**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАЕТМАТИКИ И ИНФОРМАТИКИ**

**Кафедра математического моделирования и анализа данных**

**СТАТИСТИЧЕСКИЙ АНАЛИЗ ВРЕМЕННЫХ РЯДОВ**

**С ПОМОЩЬЮ ВЕЙВЛЕТ ПРЕОБРАЗОВАНИЙ**

Курсовой проект

Цуранова Никиты Васильевича

студента 3 курса

Специальность «Прикладная математика»

Научный руководитель:

Доцент кафедры ММАД ФПМИ

В. И. Лобач

Оглавление

[Введение 3](#_Toc27117786)

[1. Вейвлеты Хаара и их свойства 4](#_Toc27117787)

[2. Преобразование Фурье и вейвлет-преобразования 5](#_Toc27117788)

[3. Задачи анализа и синтеза 6](#_Toc27117789)

[3.1 Задача анализа 6](#_Toc27117790)

[3.2 Задача синтеза 6](#_Toc27117791)

[4. Компьютерные эксперименты 7](#_Toc27117792)

[Заключение 12](#_Toc27117793)

[Список литературы 13](#_Toc27117794)

[Приложение 14](#_Toc27117795)

Введение

Вейвлет-преобразование представляет собой синтез идей, которые возникли за многие годы из разных областей, таких как математика и обработка сигналов. Вообще говоря, вейвлет-преобразование - это инструмент, который делит данные, функции или операторы на разные частотные компоненты, а затем изучает каждый компонент с разрешением, соответствующим его масштабу[1].

Таким образом, вейвлет-преобразование используется для обеспечения экономного и информативного математическое представление многих объектов, представляющих интерес[2]. В настоящее время многие компьютерные программные пакеты содержат быстрые и эффективные алгоритмы для преобразования вейвлетов. Благодаря такой легкой доступности вейвлеты быстро завоевали популярность среди ученых и инженеров, как в области теоретических исследований, так и в области применения. Прежде всего, вейвлеты широко применяются в таких областях компьютерных наук, как обработка изображений, компьютерное зрение, управление сетями и анализ данных. За последнее десятилетие интеллектуальный анализ данных или базы данных обнаружения знаний стали важной областью как в академия и в промышленности. Интеллектуальный анализ данных - это процесс автоматического извлечения новых полезных и понятных шаблонов из большой коллекции данных.

Теория вейвлетов, естественно, может сыграть важную роль в анализе данных, поскольку она хорошо обоснована и имеет очень практическое применение. У вейвлетов есть много благоприятных свойств, таких как исчезающие моменты, иерархическая структура с разложением по иерархии и многоразрешению, линейная временная и пространственная сложность преобразований, декоррелированные коэффициенты и широкий спектр базовых функций. Эти свойства могут обеспечить значительно более эффективные и эффективные решения многих проблем анализа данных. Во-первых, вейвлеты могут обеспечивать представление данных, которые делают процесс сбора данных более эффективным и точным. Во-вторых, вейвлеты могут быть включены в ядро ​​многих алгоритмов сбора данных.

Хотя стандартные вейвлет-приложения в основном используются для данных, которые имеют временную / пространственную локализацию (например, временные ряды, данные потоков и данные изображений), вейвлеты также успешно применяются в различных областях при извлечении данных. На практике широкое разнообразие методов, связанных с вейвлетами, было применено для решения целого ряда проблем интеллектуального анализа данных.

В этой работе представляются необходимые математические основы для понимания и использования вейвлетов, а также краткий обзор исследований вейвлет-приложений.

1. Вейвлеты Хаара и их свойства

Материнская функция вейвлета Хаара[3] задается следующим образом:

Масштабирующая функция определяется как:

Система базисных вейвлетов получается из материнского вейвлета путем растяжения и смещения:

Свойства вейвлетов[4]:

* абсолютно интегрируемая и принадлежит :
* Среднее равно нулю, а норма равна единицы:
* Базис Хаара так же является ортонормированным:

скалярное произведение в

Легко показать, что и что смещенные базисные функции одного уровня (с разными *b*, но одинаковыми *a*) являются ортогональными, т.к. их ненулевое пересечение пусто.

Покажем, что функции разных уровней так же являются ортогональными. Рассмотрим , где . Тогда существует 2 возможных варианта ( пересечение ненулевых элементов):

|  |  |
| --- | --- |
| Тогда очевидно, что |  |

1. Преобразование Фурье и вейвлет-преобразования

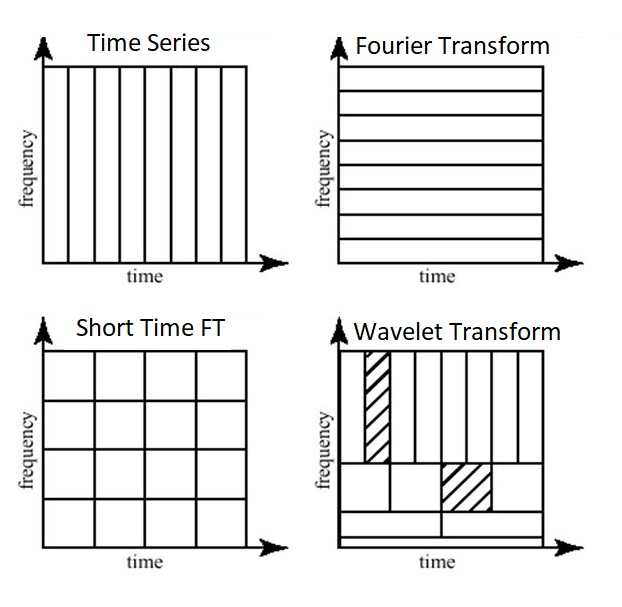
Преобразование Фурье[5] функции вещественной переменной задается формулой:

Формула обращения:

В контексте анализа временных рядов, которые можно рассматривать как сигналы, преобразование Фурье позволяет перевести сигнал из временного представления в частотное.

Но у преобразования Фурье есть недостаток. Т.к. мы получаем только частотный спектр, то результат преобразования для суммы двух синусоид и синусоиды, переходящей в другую, (с теми же частотами и на таком же временном промежутке) будет неотличим.

Если нам нужно больше, чем просто анализ спектра частот, то существует оконное преобразование Фурье, а также вейвлет-анализ.



Вейвлет-преобразование отличается от преобразования Фурье выбором анализирующей функции. Но для разложения в ряд Фурье мы накладывали условие ортогональности, и благодаря этому мы раскладывали функцию по заданному базису. С вейвлетами, в общем случае, мы так сделать не можем, но мы можем посмотреть насколько заданная функция похожа на наш вейвлет в заданный момент времени. Момент времени задается через растяжение и смещение анализирующей функции, а схожесть определяется величиной коэффициента.

Вейвлет Хаара обладает ортогональностью, а значит мы можем провести разложение по вейвлет-базису. С этим мы подходим к задаче анализа.

1. Задачи анализа и синтеза

## **3.1 Задача анализа**

Задана анализа состоит в получении коэффициентов разложения. Для работы будем использовать вейвлеты Хаара. Формулы для непрерывного преобразования:

Данные формулы работают для разложения функции на промежутке [0, 1). Для разложения на ином промежутке можно расширить базис:

* Введение в базис дополнительных вейвлетов (полученных большим смещением). Таким образом можем получить промежуток [0, n), где
* Сдвиг материнского вейвлета на константу даст промежуток [C, C + 1)
* Растяжение материнского вейвлета на константу дает промежуток [0, C)

Но на практике мы обычно имеем дело с дискретными сигналами. Рассмотрим дискретное вейвлет-преобразование.

Обычно размер данных о сигнале равен степени двойки для упрощения вычислений, тогда пусть сигнал задается как значения , где . Рассмотрим .

Где – аппроксимация сигнала, а – а детализация.

Такое разложение сигнала можно использовать рекуррентно:

Этот метод позволяет получить разложение без потери точности и без вычисления самих вейвлетов, а за счет рекурсии работает асимптотически за . Также позволяет упростить сам сигнал, оставляя просто аппроксимацию.

## **3.2 Задача синтеза**

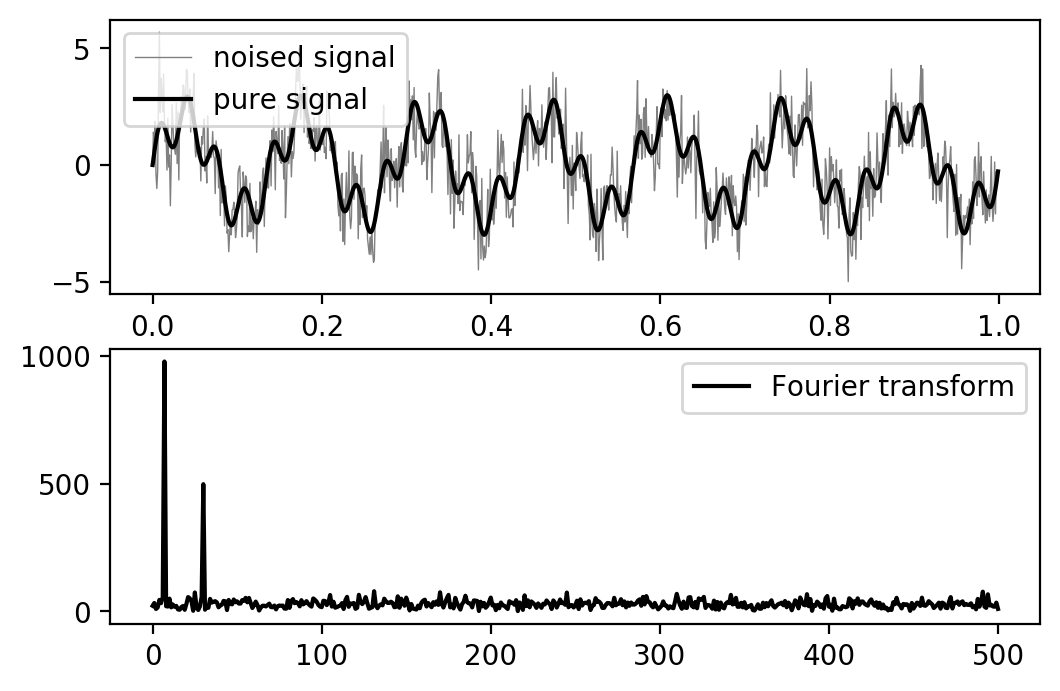
Задача синтеза состоит в восстановлении сигнала по полученным коэффициентам.

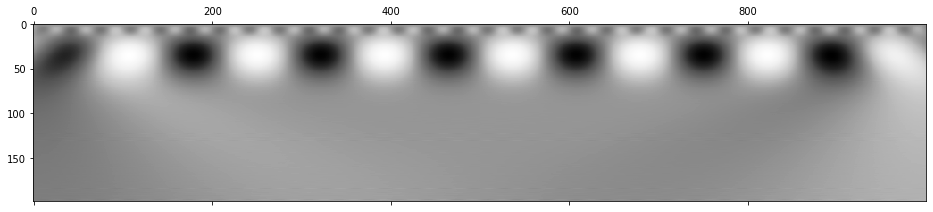
По тем формулам для вейвлета Хаара, что мы определили в 3.1. получаем:

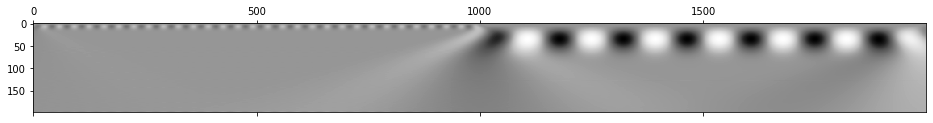
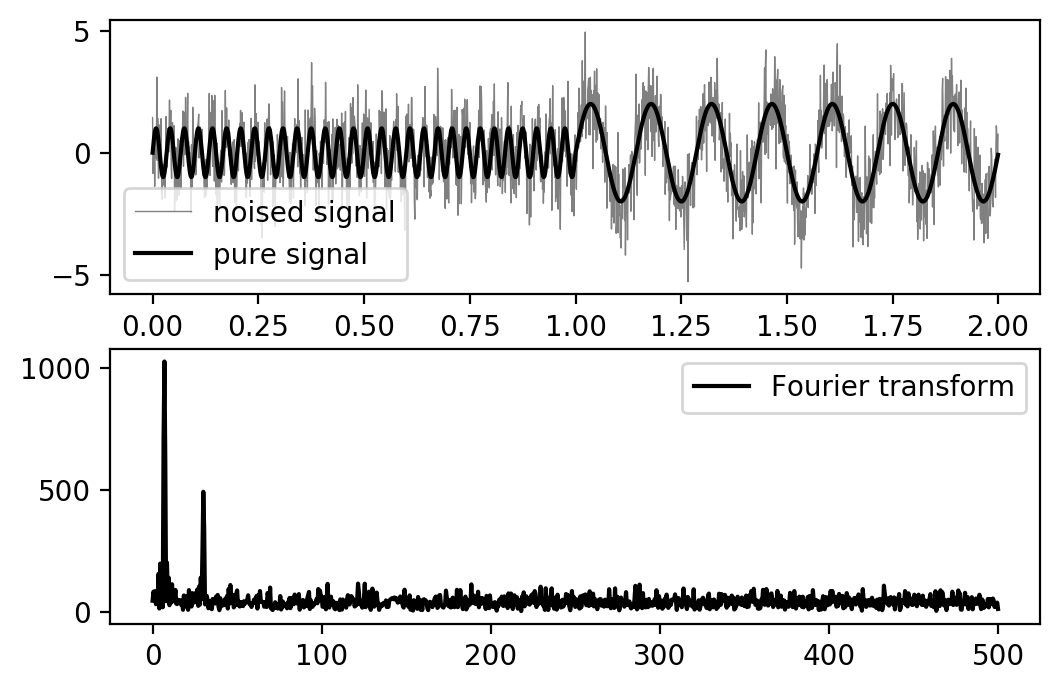
* Для непрерывного:
* Для дискретного:

1. Компьютерные эксперименты

Продемонстрируем основные отличия преобразования Фурье от вейвлет-анализа(1):



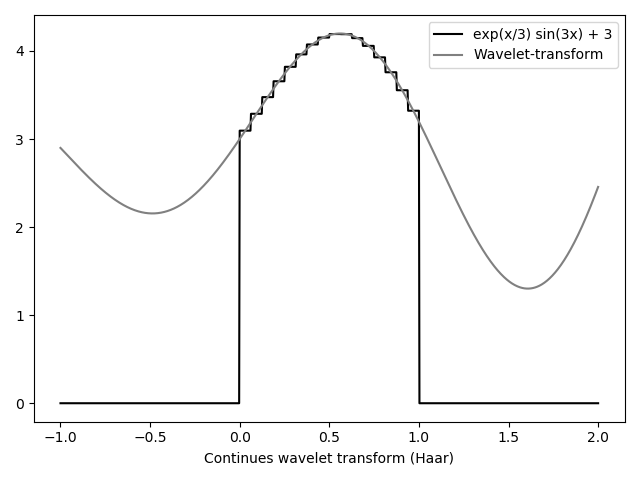




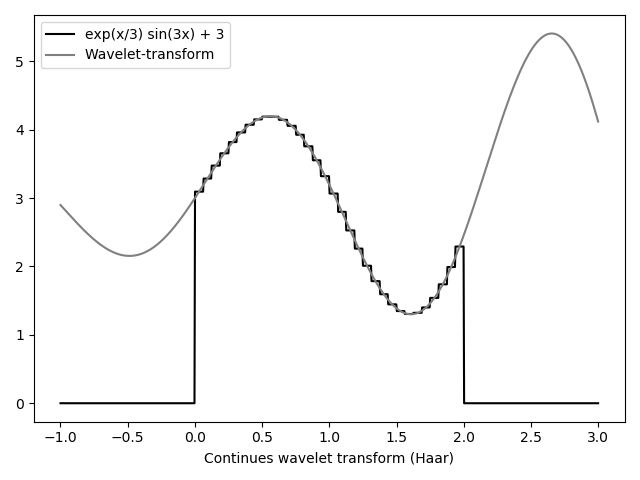
Первые графики показывают, что происходит с наложением двух синусоид, последние же с конкатенацией. Оба сигнала рассмотрены с шумом из нормального распределения. Мы можем заметить, что преобразования Фурье не отличаются значительно, потому что теряется локализация по времени. По скейлограммам же, полученных с помощью непрерывного вейвлет-преобразования, мы можем посмотреть структуру сигнала в разные моменты времени.

Дальше рассмотрим задачи анализа и синтеза(2):

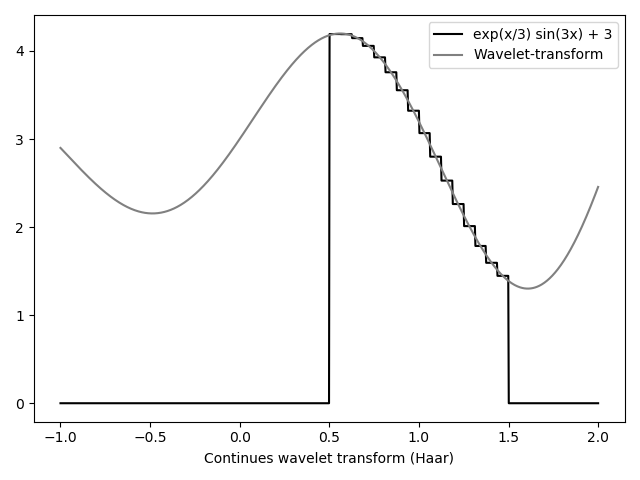
* Восстановление непрерывной функции на отрезке [0, 1] с 4 уровнями



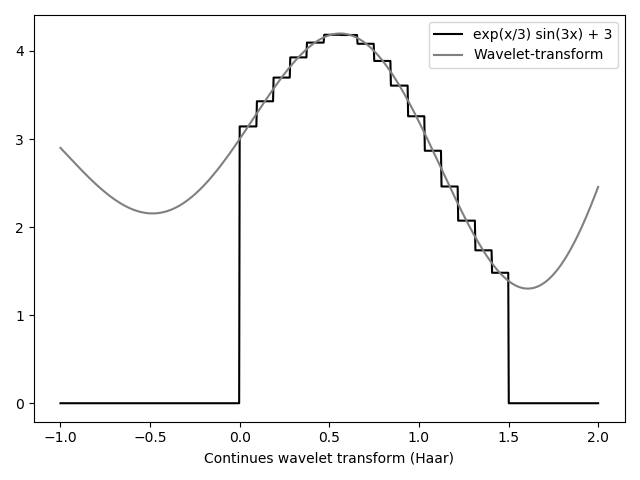
* Восстановление функции на отрезке [0, 2] полученное расширением базиса



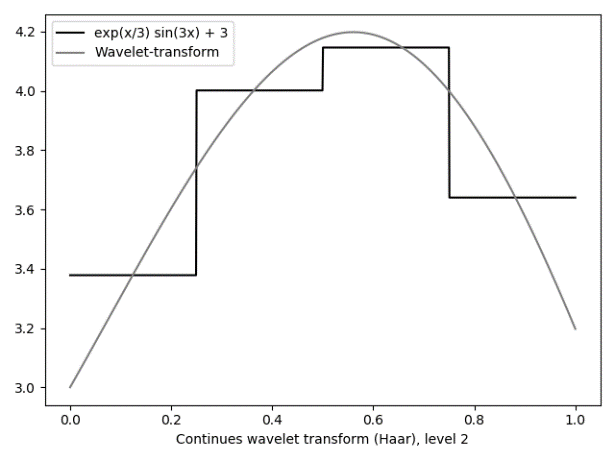
* Восстановление функции на отрезке [0.5, 1.5] через смещение материнского вейвлета и масштабирующей функции

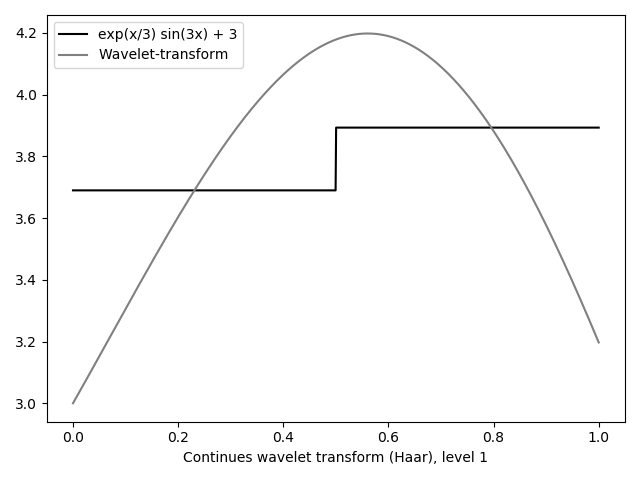


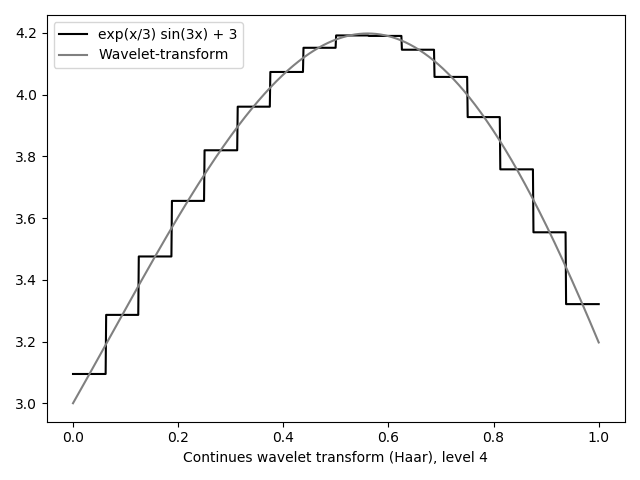
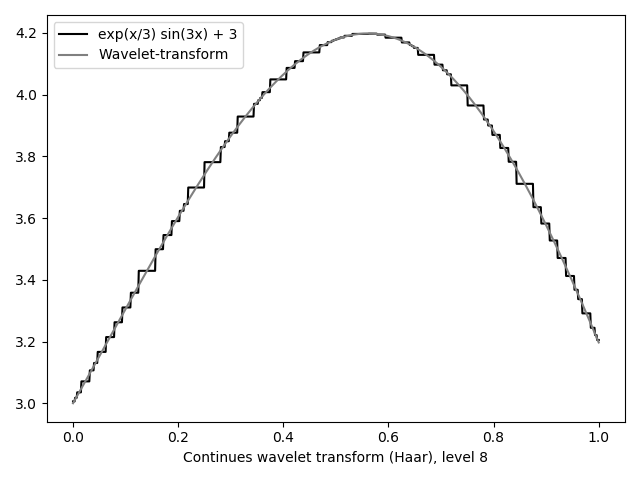
* Восстановление функции на отрезке [0, 1.5] через растяжение материнского вейвлета и масштабирующей функции



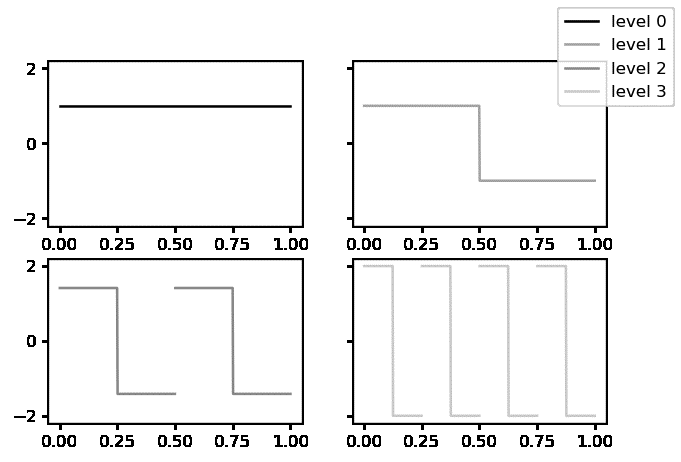
Можно использовать любые комбинации из этих способов, но самые простые в реализации последние два, также они не увеличивают количество вычислений.

Покажем, как детализация зависит от уровня:

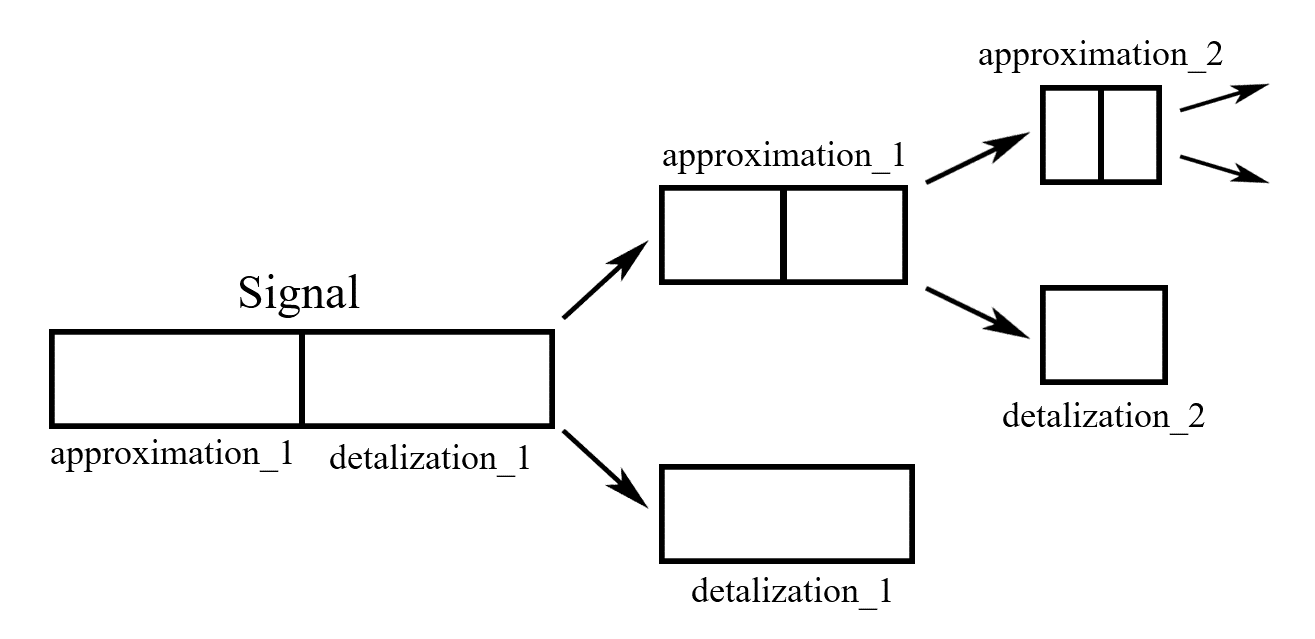


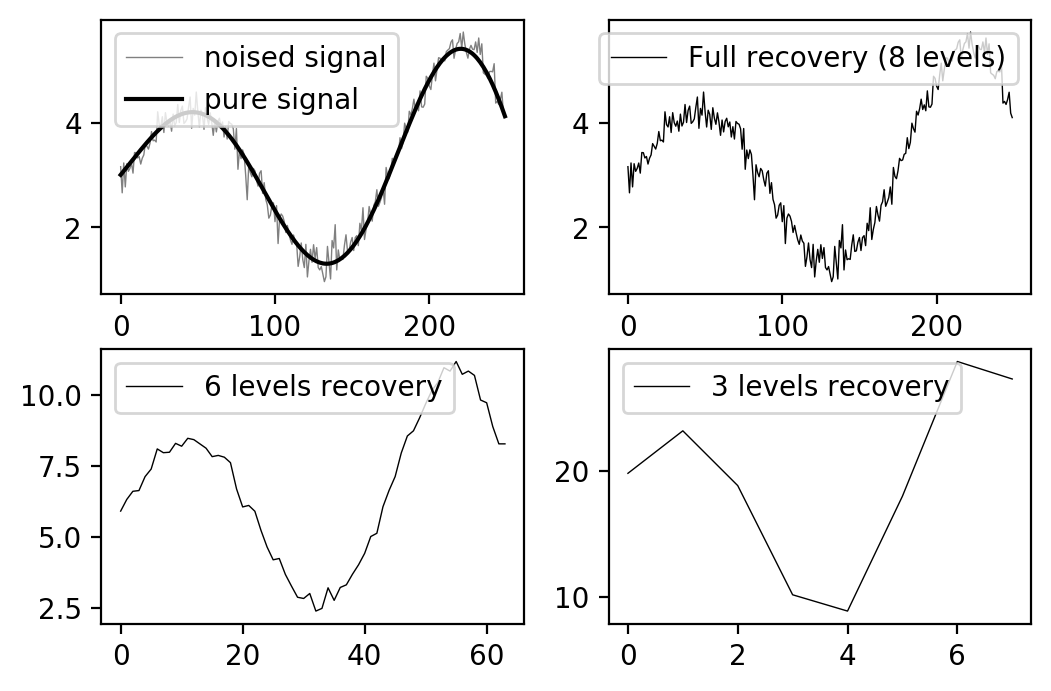


Сам базис Хаара выглядит так:



Рассмотрим дискретное преобразование Хаара. Его можно описать схемой:





На показанных графиках, мы можем сравнить различные уровни приближений(3).

Заключение

В курсовом проекте получены следующие основные результаты:

* Изучены свойства вейвлетов, в частности вейвлетов Хаара
* Приведены математические основы для работы с вейвлетами
* Проведены компьютерные эксперименты по работе с вейвлетами Хаара

Список литературы

1. I. Daubechies. Ten Lectures on Wavelets / I. Daubechies // Capital City Press, Montpelier, Vermont - 1992
2. F. Abramovich,. Wavelet analysis and its statistical applications / F. Abramovich, T. Bailey, and T. Sapatinas // JRSSD, (48):1–30 - 2000
3. Wavelet transform – Wikipedia [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Wavelet\_transform, свободный.
4. Haar wavelet – Wikipedia [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Haar\_wavelet, свободный.
5. Fourier transform – Wikipedia [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Fourier\_transform
6. Pedro A. Morettin. A wavelet analysis for time series / Pedro A. Morettin , Chang Chiann // Journal of Nonparametric Statistics - 2007
7. Charles K. Chui. An Introduction to Wavelets / Charles K. Chui // Texas A&M University, College Station, Texas - 2001
8. Tao Li. A Survey on Wavelet Applications in Data Mining / Tao Li, Qi Li, Shenghuo Zhu, Mitsunori Ogihara // ACM SIGKDD Explorations Newsletter  - 2002

Приложение

#Fourier transform (1)

import numpy as np

import matplotlib.pyplot as plt

frequency = 1000

N = 1000

def describe(signal):

n = len(signal)

signal\_with\_noise = signal + np.random.normal(0, 1, n)

plt.figure(dpi=200)

plt.subplot(2, 1, 1)

plt.plot(np.arange(n) / frequency, signal\_with\_noise,'gray',

label='noised signal', linewidth=0.5)

plt.plot(np.arange(n) / frequency, signal, 'black',

label='pure signal')

plt.legend()

plt.subplot(2, 1, 2)

specter = np.fft.rfft(signal\_with\_noise)

plt.plot(np.fft.rfftfreq(n, 1./frequency), np.abs(specter), 'black')

plt.legend(['Fourier transform'])

plt.show()

coef, freqs=pywt.cwt(signal ,np.arange(1,200),'mexh')

plt.matshow(coef, cmap='Greys')

plt.show()

def sin\_freq(t, freq):

return np.sin(2 \* np.pi \* freq \* t / frequency)

signal = np.array([sin\_freq(t, 30) + 2 \* sin\_freq(t, 7)

for t in range(N)])

describe(signal)

first\_half = np.array([sin\_freq(t, 30) for t in range(N)])

second\_half = np.array([2 \* sin\_freq(t, 7) for t in range(N)])

signal = np.append(first\_half, second\_half)

describe(signal)

# Continues Haar Transform (2)

import numpy as np  
from scipy import integrate  
import matplotlib.pyplot as plt  
  
class wavelet\_series:  
 def \_\_init\_\_(self, g, levels=8):  
 self.levels = levels  
 mother\_wavelet = lambda x: 0 if x < 0 else

1 if x < 0.5 else -1 if x <= 1 else 0  
  
 self.scaling = lambda x: 1 if 0 <= x <= 1 else 0  
  
 self.basis = [[(lambda i, j:

lambda x: 2 \*\* (i / 2) \* mother\_wavelet(2 \*\* i \* x - j))(i, j)  
 for j in range(2 \*\* i)] for i in range(levels)]  
  
 self.coef = [[integrate.quad(lambda x: g(x) \* self.basis[i][j](x),

0, 1)[0] for j in range(2 \*\* i)] for i in range(levels)]  
  
 self.scaling\_coef = integrate.quad(

lambda x: g(x) \* self.scaling(x), 0, 1)[0]

print(self.coef)  
 print(self.scaling\_coef)  
  
 def \_\_call\_\_(self, point):  
 value = 0  
 for i in range(self.levels):  
 for j in range(2 \*\* i):  
 value += self.coef[i][j] \* self.basis[i][j](point)  
 return value + self.scaling\_coef \* self.scaling(point)  
  
g = lambda x: np.exp(x / 3) \* np.sin(3 \* x) + 3  
xs = np.linspace(1e-8, 1 - 1e-8, 1000)  
  
f = wavelet\_series(g)  
plt.plot(xs, list(map(f, xs)), 'black', label='exp(x/3) sin(3x) + 3')  
plt.plot(xs, list(map(g, xs)), 'grey', label='Wavelet-transform')  
plt.legend()  
plt.xlabel('Continues wavelet transform (Haar)')  
plt.show()

#Discrete wavelet transform (3)

import numpy as np

import matplotlib.pyplot as plt

import pywt

frequency = 1000

N = 1024

def describe(signal):

n = len(signal)

wavelet = pywt.Wavelet('Haar')

signal\_with\_noise = signal + np.random.normal(0, 1, n) / 5

plt.figure(dpi=200)

plt.subplot(2, 2, 1)

plt.plot(signal\_with\_noise,'gray',

label='noised signal', linewidth=0.5)

plt.plot(signal, 'black', label='pure signal')

plt.legend()

coefs = pywt.wavedec(signal\_with\_noise, wavelet, level=8)

plt.subplot(2, 2, 2)

plt.plot(pywt.waverec(coefs, wavelet), 'black',

label='Full recovery (8 levels)', linewidth=0.5)

plt.legend()

plt.subplot(2, 2, 3)

plt.plot(pywt.waverec(coefs[:-2], wavelet), 'black',

label='6 levels recovery', linewidth=0.5)

plt.legend()

plt.subplot(2, 2, 4)

plt.plot(pywt.waverec(coefs[:-5], wavelet), 'black',

label='3 levels recovery', linewidth=0.5)

plt.legend()

plt.show()

describe((lambda x: np.exp(x / 3) \* np.sin(x \* 3) + 3)

(np.linspace(0, 3, 250)))