственных алгоритмов детектирования дефектов, формирование комплексного программного решения для детектирования аудио-

дефектов, применение параллельных вычислений для ускорения анализа аудиоданных при выполнении на массивно-параллельных микропроцессорах Intel Xeon Phi.

Исследуемые дефекты

Рассматривались дефекты следующих видов:

- щелчок (click) – короткий импульс интенсивности звука, возникший из-за некачественной склейки, статических щелчков, звуковых артефактов, чаще всего встречающихся, когда источником цифрового сигнала являются виниловые пластинки; также к ним относятся соприкосновение губ, цоканье языком и другие звуки, возникающие при работе с микрофоном и негативно влияющие на качество записи;

- глухая запись (muted) – эффект глухой записи, вызванный некорректной настройкой, режимом записи звука, неподходящим или дефектным устройством для снятия звука;

- эхо и реверберация (echo) – эффект, возникающий при попадании в приемник звука, отраженного от твердой и упругой поверхности, или созданный искусственно при обработке цифрового сигнала;

- рассинхронизация каналов звука (asnc) – нарушение синхронности стереозаписи, смещение фазы в одном из каналов;

- различия в силе звука в каналах (diff) – контроль за резкими изменениями в одном из каналов, выпадение звука, определение неестественных скачков силы звука;

- высокочастотный и низкочастотный гул, вызванный различными шумами, помехами и неполадками в электросети (hum, dense);

- перегрузка (overload) – превышение предела напряжения выходного усилителя питания, выражающееся в ограничении амплитуды сигнала;

- наличие цифровых копий фрагментов сигнала (dbl) – искусственное наложение сегмента сигнала или произвольное копирование данных на магнитной ленте.

Методы анализа дефектов

В рамках данного исследования разработано комплексное ПО, позволяющее пользователю обрабатывать массивы аудиоданных с целью детектирования дефектов. Входными данными являются коллекции аудиозаписей в формате wav [9], выходными – список описаний дефектов, каждое из которых состоит из

названия файла, номера канала, названия дефекта (если он был найден в этой записи), времени начала и конца дефекта, соответствующему примерному положению дефекта на временной шкале цифрового сигнала.

Во время работы программный комплекс последовательно выполняет поиск каждого дефекта и распараллеливает обработку аудиозаписи путем дробления файла аудиосигнала. Детектирование дефекта каждого из типов осуществляется отдельным методом. Ядро детектирования реализовано с использованием библиотеки librosa [10, 11] для языка программирования python.

Метод **click** предназначен для поиска кратковременных забросов звука примерно одного уровня по всему спектру частот. На спектрограмме частот это выражается в виде тонких вертикальных полос (рис. 1а). Дефект может возникать из-за некачественной склейки, статических щелчков, артефактов звуков и по другим причинам.

Для поиска дефекта спектрограмма нормализуется таким образом, чтобы уровень звука изменялся в пределах [0.0, 1.0]. Далее для анализа берется только верхняя часть спектрограммы (так как она более разрежена в плане полезного звука) и с помощью оператора Собеля [12] на ней выделяются вертикальные границы.

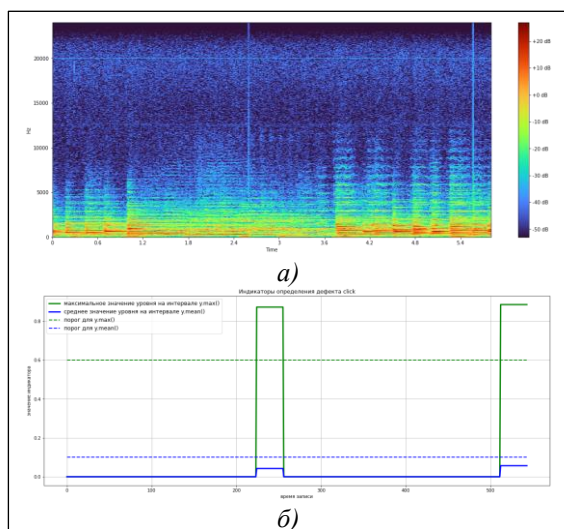


Рис. 1. Цифровой сигнал с дефектом «треск, щелчок»: а) частотная спектрограмма сигнала, б) визуализация индикаторов детектирования дефекта

Fig. 1. Digital signal with a "crack, click" defect: а) a signal frequency spectrogram, б) visualization of defect detection indicators

Вывод о присутствии дефекта в записи делается на основании двух фактов: во-первых, максимальное значение уровня звука в некотором интервале, длина которого определяется настройками, превышает определенный порог ($y_{\max}() > Thr$); во-вторых, среднее значение уровня звука в этом же интервале находится ниже второго порога ($y_{\text{mean}}() < \text{MeanThr}$). На рисунке 1б продемонстрирован факт регистрации двух щелчков, представленных на спектрограмме записи из рисунка 1а.

Метод **muted** предназначен для детектирования дефекта глухой записи, в которой на протяжении всего (или значительного) времени по каким-либо причинам отсутствуют верхние диапазоны частот (хотя они должны быть). То есть данный дефект характеризуется наличием пустых частот в верхнем диапазоне спектрограммы цифрового сигнала. Для сравнения глухой записи и исходного цифрового сигнала (не глухой записи) представлены две спектрограммы: глухой записи (рис. 2а) и исходного сигнала (рис. 2б).

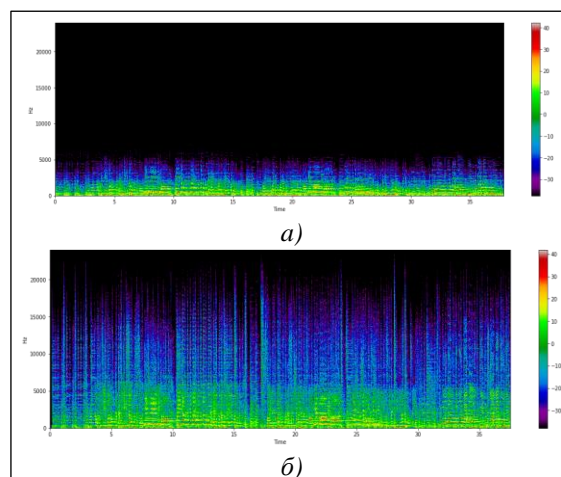


Рис. 2. Демонстрация частотной спектрограммы глухой записи: а) глухая запись, б) исходная запись

Fig. 2. Demonstration of a deaf recording frequency spectrogram: а) a deaf recording, б) the original recording

Обнаружение дефекта с использованием данного метода происходит по следующему алгоритму: решение о детектировании дефекта принимается на основе среднего значения ортоцентра нормализованной спектрограммы, при этом ортоцентр спектрограммы дает адекватное решение для участков тишины (ортоцентр на абсолютной тишине находится на уровне 50 %, что позволяет не трактовать

тишину как глухую запись), ортоцентр вычисляется из нормализованной спектрограммы прямым счетом. Причина глухоты записи не определяется.

Если ортоцентр цифрового сигнала ниже порогового значения, детектируется глухая запись, если выше – делается вывод об отсутствии исследуемого дефекта (рис. 3).

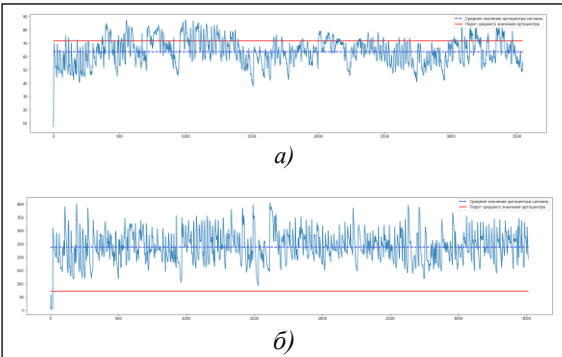


Рис. 3. Среднее значение ортоцентра:
а) глухая запись, б) нормальная запись

Fig. 3. The orthocenter average value:
а) a deaf recording, б) not a deaf recording

Метод **echo** предназначен для детектирования наличия эффекта эха в цифровом сигнале, содержащем голосовую активность человека, которое характеризуется как частичное последовательное повторение фрагмента цифрового сигнала, накладывающегося поверх основного сигнала.

Чтобы при анализе музыкальных композиций избежать ложных срабатываний, алгоритм данного модуля анализирует глобальное значение автокорреляционной функции для темпограммы цифрового сигнала, где высокое значение полученной средней величины свидетельствует о наличии сохранения темпа в записи (рис. 4). Это позволяет сделать вывод, что запись содержит музыку и велика вероятность ложного срабатывания, особенно в тех местах, где автор композиции намеренно использует эхо. Если был обнаружен данный устойчивый темп, делается вывод, что это музыка и запись не исследуется на наличие эха.

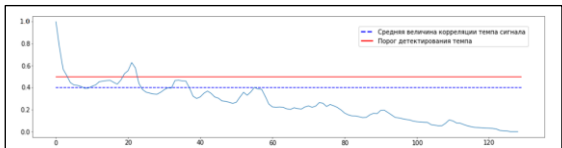


Рис. 4. Визуализация корреляции темпограммы

Fig. 4. Visualization of the tempoogram correlation

Процесс детектирования исследуемого дефекта сводится к сканированию цифрового сигнала записи небольшими окнами для детектирования высокого уровня корреляции, и при его наличии фиксируется место обнаружения признаков наличия эффекта эха (рис. 5).

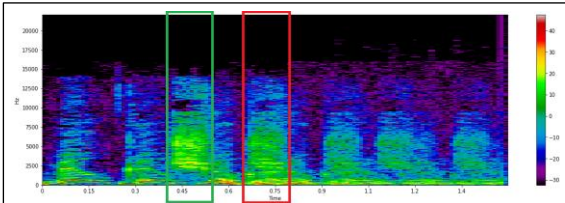


Рис. 5. Демонстрация двух окон на частотной спектрограмме с высоким значением корреляции (близким к единице)

Fig. 5. Demonstration of two windows on a frequency spectrogram with a high correlation value (close to unity)

Метод **asnc** предназначен для детектирования рассинхронизации двух каналов цифрового сигнала со смещением сигнала в одном из каналов (см. <http://www.swsys.ru/uploaded/image/2022-3/2022-3-dop/1.jpg>). Исключением для этого метода являются записи, имеющие один канал – моно. Исследуемый дефект характеризуется несовпадением фаз воспроизведения каналов в цифровом сигнале.

Методом детектирования рассинхронизации решается задача определения дефекта на основе анализа коэффициента корреляции Пирсона [13]. На основании данного коэффициента могут быть приняты следующие решения: если коэффициент близок к 1.0, то имеет место дефект ложного стерео (данное событие не фиксируется, так как практически все аудиозаписи таковы); если коэффициент слишком низкий и положительный, то имеют место эффект сильного расхождения, плывущий звук, ощутимые выпадения звука, запись на слух явно дефектная; отрицательный коэффициент означает сдвиг по фазе между каналами.

Метод **diff** ориентирован на детектирование дефектов, связанных с расхождениями в каналах цифрового сигнала, которые можно обнаружить по среднему значению силы сигнала на нормализованной спектрограмме.

Метод позволяет детектировать резкое выпадение звука на одном из каналов (рис. 6), рассинхронизацию и другие значительные отклонения в звучании каналов.

Метод **hum** определяет наличие гула в сигнале. Источником гула чаще всего являются

различные шумы, наводки, помехи и неполадки в электросети. Подобный шум проявляется в низкочастотном диапазоне.

Увидеть гул на спектре можно при увеличении спектра в районе низких частот. Подобный негативный шум проявляется в виде серии горизонтальных линий, ярко выраженных в районе 50–60 Гц.

Детектирование дефекта гула с использованием данного метода на некоторой частоте определяется на достаточно продолжительном участке записи (в качестве эмпирической настройки была выбрана продолжительность 15 секунд) и характеризуется отношением 10-го и 55-го квантилей отсортированного сигнала на спектрограмме [14]. Скачок разности оригинального и сглаженного отношений сигнализирует о гуле (рис. 7).

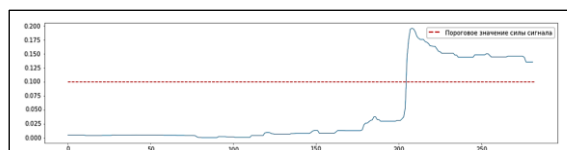


Рис. 6. Детектирование расхождения каналов записи

Fig. 6. Detecting the divergence of recording channels

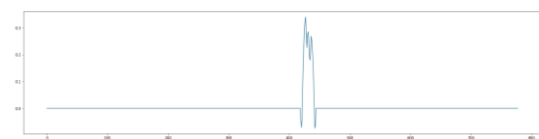


Рис. 7. Всплеск отфильтрованного значения интенсивности сигнала, характеризующий наличие hum-эффекта

Fig. 7. A spike in the filtered signal intensity value that characterizes the presence of the hum effect

Метод **dense** предназначен для детектирования высокочастотного гула в диапазоне частот от 12 000 Гц и выше (в основном в исследовании встречались записи с частотой гула до 20 000 Гц). Источником таких дефектов могут быть люминесцентные и флуоресцентные лампы, генераторы и микрофоны видеокамер, подключенных к той же электрической цепи, что и студийное оборудование.

На частотной спектрограмме такой гул проявляется в виде ровной и тонкой горизонтальной линии, проходящей в районе высоких частот на протяжении всей аудиозаписи (рис. 8а).

Данный метод работает с нормализованной

частотной спектрограммой по верхним и нижним частотам: спектрограмма сортируется по интенсивности вдоль частот, это приводит к тому, что в верхних частотах остается ярко выраженная линия, характеризующая данный дефект. Детектирование дефекта происходит путем анализа отсортированной спектрограммы в верхних частотах на предмет наличия горизонтальных линий (рис. 8б), которые возникают там лишь в двух случаях – при наличии ярко выраженного высокочастотного гула или слишком короткой по времени записи, из-за чего после сортировки частот не произошло отсеивание всех прочих частот.

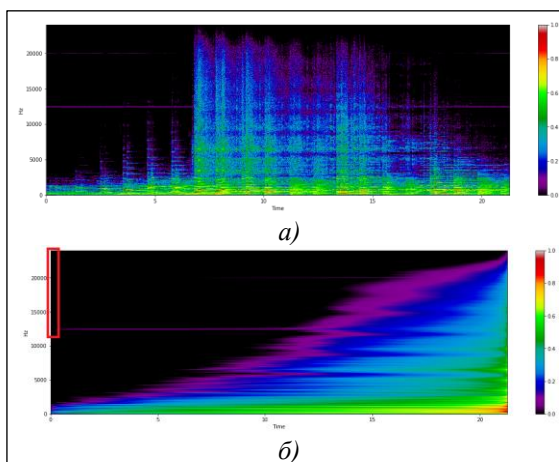


Рис. 8. Пример визуализации гула на частотной спектрограмме аудиозаписи: а) оригинальная спектрограмма записи, б) спектрограмма с отфильтрованными интенсивностями частот

Fig. 8. The example of hum visualization on the audio recording frequency spectrogram: а) the original recording spectrogram, б) a spectrogram with filtered frequency intensities

Метод **overload** предназначен для детектирования избыточной насыщенности интенсивности сигнала, вследствие чего возникает эффект перегрузки – срез амплитудных значений волны сигнала.

Основная идея алгоритма детектирования данного дефекта сводится к поиску самых высоких забросов интенсивности – громкости звука (рис. 9) и последующему анализу формы сигнала в этом месте: если будет найден срез амплитуды сигнала, то есть наклон касательной на всем исследуемом участке будет стремиться к нулю, то это перегрузка в явном виде.

Метод **dbl** предназначен для детектирования участков цифровой копии в сигнале. Дан-

ный дефект характеризуется наличием идентичной копии фрагмента цифрового сигнала (рис. 10).

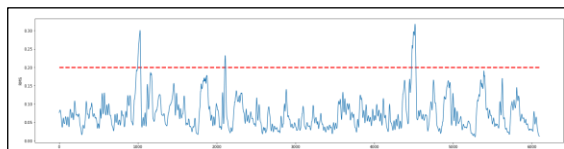


Рис. 9. Визуализация громкости цифрового сигнала тестовой записи с визуализацией порога детектирования высокого уровня интенсивности сигнала

Fig. 9. A volume visualization of the digital signal of the test recording with visualization of the threshold for detecting a high level of signal intensity

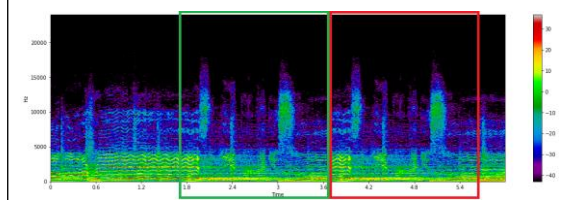
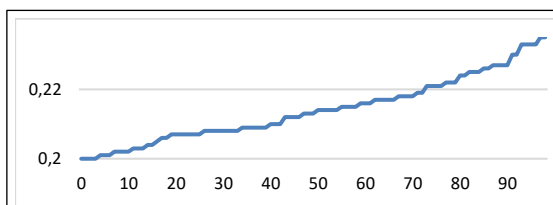


Рис. 10. Демонстрация двух идентичных фрагментов цифрового сигнала записи

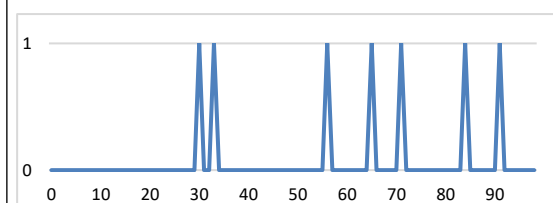
Fig. 10. Demonstration of two identical fragments of the digital recording signal

Детектирование дефекта происходит следующим образом: если в цифровом сигнале присутствует идентичная копия фрагмента сигнала, то значения амплитуд в этих копиях фрагментов имеют значение корреляции, равное единице. Для минимизации числа проверок анализируются только часть амплитуд (самые большие амплитуды) – в отсортированном массиве самых высоких амплитуд (рис. 11а) ищется общее количество совпадений соседних пар (рис. 11б), а при слишком большом количестве таких пар принимается решение о наличии дефекта – цифровой копии.

ПО, реализующее вышеописанные методы обнаружения дефектов, тестировалось на коллекции аудиоданных оцифрованных записей. По результатам тестирования была собрана следующая статистика по потреблению ресурсов каждым из методов: muted – 39 %, echo – 17 %, hum – 10 %, click – 8 %, dense – 8 %, overload – 7 %, diff – 6 %, dbl – 4 %, asnc – 1 %. Данная статистика показывает, что наиболее требовательны к вычислительному времени дефекты обнаружения глухих записей, а также поиск эха.



а)



б)

Рис. 11. Визуализация маркеров детектирования дефекта: а) сто самых высоких амплитуд тестовой записи, б) пики, демонстрирующие наличие совпадающих последовательных амплитуд из списка ста высоких амплитуд записи

Fig. 11. A visualization of defect detection markers: а) one hundred highest amplitudes of the test recording, б) peaks demonstrate the presence of coincident consecutive amplitudes from the list of one hundred high amplitudes of the recording

Для работы с архивами аудиоданных, суммарная длительность которых исчисляется часами или даже днями, важнейшее значение имеет параллельная реализация. Поэтому для работы с большими массивами данных было выполнено распараллеливание всех описанных выше методов определения дефектов для запуска на массивно-параллельных микропроцессорах Intel Xeon Phi.

Параллельные вычисления

Для реализации параллельных вычислений был разработан дополнительный функционал по распараллеливанию обработки с помощью библиотеки mpi4py [15, 16]. После считывания аудиозаписи разделялась на отдельные фрагменты, каждый из которых анализировался в своем MPI процессе. Результаты анализа всех фрагментов комплексировались в единую структуру описания дефектов. Задача по поиску дефектов не является сильно связанной (в процессе проведения анализа нет необходимости обмениваться данными между отдельными процессами), поэтому при распараллели-

вании вычислений ожидания от масштабирования задачи были высоки.

Параллельные запуски выполнялись на вычислительных ресурсах МСЦ РАН. Для тестирования параллельной версии кода были выбраны массивно-параллельные микропроцессоры с максимальным количеством вычислительных ядер, а именно вычислительные узлы сегмента MBC-10П МП2 KNL на базе микропроцессоров Intel Xeon Phi Knights Landing [17]. Каждый такой процессор содержит 72 вычислительных ядра, что позволяет использовать массивную параллельность вычислений. Были выполнены запуски на 36, 72, 108, 144 процессах при использовании одного или двух вычислительных узлов (для данной задачи межпроцессные обмены не имеют значения, поэтому большее количество узлов не рассматривалось). Во время запусков на одно вычислительное ядро не распределялось более одного процесса. Результаты запусков представлены на рисунке 12.

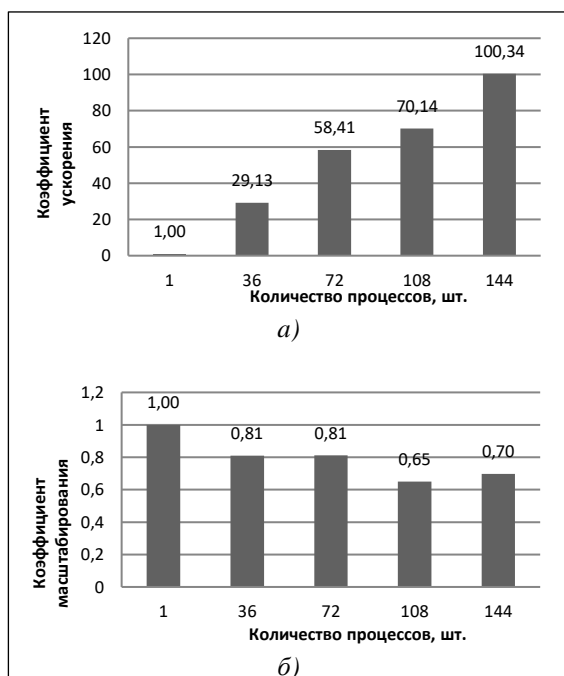


Рис. 12. Результаты тестирования программного комплекса: а) коэффициент ускорения, б) коэффициент масштабирования

Fig. 12. Software package testing results: а) an acceleration coefficient, б) a scaling factor

Во время запусков измерялись коэффициенты ускорения и масштабируемости вычислений относительно эталонного запуска на одном процессе (версия без распараллеливания). Данные коэффициенты вычислялись следующим образом. Если время анализа программным комплексом фиксированного множества входных данных при распараллеливании на n процессов обозначить через $T(n)$, то под коэффициентом ускорения понимается величина $s(n) = T(1)/T(n)$, а под коэффициентом эффективности масштабирования – величина $e(n) = s(n)/n$. Коэффициент эффективности масштабирования вычислений является удобной характеристикой, так как для идеального распараллеливания данный коэффициент точно равен единице [18]. Для рассмотренной задачи определения дефектов в аудиосигналах коэффициент эффективности масштабирования оказался достаточно высоким (0,7 для 144 MPI процессов), что позволяет применять разработанный программный комплекс на больших массивах данных.

Заключение

В результате анализа предметной области был выявлен ряд научных и профессиональных работ, посвященных нахождению и устранению аудиодефектов в цифровом сигнале.

Для детектирования исследуемых дефектов реализованы существующие алгоритмы и разработан ряд новых для повышения качества детектирования и снижения процентного соотношения ложного срабатывания различных методов детектирования при наличии или отсутствии в цифровом сигнале нескольких дефектов.

Результатом данной работы стало ПО, которое позволяет оперативно оценивать качество больших массивов оцифрованных аудиоданных, выявляя такие дефекты, как щелчок, глухая запись, эхо, рассинхронизация каналов звука и отдельных фрагментов в каналах звука, высокочастотный и низкочастотный гул, перегрузка, цифровые копии сигнала.

Для разработанного программного комплекса было выполнено распараллеливание, что позволило осуществлять параллельную обработку аудиоданных на массивно-параллельных микропроцессорах Intel Xeon Phi с высоким показателем эффективности масштабирования вычислений.

Работа выполнена в МСЦ РАН в рамках государственного задания по теме FNEF-2022-0016. В исследованиях использовался суперкомпьютер MBC-10П.

Литература

1. Nowak N., Zabierowski W. Methods of sound data compression – Comparison of different standards. Proc. XI Int. Conf. CADSM, 2011, pp. 431–434.
2. Абдуллаев У.М. Методы и алгоритмы обработки звуковых сигналов // Бюллетень науки и практики. 2020. Т. 20. № 6. С. 25–30. DOI: 10.33619/2414-2948/55/03.
3. Коробейников А.Г., Даурских А.Г., Павлова Н.В. Встраивание цифровых водяных знаков в аудиосигнал методом расширения спектра // Науч.-технич. вестн. СПбГУ ИТМО. 2009. № 1. С. 82–88.
4. Kot V. Digital sound effects echo and reverb based on non-exponentially decaying comb filter. Proc. XIV EUSIPCO, 2006, pp. 1–5.
5. Гаврилович Н.В., Сейтвелиева С.Н. Методы распознавания речи и их классификация // Таврический научный обозреватель. 2016. № 6. С. 206–212. URL: <http://tavr.science/stat/2016/06/30-Gavrilovich.pdf> (дата обращения: 01.04.2022).
6. Smith S.W. The Scientist and Engineer's Guide to DSP. CA, San Diego, California Technical Publ., 1999, 650 p.
7. Zoican S. Audio signals noise removal real time system. Proc. VIII Int Conf. on Communications, 2010, pp. 25–28. DOI: 10.1109/ICCOMM.2010.5509098.
8. Галяшина Е.И. Диагностика аутентичности цифровых фонограмм в фоноскопической экспертизе: возможности и пределы исследования // Вестн. экономической безопасности. 2018. № 1. С. 34–41.
9. Melnyk A., Shevchuk R., Sapozhnyk H. Multichannel mixing of speech signals accordant with the method of multistage mixing. Proc. Int. Conf. IEEE Cat. no. 02EX542, 2006, pp. 169–172. DOI: 10.1109/TCSET.2006.4404484.
10. McFee B., Raffel C., Liang D., Ellis D.P.W., McVicar M. et al. Librosa: Audio and music signal analysis in Python. Proc. XIV SCIPY, 2015, pp. 18–24. DOI: 10.25080/MAJORA-7B98E3ED-003.
11. Raguraman P., Mohan R., Vijayan M. Librosa based assessment tool for music information retrieval systems. Proc. II IEEE Conf. MIPR, 2019, pp. 109–114. DOI: 10.1109/MIPR.2019.00027.
12. Мартынова А.В. Сравнительный анализ градиентных методов выделения границ для распознавания лиц // Безопасность информационного пространства: матер. Всерос. научн.-практич. конф. 2014. С. 39–45.
13. Наследов А.Д. Математические методы психологического исследования. Анализ и интерпретация данных. СПб: Речь, 2012. 392 с.
14. Brandt M., Bitzer J. Automatic detection of hum in audio signals. J. of the AES, 2014, vol. 62, no. 9, pp. 584–595. DOI: 10.17743/jaes.2014.0034.
15. Fink Z., Liu S., Choi J., Diener M., Kale L.V. Performance evaluation of Python parallel programming models: Charm4Py and mpi4py. Proc. IEEE/ACM VI Int. Workshop ESPM2, 2021, pp. 38–44. DOI: 10.1109/ESPM254806.2021.00010.
16. Smith R. Performance of MPI Codes written in Python with NumPy and mpi4py. Proc. VI Workshop PyHPC, 2016, pp. 45–51. DOI: 10.1109/PyHPC.2016.010.
17. Jeffers J., Reinders J., Sodani A. Intel Xeon Phi Processor High Performance Programming. Morgan Kaufmann Publ., 2016, 632 p.
18. Shabanov B.M., Rybakov A.A., Shumilin S.S., Vorobyov M. Yu. Scaling of supercomputer calculations on unstructured surface computational meshes. Lobachevskii J. of Math., 2021, vol. 42, no. 11, pp. 2571–2579. DOI: 10.1134/S1995080221110202.

Software & Systems
DOI: 10.15827/0236-235X.139.428-437

Received 29.04.22
2022, vol. 35, no. 3, pp. 428–437

Automatic detection of audio defects using parallel computing

A.A. Rybakov¹, *Ph.D. (Physics and Mathematics), rybakov@jscs.ru, lead researcher*
S.A. Freylekhan¹, *Ph.D. (Engineering), freysa@jscs.ru, researcher*

¹ JSCC RAS – Branch of SRISA RAS, Moscow, 117218, Russian Federation

Abstract. The paper is devoted to research aimed at automatic detection of defects and anomalies that occur in the audio record digital signal. Defect detection methods are mainly used in digitizing analog audio records

and restoring damaged signals. Anomaly detection methods have a wide range of applications including the development of security and environmental monitoring systems, the identification of artificially edited records, restoration of archival audio recordings of cultural value for a certain time period of society development and formation, the fight against so-called deepfakes, encryption and decryption of classified information encoded in audio data and much more.

Modern technologies and techniques make it possible to efficiently eliminate the found defects by mathematical manipulation of the signal by an audio engineer or using smart and adaptive digital signal editing tools. However, for this purpose, the defect must be accurately detected and localized, its type and the possible origin must also be determined.

There is special software developed within the framework of this work to solve the problem of automatic detection of defects in a digital signal of an audio record. It was verified on digitized audio records of various quality. Since digital media data including audio records have a large size, the aspect of parallel distributed processing is of particular importance during the analysis. Due to this fact, the developed defect detection code was upgraded to take into account the need to run on Intel Xeon Phi Knights Landing massively parallel microprocessors and demonstrated high scaling efficiency.

Keywords: audio record, defect, anomaly, spectrogram, Intel Xeon Phi.

Acknowledgements. *The work has been done in the Joint Supercomputer Center of the Russian Academy of Sciences within the framework of the state assignment on the topic FNEF-2022-0016. The research involved using the MVS-10P supercomputer.*

References

1. Nowak N., Zabierowski W. Methods of sound data compression – Comparison of different standards. *Proc. XI Int. Conf. CADSM*, 2011, pp. 431–434.
2. Abdullayev U. Methods and algorithms sound signals processing. *Bull. of Science and Practice*, 2020, vol. 20, no. 6, pp. 25–30. DOI: 10.33619/2414-2948/55/03 (in Russ.).
3. Korobeynikov A., Dauriskikh A., Pavlova N. Integration of digital watermarks into audio signal using the method of spectrum expansion. *Sci.Tech. J. Inf. Technol. Mech. Opt.*, 2009, no. 1, pp. 82–88 (in Russ.).
4. Kot V. Digital sound effects echo and reverb based on non-exponentially decaying comb filter. *Proc. XIV EUSIPCO*, 2006, pp. 1–5.
5. Gavrilovich N.V., Seytvelieva S.N. Speech recognition methods and their classification. *Tauride Scientific Observer*, 2016, no. 6, pp. 206–212. Available at: <http://tavr.science/stat/2016/06/30-Gavrilovich.pdf> (accessed April 01, 2022) (in Russ.).
6. Smith S.W. *The Scientist and Engineer's Guide to DSP*. CA, San Diego, California Technical Publ., 1999, 650 p.
7. Zoican S. Audio signals noise removal real time system. *Proc. VIII Int Conf. on Communications*, 2010, pp. 25–28. DOI: 10.1109/ICCOMM.2010.5509098.
8. Galyashina E.I. Diagnostics of digital phonogram authenticity in phonoscopic examination: capabilities and limits of research. *Vestn. of Economic Security*, 2018, no. 1, pp. 34–41 (in Russ.).
9. Melnyk A., Shevchuk R., Sapozhnyk H. Multichannel mixing of speech signals accordant with the method of multistage mixing. *Proc. Int. Conf. IEEE Cat. no. 02EX542*, 2006, pp. 169–172. DOI: 10.1109/TCSET.2006.4404484.
10. McFee B., Raffel C., Liang D., Ellis D.P.W., McVicar M. et al. Librosa: Audio and music signal analysis in Python. *Proc. XIV SCIPY*, 2015, pp. 18–24. DOI: 10.25080/MAJORA-7B98E3ED-003.
11. Raguraman P., Mohan R., Vijayan M. Librosa based assessment tool for music information retrieval systems. *Proc. II IEEE Conf. MIPR*, 2019, pp. 109–114. DOI: 10.1109/MIPR.2019.00027.
12. Mart'yanova A.V. Comparative analysis of gradient methods of border selection for face recognition. *Proc. Sci-Pract. Conf. Information Space Security*, 2014, pp. 39–45 (in Russ.).
13. Nasledov A.D. *Mathematical Methods of Psychological Research. Data Analysis and Interpretation*. St. Petersburg, 2012, 392 p. (in Russ.).
14. Brandt M., Bitzer J. Automatic detection of hum in audio signals. *J. of the AES*, 2014, vol. 62, no. 9, pp. 584–595. DOI: 10.17743/jaes.2014.0034.
15. Fink Z., Liu S., Choi J., Diener M., Kale L.V. Performance evaluation of Python parallel programming models: Charm4Py and mpi4py. *Proc. IEEE/ACM VI Int. Workshop ESPM2*, 2021, pp. 38–44. DOI: 10.1109/ESPM254806.2021.00010.
16. Smith R. Performance of MPI Codes written in Python with NumPy and mpi4py. *Proc. VI Workshop PyHPC*, 2016, pp. 45–51. DOI: 10.1109/PyHPC.2016.010.

17. Jeffers J., Reinders J., Sodani A. *Intel Xeon Phi Processor High Performance Programming*. Morgan Kaufmann Publ., 2016, 632 p.

18. Shabanov B.M., Rybakov A.A., Shumilin S.S., Vorobyov M.Yu. Scaling of supercomputer calculations on unstructured surface computational meshes. *Lobachevskii J. of Math.*, 2021, vol. 42, no. 11, pp. 2571–2579. DOI: 10.1134/S1995080221110202.

Для цитирования

Рыбаков А.А., Фрейлекхман С.А. Автоматическое детектирование аудиодефектов с применением параллельных вычислений // Программные продукты и системы. 2022. Т. 35. № 3. С. 428–437. DOI: 10.15827/0236-235X.139.428-437.

For citation

Rybakov A.A., Freylekhman S.A. Automatic detection of audio defects using parallel computing. *Software & Systems*, 2022, vol. 35, no. 3, pp. 428–437 (in Russ.). DOI: 10.15827/0236-235X.139.428-437.