



Lomonosov Moscow State University

Московский Государственный Университет имени М.В. Ломоносова

**Adaptive noise reduction system
based on multi-resolution FFT**

/

**Система адаптивного подавления
нестационарных шумов на основе
многомасштабных преобразований**

russian edition

Tkachenko Maxim Sergeevich / Ткаченко Максим Сергеевич

makseq@makseq.com

Moscow, Russian Federation / Москва, Россия

Май 2010 / May 2010

Оглавление

1	Аннотация	4
2	Введение	5
3	Постановка задачи.....	6
3.1	Шумоподавление при известном спектре шума	6
3.2	Шумоподавление с адаптацией к меняющемуся спектру шума	7
3.3	Актуальность и новизна.....	8
4	Обзор существующих решений	9
4.1	Шумоподавление при фиксированном спектре шума.....	10
4.1.1	<i>Метод спектрального вычитания</i>	<i>10</i>
4.1.2	<i>Метод рекурсивного сглаживания по времени.....</i>	<i>12</i>
4.1.3	<i>Методы шумоподавления для изображений</i>	<i>12</i>
4.1.4	<i>Многомасштабные преобразования</i>	<i>15</i>
4.2	Шумоподавление с адаптацией к меняющемуся спектру шума	19
4.2.1	<i>Составление слепка шума с помощью адаптивной оценки энергии спектра</i>	<i>19</i>
4.2.2	<i>Составление слепка шума с помощью операций морфологии</i>	<i>21</i>
4.2.3	<i>Поиск связных областей на изображениях.....</i>	<i>22</i>
4.2.4	<i>Распознавание образов на основе байесовской теории решений</i>	<i>24</i>

5	Исследование и построение решения задачи.....	27
5.1	Шумоподавление при известном спектре шума	27
5.1.1	<i>Оценка спектра шума с использованием минимума энергии</i>	<i>28</i>
5.1.2	<i>Оценка спектра шума с использованием разреженности спектра</i>	<i>29</i>
5.1.3	<i>Многомасштабные преобразования</i>	<i>30</i>
5.1.4	<i>Результаты работы системы.....</i>	<i>31</i>
5.2	Шумоподавление с адаптацией к меняющемуся спектру шума	32
5.2.1	<i>Поиск сигнала и шума с помощью оценки разреженности спектрограммы</i>	<i>33</i>
5.2.2	<i>Поиск сигнала и шума на основе выделения связных областей по энергии.....</i>	<i>34</i>
5.2.3	<i>Дополнительная обработка полученного спектра шума</i>	<i>38</i>
5.2.4	<i>Результаты работы системы.....</i>	<i>41</i>
6	Описание практической части.....	43
6.1	Шумоподавление при известном спектре шума	43
6.1.1	<i>Общая структура программы.....</i>	<i>43</i>
6.1.2	<i>Оконное преобразование Фурье.....</i>	<i>44</i>
6.1.3	<i>Модули шумоподавления.....</i>	<i>44</i>
6.1.4	<i>Анализатор и микшер.....</i>	<i>46</i>
6.2	Шумоподавление с адаптацией к меняющемуся спектру шума	49
6.2.1	<i>Поиск сигнала и шума с помощью оценки разреженности спектрограммы</i>	<i>51</i>
6.2.2	<i>Поиск сигнала и шума на основе выделения связных областей по энергии.....</i>	<i>53</i>

6.3	Web приложение	54
6.4	Инструментарий разработки	55
7	Заключение	56
8	Литература	57
9	Приложение А. Нахождение транзиентов системой шумоподавления	59
10	Приложение Б. Адаптивный слепок шума	60

1 Аннотация

Основная цель работы – построение, исследование и реализация алгоритмов очистки звукового сигнала от шума. Рассматриваемые методы ориентированы на музыкальные записи и использование в профессиональной звукорежиссерской практике.

В первой части работы описываются простые способы очистки фонограмм от аддитивных стационарных помех, шумоподавление для изображений, адаптированное под аудиозадачи, а также многомасштабные преобразования, позволяющие достигать более качественных результатов при использовании оконного преобразования Фурье.

Во второй части строятся оригинальные подходы, с помощью которых можно отделить нестационарный шум от полезного сигнала, а затем наиболее эффективно использовать алгоритмы, описанные в первой части.

На основе изучаемых методов разработана система, обладающая рядом новшеств и ориентированная на решение широкого класса задач очистки звука от шума.

2 Введение

Интенсивный прогресс технологий аудио- и киноиндустрии, обработки медицинских данных, radar и sonar систем, а также телекоммуникаций стимулирует активное развитие алгоритмов цифровой обработки звука, одной из самых востребованных областей которых является шумоподавление.

Шум – это нежелательный сигнал, который возникает при передаче или измерении другого, чистого сигнала. Зашумление может появляться из-за некачественной линии передачи информации или интерференции с другими сигналами-наводками [1].

Искажения и помехи являются главными ограничивающими факторами в коммуникационных и измерительных системах, по этой причине теория и практика устранения шума является важными в разработке подобных систем.

Шум бывает разных видов: акустический, электромагнитный, электростатический и т.д. Также различают помехи по их спектральным свойствам: стационарные (не меняющиеся во времени) и нестационарные.

Аддитивным называют шум, который можно представить так: $x(t) = s(t) + w(t)$, где $x(t)$ – зашумлённый сигнал, $s(t)$ – чистый сигнал, $w(t)$ – сигнал шума, не зависящий от чистого сигнала. Примерами аддитивных стационарных шумов могут являться постоянное шипение микрофона или усилительной аппаратуры, гул электросети. Работа различных приборов, не меняющих звучания во времени (вентиляторы, бытовые устройства), также может создавать шумы, близкие к стационарным [1, 2].

В данной работе будут рассмотрены методы подавления аддитивных стационарных и нестационарных шумов.

3 Постановка задачи

Основная задача – разработка системы шумоподавления. В качестве входных данных программе задается зашумлённый сигнал. Затем производится автоматический поиск *слепок шума* (шумовых характеристик, footprint). После этого необходимо выполнить очистку сигнала, в соответствии с указанными ниже особенностями и свойствами. Система включает два класса алгоритмов:

1. Шумоподавление при известном фиксированном спектре шума.
2. Шумоподавление с адаптацией к меняющемуся спектру шума.

3.1 Шумоподавление при известном спектре шума

Для получения качественных результатов необходимо задействовать банки фильтров с адаптивным частотно-временным разрешением (multi-resolution FFT [3]), а в качестве фильтра для шумоподавления использовать несколько подходов:

- Простой метод спектрального вычитания [4, 5],
- Рекурсивный метод спектрального вычитания,
- Метод шумоподавления для изображений [6].

Программа должна иметь гибкую систему конфигурирования и позволять варьировать её многочисленные параметры. А также удовлетворять следующим требованиям:

- Работа алгоритма осуществляется с высокой скоростью (в реальном времени),
- Интерфейс программы имеет возможность графического исследования всплесков энергии, т.н. транзиентов.

3.2 Шумоподавление с адаптацией к меняющемуся спектру шума

Необходимо разработать алгоритмы, способные различать шум и полезный сигнал, после чего шум можно будет исключить. Спектр шума меняется во времени, что усложняет поставленную задачу.

Допускаются разные решения этой задачи, в том числе и те, которые основаны на цепях Маркова, классификаторах и т.д. [7, 1]. Важно, чтобы алгоритм верно распознавал зашумленные участки сигнала и принимал решение о том, какой слепок шума использовать для очистки в каждый конкретный момент времени.

Когда решение принято, с подобранными слепками шума работает система, описанная в п. 3.1 (см. рис. 3.1).

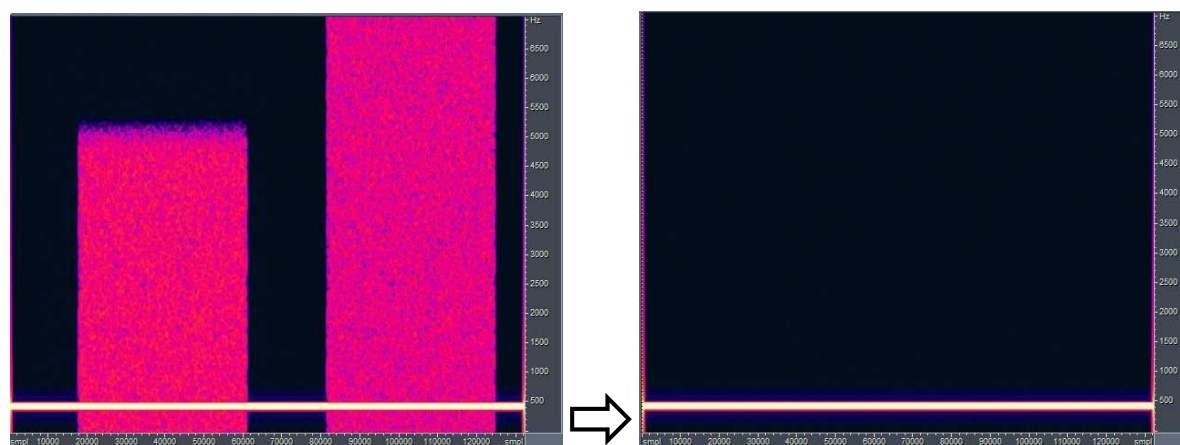


Рис. 3.1. Пример восстановления аудио-сигнала с использованием адаптивно выбираемого слепка шума. Слева – зашумлённая спектрограмма, справа – желаемый результат после очистки.

3.3 Актуальность и новизна

Адаптация методов из области обработки изображений под задачи аудио на данный момент активно развивающаяся область. Соответствующие алгоритмы пока что нечасто встречаются в коммерческих программах. Исследование их работы – важная и интересная задача.

Многомасштабные преобразования – эффективный инструмент для работы со звуком. С их помощью можно наиболее точно обработать сигнал. Не все тонкости функционирования этого механизма хорошо изучены.

На данный момент не существует общепризнанного подхода в нахождении меняющегося во времени шума, и практически все имеющиеся решения, в т. ч. коммерческие продукты, не предоставляют качественных средств по устранению нестационарных помех. В связи с этим получение новых научных результатов и методов в этой области – новая и актуальная задача.

4 Обзор существующих решений

Рынок систем цифрового шумоподавления насыщен и не перестаёт развиваться. Но немногие компании располагают действительно качественными продуктами и разрабатывают принципиально новые эффективные алгоритмы. Среди успешных разработок нужно отметить TC Electronic BackDrop, Waves Z-Noise, Algorithmix NoiseFree, iZotope RX, SoundSoap 2, разработки CEDAR.

Также есть множество алгоритмов для изображений, позволяющих добиться высокого качества устранения шума при адаптации их к аудиосигналам (к примеру, Non-Local Means).

Существуют и принципиально другие способы устранения шума, к примеру, с помощью широкополосного компрессора. Такие системы основаны не на методе спектрального вычитания слежка шума, а на эффекте маскировки и приносят в результирующий сигнал нежелательные модуляционные искажения [8].

Систем с адаптивным выбором спектра шума достаточно мало, и они опираются на простые принципы отделения помех от сигнала. Обычно классификация производится по энергии: если она опускается ниже некоторого порогового значения, то фрагмент помечается как зашумленный.

Подобным образом устроена одна из американских запатентованных систем: если энергия в середине частотного диапазона превышает некоторую константу, то низкочастотный фильтр отключается [9]. Это позволяет избавиться, к примеру, от гула электросети в паузах между речевыми сигналами.

Рассмотрим подходы, которые наиболее приближены к изучаемой тематике. Все главы для удобства разбиты на два раздела:

1. Шумоподавление при известном спектре шума;
2. Шумоподавление с адаптацией к меняющемуся шуму.

4.1 Шумоподавление при фиксированном спектре шума

В основе этого раздела положено предположение, что в среднем спектр шума не изменяется во времени. Его следует задать один раз (к примеру, с помощью усреднения вручную указанного фрагмента, содержащего только помехи), после чего с ним производится спектральное вычитание для всей аудиозаписи.

Метод спектрального вычитания не всегда оказывается достаточно эффективным и для его усовершенствования возможно использование многомасштабных преобразований и различных методов сглаживания спектра, которые рассматриваются ниже.

4.1.1 Метод спектрального вычитания

Метод спектрального вычитания является широко используемым и распространённым [1, 4, 5].

Аддитивные стационарные шумы – порождаемые окружающей средой, звукозаписывающей аппаратурой и т. д. Стационарность означает, что свойства шума (мощность, спектральный состав) не меняются во времени. Аддитивность означает, что шум суммируется с "чистым" сигналом $y[t]$ и не зависит от него:

$$x[t] = y[t] + noise[t], \text{ где } t - \text{ время.}$$

Для подавления аддитивных стационарных шумов используется алгоритм спектрального вычитания. Он состоит из следующих стадий:

1. Разложение сигнала с помощью кратковременного (оконного) преобразования Фурье (STFT), компактно локализирующего энергию сигнала [2].
2. Составление вычитаемого слепок шума (noise footprint). Слпок шума получается усреднением по времени амплитуд спектра,

взятых из заранее подготовленного участка шума, не содержащего полезного сигнала.

$$footprint[f] = \left(\sum_{t=1}^K noise[f, t] \right) / K, \text{ где}$$

$noise[f, t]$ – спектр шума, f – индекс преобразования Фурье, соответствующий частоте, t – номер текущего STFT окна, K – количество окон в участке с шумом.

3. "Вычитание" (в обобщенном смысле) амплитудного спектра шума из амплитудного спектра сигнала.
4. Обратное преобразование STFT - синтез результирующего сигнала.

Вычитание амплитудных спектров осуществляется по формуле:

$$Y[f, t] = \max\{X[f, t] - k \cdot W[f, t], 0\}$$

Здесь $X[f, t]$ и $W[f, t]$ – амплитудные спектры сигнала и шума соответственно, $Y[f, t]$ – амплитудный спектр результирующего очищенного сигнала, а k – коэффициент подавления. Фазовый спектр очищенного сигнала полагается равным фазовому спектру зашумленного сигнала. Результат работы МСВ см. на рис. 4.1.

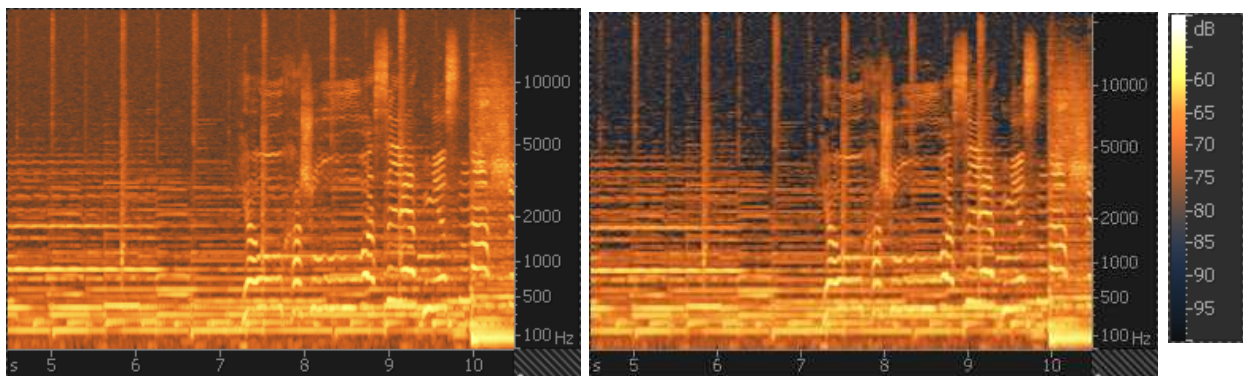


Рис. 4.1. Спектрограммы зашумленного сигнала (слева) и очищенного сигнала (справа)

4.1.2 Метод рекурсивного сглаживания по времени

Одна из проблем метода спектрального вычитания – "музыкальный шум". Он появляется вследствие того, что коэффициенты STFT шумовых сигналов статистически случайны. Это приводит к их неравномерному подавлению. В результате, очищенный сигнал содержит кратковременные и ограниченные по частоте всплески энергии, которые на слух воспринимаются как "колокольчики" или "льющаяся вода" (см. рис.4.2).

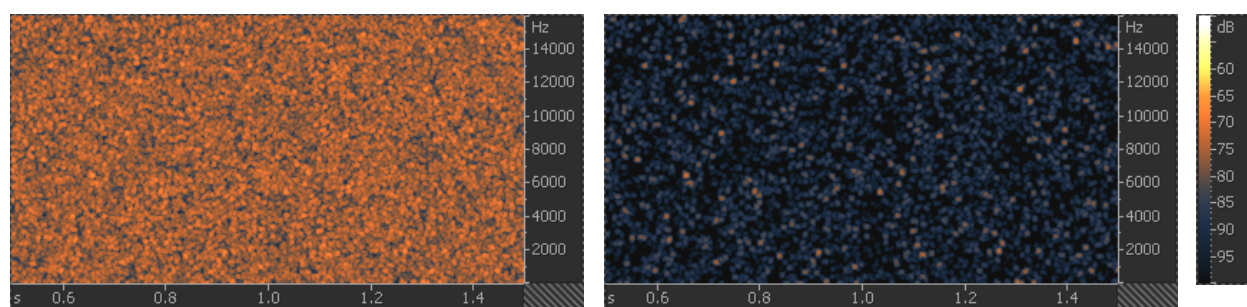


Рис. 4.2. Слева – белый шум. Справа - музыкальный шум, полученный после применения простого метода спектрального вычитания

Для подавления этого артефакта применяется метод рекурсивного сглаживания первого порядка по времени [1]:

$$S[f,t] = S[f,t-1] + \alpha(X[f,t] - S[f,t-1]),$$

где $S[f,t]$ – сглаженный спектр сигнала, а $X[f,t]$ – f -тая амплитуда спектра сигнала в момент времени t , α – коэффициент усреднения, f – индекс преобразования Фурье, соответствующий частоте, t – время.

Далее из усредненного спектра $S[f,t]$ вычитается слепок шума и производится обратное преобразование Фурье (iSTFT) с учетом перекрытия окон.

4.1.3 Методы шумоподавления для изображений

Аудиозапись может быть представлена в виде спектрограммы, которая можно рассматривать, как изображение. Шум на ней по своим

свойства схож с шумом на изображениях, так как имеет одинаковую природу происхождения.

Таким образом, для очистки спектрограммы могут быть использованы те же методы, что и для изображений: начиная от простейшей медианной фильтрации, заканчивая высокоэффективными Non-Local Means (NLM) и BM3D алгоритмами (рис. 4.3) [10, 11].



Рис. 4.3. Примеры работы алгоритмов шумоподавления для изображений. Слева - направо, сверху - вниз: оригинальное зашумленное изображение, Гауссова фильтрация, анизотропная фильтрация, метод полной вариации, билатеральная фильтрация, Non-Local Means

Основная идея NLM и подобных ему методов заключается в усреднении каждого пикселя, с учетом окружающих соседей или соседних блоков. Благодаря такому подходу сохраняется геометрическая структура изображения, не размываются границы, появляется возможность восстанавливать поврежденные участки изображения.

Рассмотрим более подробное описание работы алгоритма NLM [10]. Предположим, что $v[i]$ – зашумленное изображение, $u[i]$ – исходное изображение, $noise[i]$ – наложенный шум.

$$v[i] = u[i] + noise[i]$$

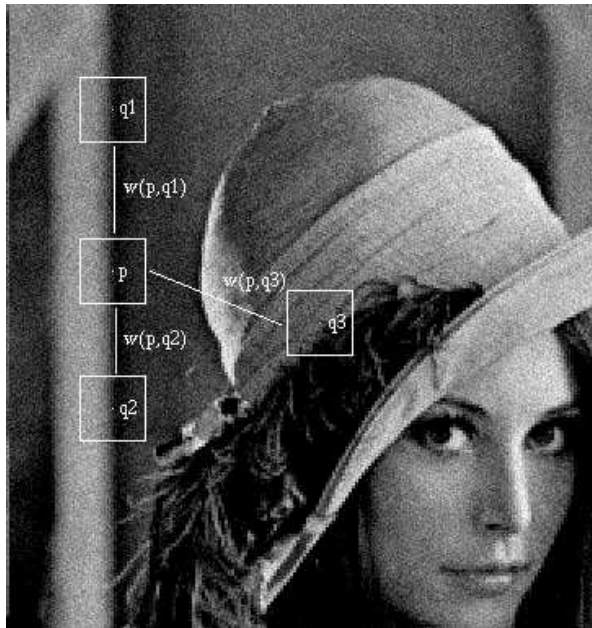


Рис. 4.4. Non-Local Means. $w(i,j)$ – веса для усреднения, $q1, q2, q3$ – анализируемые блоки, p –пиксель, для которого вычисляется вес $w(i,j)$.

Пусть i – номер обрабатываемого пикселя, а I – множество всех пикселей изображения. η_i – текущий блок, содержащий анализируемый пиксель, η_j – анализируемый блок, с которым происходит попиксельное сравнение текущего η_i (см. рис. 4.4).

Тогда для любого i справедливо:

$$NL[v](i) = \sum_{j \in I} w(i, j)v(j), \quad \text{где}$$

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(\eta_i) - v(\eta_j)\|_{2,\alpha}^2}{h^2}} \quad \text{– вес анализируемого блока и}$$

$$Z(i) = \sum_j \frac{1}{Z(i)} e^{-\frac{\|v(\eta_i) - v(\eta_j)\|_{2,\alpha}^2}{h^2}} \quad \text{– нормировка, для того, чтобы}$$

избежать изменения интенсивности всего изображения.

Для $w(i, j)$ всегда выполняется

- $0 \leq w(i, j) \leq 1$
- $\sum_j w(i, j)$

Подобие между блоками

$$\|v(\eta_i) - v(\eta_j)\|_{2,\alpha}^2$$

является нормой в евклидовом пространстве, где α – стандартное отклонение ядра Гаусса.

NLM широко используется в обработке изображений и является одним из самых эффективных алгоритмов устранения шума. Разработаны его различные модификации, которые позволяют производить обработку в режиме реального времени.

4.1.4 Многомасштабные преобразования

При использовании банков с постоянным разрешением невозможно получить одновременно хорошее разрешение как по частоте, так и по времени. Фильтры с фиксированным разрешением приводят к нежелательному размытию сигнала в частотной и временной областях (см. рис. 4.5).

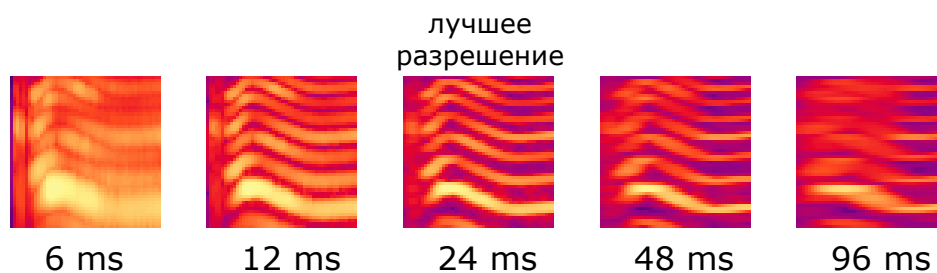


Рис. 4.5. Размытие спектра и выбор лучшего разрешения

Банки фильтров с адаптивным частотно-временным разрешением, в частности многомасштабные преобразования (multi-resolution FFT = MR FFT), помогают решить эту проблему [3, 12].

В основе MR FFT лежат

- вычисление STFT, взятых от одного сигнала, с разными разрешениями (размерами окна),
- оценка степени разреженности соответствующих областей спектра,
- микширование спектрограмм с учетом их разреженности.

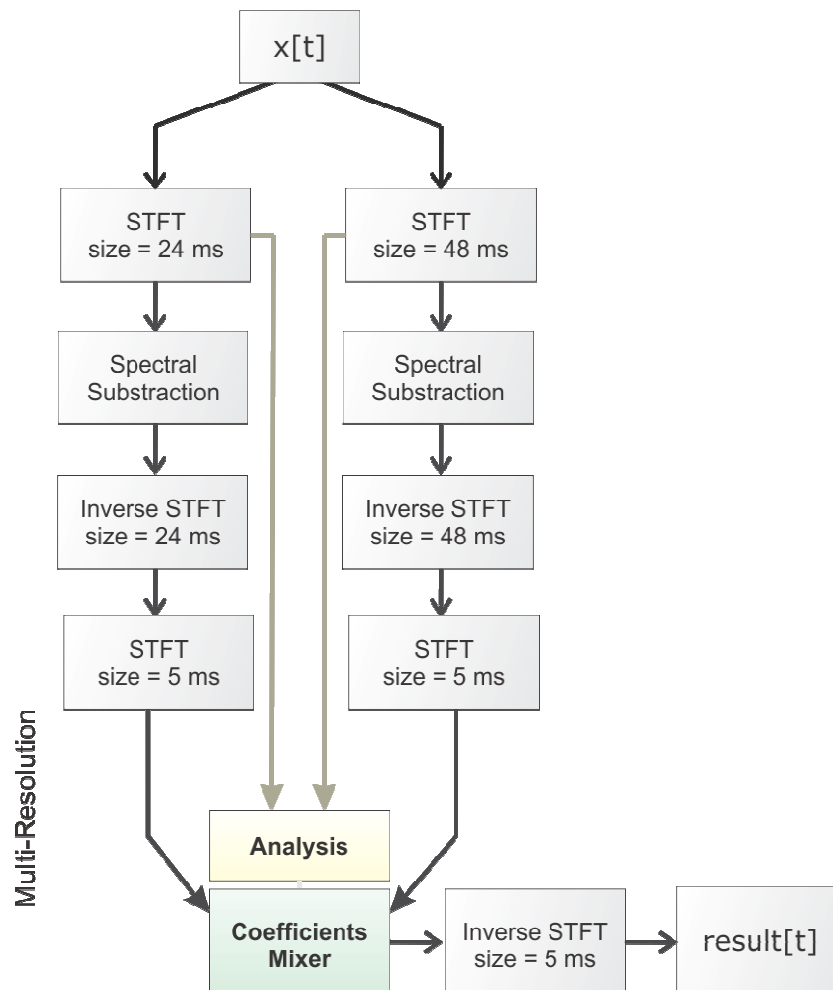


Рис. 4.6. Схема работы многомасштабных преобразований

Вначале выполняется независимая друг от друга работа эффектов (фильтров), с фиксированными разрешениями (размерами FFT):

$$STFT \Leftrightarrow \text{Применение Эффекта} \Leftrightarrow \text{inverse STFT}.$$

Результирующий сигнал каждого из эффектов передаётся в микшер.

Запускается анализатор, рассчитывающий коэффициенты микширования, опираясь на степени разреженности областей спектрограмм (см. рис. 4.6).

Степень разреженности (размытия) для каждой области Ω вычисляется, как отношение среднего арифметического модулей коэффициентов $a_i \in \Omega$ к их среднему квадратичному.

$$S_{\Omega} = \frac{\sum_{i=1}^N a_i}{\sqrt{N} \sqrt{\sum_{i=1}^N a_i^2}}$$

В дальнейшем такая мера часто используется, поэтому рассмотрим её свойства более подробно. Так как среднее арифметическое всегда меньше или равно среднему квадратичному, то $S_{\Omega} \leq 1$. В данной работе S_{Ω} всегда вычисляется для спектрограмм, амплитудные коэффициенты которых больше или равны 0. Следовательно, $0 \leq S_{\Omega} \leq 1$.

Чем меньше $|a_i - a_j|$ для любых i, j из Ω (т.е. все значения достаточно близки друг к другу), тем сильнее S_{Ω} приближается к 1. Это соответствует размытому спектру, как, например, у белого шума. При наличии малого числа больших по величине коэффициентов a_i мера S_{Ω} становится малой, что показывает компактную локализацию энергии спектра в малом числе коэффициентов, как, например, для тональных сигналов или коротких всплесков энергии.

Поведение меры размытия от двух переменных см. на рис. 4.7. При большом количестве значений в области Ω , минимум S_Ω стремится к 0.

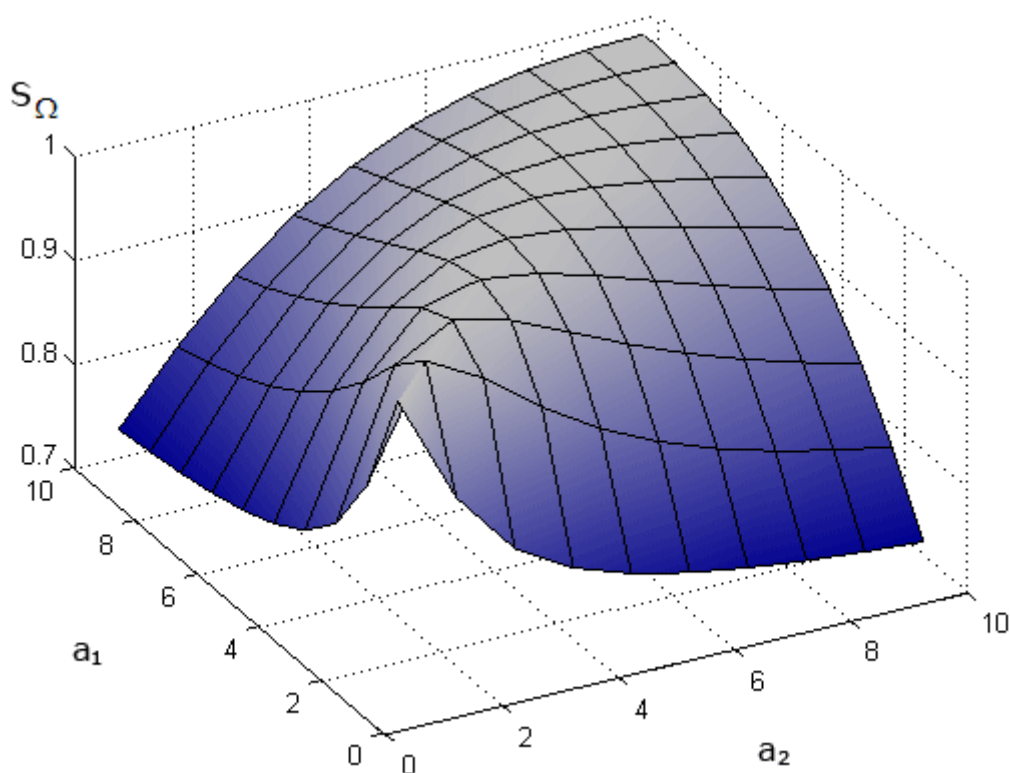


Рис. 4.7. Мера размытия, зависящая от двух коэффициентов.

Для каждой из спектрограмм вычисляется свой коэффициент микширования: $\alpha = 1/S_\Omega$ и $\sum_{i=1}^N \alpha_i = 1$, где N – количество анализируемых спектрограмм (т.е. количество эффектов).

Таким образом, для длинных тональных сигналов будет подбираться FFT с наибольшим разрешением по частоте, а для резких всплесков энергии (щелчков, событий с быстрой атакой) – с наименьшим по частоте, но наилучшим по времени.

Использование этого подхода весьма эффективно, и его работа даже в совокупности с простым методом спектрального вычитания оказывается высококачественной.

4.2 Шумоподавление с адаптацией к меняющемуся спектру шума

На данный момент стандартного решения этой задачи не существует, но есть множество способов, позволяющих при некоторых предположениях классифицировать шумы и полезные сигналы.

Необходимо определить критерии, по которым различаются шум и сигналы. Предполагается, что полезные аудиофрагменты воспринимаются на слух громче, чем помехи. Следовательно, по уровню энергии они стоят на несколько децибел выше. В основе подразделов 4.2.1 "Составление слепка шума с помощью адаптивной оценки энергии спектра" и 4.2.2 "Составление слепка шума с помощью операций морфологии" лежит именно такое предположение.

Части 4.2.3 "Поиск связных областей на изображениях" и 4.2.4 "Распознавание образов на основе байесовской теории решений" используют иные принципы классификации, хотя основным главным критерием остаётся энергия спектра.

4.2.1 Составление слепка шума с помощью адаптивной оценки энергии спектра

Критерий отличия шума от полезного сигнала: все частоты спектра, которые по амплитуде меньше некоторого порогового значения, подлежат удалению.

Можно получить более хороший результат, если модифицировать метод: разбить спектр на несколько частотных полос и для каждой из них подбирать свои пороговые значения.

Авторы статьи Noise Reduction of Speech Signals предлагают следующий подход [13].

Вначале необходимо провести классификацию шума и сигнала. Для каждого i -того фрейма вычисляется $E^2(i)$,

$$E^2(i) = \frac{1}{k_1 - k_0} \sum_{k=k_0}^{k_1} [|X(i, k) - \bar{X}(i, k)]^2,$$

$$\bar{X}(i) = \frac{1}{k_1 - k_0 + 1} \sum_{k=k_0}^{k_1} |X(i, k)|,$$

где $X(i)$ спектр сигнала, k_0 и k_1 – константы, выбираемые в зависимости от конкретной задачи (для речи эти константы пропорциональны 300 и 3000 Гц соответственно).

После того, как $E^2(i)$ рассчитаны, производится отсечение по пороговому значению: если $E(i) > T$, то фрейм i классифицируется, как сигнал. Иначе фрейм принимается за шум. Значение T находится из следующей формулы:

$$T = E_{\min} + \delta(E_{\max} - E_{\min}),$$

$$E_{\min} = \min\{E(j)\}, \quad E_{\max} = \max\{E(j)\},$$

$$j = i - W_e, i + 1 - W_e, \dots, i - 1.$$

Здесь W_e – размер окна, δ – константа. Благодаря увеличению размера окна, можно избежать удаления тихих по энергии участков сигнала.

После классификации используется винеровская фильтрация, являющаяся аналогом спектрального вычитания.

Алгоритм решает проблему нахождения и удаления нестационарного шума, однако он хорошо работает только с речевыми, плавно меняющимися во времени сигналами. Кроме того, происходят сильные искажения амплитуды, что оказывается недопустимым в профессиональной звукорежиссерской практике.

4.2.2 Составление слепка шума с помощью операций морфологии

Составление слепка шума возможно при помощи операций математической морфологии. Рассмотрим спектр сигнала в момент времени t и применим к амплитудным коэффициентам фильтр бегущего минимума, который является аналогом операции эрозии [14].

Изначально положим, что всё множество амплитудных коэффициентов спектра – помехи, и они все подходят для слепка шума. Попытаемся исключить из этого множества всё, что может оказаться сигналом, опираясь на предположение о его высокой энергии в сравнении с шумом.

Размер окна фильтра бегущего минимума равен W , которое выбирается в соответствии с размерами FFT. Найдём максимум, среди значений спектра $X[f,t]$ (f – текущая частота):

$$p = \arg \max \{ X[f,t], X[f+1,t], \dots, X[f+W,t] \}$$

Зная, что локальный максимум располагается в точке p , можно заменить его локальным минимумом. Смещая f на половину W до конца частотного диапазона, производя такую замену, получим спектр шума.

Алгоритм можно усовершенствовать, если оперировать не со спектром, а со спектрограммой как с изображением: искать локальный максимум не в одномерном пространстве, а в двумерном. Интересных результатов можно добиться при вариации W , а также при использовании второго дополнительного окна, сглаживающего по краям фильтр бегущего минимума (к примеру, это может быть окно Ханна).

4.2.3 Поиск связных областей на изображениях

В дальнейшем в работе будут использоваться понятия смежности пикселей, связности и т. д. Рассмотрим эти основы цифрового представления изображений [15].

Чтобы установить, что два элемента изображения являются смежными, необходимо, чтобы они были соседями и их уровни яркости удовлетворяли заданному критерию сходства (скажем, были равны друг другу).

Пусть V — множество значений яркости, используемое при определении понятия смежности. В бинарном изображении $V = \{1\}$, если смежными считаются соседние пиксели с единичным значением яркости. Для полутоновых изображений идея та же, но множество V обычно состоит из большего числа элементов. Например, при определении понятия смежности для пикселей с диапазоном возможных значений яркости от 0 до 255 множество V может быть любым подмножеством этих 256 значений. Мы будем рассматривать три вида смежности:

1) *4-смежность*. Два пикселя p и q со значениями из множества V являются 4-смежными, если q входит в множество $N_4(p)$;

2) *8-смежность*. Два пикселя p и q со значениями из множества V являются 8-смежными, если q входит в множество $N_8(p)$;

3) *m -смежность (смешанная)*. Два пикселя p и q со значениями из множества V являются m -смежными, если:

- элемент q входит в множество $N_4(p)$, или
- элемент q входит в множество $N_D(p)$ и множество $N_4(p) \cap N_4(q)$ не содержит элементов изображения со значением яркости из множества V .

Смешанная смежность представляет собой модификацию 8-смежности с целью исключения неоднозначности, часто возникающей

при использовании 8-смежности в чистом виде. Рассмотрим, например, изображенную на рис. 4.8 (а) конфигурацию пикселей при $V = \{1\}$. Три элемента в верхней части рис. 4.8 (б) демонстрируют неоднозначную 8-смежность, как указано пунктирными линиями. Эта неоднозначность устраняется при использовании m -смежности, что иллюстрирует рис. 4.8 (в). Два подмножества пикселей S_i и S_j называются *смежными*, если некоторый пиксель из S_i является смежным с некоторым пикселем из S_j .

Дискретным путем (или кривой) от пикселя p с координатами (x, y) до пикселя q с координатами (s, t) называется неповторяющаяся последовательность пикселей с координатами $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, где $(x_0, y_0) = (x, y)$, $(x_n, y_n) = (s, t)$ и пиксели (x_i, y_i) и (x_{i-1}, y_{i-1}) являются смежными при $1 < i < n$. В этом случае n называется длиной пути.

Пусть S — некоторое подмножество элементов изображения. Два его элемента p и q называются *связными* в S , если между ними существует путь, целиком состоящий из элементов подмножества S . Для любого пикселя p из S множество всех пикселей, связанных с ним в S , называется *связной компонентой* (или компонентой связности) S . Если множество S содержит только одну компоненту связности, оно называется *связным множеством*.

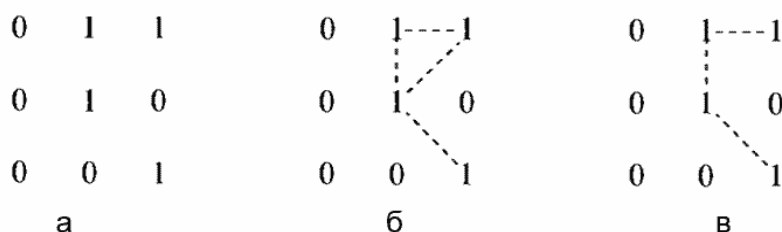


Рис 4.8. (а) Конфигурация пикселей, (б) Элементы, являющиеся 8-смежными между собой (показано пунктиром), (в) Отношения m -смежности

Оперируя вышеизложенными понятиями можно разработать множество алгоритмов для поиска связанных областей [19].

4.2.4 Распознавание образов на основе байесовской теории решений

Обратимся к вероятностным моделям и рассмотрим байесовскую теорию решений, как возможный вариант классификации шума и сигнала [16].

Байесовская теория принятия решений составляет основу статистического подхода к задаче классификации образов. Этот подход основан на предположении, что задача выбора решения сформулирована в терминах теории вероятностей и известны все представляющие интерес вероятностные величины.

Остановимся на конкретном примере. Построим классификатор, различающий два вида древесины — ясень и березу. Предположим, что наблюдателю, следящему за выпуском древесины с завода, представляется настолько трудным предсказать, какого вида древесина появится следующей, что последовательность ее видов кажется ему случайной. Используя терминологию теории решений, можно сказать, что появление куска древесины того или иного вида означает, что природа пришла в одно из двух состояний — древесина оказывается либо ясенем, либо березой. Обозначим состояния природы символом w , причем для ясеня $w = w_1$, а для березы $w = w_2$. Состояние w может рассматриваться как случайная величина в том смысле, что состояние природы не предсказуемо.

Если фабрика выпускает ясеня столько же, сколько березы, то можно сказать, что следующий кусок в равной мере может оказаться или ясенем, или березой. В общем случае предположим, что существует некоторая априорная вероятность $P(w_1)$ того, что следующий кусок окажется ясенем, и $P(w_2)$ — что это будет береза. Эти априорные вероятности отражают исходное знание того, с какой степенью уверенности можно предсказать ясень или березу до их действительного появления. Предполагается, что величины $P(w_1)$ и $P(w_2)$ неотрицательны и сумма их равна единице.

Допустим сначала, что требуется решить, какой из видов древесины появится следующим, не видя ее. Единственная информация, которой мы располагаем, это величины априорных вероятностей. Если решение необходимо принять, исходя из столь малой информации, то разумно воспользоваться следующим решающим правилом: принять решение w_1 , если $P(w_1) > P(w_2)$, и w_2 в противном случае.

Эта процедура может показаться странной в том смысле, что в любом случае принимается одно и то же решение, хотя и известно о возможности появления обоих видов древесины. Насколько она хороша, зависит от величины априорных вероятностей. Если $P(w_1)$ намного больше, чем $P(w_2)$, то наше решение отдавать предпочтение w_1 должно большей частью оправдываться. Если $P(w_1) = P(w_2)$, то у нас 50 шансов из 100 быть правыми. Вообще вероятность ошибки равна в данном случае меньшей из величин $P(w_1)$ и $P(w_2)$, а из дальнейшего станет видно, что при таких условиях никакое другое правило решения не даст меньшей вероятности ошибки.

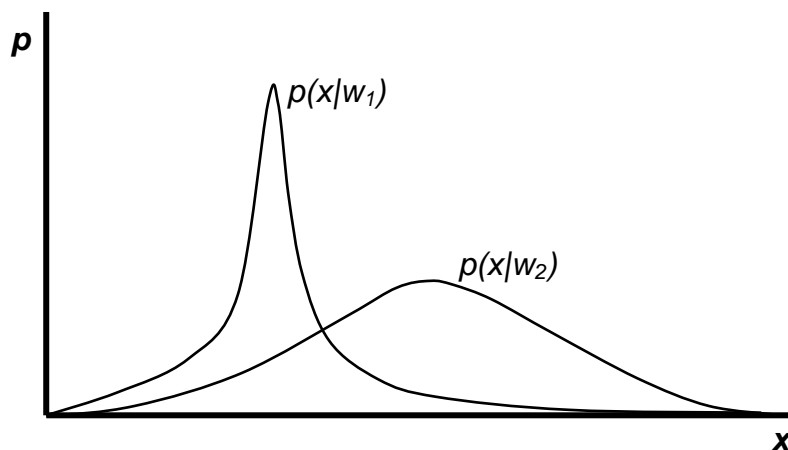


Рис 4.9. Пример плотности распределения, условной по классу

В большинстве случаев при выборе решения не ограничиваются столь малой информацией. В нашем примере в качестве определяющего признака можно взять яркость x оттенка древесины. Разные куски древесины выглядят светлее или темнее, так что естественно выразить это различие с помощью вероятностных

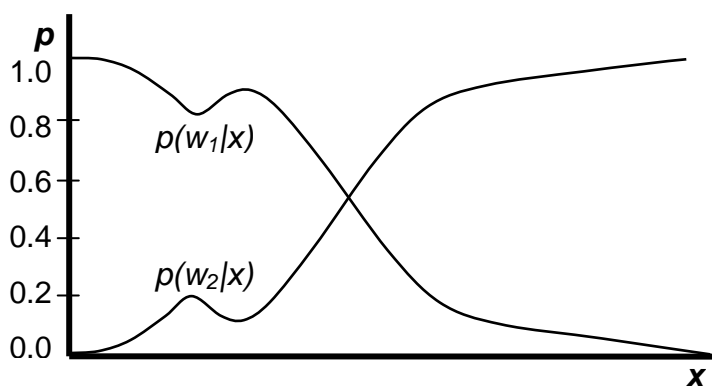
законов, а x рассматривать как непрерывную случайную величину, распределение которой зависит от состояния природы. Пусть $p(x|w_j)$ — условная плотность распределения величины x в состоянии w_j , т. е. функция плотности распределения случайной величины x при условии, что состояние природы есть w_j . В этом случае различие между $p(x|w_1)$ и $p(x|w_2)$ отражает различие яркости оттенков ясеня и березы (рис. 4.9).

Допустим, что известны как априорные вероятности $P(w_j)$, так и условные плотности $p(x|w_j)$. Предположим далее, что мы измеряем яркость оттенка древесины и находим, что это есть x . Ответ на вопрос, в какой мере это измерение повлияет на наше представление об истинном состоянии природы, дает *правило Байеса*:

$$P(w_j | x) = \frac{p(x | w_j)P(w_j)}{p(x)},$$

$$p(x) = \sum_{j=1}^2 p(x | w_j)P(w_j).$$

Правило Байеса показывает, как наличие измеренной величины x позволяет из априорной вероятности $P(w_j)$ получить *апостериорную вероятность* $P(w_j|x)$. Зависимость $P(w_j|x)$ от x для случая $P(w_1)=2/3$ и $P(w_2)=1/3$ показана на рис. 4.10. Если при наблюдении получено значение x , для которого $P(w_1|x)$ больше, чем $P(w_2|x)$, то естественно



склониться к решению, что истинное состояние природы есть w_1 . Аналогично, если $P(w_2|x)$ больше, чем $P(w_1|x)$, то естественно склониться к выбору w_2 .

Рис 4.10. Апостериорные вероятности

5 Исследование и построение решения задачи

5.1 Шумоподавление при известном спектре шума

В данном разделе делается акцент на методы удаления помех из фонограмм, а не способы нахождения шума для его последующего устранения. Предполагается, что шум не изменяется во времени и найти его усредненный спектр не составляет особого труда. В таком случае задача шумоподавления сводится к четырём подзадачам:

1. Заданию слепка шума для стационарных шумов (слепок не зависит от времени),
2. Сглаживанию спектрограммы зашумленного сигнала,
3. Спектральному вычитанию слепка шума из зашумленного сигнала,
4. Использованию многомасштабных преобразований.

Задание слепка шума может производиться несколькими способами. Простейший из них – указание пользователем фрагмента сигнала, содержащего только помехи. Подробнее об этом можно прочесть в обзоре литературы (п. 4.1.1). Ниже рассмотрены автоматические способы составления слепка шума, не меняющегося во времени.

Сглаживание спектрограммы и использование многомасштабных преобразований, которым посвящен этот раздел, повышают качество очистки, но не являются обязательными в использовании.

Раздел завершается обзором полученных результатов очистки фонограмм и особенностями работы алгоритмов, производящих шумоподавление.

В пункте 6 "Описание практической части" рассматриваются величины параметров для методов из этой главы.

5.1.1 Оценка спектра шума с использованием минимума энергии

Оценка спектра шума с использованием минимума энергии широко распространена и позволяет добиться в большинстве случаев хорошего результата.

Введём понятие среднеквадратичной энергии спектра (Root mean square, RMS). Допустим, есть множество амплитудных значений спектра сигнала $X_t = \{x_1, x_2, \dots, x_N\}$ в момент времени t . Тогда

$$RMS[X_t] = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_N^2}{N}}.$$

Вычислим для всех t $RMS[X_t]$ и упорядочим получившиеся значения по возрастанию. В результате получим множество $R = \{ (rms_1, t_1), (rms_2, t_2), \dots, (rms_T, t_T) \}$, где T - длина сигнала.

Введем несколько вспомогательных констант:

- k - определяет верхний порог для RMS ($rms_1 \cdot k$, k выбирается опытным путем),
- n - указывает достаточное количество спектров для усреднения,
- r - количество спектров шума, стоящих поблизости.

Переберем все значения из множества R . Если для какого-нибудь i верно $rms_i < rms_1 \cdot k$, и для всех rms_j из R справедливо $rms_j < rms_1 \cdot k$, $|t_i - t_j| < r$ - то спектры X_i и X_j добавляются к множеству спектров шума $noise$ (рис. 5.1). При этом $noise$ не может содержать больше n элементов.

После этого производится усреднение $noise$ и получается неменяющийся во времени спектр шума $footprint[f]$ (подробнее см. п. 4.1.1), где f - частота.

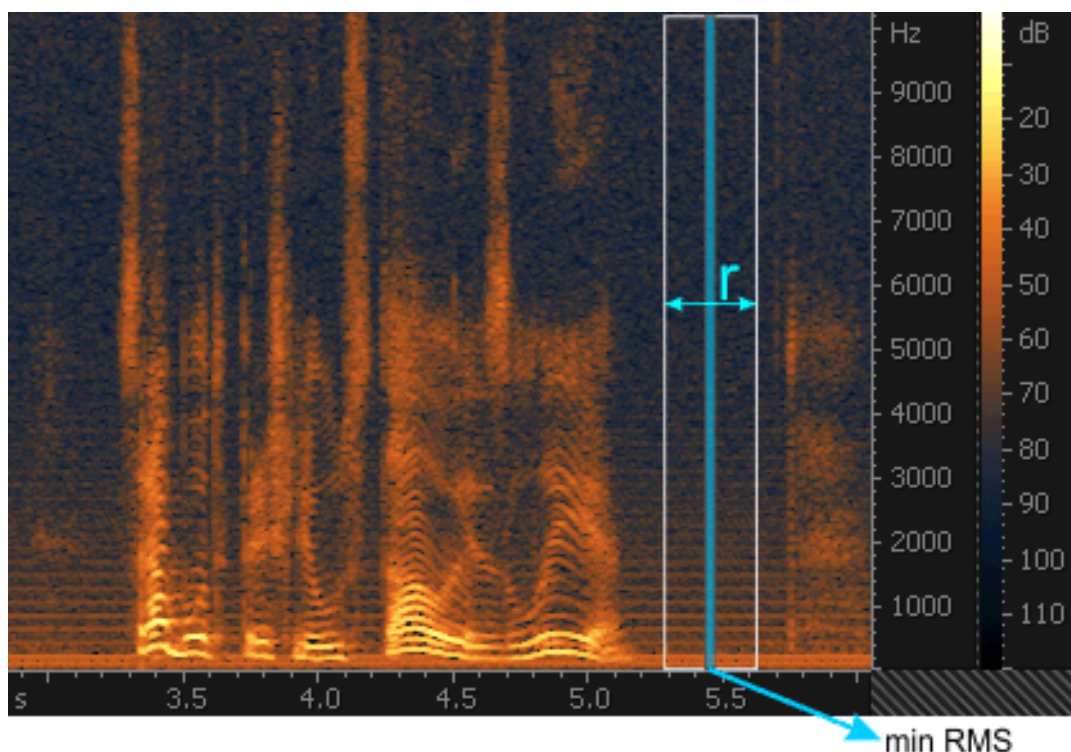


Рис 5.1. Нахождение множества noise. Все RMS спектров из области в прямоугольнике достаточно малы. Вероятно, из них будет составлен слепок шума

5.1.2 Оценка спектра шума с использованием разреженности спектра

Процесс составления спектра шума с использованием разреженности (sparsity) спектра очень похож на составление слепка шума с использованием минимума энергии (предыдущий пункт).

Меняется лишь тип меры, используемый для оценки шума. Предлагается использовать степень размытия для этой цели. Такая оценка позволит наиболее точно определить широкополосные шумы, так как они имеют высокую степень размытия.

Подробно о мере размытия можно прочесть в п. 4.1.4 "Многомасштабные преобразования".

Возможны различные модификации приведенных алгоритмов, но значительных улучшений результата они не дают.

5.1.3 Многомасштабные преобразования

Математическая модель многомасштабных преобразований подробно изложена в обзоре литературы (п. 4.1.4). Опишем некоторые тонкости процессов анализа и микширования спектрограмм.

В результате исследований установлено, что высокое качество очистки достигается при обработке и анализе как минимум трёх спектрограмм с разными размерами STFT. Размер каждого окна STFT выбирается в два раза больше предыдущего, начиная с 12 мс.

Спектрограммы разбиваются по времени и частоте на несколько прямоугольных областей Ω_r . Важный этап – это сопоставление Ω_r и их размеров на разных спектрограммах. Каждая из них имеет размер по частоте в 1/16 от количества частотных полос для конкретной спектрограммы и по времени w :

$$v = (\text{Size}-1) \cdot \frac{\text{MixFFT}}{\text{CurrentFFT}} + \left\lceil \frac{\text{MixFFT}-\text{CurrentFFT}+1}{\text{CurrentFFT} \cdot \text{InvOverlap}} \right\rceil$$
$$w = \begin{cases} v, & v > 1 \\ 1, & v \leq 1 \end{cases}$$

где MixFFT – размер STFT для спектрограммы, в которую суммируется результат, CurrentFFT – размер STFT исследуемой спектрограммы, InvOverlap – обратная величина от степени перекрытия окон STFT (1/4, по умолчанию). Все величины задаются в виде отчетов (samples). Size – размер области Ω_r для спектрограммы с минимальным размером STFT. Параметр Size равный 64 оказывается достаточным для корректной работы анализатора.

Отметим, что суммирование производится в спектрограмму с наименьшим размером STFT, иначе переключение между областями будет происходить не достаточно быстро.

Благодаря ходу с перекрытием на 1/4 по частоте и времени всех Ω_r , возможно плавное переключение при микшировании спектрограмм.

5.1.4 Результаты работы системы

В таблице 1 приведены результаты исследования работы полученной системы. Тестирования проводились с автоматическим поиском спектра шума по минимальной энергии.

<i>Метод</i>	<i>Подавление шума</i>	<i>Комментарий</i>
МСВ 11 мс	13.10 дБ	Размер окна – 11 мс. Недостаточная степень устранения шума.
МСВ 93 мс	13.15 дБ	Размер окна – 93 мс. Появление эхо.
МСВ на основе многомасштабных преобразований	16.10 дБ	Остаётся музыкальный шум. Запись кажется приглушенной.
Метод нелокального среднего на основе многомасштабных преобразований	16.70 дБ	Минимальное количество артефактов.

Таблица 1. Результаты тестирования системы шумоподавления

Система работает в режиме реального времени (поточный режим). Если используются многомасштабные преобразования вместе с обычным МСВ и ручным указанием слепка шума, то задержка при выдаче результата равняется размеру максимальной используемой длине окна эффектов. К примеру, если используется несколько фильтров с размерами FFT 6 мс, 12 мс, 24 мс, то задержка равна 24 мс.

Автоматической системе поиска шума требуется время для обучения. При работе с ней задержка составляет не менее $100 \cdot FFTSize \cdot (1/Overlap + 1)$, где $FFTSize$ – размер окна, $Overlap$ – степень перекрытия окон.

5.2 Шумоподавление с адаптацией к меняющемуся спектру шума

Шумоподавление с адаптацией к меняющемуся спектру шума расширяет п. 5.1. "Шумоподавление при известном спектре шума". Решаемые подзадачи остаются прежними, но оценка слепка шума производится иначе. Основное внимание в этом разделе уделяется поиску нежелательных помех в аудиосигнале.

Предлагается рассматривать спектрограмму, как изображение $I(x,y)$: по оси x откладывается время, а по y частота. Формально – $I(x,y): N^2 \Rightarrow R$, $x=0..m$, $y=0..n$, m , n – ширина и высота изображения. Гораздо удобнее искать полезный сигнал, а не шум. Если удастся определить множество $S \subset I$, относящееся к полезному сигналу, то автоматически находится множество шума $N \subset I$: $N = I \setminus S$.

Введём вспомогательное отображение $M(x,y): N^2 \Rightarrow R$ такое, что $\forall(x,y) \in N^2$ выполняется: $0 \leq M(x,y) \leq 1$. Назовём $M(x,y)$ вероятностной маской шума. Если $M(x,y)=1$, то пиксель изображения $I(x,y)$ – будем считать шумом, если $M(x,y)=0$, то $I(x,y)$ – полезный сигнал. Промежуточные значения $M(x,y)$ указывают на вероятность принадлежности $I(x,y)$ к шуму.

Отметим, что в п. 5.2.2 "Поиск сигнала и шума на основе выделения связанных областей" удобнее использовать обратное представление: если $M(x,y)=0$, то $I(x,y)$ – шум, и если $M(x,y)=1$, то $I(x,y)$ – сигнал.

Предположим, что вероятностная маска шума уже получена. Теперь необходимо рассчитать окончательный спектр шума $Footprint(x,y)$, x – время, y – частота. Для этого необходимо перемножить $I(x,y)$ и $M(x,y)$ попиксельно: $Footprint(x,y) = I(x,y) \cdot M(x,y)$. Далее можно использовать систему шумоподавления, описанную в разделе 5.1, но уже с зависящим от времени слепком шума.

Построим несколько методов нахождения вероятностной маски шума.

5.2.1 Поиск сигнала и шума с помощью оценки разреженности спектрограммы

Для детектирования широкополосных помех удобно использовать оценку разреженности спектрограммы. Принцип её работы очень похож на функционирование анализатора в многомасштабных преобразованиях (см. п. 4.1.4).

Зафиксируем центральный пиксель изображения $I(x_0, y_0)$ и вычислим степень размытия в окрестности $(x_0 + \varepsilon, y_0 + \gamma)$, $-M < \varepsilon < M$, $-N < \gamma < N$, M, N – ширина и высота области:

$$Smooth[I(x_0, y_0)] = \frac{\sum_{x=x_0-M}^{x_0+M} \sum_{y=y_0-N}^{y_0+N} I(x, y)}{\sqrt{M \cdot N} \sqrt{\sum_{x=x_0-M}^{x_0+M} \sum_{y=y_0-N}^{y_0+N} I(x, y)^2}}$$

Результат $Smooth[I(x_0, y_0)]$ можно присвоить маске $M(x_0, y_0)$. Если выполнить эти операции для каждого пикселя из I , то получится вероятностная маска шума (рис. 5.2).

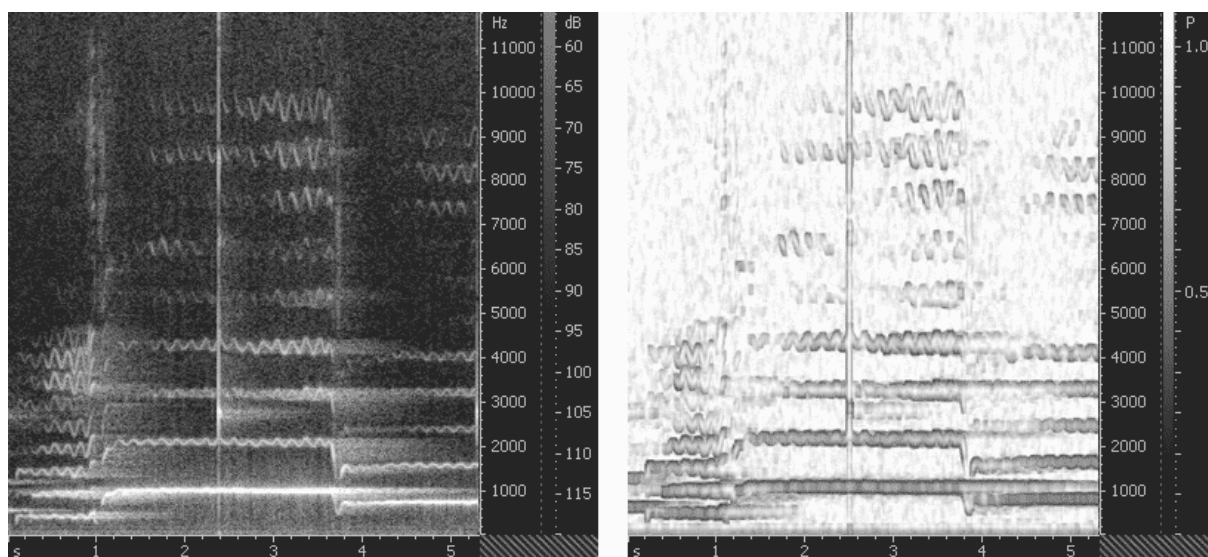


Рис. 5.2. Слева – зашумленный сигнал, справа – вероятностная маска шума, полученная с помощью оценки размытия. Чем белее пиксель маски, тем выше вероятность, что это шум.

5.2.2 Поиск сигнала и шума на основе выделения связанных областей по энергии

Составление вероятностной маски шума состоит из нескольких этапов, которые производятся для всей спектрограммы $I(x,y)$:

1. На спектрограмме $I(x,y)$ выбирается некоторая область Ω ;
2. Находится локальный максимум $I(x_{max}, y_{max}) = \max$ на Ω ,
3. Производится оценка максимума,
4. Производится оценка степени размытия области $\theta \subset \Omega$ с центром в точке (x_{max}, y_{max}) ,
5. В случае положительных результатов оценок запускается составление маски, с помощью рекурсивного поиска связанных областей по энергии,
6. Из маски удаляются множества связанных пикселей, имеющие маленькую мощность,
7. Из маски удаляются "всплески", область вокруг которых имеет низкую энергию.

1) *Выбор области Ω* . Область Ω должна иметь размер, при котором сигнал разбивается на некоторые логические конструкции.

Область Ω обходит изображение и выполняются этапы 1-5. За счет этого производится его полный анализ и расчет вероятностной маски. Как показали исследования, более лучшего результата можно добиться при перекрытии областей на один отчет по оси x и y .

2) *Нахождение локального максимума $I(x_{max}, y_{max}) = \max$ на Ω* . Основной критерий определения полезного сигнала на спектрограмме – высокое значение пикселя $I(x,y)$. Это указывает на то, что в некоторой окрестности в $I(x,y)$ будет достигаться максимум (см. рис. 5.3).

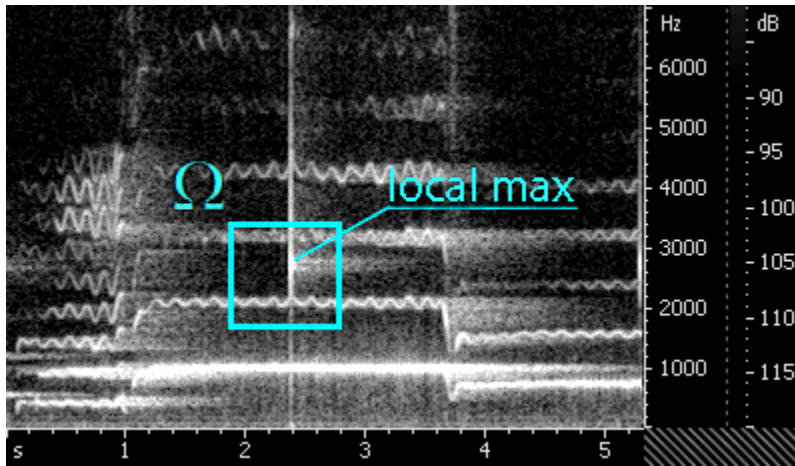


Рис. 5.3. Нахождение локального максимума на спектрограмме в области Ω

Для того, чтобы исключить нахождения одного и того же максимума (x_{\max}, y_{\max}) у расположенных поблизости областей Ω , предварительно необходимо умножить Ω на окно, спадающее по краям (к примеру, окно Ханна).

3) *Оценка максимума.* Важно понять, при каком условии локальный максимум не является всплеском шума, то есть необходимо исключить "ложные срабатывания".

Предположим, что в области оказался исключительно шумовой фрагмент. Тогда значения $I(x, y)$ из Ω будут близки друг к другу. Это означает, что $|I(x, y) - a| < \varepsilon$ для $\varepsilon > 0$, где a – среднее арифметическое значений $I(x, y)$ на Ω (см. рис. 5.4).



Рис. 5.4. Фрагмент шума в области Ω

Предлагается сравнивать локальный максимум из Ω со средним арифметическим значений $I(x, y)$ на Ω . Это позволит исключить большинство ложных срабатываний рекурсивного поиска связанных областей.

4) *Оценка степени размытия.* Ещё одним критерием, который позволяет лучше детектировать полезный сигнал, является мера размытия. Чем выше степень размытия для окрестности точки (x_{\max}, y_{\max}), тем вероятнее, что в этой окрестности находится шум.

5) *Рекурсивный поиск связных областей по энергии.* При помощи двух оценок, рассмотренных выше, можно качественно определить, принадлежит ли локальный максимум полезному сигналу. Если это так, то необходимо запустить алгоритм поиска связных областей по энергии для точки (x_{max}, y_{max}) .

Многие аудио-сигналы имеют либо нарастающую, либо затухающую энергетическую природу. Такие процессы можно приближенно описать при помощи первой производной. К примеру, энергию эха от удара камень о камень на спектрограмме $\bar{I}(x, y)$ можно выразить так (предполагается, что $\bar{I}(x, y)$ непрерывна, x – время, y – частота, τ – приращение времени):

$$\bar{I}(x + \tau, y) = \bar{I}(x, y) + \frac{\partial \bar{I}(x, y)}{\partial x} \cdot \tau$$

Этот критерий принимается за основной для поиска связных областей. Алгоритм делает рекурсивный обход области Ω , начиная с точки $p_0 = (x_0, y_0) = (x_{max}, y_{max})$ и составляя вероятностную маску шума. При этом используется m -связность: обрабатываются не только ближайшие соседи пикселя, но и те точки, которые удалены на радиус r [15].

Каждая следующая точка обозначается как $p_k = (x_k, y_k)$, а предыдущая – p_{k-1} . Если соседний пиксель $p_{k+1} = (x_k + i, y_k + j)$, $0 \leq i, j \leq r$ удовлетворяет критерию

$$I(p_{k+1}) \geq I(p_k) + [I(p_k) - I(p_{k-1})] \quad (1)$$

то к $M(p_{k+1})$ добавляется число *value*.

При запуске поиска связных областей *value* предполагается равным некоторому значению (к примеру, 0.5). Чем больше рекурсивный алгоритм уходит вглубь и чем дальше расстояние от точки (x_{max}, y_{max}) , тем меньшим становится *value*. Это позволяет избежать зацикливаний алгоритма и делает возможным нахождение не просто битовой маски шума (либо 0, либо 1), а вероятностной (от 0.0 до 1.0).

Имеет смысл ввести систему наказаний и поощрений на этапе проверки (1). Достаточно ослабить критерий (1) и всем пикселям M из окрестности p_k , удовлетворяющим ослабленному критерию (1), прибавлять небольшое число (к примеру, $value/10$). Если $M(p_{k+1}) > 0.5$ (т.е. как минимум один раз данный пиксель был помечен, как сигнал), то предлагается проверить, насколько сильно p_k отличается от p_{k+1} . В случае сильной разницы из $M(p_{k+1})$ вычитается штраф. Значение штрафа выбирается по аналогии с $value$.

Возможны и иные модификации этого алгоритма. К примеру, в алгоритме рекурсивного поиска связных областей значительных улучшений можно добиться при использовании принципов сравнения блоков (Block Matching) [17]. К тому же, внедрение других частотных шкал (к примеру, шкалы Мела вместо линейной) должно давать более качественный результат поиска.

б) Удаление множеств связных пикселей, имеющих маленькую мощность. Операции и анализ на этом этапе производятся с вероятностной маской. Необходимо отсеять ложные срабатывания, образовавшиеся после работы рекурсивного поиска связных областей (рис. 5.5).

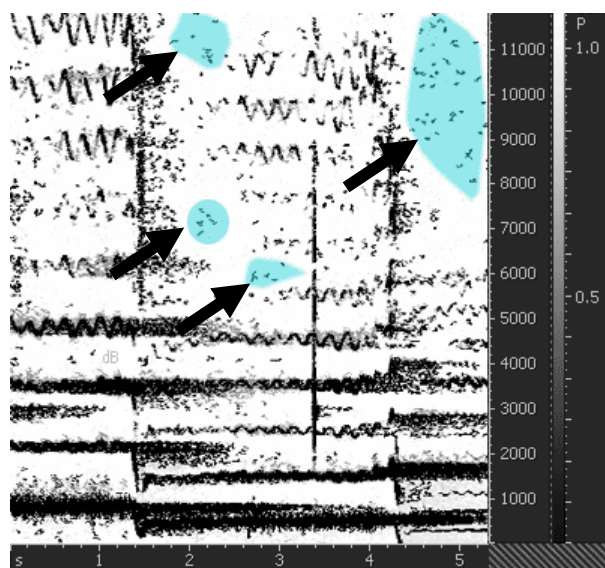


Рис. 5.5. Вероятностная маска шума. Указанные области содержат неверно помеченные участки работы алгоритма рекурсивного поиска связных областей по энергии.

Маска $M(x,y)$ подвергается анализу: на ней выделяются все множества связных пикселей для которых $M(x,y) > 0.5$. После этого

множества, имеющие маленькую мощность, удаляются (т.е. для всех $M(x,y) > 0.5 \Rightarrow M(x,y) = 0$).

7) Удаление "всплесков" вокруг области, которая имеет низкую энергию. Этот этап похож на предыдущий, и его цель такая же: удаление лишних участков на маске.

Выберем небольшую область Ω на $M(x,y)$ (её размеры устанавливаются опытным путём в зависимости от параметров спектрограммы). Если выполняется неравенство:

$$\int_{\Omega} M(x, y) ds < c \cdot \int_{\Omega} ds,$$

(где $0 < c < 1$, обычно $c = 0.1$ достаточно для хорошей очистки)

то значения маски в области Ω можно ослабить.

5.2.3 Дополнительная обработка полученного спектра шума

Дополнительная обработка вероятностной маски шума. Основной недочет, появляющийся при нахождении маски шума разными методами – ложные срабатывания, которые выглядят как мелкие всплески шума (см. рис. 5.5).

В целях борьбы с этим артефактом вводится понятие *меры плотности населения группами* некоторой прямоугольной области Ω размерами $m \times n$. Под группой подразумевается связанное множество пикселей (8-смежное).

Максимальное количество возможных несвязных (по 8-смежности) пикселей в области Ω : $G = S(\Omega)/9 = m \cdot n/9$. В дальнейшем G понадобится с целью нормировки.

Находятся все возможные несвязные группы в Ω и подсчитывается их количество K . Поиск удобно осуществлять с помощью рекурсивного алгоритма.

Отношение $P=K/G$ показывает меру плотности населения группами. Если P близка к 1.0, то это означает, что в области Ω много мелких всплесков шума и их следует удалить.

Дополнительная обработка слепок шума. Один из самых эффективных способов дополнительной обработки полученного слепок шума $Footprint(x,y)$ является медианная или гауссова фильтрация. Эти методы широко распространены и известны [15]. Автором предлагается использовать медианную фильтрацию: она сохраняет резкие перепады (к примеру, транзиенты) и более эффективна при сглаживании импульсных шумов, которые возникают как дефект работы рекурсивного поиска связных областей [18].

Адаптивная медианная фильтрация позволяет улучшить результаты [19] (рис. 5.6). Предпочтительнее всего варьировать радиус медианного фильтра в зависимости от энергии M для некоторой окрестности Ω с центром в точке (x,y) . Т.е. вначале необходимо вычислить:

(т.к. $0.0 \leq M(x,y) \leq 1.0$ для любых пар (x,y) , то $\int_{\Omega} 1.0 ds$ – нормировка)

$$E = \frac{\int_{\Omega} M(x,y) ds}{\int_{\Omega} ds}.$$

Затем в соответствии с параметрами спектрограммы выбирается некоторый радиус размытия R . В областях локализации сигнала следует использовать размытие слепок шума высокой степени, чтобы

минимально задевать полезный сигнал. В области шума – нужно точнее выполнить спектральное вычитание, но в тоже время исключить музыкальный шум. Поэтому радиус адаптивного медианного фильтра будет таким:

$$\begin{cases} r = R \cdot E, & R \cdot E > r_{\min} \\ r = r_{\min}, & R \cdot E \leq r_{\min} \end{cases}$$

Обычно r_{\min} считается равным степени перекрытия окон спектрограммы.

В завершении к пикселю $Footprint(x,y)$ применяется медианный фильтр с радиусом r . Чтобы избежать разрывов и артефактов после обработки, необходимо применять к области Ω окно, спадающее по краям.

Для ускорения процесса расчета можно использовать оптимизированные алгоритмы медианной фильтрации, изложенные в [18].

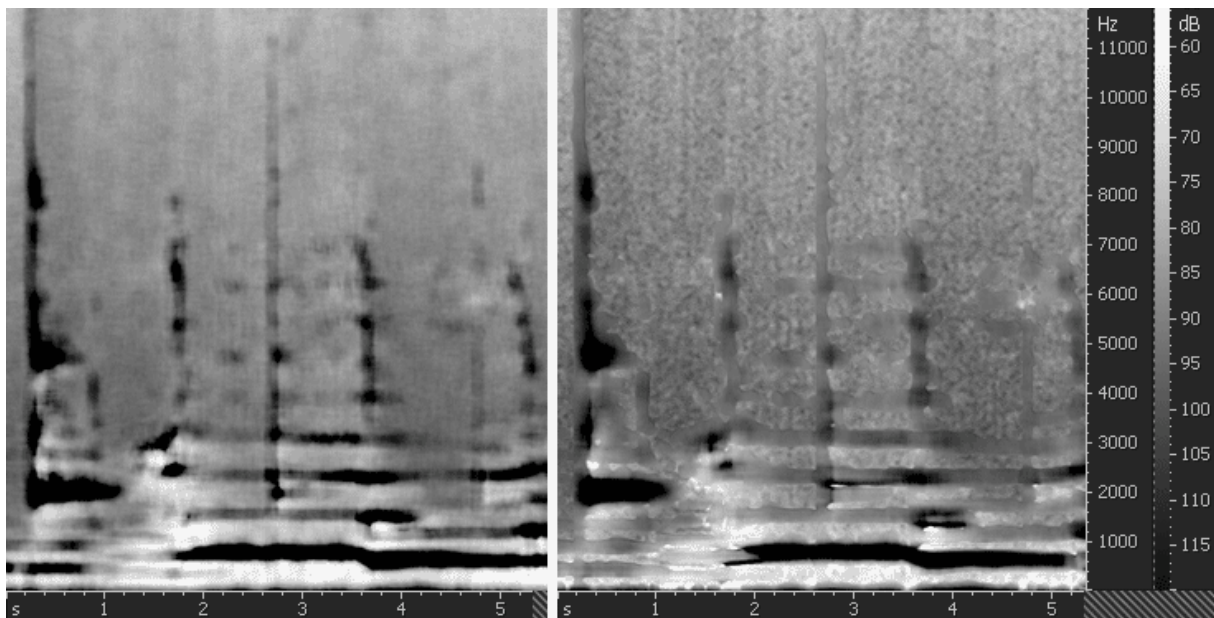


Рис. 5.6. Слева – простая медианная фильтрация, справа адаптивная медианная фильтрация (с варьируемым радиусом)

5.2.4 Результаты работы системы

Наглядно представленные результаты исследования функционирования основных методов, изложенных в разделе 5.2, можно найти в п. 9 "Приложение Б. Адаптивный слепок шума".

Тестирование с нестационарным широкополосным шумом (см. таблицу 2). Сгенерирован белый шум с амплитудной огибающей, схожей с функцией $\sin(x)$: помехи то исчезают, то появляются с ещё большей силой. В качестве сигнала выбран набор синусоид с разными частотами и модуляциями, а также добавлен звук удара по железной тарелке.

Метод	PSNR, дБ	Комментарий
МСВ 46 мс	39.80	Слабое подавление шума
МСВ на основе многомасштабных преобразований	42.85	Хорошее удаление шума в областях, близких к фиксированному слепку шума
МСВ на основе многомасштабных преобразований с адаптацией к меняющемуся шуму	48.51	Отличная степень удаления шума во всём временном диапазоне
Коммерческие продукты		
BIAS SoundSoap 2	40.56	Сильные искажения амплитуды сигнала
iZotope RX	44.93	Отличное удаление шума в областях, близких к фиксированному слепку шума

Таблица 2. Результаты тестирования системы шумоподавления, снабженной адаптивной оценкой спектра шума на основе выделения связных областей по энергии. Отношение сигнал/шум: 29.15 дБ

Тестирование со стационарным широкополосным шумом (см. таблицу 3). Использовался тот же сигнал, что описан выше, но помехи не меняли средней энергии во времени.

Метод	PSNR, дБ	Комментарий
МСВ 46 мс	40.90	Недостаточное подавление шума, появляется размытие спектра
МСВ на основе многомасштабных преобразований	41.24	Не во всем частотном диапазоне шум подавляется равномерно
МСВ на основе многомасштабных преобразований с адаптацией к меняющемуся шуму	43.58	Отличная степень удаления шума
Коммерческие продукты		
Waves X-Noise	38.35	Слабое шумоподавление
iZotope RX	41.94	Отличное удаление шума

Таблица 3. Результаты тестирования системы шумоподавления, снабженной адаптивной оценкой спектра шума на основе выделения связанных областей по энергии. Отношение сигнал/шум: 23.14 дБ

Произведены также и субъективные оценки качества на основе экспертных прослушиваний на сайте <http://tka4.org/adaptivenoisetest> (MUSHRA – MULTiple Stimuli with Hidden Reference and Anchor). Опрос показал, что 51% людей склоняется к результату очистки разрабатываемой системы, 38% проголосовали за работу iZotope RX и 10% отдали предпочтение неочищенному оригинальному аудиосигналу. В качестве фонограммы был выбран зашумленный женский оперный вокал.

6 Описание практической части

6.1 Шумоподавление при известном спектре шума

6.1.1 Общая структура программы

На вход программе подаётся wav файл [20]. Для работы с иными форматами предлагается использование конвертеров, к примеру, свободно распространяемого ffmpeg. Считанные из него данные посылаются в аудио поток (Audio Stream), работающий в режиме буферизации, благодаря чему возможна обработка больших записей, требующих в обычном режиме работы большое количество оперативной памяти.

Из конфигурационного файла считывается набор из N эффектов и их параметры. В качестве эффектов в данной версии программы рассматриваются простой метод спектрального вычитания (МСВ), рекурсивный МСВ, адаптивный рекурсивный МСВ и метод нелокального среднего (Non-Local Means) [6]. Кусок аудио из буфера передаётся на вход каждому из эффектов, обрабатывается им и отправляется для анализа в модуль Multi-Resolution (рис. 4.6).

Анализатор, встроенный в Multi-Resolution, выбирает наиболее подходящий результат работы одного из эффектов, а микшер производит его суммирование с выходным аудио сигналом.

Вся настройка системы производится с помощью главного файла *settings.ini*, в котором указываются все возможные параметры системы. Из основных следует выделить: размер FFT результирующей спектрограммы (используемой для микширования), размеры области Ω (т.е. разбиение спектрограммы по частоте и по времени), степени перекрытия анализируемых областей.

Ниже представлено более детальное описание работы модулей системы.

6.1.2 Оконное преобразование Фурье

Оконное преобразование Фурье (Short Time Fourier Transform, STFT) – "сердце" программы. Для вычисления STFT необходимо умножить сигнал на окно и произвести преобразование Фурье:

$$X[f, t] = \sum_{m=0}^N x[t + m] \cdot w[m] \cdot e^{-ifm}, \text{ здесь}$$

- $x[t]$ – амплитуда сигнала в отчете t ,
- $X[f, t]$ – дискретный спектр сигнала с $f=0..N/2+1$ частотами,
- $w[m]$ – оконная функция, окно (её основное назначение – сглаживание сигнала по краям окна до 0-го значения, в данной разработке используется функция Ханна),
- N – размер окна. При размере окна около 12 мс (512 отчетов для частоты дискретизации 44,100 Гц) можно добиться хороших результатов.

Для составления спектрограммы выбирается некоторое перекрытие спектров и рассчитывается шаг $step$. В разрабатываемой системе по умолчанию используется $step = 1/4 \cdot N$. Тогда спектрограмма вычисляется так: $Spectrogram[x, f] = X[f, x \cdot step]$, где $x = 1..n - step$, n – длина сигнала $x[t]$ в отчетах.

6.1.3 Модули шумоподавления

Конфигурация эффектов задаётся в отдельном ini файле. Ссылка на него указывается в settings.ini:

```
[Advanced]
use_effects_file = effects.ini
```

Каждый эффект имеет свой тип (1 – пустой эффект, 2 – обычный МСВ, 3 – рекурсивный МСВ, 4 – адаптивный рекурсивный МСВ, 5 – метод нелокальных средних и т.д.) и размер FFT. Далее настройки

зависят от конкретно используемого эффекта. Для простого метода спектрального вычитания дополнительно задаётся сила подавления шума, а для рекурсивного – коэффициент сглаживания спектра по времени и т.д. Пример:

```
[Effect 0]
type = 3;           Effect type
fft_size = 128;    Effect FFT size
suppression = 0.95;  Suppression power of the denoiser
alpha = 256;       Smoothing parameter
```

Любой эффект представляется своим классом C++, наследуемым от *CSimpleEffect* (рис. 6.1). Принцип работы *CSimpleEffect*:

1. Вычисляется STFT аудио-сигнала,
2. К полученному спектру применяется эффект,
3. Результат добавляется в анализируемую спектрограмму (*Analyzable Spectrogram*),
4. Производится inverse STFT, полученный сигнал оказывается на выходе эффекта.

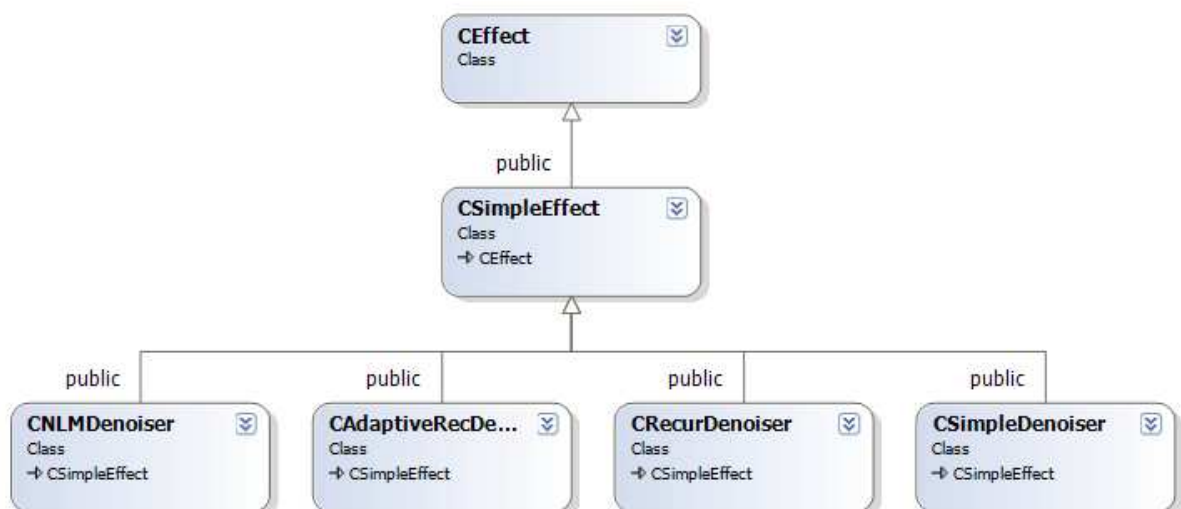


Рис. 6.1. Диаграмма наследования классов

Анализируемая спектрограмма является частью модуля анализатора Multi-Resolution (рис. 4.6). И её использование рассматривается в ниже.

Для шумоподавления адаптированного с методов для изображений алгоритм работы эффекта выглядит немного иначе: сначала производятся все STFT преобразования, результат которых сохраняется в анализируемой спектрограмме. Затем к ней применяется метод шумоподавления для изображения. И только после этого происходит обратное преобразование iSTFT.

В программе задействован метод Non-Local Means, алгоритм которого изложен в разделе "Обзор существующих решений".

Работа каждого из эффектов происходит независимо, поэтому её можно выделить в отдельную программную нить, т.е. параллельно выполнять работу всех эффектов. Для этого задействованы средства OpenMP, которые позволяют

- добиться хорошей производительности,
- получить легко читаемый программный код,
- компилировать программу под любую операционную систему, в которой имеется поддержка OpenMP.

6.1.4 Анализатор и микшер

Основная задача анализатора заключается в том, чтобы определить транзиент – резкий всплеск энергии в спектре, и более компактно представить спектр. Транзиенты предпочтительно обрабатывать эффектом с коротким размером FFT. А места, где энергия локализована равномерно – длинным (см. приложение А).

Сопоставление спектрограмм. В начале необходимо выяснить, как сопоставляются между собой спектрограммы разных размеров.

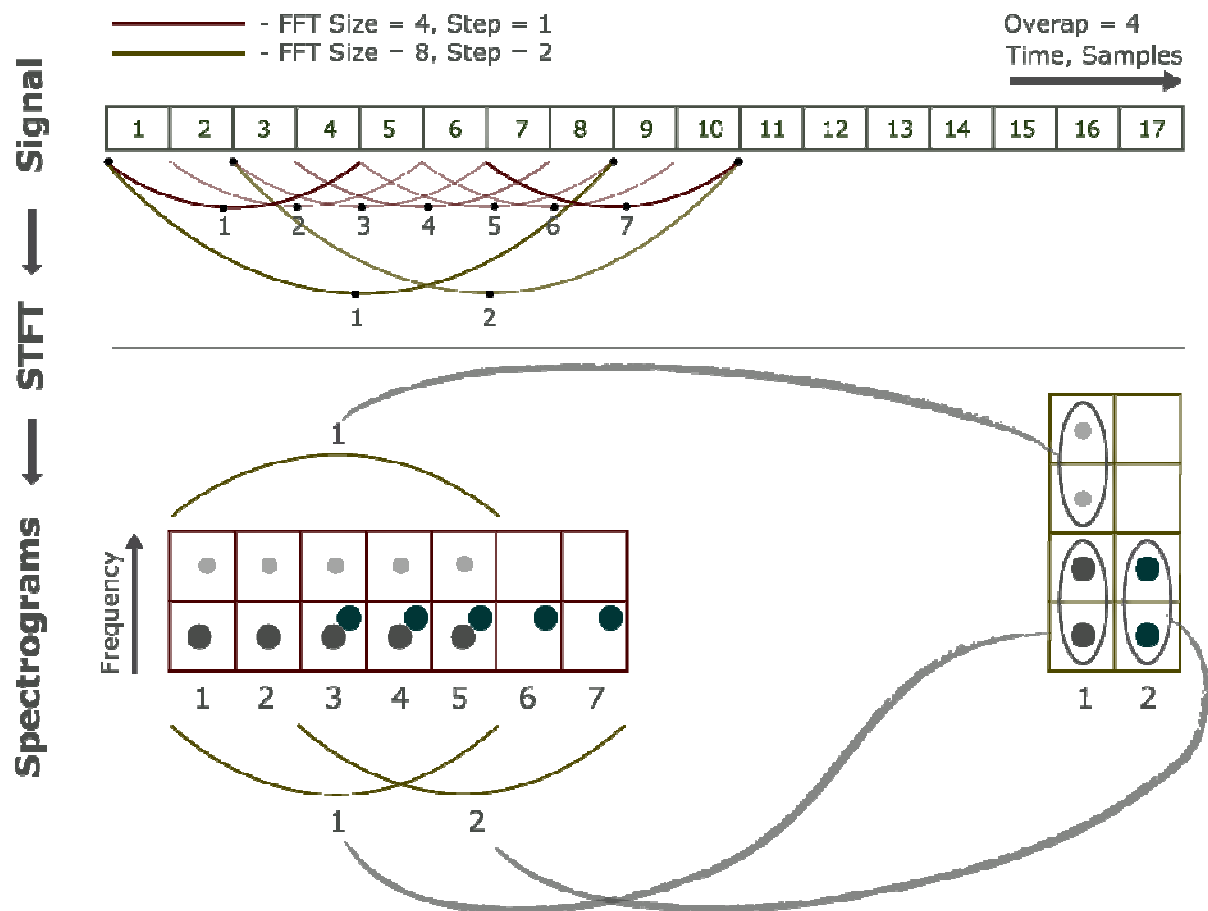


Рис. 6.2. Сопоставление спектрограмм разных размеров

На рисунке 6.2 изображены две спектрограммы (слева и справа) и отмечено соответствие областей на них. Здесь *FFT Size* – размеры FFT, *Step* – шаг по времени, определяемый по формуле $FFT\ Size / Overlap$, где *Overlap* – степень перекрытия.

```
float inv_over = 1.0/Overlap;
float step_x=(FFTSize2*inv_over)/(FFTSize1*inv_over);
float coef_x=(FFTSize2-FFTSize1)/(FFTSize1*inv_over)+1;
float step_y=CurrentFFT/MixFFT;
float coef_y=1;
```

С помощью указанных формул можно рассчитать, какая область соответствует на спектрограмме с большим размером $FFTSize2$ спектрограмме с меньшим размером $FFTSize1$:

```
float x1_FFTSize1 = x_FFTSize2 * step_x;  
float y1_FFTSize1 = y_FFTSize2 * step_x;  
float x2_FFTSize1 = x_FFTSize2 * step_x + coef_x;  
float y2_FFTSize1 = y_FFTSize2 * step_x + coef_y;
```

Анализ областей. Анализатор рассматривает исключительно Analyzable Spectrograms (см. рис. 4.6). Две области сопоставляются друг другу и затем вычисляется их степень размытия. Области являются прямоугольными и "обходят" всю спектрограмму с перекрытием по частоте и времени. Размеры области и степени перекрытия указываются в общем конфигурационном файле.

Формула для расчёта размытия была рассмотрена в разделе "Обзор существующих методов: Многомасштабные преобразования"

В случае белого шума в области – степень размытия будет приближена к единице. Это означает, что энергия локализована равномерно по всей области.

Микширование спектрограмм. Операция суммирования (или микширования) производится с *Mixable Spectrograms* (см. рис. 4.6).

Область, для которой рассчитана степень размытия, сопоставляется области на микшируемой спектрограмме. Затем коэффициенты этой спектрограммы микшируются в результирующую (Resulting Spectrogram), дополнительно умноженные на

$$\alpha = 1.0 / (S_{\Omega_r}^8 + \epsilon).$$

6.2 Шумоподавление с адаптацией к меняющемуся спектру шума

При программировании такой системы предлагается использовать два интерфейса (абстрактных класса). Первый – для поисковика маски шума, второй – для поисковика слежка шума. Вначале необходимо вычислить маску шума с помощью *INoisePredictor*.

```
class INoisePredictor {  
    /// Возвращает вероятностную маску шума  
    getMask (int time, int frequency) = 0;  
    /// Захват слежка шума для данного предиктора  
    capture () = 0;  
}
```

IFootprint формирует окончательный спектр шума и возвращает его объекту, непосредственно производящему очистку аудиосигнала.

```
class IFootprint {  
    /// Основная функция, возвращающая слепок шума  
    virtual float getFootprint (int time, int frequency) = 0;  
    /// Захват слежка шума  
    virtual void capture () = 0;  
}
```

В итоге, *AdaptiveFootprint* связывает эти два интерфейса. Приведем пример с простейшей реализацией такого класса:

```
class AdaptiveFootprint : public IFootprint {  
protected:  
    INoisePredictor      *predictors;  
    float                 **spectrogram;  
    int                  predictorsSize;  
  
public:  
    virtual float getFootprint (int time, int frequency) {  
        float res = 1;  
        for (int i=0; i<predictorsSize; i++)  
            res *= predictors[i]->getMask(time, frequency);  
        return spectrogram[time][frequency] * res;  
    }  
}
```

```
virtual void capture () {
    for (int i=0; i<predictorsSize; i++)
        predictors[i]->capture();
}
```

Так как *INoisePredictor* возвращает вероятность нахождения шума от 0 до 1 в точке *(time, frequency)*, то результаты работы нескольких предикторов можно комбинировать между собой, к примеру, с помощью их перемножения.

После вычисления комбинированной вероятности от нескольких предикторов в *(time, frequency)*, она перемножается со значением спектрограммы в этой точке и передается в блок очистки сигнала.

Также внутри *AdaptiveFootprint* заложены методы постобработки полученных маски и слепка шума такие, как медианная фильтрация и удаление участков маски по мере плотности населения группами. Создаются отдельные двумерные массивы для результирующих маски и слепка шума, хранящиеся в *AdaptiveFootprint*, и все дальнейшие действия производятся уже с ними, как с изображениями.

Поиск сигнала и шума с помощью различных методов реализуется с помощью наследования от *INoisePredictor*. Классы предикторов на основе оценки степени разреженности и на основе выделения связных областей по энергии именуется как *SmoothNoisePredictor* и *ConnectionsNoisePredictor* соответственно.

Рассмотрим устройство и параметры классов *SmoothNoisePredictor* и *ConnectionsNoisePredictor*.

6.2.1 Поиск сигнала и шума с помощью оценки разреженности спектрограммы

В этом методе для каждого пикселя маски вычисляется степень размытия спектрограммы в его окрестности Ω . Важная задача – выбор области омега. Этот размер зависит от степени перекрытия окон при составлении спектрограммы и частотного диапазона спектра.

Обычно степень перекрытия выбирается равной $1/4$ от размера FFT. Кратковременный всплеск энергии в один отчет попадает сразу в четыре спектральных окна (см. рис. 6.5). Поэтому на спектрограмме транзиенты занимают не менее четырёх пикселей по времени.

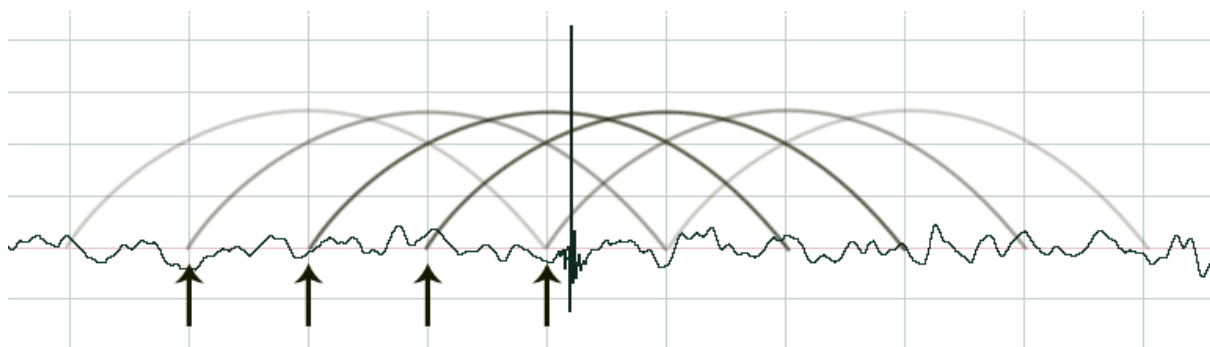


Рис. 6.5. Перекрытие транзиента окнами. Стрелками указаны окна, в которые попадает кратковременный всплеск.

В звуковой природе нечасто встречаются настолько короткие транзиенты (1 отчет при частоте дискретизации 44100 Гц – меньше 1 мс). Максимальная длина аудиофрагмента, который воспринимается на слух как щелчок, – около 30 мс, что соответствует ~ 2.5 пикселям по времени на спектрограмме с разрешением 46 мс. Т.е. каждый отчет, относящийся к щелчку, нужно "закрывать" окном. Это число можно принять за минимальный радиус по времени для Ω .

Опытным путем было установлено, что наиболее хорошие результаты работы *SmoothNoisePredictor* достигаются при выборе прямоугольной области Ω , равной 25 пикселям по времени и частоте при разрешении спектра 46 мс и степени перекрытия $1/4$.

Такой выбор Ω связан с нашим психоакустическим восприятием звука: 25 пикселей по времени эквивалентны ~ 300 мс. Именно за этот минимальный промежуток времени мозг способен распознать сигнал [21].

Предпочтительнее использовать варьируемый радиус по частоте для Ω . Так как большая часть значимого для человека сигнала содержится в диапазоне от 300 до 5000 Гц, то в нём целесообразно уменьшать этот параметр, чтобы выявить большее количество мелких изменений в сигнале. К тому же разрешающая способность уха в районе 3000 Гц гораздо выше, чем около 10000 Гц [22].

Учитывая сказанное выше, запишем итоговые формулы для выбора области Ω :

$$p = \left(\frac{t \cdot D}{FFTSize} - 1 \right) \cdot overlap,$$

$$\begin{cases} r_x = p, & \text{если } p > 2.5, \\ r_x = 2.5, & \text{иначе.} \end{cases}$$

$$\begin{cases} c = 1, & \text{если } 150 < \frac{y}{FFTSize} \cdot D < 5000 \\ c = 2, & \text{иначе.} \end{cases}$$

$$r_y = 25 \cdot \frac{FFTSize}{2048} \cdot c$$

Здесь r_x , r_y – радиусы Ω по времени и частоте, t – время в секундах, D – частота дискретизации, $overlap$ – перекрытие окон (2, 4, ...), y – обрабатываемая частота.

Время t полагается равным 0.3 секунды. Хотя эта величина тоже может варьироваться для достижения наилучшего результата.

6.2.2 Поиск сигнала и шума на основе выделения связанных областей по энергии

Огромную роль в данном предикторе играет выбор области Ω . От её размера зависит общее количество найденных локальных максимумов. Идеи и принципы остаются такими же, как в предыдущем подразделе. Перекрытие областей при поиске максимума делается попиксельно.

После действий, описанных в п. 5.2.2 "Поиск сигнала и шума на основе выделения связанных областей по энергии", производится дополнительная обработка, встроенная в *ConnectionsNoisePredictor*: удаление областей на маске шума, имеющих низкую мощность, и усиление уже найденных областей сигнала на маске.

Удаление областей на маске шума, имеющих низкую мощность производится с помощью рекурсивного поиска. Выбирается некоторый пиксель на маске, помеченный, как сигнал и с ним срабатывает указанный метод.

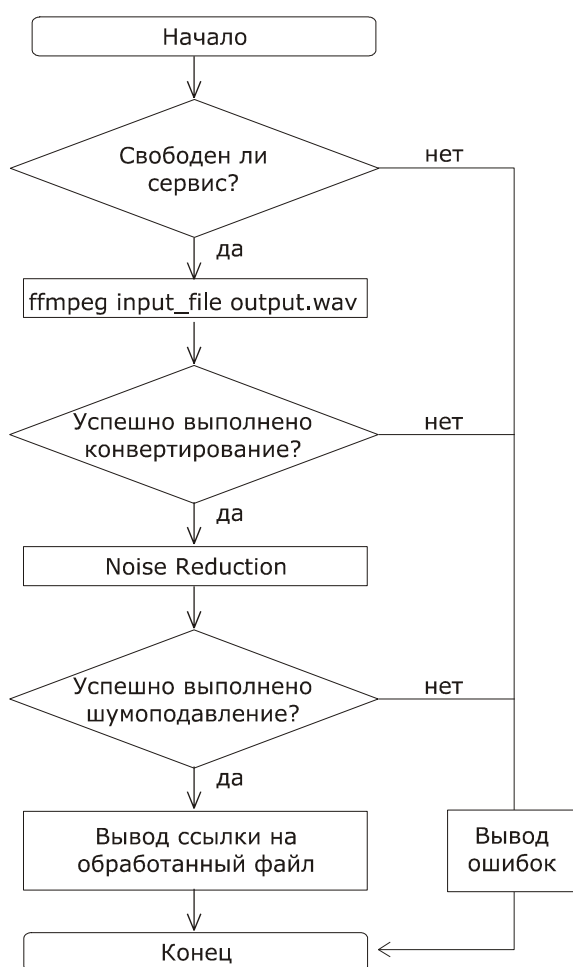
Усиление уже найденных областей сигнала на маске выполняется так: на $M(x,y)$ находится пиксель (k, l) , принадлежащий сигналу. Если $M(k, l) > 0.5$ и медиана, деленная на $2 \cdot (FFTSize/2 + 1)$, для $M(i, j)$, $i = \{k-1, k, k+1\}$, $j = \{0..FFTSize/2+1\}$ меньше $M(k, l)$, то к $M(k, l)$ добавляется 0.5. Таким образом, области сигнала на маске выделяются ещё сильнее и искажения при спектральном вычитании минимизируются.

6.3 Web приложение

За основу web приложения, с помощью которого пользователь взаимодействует с системой, взята технология AJAX. Благодаря ей сервер передаёт клиенту (Интернет браузеру) информацию о процессе выполнения стадий запроса: загрузки аудио-файла на сервер, конвертирования исходного аудио в wav, работы системы шумоподавления.

Серверная часть написана на языке PHP. Реализован механизм простой очереди: если попытаться обратиться к сервису в момент его использования другим пользователем, то запрос будет отклонён с просьбой обратиться к онлайн-денойзеру через несколько минут.

После того, как пользователь выбрал файл и нажал кнопку "Upload and process" срабатывает следующая схема:



input_file – загруженный на сервер файл пользователя;

output.wav – конвертированный аудио-файл;

ffmpeg – стандартная утилита для конвертации аудио- и видео- файлов;

Noise Reduction – разрабатываемая система шумоподавления.

Схема 1. Основные стадии работы онлайн-денойзера [23]

6.4 Инструментарий разработки

При построении некоторых методов использовалась математическая среда The MathWorks Matlab, но окончательное решение программировалось на языках C++ с использованием расширения STL и C#.

Для получения оптимального кода программы задействован Intel C++ Compiler 11 [24]. Это позволило ускорить время просчета в среднем на 20%, а также обеспечило кроссплатформенную совместимость кода.

Использование средств OpenMP повысило скорость работы в несколько раз на многопроцессорных и многоядерных компьютерах. OpenMP очень удобно использовать для многомасштабных преобразований, так как все их эффекты шумоподавления работают параллельно [24, 25].

При помощи AJAX и PHP разработаны веб-интерфейс и серверная часть онлайн-декойзера. Программа работает под управлением Ubuntu Linux.

Для исследований полученных результатов применялись Adobe Audition, Adobe Photoshop, iZotope RX. Разработана собственная система визуализации, удобно представляющая различные спектральные и иные данные в трёхмерном и двухмерном виде.

7 Заключение

Разработана система шумоподавления, сочетающая в себе множество качественных и новых подходов для очистки аудиосигналов от шумов. Исследованы параметры их функционирования. Проведен ряд тестов, подтверждающих высокую эффективность полученных алгоритмов.

В систему интегрированы несколько модификаций метода спектрального вычитания и метод нелокального усреднения, адаптированный из области шумоподавления для изображений [8]. Из таблицы 1 видно, что использование многомасштабных преобразований в совокупности с этими подходами значительно повышает степень удаления шума, не нарушая структуры сигнала.

Достигнута высокая производительность при работе с фиксированным спектром шума, возможна очистка аудио в реальном времени. На основе такой системы, оснащенной автоматическим поиском слежка шума, создан онлайн интернет-сервис [23].

Построены оригинальные адаптивные методы для поиска шума, меняющегося во времени. Благодаря им стало возможно не только подавлять нестационарные широкополосные шумы, но и повысить степень очистки от стационарных помех (см. таблицы 2 и 3).

Введено понятие вероятностной маски шума, с помощью которого удобно оперировать с шумовым профилем и смешивать результаты работы нескольких предсказателей шума. К примеру, совокупность предикторов шума на основе оценки разреженности спектрограммы и выделения связных областей по энергии даёт в ряде случаев лучший результат, чем использование их по отдельности.

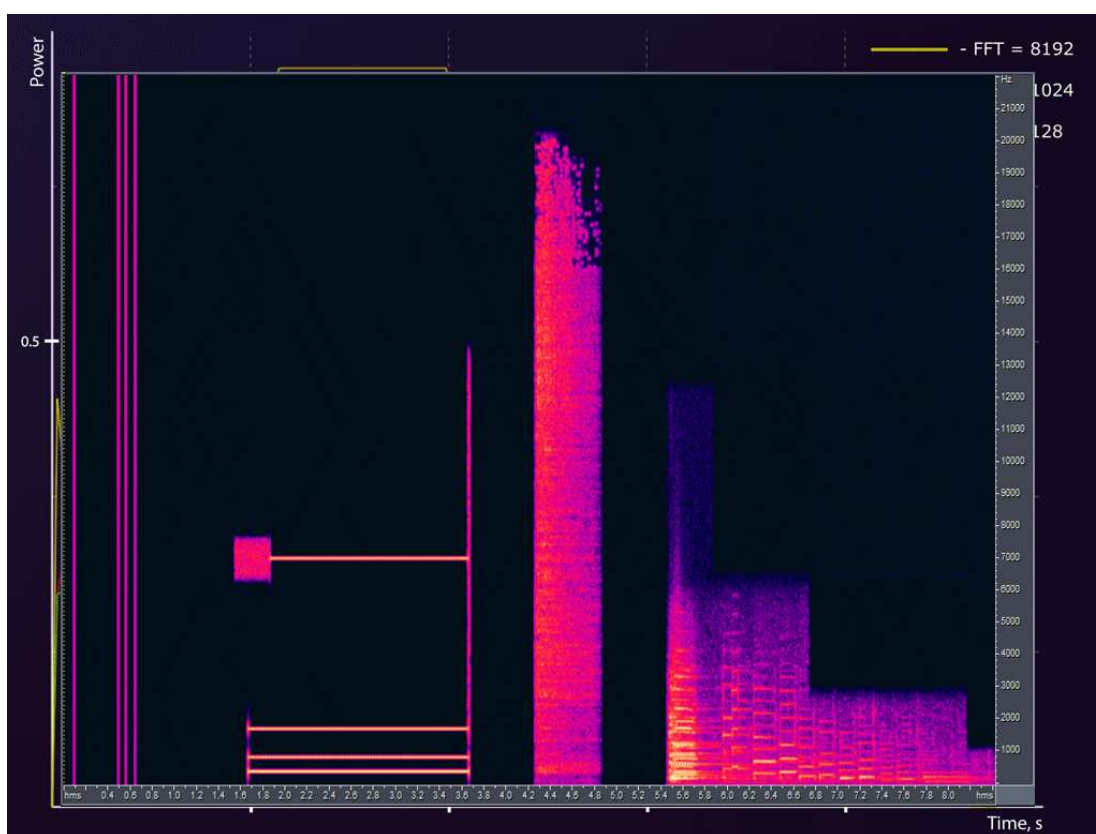
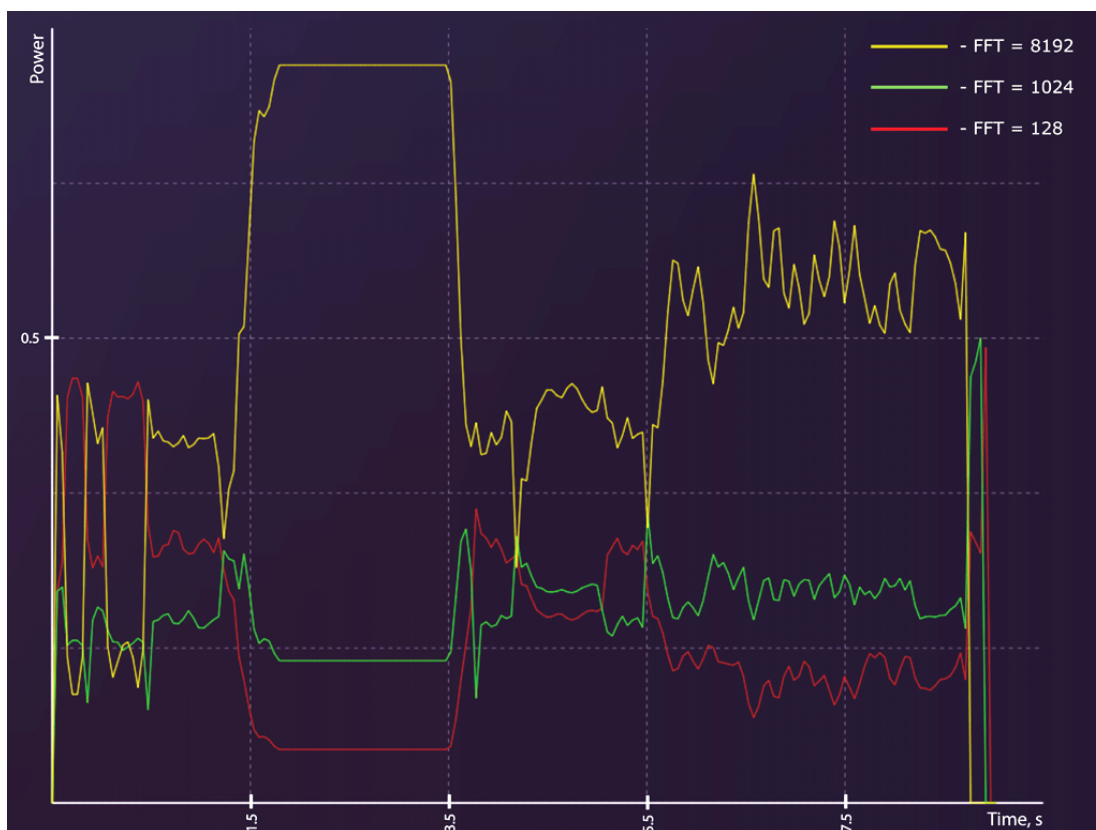
Работа отмечена дипломом на международной конференции и выставке Digital Signal Processing and its Applications 2010.

8 Литература

- [1] Saeed V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons Ltd, 2000, 465 p.
- [2] Алексей Лукин. Введение в цифровую обработку сигналов (Математические основы). Методическое пособие, 2002, 44 с.
<http://audio.rightmark.org/lukin/dspcourse/dspcourse.pdf>
- [3] Alexey Lukin, Jeremy Todd. Adaptive Time-Frequency Resolution for Analysis and Processing of Audio. *Proceeding of the Audio Engineering Society (AES) Convention, Paris, May 2006*, 10 p.
- [4] Joachim Thiemann. Acoustic Noise Suppression for Speech Signals using Auditory Masking Effects. *Department of Electrical & Computer Engineering. McGill University. Montreal. 2001*, 83 p.
- [5] S.F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. Acoust., Speech, Signal Process.*, vol.27, Apr. 1979, pp. 113-120.
- [6] Arthur Szlam. Non-Local Means for Audio Denoising. *Recent UCLA Computational and Applied Mathematics Reports*, 2008, 5 p.
- [7] Кельберт М.Я., Сухов Ю.М. Вероятность и статистика в примерах и задачах. Т. II: Марковские цепи как отправная точка теории случайных процессов и их приложения. МЦНМО, Москва, 2009, 295 с.
- [8] Кен Гендри. Системы шумоподавления. Журнал "Звукорежиссёр" выпуск №6, <http://rus.625-net.ru/archive/z0604/noise.htm>, 2004.
- [9] John Elliott. Adaptive noise reduction filter with continuously variable sliding bandwidth. *USA Patent № 6154547*, 2000.
- [10] Antoni Buades, Bartomeu Coll, Jean-Michel Morel. A non-local algorithm for image denoising. *CVPR'05, Vol. 2, 2005*, pp 60-65.
- [11] Dabov, K., A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising with block-matching and 3D filtering. *Electronic Imaging'06, Proc. SPIE 6064*, no. 6064A-30, San Jose, California USA, 2006.

- [12]** Karin Dressler. Sinusoidal Extraction Using an Efficient Implementation of a Multi-Resolution FFT. *DAFx-06, 2006, 6 p.*
- [13]** Wenqing Jiang, Henrique Malvar. Adaptive Noise Reduction of Speech Signals. *Microsoft Research, 2000, 9 p.*
- [14]** Frank Y. Shin. Image processing and mathematical morphology. Fundamentals and applications. *CRC Press, 2009, 442 p.*
- [15]** Р. Гонсалес, Р. Вудс. Цифровая обработка изображений. *ТЕХНОСФЕРА, Москва, 2005, 1073 с.*
- [16]** Р. Дуда. П. Харт. Распознавание образов и анализ сцен. *Мир, Москва, 1976, 507 с.*
- [17]** College of CSS Engineering. Digital Image Processing. <http://www.icaen.uiowa.edu/~dip/LECTURE/Segmentation.html>
- [18]** Хуанг Т.С. Быстрые алгоритмы в цифровой обработке изображений. *Радио и связь, Москва, 1984, 224 с.*
- [19]** Яровой Н.И. Адаптивная медианная фильтрация. <http://www.controlstyle.ru/articles/science/text/amf/>
- [20]** Issued as a joint design by IBM Corporation and Microsoft Corporation. Multimedia Programming Interface and Data Specifications. *1991, 97 p.*
- [21]** John G. Nicholls, A. Robert Martin, Bruce G. Wallace, Paul A. Fuchs. From Neuron to Brain. *Sinauer Associates, Inc. Sunderland, Massachusetts, 2001, 671 с.*
- [22]** Алдошина И.А. Основы Психоакустики. *Подборка статей с сайта <http://www.625-net.ru>, 154 с., 2009.*
- [23]** Демонстрационная версия онлайн-декойзера: <http://makseq.com/makseq>
- [24]** Intel C++ Compiler for Linux Systems. User's Guide. *2009, 440 p.*
- [25]** В.В. Воеводин, Вл.В. Воеводин. Параллельные вычисления. *BHV, Москва, 2004, 608 с.*

9 Приложение А. Нахождение транзиентов системой шумоподавления



10 Приложение Б. Адаптивный слепок шума

