

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования**
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ РАБОТАМ

«Методы обработки сигналов и изображений»

Направление подготовки 09.04.02 «Информационные системы и технологии»
Направленность (профиль) «Искусственный интеллект, математическое моделирование и
суперкомпьютерные технологии в разработке информационных систем»
Квалификация выпускника: магистр

Ставрополь
2026

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ЛАБОРАТОРНАЯ РАБОТА 1 Исследование и применение дискретного преобразования Фурье	8
ЛАБОРАТОРНАЯ РАБОТА 2 Исследование свойств различных видов окон	12
ЛАБОРАТОРНАЯ РАБОТА 3 Синтез и исследование КИХ-фильтров	17
ЛАБОРАТОРНАЯ РАБОТА 4 Синтез и исследование БИХ-фильтров.....	22
ЛАБОРАТОРНАЯ РАБОТА 5 Исследование алгоритмов изменения частоты дискретизации.....	27
ЛАБОРАТОРНАЯ РАБОТА 6 Спектральный анализ и синтез сигналов средствами MATLAB	32
ЛАБОРАТОРНАЯ РАБОТА 7 Вейвлет-анализ и синтез сигналов в MATLAB	45
ЛАБОРАТОРНАЯ РАБОТА 8 Вейвлет-обработка и компрессия сигналов в MATLAB.....	56
ЛАБОРАТОРНАЯ РАБОТА 9 Основные приложения и команды по обработке изображений в Matlab	66
ЛАБОРАТОРНАЯ РАБОТА 10 Преобразование яркости изображений.....	75
ЛАБОРАТОРНАЯ РАБОТА 11 Пространственная фильтрация изображений.....	79
ЛАБОРАТОРНАЯ РАБОТА 12 Восстановление изображений.....	87
ЛАБОРАТОРНАЯ РАБОТА 13 Обработка цветных изображений.....	93
ЛАБОРАТОРНАЯ РАБОТА 14 Сжатие изображений	100
ЛАБОРАТОРНАЯ РАБОТА 15 Сегментация изображений	105
ЛАБОРАТОРНАЯ РАБОТА 16 Коррекция неравномерной засветки изображения.....	111
ЛАБОРАТОРНАЯ РАБОТА 17 Формирование цифровых латентных изображений.....	117
ЛАБОРАТОРНАЯ РАБОТА 18 Методы выявления скрытой информации в латентных изображениях.....	123
ЛАБОРАТОРНАЯ РАБОТА 19 Кластеризация данных алгоритмом FOREL	129
ЛАБОРАТОРНАЯ РАБОТА 20 Нахождения дискриминантной функции по прецедентам методом потенциальных функций.....	135
ЛАБОРАТОРНАЯ РАБОТА 21 Вычисление дискриминантной функции методом потенциальных функций.....	139
ЛАБОРАТОРНАЯ РАБОТА 22 Вычисление дискриминантной функции путем представления потенциальной с помощью системы базисных функций	143
ЛАБОРАТОРНАЯ РАБОТА 23 Простейшая гистограмма с регулярными ячейками	149
ЛАБОРАТОРНАЯ РАБОТА 24 Вычисление гистограммы с нерегулярными ячейками ..	154
ЛАБОРАТОРНАЯ РАБОТА 25 Вычисление гистограммы с нерегулярными ячейками ..	159
ЛАБОРАТОРНАЯ РАБОТА 26 Подгонка кривых.....	167
ЛАБОРАТОРНАЯ РАБОТА 27 Оценка качества подгонки кривых.....	172
ЛАБОРАТОРНАЯ РАБОТА 28 Частотно-временной анализ нестационарных сигналов методом Гильберта-Хуанга	175
ЛАБОРАТОРНАЯ РАБОТА 29 Алгоритм эмпирической декомпозиции сигнала.....	181

ЛАБОРАТОРНАЯ РАБОТА 30 Построение Байесовского классификатора по выборке двумерных нормально распределенных векторов. Построение Байесовского классификатора по прецедентам	192
ЛИТЕРАТУРА И ИСТОЧНИКИ	201

Введение

Целью освоения дисциплины «Методы обработки сигналов и изображений» является получение знаний, умений и навыков по использованию методов и технологий искусственного интеллекта, применению сквозных цифровых технологий искусственного интеллекта в прикладных, а также формирование на их основе компетенций будущего магистра по направлению подготовки 09.04.02. – Информационные системы и технологии и направленности (профилю) «Искусственный интеллект, математическое моделирование и суперкомпьютерные технологии в разработке информационных систем». Приобретение навыков профессионального использования сквозных цифровых технологий искусственного интеллекта в прикладных областях.

Задачи дисциплины: использование математического аппарата в контексте аналитических работ в информационно-технологических проектах; изучение современных методов и технологий в области искусственного интеллекта; использование сквозных цифровых технологий искусственного интеллекта в прикладных областях.

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-8 – Способен осуществлять руководство по созданию и развитию систем и комплексов обработки данных, в том числе больших данных, для корпоративных и государственных заказчиков	ПК-8ид-8.1 - Участвует в создании (модернизации) общедоступных платформ для хранения наборов данных, соответствующих методологиям описания, сбора и разметки данных; хранения наборов данных (в том числе звуковых, речевых, медицинских, метеорологических, промышленных данных и данных	Правильно выбирает общедоступные платформы для хранения наборов данных, соответствующих методологиям описания, сбора и разметки данных (в том числе звуковых, речевых, медицинских, метеорологических, промышленных данных и данных систем видеонаблюдения) и участвует в их создании или модернизации для обеспечения потребностей организаций разработчиков в области искусственного интеллекта.

	систем видеонаблюдения) на общедоступных платформах для обеспечения потребностей организаций разработчиков в области искусственного интеллекта	
	ПК-8 _{ид-8.2} - Осуществляет руководство по созданию и развитию систем и комплексов обработки данных, в том числе больших данных, для корпоративных и государственных	Осуществляет руководство по созданию и развитию систем и комплексов обработки данных, в том числе больших данных
ПК-10 – Способен руководить проектами по созданию, внедрению и использованию одной или нескольких сквозных цифровых технологий искусственного интеллекта в прикладных областях	ПК-10 _{ид-10.1} . Исследует и анализирует развитие новых направлений и перспективных методов и технологий в области искусственного интеллекта, участвует в исследовательских проектах по развитию новых направлений в области искусственного интеллекта (алгоритмическая имитация биологических систем принятия решений, автономное самообучение и развитие	Правильно выбирает и анализирует развитие новых направлений и перспективных методов и технологий в области искусственного интеллекта. Участвует в исследовательских проектах по развитию новых направлений в области искусственного интеллекта выбирает, разрабатывает и проводит экспериментальную проверку работоспособности программных компонентов систем.

	адаптивности алгоритмов к новым задачам, автономная декомпозиция сложных задач, поиск и синтез решений)	
	ПК-10 ^{ид-10.2} – Руководит проектами по созданию, внедрению и использованию сквозных цифровых технологий искусственного интеллекта в прикладных областях	Участвует и осуществляет руководство проектами по созданию, внедрению и использованию сквозных цифровых технологий искусственного интеллекта в прикладных областях.
ПК-12 – способен разрабатывать и исследовать теоретические и экспериментальные модели объектов профессиональной деятельности на основе искусственного интеллекта, математического моделирования и суперкомпьютерных технологий и методов искусственного интеллекта	ПК-12 ^{ид-12.1} – Разрабатывает методику выполнения аналитических работ в контексте исследования модели объектов профессиональной деятельности на основе методов математического моделирования и искусственного интеллекта.	Правильно выбирает и грамотно применяет методы математического моделирования и искусственного интеллекта для разработки методики выполнения аналитических работ при исследовании моделей объектов профессиональной.
	ПК-12 ^{ид-12.2} – Планирует и организует аналитические работы в информационно-технологическом проекте на основе суперкомпьютерных технологий и методов искусственного интеллекта	Грамотно планирует и организует аналитические работы в информационно-технологическом проекте на основе суперкомпьютерных технологий и методов искусственного интеллекта

ЛАБОРАТОРНАЯ РАБОТА 1

Исследование и применение дискретного преобразования Фурье

Цель: приобретение практических навыков работы с дискретным преобразованием Фурье (ДПФ).

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Дискретное преобразование Фурье применяется для дискретизированных во времени периодических процессов. Допустим, имеется некоторая выборка из n временных отсчетов $x(n)$, $n = 0, \dots, N-1$. Причем, предполагается, что за пределами указанного интервала данная выборка периодически повторяется. Тогда ДПФ вычисляется по формуле:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \quad (1)$$

и представляет собой последовательность комплексных дискретных спектральных отсчетов равномерно распределенных на отрезке частотной оси от 0 до частоты дискретизации F_s . За пределами указанного интервала эта последовательность также периодически повторяется.

Обратное ДПФ (ОДПФ) вычисляется по формуле:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N} \quad (2)$$

Таким образом, существует взаимно однозначное соответствие между периодическим дискретным процессом и его периодическим дискретным спектром. Причем как во временной, так и в спектральной области длина периода составляет N отсчетов.

Из формул (1) и (2) видно, что ДПФ и ОДПФ требует N^2 арифметических операций. Однако, если $N=2m$, где m – любое натуральное число возможно применение быстрого преобразования Фурье (БПФ), и обратного быстрого преобразования Фурье (ОБПФ), для выполнения которых требуется приблизительно $\frac{N}{2} \cdot \log_2 N$ операций. Если $N=210$, то БПФ работает приблизительно в 200 раз быстрее по сравнению с ДПФ. И этот выигрыш возрастает с ростом N . При этом результаты ДПФ и БПФ полностью совпадают.

Важным практическим применением преобразования Фурье является спектральный анализ аддитивной смеси сигнала и помех. Если ширина спектра и амплитуда полезного сигнала и помехи заметно различаются, это может быть использовано для подавления помех. Предположим, что имеется узкополосная, большая по амплитуде помеха на фоне широкополосного сигнала. В этом случае подавление помехи может осуществляться методом режекции спектральных составляющих, превышающих некоторый порог.

На границах временной выборки функция $x(n)$, как правило, испытывает разрыв. Это приводит к расширению основания спектра и затрудняет подавление помех. Эффект расширения спектра можно ослабить путем плавного уменьшения отсчетов выборки при движении от ее центра к краям, что осуществляется за счет умножения отсчетов временной выборки на отсчеты функции окна. Более подробно свойства различных функций окон будут исследованы в следующей лабораторной работе.

В MATLAB вычисление спектра выполняется с помощью процедуры FFT (Fast Fourier Transform): $X = \text{FFT}(x)$. $\text{FFT}(x)$ вычисляет ДПФ от последовательности $x(n)$.

Если длина $x(n)$ равна $2m$, где m – целое, то для вычисления спектра процедура FFT использует более быстрый алгоритм – быстрое преобразование Фурье (БПФ).

Если длина $X(k)$ равна $2m$, где m – целое, то для вычисления спектра процедура IFFT использует обратное быстрое преобразование Фурье (ОБПФ).

Функция окна Хэмминга вычисляются в виде вектора-столбца с помощью процедуры $W = \text{hamming}(n)$, где n – длина последовательности. При умножении на последовательность $x(n)$ необходимо использовать транспонированный вектор окна W' .

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса.

Методика и порядок выполнения работы

1. Задайте входной процесс в виде суммы полезного ЧМ сигнала и АМ помехи.
2. Найдите спектр входного процесса и спектр помехи.
3. Исключите помеху методом непосредственной обработки спектра.
4. Перейдите обратно во временную область.
5. Найдите разность полезного и выходного сигнала.
6. Сравните временные диаграммы исходного процесса, полученного выходного сигнала, исходного полезного сигнала (без помехи) и разности полезного и выходного сигнала.

7 Рассматривая вышеприведенную разность как аддитивную помеху, вычислить отношение сигнал/помеха в децибелах до и после подавления помехи.

8. Умножьте входной процесс на окно Хэмминга и повторите п.п. 1.2 – 1.7

9. Сравните результаты подавления помехи с применением и без применения окна.

Порядок выполнения:

1. Задайте входной процесс $s_{\text{inp}}(n)$ в виде суммы полезного ЧМ сигнала:

$$\text{signal} = \sin(w_0 * ts * n + (w_d/w_m) * \sin(w_m * ts * n) + 1),$$

и помехи:

$$\text{interference} = A * (1 + 0.5 * \sin(2 * \pi * 0.01 * F_s * ts * n)) * \sin((w_0 + 0.7 * w_d) * ts * n),$$

где $w_0 = 2 * \pi * (0.1 + \# * 0.005) * F_s$, $w_d = (0.07 + \# * 0.003) * F_s$, $A = 2.5 + \# * 0.05$,
 $\#$ - номер варианта, $w_m = 2 * \pi * 0.001 * F_s$, $F_s = 4 \text{ МГц}$ – частота дискретизации, $ts = 1/F_s$, $n = 0, \dots, 10000$,

2. Найдите спектр входного процесса и спектр помехи с помощью процедуры `fft` и построить график его модуля: `abs(fft(...))`

3. Проанализируйте полученный спектр и определить наименьший пороговый уровень, выше которого располагаются только спектральные составляющие помехи. С помощью процедуры `find` обнулить все спектральные составляющие, лежащие выше порога. (Графики спектров входного процесса, помехи и процесса с подавленной помехой построить в одном окне с использованием функции `subplot(3,1,*)`).

4. Перейдите от спектра к временному процессу с помощью процедуры `ifft`.

5. Найдите разность исходного полезного сигнала и полученного процесса с подавленной помехой (при идеальном подавлении помехи эта разность оказалась бы нулевой). Построить в одном окне графики исходного процесса, полезного сигнала, процесса с подавленной помехой и найденной разности.

6. Найдите отношение сигнал/помеха до и после подавления помехи в децибелах: $(S/I) = 10 \log_{10}(S/I)$, где S – энергия полезного сигнала, рассчитанная как сумма квадратов временных отсчетов сигнала по формуле `sum(signal.^2)`, I – энергия помехи. (После подавления помехи в качестве I рассчитывается энергия разности сигнала и процесса с подавленной помехой).

7. Умножьте сигнал и помеху на окно Хэмминга после чего повторите п.п. 2.1 – 2.5. При выполнении п.2.4 и п.2.5 использовать полезный сигнал и помеху, умноженные на окно.

8. Сравните спектры, временные диаграммы и значения (S/I) до и после применения окна.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Почему разность сигнала и процесса с подавленной помехой не равна нулю?
2. Каким образом можно автоматизировать вычисление порога для подавления помехи, если заранее известно, что спектр сигнала значительно шире спектра помехи, а амплитуда помехи больше амплитуды сигнала?
3. По какой причине пороговый уровень следует минимизировать?
4. За счет чего улучшается подавление помехи при использовании окна?

ЛАБОРАТОРНАЯ РАБОТА 2

Исследование свойств различных видов окон

Цель: приобретение практических навыков применения различных видов окон при исследовании спектров сигналов.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Известно, что для нахождения спектра сигнала используется преобразование Фурье (для дискретизированных сигналов это Дискретное Преобразование Фурье или ДПФ). Как правило, преобразование Фурье берется от конечных временных выборок сигнала, что приводит к искажению спектра. Причина в том, что взятие конечной временной выборки от длительного сигнала эквивалентно умножению его на прямоугольную функцию – функцию окна, равную единице в пределах выборки и равную нулю за ее пределами.

При умножении двух функций во временной области из спектр находится как свертка спектров исходных функций. Поэтому на спектр длительного процесса накладывается функция $\sin(x)/x$ – спектр прямоугольного окна, боковые выбросы которой затухают достаточно медленно, со скоростью 6 дБ на октаву. Образно говоря, резкое ограничение временной выборки приводит к звону в спектре, что затрудняет различение близко лежащих спектральных компонент.

Исключение составляет тот случай, когда в выборке, взятой из периодического процесса, укладывается целое количество периодов от периодического процесса.

Для того, чтобы ослабить искажения спектра, при формировании временных выборок применяются другие оконные функции, плавно затухающие по краям. В спектре таких функций боковые выбросы малы по сравнению с главным (центральным лепестком) и быстро затухают при удалении от центра.

Таблица 3.1 Виды окон и соответствующие им стандартные процедуры MATLAB

Название окна	Процедура MATLAB
Прямоугольное	$w = \text{boxcar}(n)$
Треугольное	$w = \text{triang}(n)$
Cos2 (фон Ханна)	$w = \text{hanning}(n)$
Хэмминга	$w = \text{hamming}(n)$
Блэкмана	$w = \text{blackman}(n)$
Кайзера	$w = \text{kaiser}(n, \beta)$

1. Параметр β ($\beta > 1$) в окне Кайзера определяет высоту боковых выбросов. С ростом β боковые выбросы уменьшаются, а главный лепесток расширяется.

2. Окна задаются в виде столбца из n элементов, поэтому перед умножением на сигнал w необходимо транспонировать.

Таблица 3.2. Сравнительные характеристики окон

Окно:	П рямо- угольн ое	Тр еуголь- ное	ф он Ханна	Хэ мминга	Б лэкман а	К айзера α = 3.0
Ширина главного лепестка по уровню 6дБ	0. 0242	0. 0356	0 .04	0. 036	0 .047	0 .0478
Полная ширина главного лепестка	0. 04	0. 08	0 .08	0. 1	0 .14	0 .14 - 70дБ
Высота наибольшего бокового выброса (дБ)	- 13	- 27	- 32	- 43	- 58	- 69
Скорость затухания боковых лепестков (дБ/окт)	- 6	- 12	- 18	-6	- 18	- 6

Примечание: ширина главного лепестка приводится в относительных частотах (при $F_s = 1$).

Для окна фон Ханна существует экономичный способ нахождения спектра методом свертки спектра окна со спектром не взвешенной (с прямоугольным окном) последовательности по формуле:

$$Y_w(k) = \frac{1}{2} \left\{ Y(k) - \frac{1}{2} [Y(k-1) + Y(k+1)] \right\}, \quad (1)$$

где $Y_w(k)$ – искомый спектр с окном фон Ханна, $Y(k)$ – спектр не взвешенной последовательности.

Таким образом, вместо n операций умножения исходной выборки на окно выполняется сложение смещенных на один отсчет спектров и два деления на 2, которые в цифровой обработке выполняются простым сдвигом на один разряд влево.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса.

Методика и порядок выполнения работы

1. Найдите спектры перечисленных во введении видов окон
2. Постройте графики спектров всех окон в дБ. Сравнить параметры окон с табличными (Табл.2).
3. Постройте в одних осях спектры окна Кайзера для нескольких β . Найдите β , при котором высота первого бокового выброса не превышает -66дБ относительно главного лепестка.
4. Задайте входной процесс в виде суммы АМ сигнала, тонального сигнала и шума.
5. Найдите спектры для входного процесса, применяя все перечисленные окна. Постройте графики спектров в одном окне и сравните спектры между собой. Обратите внимание на соотношение уровней сигнала и наличие боковых составляющих АМ сигнала.
6. Найдите спектр процесса с окном фон Ханна по формуле (1), построить его в одних осях со спектром, найденным обычным способом и сравнить.
7. Повторите п.п.1.4, 1.5 для $\# = 0$.

Порядок выполнения:

1. Задайте временные последовательности окон w , приведенных в таблице 1. Длина окна $n=50$. Транспонировать полученные последовательности и дополнить их последовательностью из 500 нулей: $[w' z(1,500)]$. При дополнении временной последовательности нулями форма ее спектра не меняется. Увеличивается только количество точек в спектре, что позволяет более подробно отобразить форму спектра.
2. Найдите спектры всех графиков с помощью `fft`. Постройте графики всех спектров в дБ в одном окне с использованием оператора `subplot(3,2,k)`. Озаглавьте каждый график, используя оператор `title`. Для удобства сравнения все графики построите в одинаковом масштабе по оси ординат, используя оператор `axis`. Масштаб по оси абсцисс задать в относительных частотах (от 0

до 1). Для того, чтобы главный лепесток спектра окна располагался в центре графика, примените оператор `fftshift(fft(...))`. Сравните параметры окон с табличными (Табл.2).

3. Постройте в одних осях спектры окна Кайзера для нескольких β . Найдите β , при котором высота первого бокового выброса не превышает -66дБ относительно главного лепестка. При анализе спектров использовать оператор `zoom`.

4. Задайте входной процесс в виде суммы АМ сигнала, тонального сигнала и нормально распределенного белого шума: частота дискретизации: $F_s = 2\text{МГц}$, частота АМ сигнала $f_1 = (100+\#)\text{кГц}$, частота модуляции АМ сигнала $f_m = 0.01F_s$, глубина модуляции АМ сигнала 30%, средняя амплитуда АМ сигнала $A_1=1$, частота тонального сигнала $f_2=(160+\#)$, амплитуда тонального сигнала $A_2=0.01$, амплитуда шума (СКО) $A_n=0.0005$, количество отсчетов процесса $N=300$, $\#$ - номер варианта.

5. Перейдите к относительным частотам f/F_s . Найдите спектры для входного процесса, применяя все перечисленные окна. Постройте графики спектров в одном окне, используя операторы `subplot(3,2,k)`. Озаглавьте каждый график, используя оператор `title`. Сравните спектры между собой. Для удобства сравнения все графики постройте в одинаковом масштабе по оси ординат, используя оператор `axis`. Ось абсцисс взять в относительных частотах.

6. Найдите спектр процесса с окном фон Ханна по формуле (1), постройте его в одних осях со спектром, найденным обычным способом и сравните. В качестве $Y(k)$ выберите спектр с прямоугольным окном. При сложении со смещенными спектрами недостающие крайние отсчеты считать нулевыми.

7. Повторить п.п.1.4, 1.5 для $\#=0$. Сопоставить спектры с прямоугольным окном и графики временных процессов для $\#=0$ и вашего $\#$.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. С каким из окон наблюдается наилучшее разделение сигналов?
2. С каким из окон наблюдается наилучшее разделение несущей и боковых составляющих АМ сигнала? В чем причина?

3. Выберите окно с которым достигается компромисс между разделением двух сигналов и выявлением структуры АМ сигнала.
4. В чем причина медленного спада боковых выбросов окна Хэмминга?
5. В каком из окон быстрый спад боковых выбросов сочетается с узким центральным лепестком? В чем недостаток этого окна?
6. В чем причина резкого улучшения разрешающей способности в спектре с прямоугольным окном при $\beta=0$?

ЛАБОРАТОРНАЯ РАБОТА 3

Синтез и исследование КИХ-фильтров

Цель: приобретение практических навыков разработки КИХ-фильтров.
Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Наиболее простым и наглядным методом разработки КИХ-фильтров является метод окна. Известно, что отсчеты импульсной характеристики цифрового фильтра $h(n)$ связаны с его частотной характеристикой $H(\omega)$ соотношением дискретного ряда Фурье:

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(\omega) e^{j\omega n} d\omega, \quad (1)$$

Для фильтра нижних частот с идеально прямоугольной амплитудно-частотной характеристикой (АЧХ) и частотой среза ω_0 получаем:

$$h(n) = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} e^{j\omega n} d\omega = \frac{\omega_0}{\pi} \text{Sinc}\left(\frac{\omega_0}{\pi} n\right), \quad (2)$$

где n изменяется от $-\infty$ до $+\infty$.

Очевидно, что реализация КИХ-фильтра с такой характеристикой невозможна. Для построения реального КИХ-фильтра необходимо ограничить $h(n)$ конечным числом отсчетов N , где N – порядок фильтра. При этом линейность фазочастотной характеристики (ФЧХ) обеспечивается за счет симметрии отсчетов импульсной характеристики $h(n)$ относительно центра, а максимум огибающей $h(n)$ определяет групповую задержку фильтра.

Импульсная характеристика, полученная таким образом, имеет резкие спады по краям, что приводит к появлению выбросов в АЧХ. Для сглаживания этих выбросов необходимо перемножить вектор его коэффициентов $h(n)$ с одним из окон.

Более точное отображение желаемой АЧХ достигается оптимизационными методами: методом Чебышевской аппроксимации, использующим алгоритм Ремеза и методом наименьших квадратов.

Алгоритм Ремеза основан на теории наилучшей равномерной аппроксимации Чебышева и позволяет проектировать оптимальные фильтры. Фильтры оптимальны в том смысле, что минимизируются максимальные отклонения действительной АЧХ от желаемой. При этом все максимальные отклонения оказываются равными между собой. Соотношение ошибок аппроксимации в полосе пропускания и полосе подавления задается с помощью весовой функции. Например, если в полосе пропускания значение весовой функции равно 1, а в полосе подавления равно 10, то точность аппроксимации в полосе подавления будет в 10 раз выше.

Метод наименьших квадратов, в отличие от алгоритма Ремеза, минимизирует суммарное среднеквадратичное отклонение желаемой АЧХ от действительной и в этом смысле является оптимальным. Это означает, что отдельные отклонения АЧХ, полученной по методу наименьших квадратов могут оказаться больше, чем в АЧХ, полученной по алгоритму Ремеза. Но общее среднеквадратичное отклонение будет наименьшим.

Среди перечисленных методов метод Ремеза обеспечивает наибольшую крутизну спада АЧХ при прочих равных условиях. Групповая задержка D всех видов КИХ-фильтров с линейной ФЧХ зависит только от порядка N и определяется по формуле:

$$D = (N-1)/2. \quad (3)$$

При выполнении рабочего задания потребуются следующие стандартные процедуры MATLAB.

Процедура `fir2` рассчитывает коэффициенты фильтра методом окна.

Например, последовательность операторов:

`N = 30;`

`F = [0 0.2 0.3 0.5];`

`A = [1 1 0 0];`

`H = fir2(N-1, f*2, a, hanning(N));`

позволяет рассчитать коэффициенты фильтра h 30 порядка, желаемая АЧХ которого равна 1 при частотах от 0 до $0.2 \cdot F_s$ (полоса пропускания), и 0 при частотах от $0.3 \cdot F_s$ до $0.5 \cdot F_s$ (полоса подавления). В полосе от $0.2 \cdot F_s$ до $0.3 \cdot F_s$ на АЧХ не накладывается ограничений. F_s – частота дискретизации в Гц.

В процедуре `fir2`, как и в других подобных процедурах MATLAB все частоты пронумерованы относительно частоты Найквиста: $F_n = F_s/2$. Поэтому в качестве аргумента частот в `fir2` подставляется не f , а $2 \cdot f$. Кроме того, вместо порядка фильтра N , равного длине импульсной характеристики, подставляется номер ее максимального отсчета при нумерации от 0 до $N-1$.

Процедура `firls` рассчитывает коэффициенты фильтра методом наименьших квадратов.

Например, последовательность операторов:

`N=30;`

`F = [0 0.2 0.3 0.5];`

`A = [1 1 0 0];`

`Weigh = [5, 1];`

`H = fir2(N-1, f*2, a, weight);`

позволяет рассчитать коэффициенты фильтра h 30 порядка, желаемая АЧХ которого равна 1 при частотах от 0 до $0.2 \cdot F_s$, и 0 при частотах от $0.3 \cdot F_s$ до $0.5 \cdot F_s$. Причем точность аппроксимации в полосе пропускания в 5 раз выше, чем в полосе подавления за счет применения весовой функции `weight`. При отсутствии весовой функции среди аргументов процедуры `firls` точность аппроксимации одинакова во всех полосах.

Процедура `remez` рассчитывает коэффициенты фильтра методом чебышевской аппроксимации по алгоритму Ремеза.

Например, последовательность операторов:

$N = 30;$

$F = [0 \ 0.2 \ 0.3 \ 0.5];$

$A = [1 \ 1 \ 0 \ 0];$

$Weight = [5, 1];$

$H = \text{remez}(N-1, f*2, a, \text{weight});$

позволяет рассчитать коэффициенты фильтра h 30 порядка, требуемая АЧХ которого равна 1 при частотах от 0 до $0.2*F_s$, и 0 при частотах от $0.3*F_s$ до $0.5*F_s$.

Во всех перечисленных процедурах может быть задана и более сложная АЧХ за счет соответствующих векторов f и a .

Процедура $[h, n] = \text{impz}(b, a);$ позволяет находить отсчеты импульсной характеристики h и их номера n для цифровых фильтров по их коэффициентам b и a . Для КИХ-фильтров $a = 1$, а $h = b$.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

1. Рассчитайте коэффициенты КИХ-фильтра нижних частот заданного порядка, с помощью формулами (1) и (2) с использованием окна и без него. Постройте частотные и импульсные характеристики для обоих случаев.

2. Синтезируйте три фильтра нижних частот с одинаковой частотой среза, используя методом окна, Ремеза и наименьших квадратов и построить их АЧХ и импульсные характеристики.

3. Увеличьте порядок фильтров синтезированных по методу окна и методу наименьших квадратов таким образом, чтобы крутизна спада АЧХ сравнялась с крутизной фильтра Ремеза. Постройте импульсные характеристики трех фильтров. По импульсным характеристикам определите задержки для трех фильтров и сравните их с расчетными, полученными из формулы (3).

4. Для фильтров, синтезированных по методу Ремеза и наименьших квадратов, ввести некоторую весовую функцию. Проанализируйте ее влияние на АЧХ.

Порядок выполнения:

1. Рассчитайте коэффициенты $h(n)$ фильтра порядка $N = 60 - \#$, где $\#$ – номер варианта, и частотой среза $F = 0.12 + \#/100$, пользуясь формулой (2). Постройте график импульсной характеристики фильтра. Она должна быть симметричной. Для достижения симметрии при четном N использовать сдвиг n на $1/2$. Постройте АЧХ фильтра. Для сглаживания выбросов АЧХ применить одно из окон. Постройте импульсную характеристику и АЧХ на тех же графиках, что и для предыдущего случая. Оцените, как влияет форма импульсной характеристики на форму АЧХ.

2. Синтезируйте три ФНЧ: методом окна (по процедуре `fir2`), методом наименьших квадратов (по процедуре `firls`) и методом чебышевской аппроксимации (по процедуре `remez`). Полоса пропускания всех ФНЧ - от 0 до F , полоса подавления - от $F+0.025$ до 0.5. Постройте АЧХ трех фильтров на одном графике и сравните их между собой.

3. Увеличьте порядок фильтров синтезированных по методу окна и методу наименьших квадратов таким образом, чтобы крутизна спада АЧХ сравнялась с крутизной фильтра Ремеза. Постройте АЧХ и импульсные характеристики трех фильтров. По импульсным характеристикам определите задержки для трех фильтров и сравните их с расчетными, полученными из формулы (3).

4. Для фильтров, синтезированных по методу Ремеза и наименьших квадратов, введите некоторую весовую функцию. Проанализируйте ее влияние на АЧХ. Постройте графики АЧХ.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Чем отличаются КИХ-фильтры I и II типов. В чем отличие их АЧХ?
2. Какова связь между величиной выбросов АЧХ и формой импульсной характеристики фильтра?

3. Убедитесь в том, что фильтр Ремеза действительно обеспечивает наименьшие абсолютные отклонения АЧХ от заданной по сравнению с фильтром наименьших квадратов при одинаковом порядке. Оцените, насколько максимальное отклонение АЧХ фильтра по методу наименьших квадратов больше, чем у фильтра Ремеза в вашем случае.

4. Как влияет весовая функция на форму АЧХ?

ЛАБОРАТОРНАЯ РАБОТА 4

Синтез и исследование БИХ-фильтров

Цель: приобретение практических навыков разработки БИХ-фильтров на базе их аналоговых прототипов.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Основное преимущество фильтров с бесконечной импульсной характеристикой (БИХ-фильтров) состоит в том, что при тех же требованиях к амплитудно-частотной характеристике их порядок оказывается значительно меньше, чем у КИХ-фильтров. Однако фазо-частотная характеристика БИХ-фильтров - нелинейна.

Распространенный метод расчета БИХ-фильтров основан на преобразовании аналоговых фильтров в цифровые. Это связано с тем, что методы расчета аналоговых фильтров хорошо известны, просты и эффективно программируются.

Последовательность действий расчета БИХ-фильтров по их аналоговым прототипам.

1. Расчет аналогового ФНЧ заданного порядка типа Баттерворта, Чебышева I или II типа, или эллиптического с единичной частотой среза.

2. Пересчет фильтра на реальную частоту и преобразование ФНЧ в ФВЧ, полосовой или режекторный фильтр, если это требуется.

3. Дискретизация аналогового прототипа методом инвариантного преобразования импульсных характеристик или методом билинейного преобразования.

Метод инвариантного преобразования импульсных характеристик основан на непосредственной дискретизации импульсных характеристик аналоговых фильтров. Если аналоговая цепь описывается импульсной характеристикой вида

$$h_a(t) = e^{-at},$$

и передаточной функцией

$$H(p) = \frac{1}{p + a},$$

то после дискретизации имеем:

$$h(n) = e^{-a\tau n},$$

где τ - период дискретизации, и

$$H(z) = \frac{1}{1 - e^{-a\tau} z^{-1}}$$

В общем случае передаточную функцию аналогового устройства (фильтра), методом разложения на простые дроби, можно представить в виде

$$H(p) = \sum_{i=1}^M \frac{A_i}{p + p_i}.$$

Тогда, по аналогии с предыдущими формулами, переходим к передаточной функции цифрового БИХ-фильтра:

$$H(z) = \sum_{i=1}^M \frac{A_i}{1 - e^{-p_i \tau} z^{-1}}.$$

В результате такого перехода участок мнимой оси на p -плоскости, ограниченный частотами $\pm fs/2$ ($fs=1/\tau$ - частота дискретизации), отображается на единичную окружность z -плоскости. Это означает, что частотные характеристики аналогового прототипа после его дискретизации периодически повторяются с перекрытием на частотах, кратных $fs/2$. Если АЧХ аналогового прототипа имеет достаточно большое значение за пределами $fs/2$, то применение данного метода дискретизации приводит к значительным искажениям характеристик БИХ-фильтра по сравнению с прототипом.

От указанного недостатка свободен метод билинейного преобразования, который основан на математическом отображении p -плоскости на z -плоскость по следующим правилам:

$$H(z) = H(p) \Big|_{p=2fs \frac{z-1}{z+1}}.$$

Такое преобразование отображает ось $j\Omega$ (от $\Omega = \infty$ до $\Omega = -\infty$) на p -плоскости в единичную окружность $e^{j\omega}$ (от $\omega = -\pi$ до $\omega = \pi$) на z -плоскости. (За пределами интервала от $-\pi$ до π частотные характеристики цифрового фильтра периодически повторяются.) При этом

$$\omega = 2 \arctg\left(\frac{\Omega}{2fs}\right).$$

Таким образом, частотная ось Ω аналогового фильтра отображается на частотную ось ω цифрового фильтра по нелинейному закону – с ростом частоты ось ω сжимается. Для ФНЧ, например, это приводит к более крутому спаду АЧХ цифрового фильтра по сравнению с аналоговым прототипом.

Иногда, при использовании метода билинейных преобразований, в АЧХ аналогового прототипа вводятся предискажения для получения заданных частот среза в цифровом фильтре:

$$\Omega_a = 2 * Fs * \tan(\pi * Fd / Fs), (1)$$

где Ω_a – частота (частоты) на АЧХ аналогового прототипа с учетом предискажений, Fd – требуемая частота (частоты) цифрового фильтра, Fs – частота дискретизации.

При использовании языка MATLAB расчет аналоговых прототипов на этапе (1) выполняется с помощью стандартных процедур `buttap`, `cheblap`, `cheb2ap`, `ellipap`. Затем, на этапе (2), с помощью процедур `lp2lp`, `lp2hp`, `lp2bp`, `lp2bs`, осуществляется пересчет частоты и преобразование типа фильтра. И на этапе (3), с помощью процедур `impinvar` и `bilinear` аналоговые фильтры преобразуются в цифровые.

Процедуры `butter`, `cheby1`, `cheby2`, `ellip` выполняют все три этапа в комплексе, объединяя вышеприведенные процедуры. Причем преобразование в цифровые фильтры выполняется методом билинейного преобразования

(bilinear) с предискажением АЧХ аналогового прототипа. В большинстве практических случаев пользуются именно этими процедурами. Если в параметрах указанных процедур присутствует указатель 's', то этап (3) не выполняется, и по ним рассчитываются аналоговые фильтры.

Процедура `butter` рассчитывает коэффициенты передаточной функции $H(z)$ цифрового БИХ-фильтра Баттерворта.

Например, последовательность операторов:

`N = 6;` % Порядок фильтра

`F = 0.25;` % Частота среза.

`[b, a] = butter(N-1, F*2);`

позволяет рассчитать коэффициенты ФНЧ порядка $N=6$ с относительной частотой среза на уровне -3дБ (0,707), равной F .

В процедуре `butter`, как и в других подобных процедурах MATLAB все частоты пронормированы относительно частоты Найквиста: $F_n = F_s/2$. Поэтому в качестве аргумента частот в `fir2` подставляется не f , а $2*f$. Кроме того, вместо порядка фильтра N подставляется максимальный номер коэффициента при нумерации от 0 до $N-1$.

Процедура `cheby1` рассчитывает коэффициенты передаточной функции $H(z)$ цифрового БИХ-фильтра Чебышева типа 1.

Например, последовательность операторов:

`N = 6;` % Порядок фильтра

`Rp = 3;` % Неравномерность в полосе пропускания

`F = 0.25;` % Частота среза.

`[b, a] = cheby1(N-1, Rp, F*2);`

позволяет рассчитать коэффициенты ФНЧ порядка N с относительной частотой среза F на уровне -3дБ (0,707).

Процедура `cheby2` рассчитывает коэффициенты передаточной функции $H(z)$ цифрового БИХ-фильтра Чебышева типа 2.

Например, последовательность операторов:

`N = 6;` % Порядок фильтра

`Rs = 20;` % Неравномерность в полосе подавления

`F = 0.25;` % Частота среза.

`[b,a] = cheby2(N-1, Rp, F*2);`

позволяет рассчитать коэффициенты ФНЧ порядка N с относительной частотой среза F на уровне -20дБ (0,1).

Процедура `ellip` рассчитывает коэффициенты передаточной функции $H(z)$ цифрового эллиптического БИХ-фильтра.

Например, последовательность операторов:

`N = 6;`

`Rp = 3;` % Неравномерность в полосе пропускания

`Rs = 20;` % Неравномерность в полосе подавления

`F = 0.25;`

`[b, a] = ellip(N-1, Rp, Rs, F*2);`

позволяет рассчитать коэффициенты ФНЧ порядка N с относительной

частотой среза F на уровне -3дБ ($0,707$).

Для расчета полосовых фильтров параметр F должен быть вектором из 2-х элементов. При расчете ФВЧ или режекторных фильтров вводятся параметры «high» и «stop».

Все перечисленные процедуры могут быть использованы для расчета передаточных функций $H(p)$ соответствующих аналоговых фильтров. Для этого вводится параметр 's'. Например, процедура

$[b, a] = \text{ellip}(N-1, R_p, R_s, 2*\pi*F, 's')$

рассчитывает параметры аналогового эллиптического ФНЧ.

Причем частота среза в аналоговом фильтре должна быть задана в виде круговой частоты $\Omega = 2*\pi*f$.

Процедура `bilinear` пересчитывает коэффициенты аналогового фильтра в коэффициенты цифрового фильтра:

$[bz, az] = \text{bilinear}(b, a, Fs)$.

Аналогично действует и процедура `impinvar`:

$[bz, az] = \text{impinvar}(b, a, Fs)$.

Процедура расчета частотных характеристик аналоговых фильтров $[H, \Omega] = \text{freqs}(b, a, 512)$ действует аналогично процедуре `freqz`. При этом вектор частот Ω состоит из отсчетов круговой частоты.

Процедура `zplane(b, a)` отображает на графике нули и полюса.

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

1. Рассчитайте коэффициенты четырех видов цифровых БИХ-фильтров нижних частот: Баттерворта, Чебышева типа 1 и 2, и эллиптического. Построить АЧХ, ФЧХ, диаграммы нулей и полюсов.

2. Рассчитайте цифровой полосовой фильтр Баттерворта по его аналоговому прототипу методом инвариантного преобразования импульсной характеристики и методом билинейного преобразования. Постройте график АЧХ аналогового прототипа и цифровых фильтров, рассчитанных двумя способами.

3. Введите предысказания в частоты аналогового прототипа, после чего вновь перейти к цифровому фильтру по методу билинейного преобразования.

Порядок выполнения:

1. Рассчитайте коэффициенты четырех видов цифровых БИХ-фильтров нижних частот: Баттерворта, Чебышева типа 1 и 2, и эллиптического, порядка $N = 5$ и частотой среза $F = 0.12 + \#/100$, где $\#$ – номер варианта. Используйте процедуры MATLAB: `butter`, `cheby1`, `cheby2`, `ellip`. Постройте в одних осях АЧХ и ФЧХ для четырех видов фильтров. Сравните полученные характеристики. Постройте диаграммы нулей и полюсов для четырех видов фильтров. Сопоставьте расположение нулей и полюсов с видом АЧХ. Обратите внимание на те места, где нули соседствуют с полюсами.

2. Рассчитайте аналоговый полосовой фильтр Баттерворта с частотами среза $F1 = 0.12 + \#/100 - 0.05$ и $F2 = 0.12 + \#/100 + 0.05$, используя процедурой `butter` с параметром 's'. Вместо линейных частот F при этом необходимо подставлять круговые частоты $\Omega = 2\pi F$. Постройте АЧХ аналогового прототипа с помощью процедуры `freqs`, но в зависимости от линейной частоты F . Используя процедуры `bilinear` и `impinvar`, преобразовать аналоговый фильтр в цифровые. Постройте АЧХ этих фильтров на одном графике и сравните между собой и с АЧХ аналогового фильтра.

3. Используя формулы (1), ввести предискажения частот аналогового фильтра после чего повторить п.2.2.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Сравните частотные характеристики фильтра Баттерворта и эллиптического фильтра. В чем их сильные и слабые стороны?

2. Как по виду диаграммы нулей полюсов эллиптического фильтра определить его частоту среза?

3. Каким образом по результатам п.2.1 можно построить схемы цифровых фильтров?

4. Для п. 2.2 опишите характер и причины искажений АЧХ цифровых фильтров в том и другом случае по сравнению с аналоговым прототипом?

ЛАБОРАТОРНАЯ РАБОТА 5

Исследование алгоритмов изменения частоты дискретизации

Цель: приобретение практических навыков применения алгоритмов изменения частоты дискретизации цифровых сигналов.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Алгоритмы изменения частоты дискретизации первоначально применялись для сопряжения цифровых систем с различной тактовой частотой. В дальнейшем они стали использоваться для снижения вычислительных затрат при реализации алгоритмов ЦОС. Процесс понижения частоты дискретизации называют децимацией, а процесс повышения частоты дискретизации – интерполяцией.

Децимация выполняется с помощью операции прореживания: Процесс с пониженной частотой дискретизации (децимированный процесс) формируется путем выборки каждого k -го отсчета из процесса с высокой частотой дискретизации. Коэффициент k называется коэффициентом децимации.

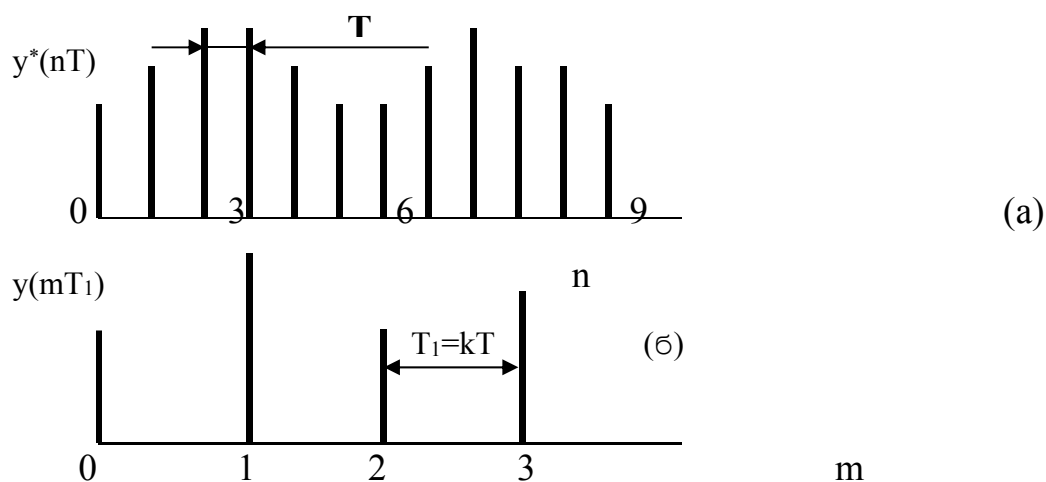


Рис. 7.1

На рис.7.1 показан пример децимации при $k=3$ в результате чего процесс $y^*(nT)$, рис.7.1а, преобразуется к процессу $y(mT_1)$, рис.7.1б. Следует учитывать, что процесс должен соответствовать теореме Котельникова, как до понижения частоты дискретизации, так и после нее. Таким образом, перед выполнением децимации необходимо убедиться в том, что наивысшая частота в спектре процесса не превышает половины новой (пониженной) частоты дискретизации. Если это условие не выполняется, необходимо ограничить спектр процесса с помощью фильтрации. В противном случае децимация приведет к искажению сигнала за счет перекрытия спектров.

При интерполяции прореженные отсчеты должны быть восстановлены. Существует множество способов интерполяции. Наиболее распространены способы повышения частоты дискретизации: за счет добавления нулевых отсчетов; за счет повторения отсчетов.

При первом способе для повышения частоты дискретизации достигается за счет добавления нулевых отсчетов между отсчетами исходного процесса. Например, при интерполяции в три раза между отсчетами исходного процесса добавляется по два нулевых отсчета. Частота дискретизации при этом увеличивается, однако фактическая реализация процесса не меняется. Поэтому и вид спектра процесса с добавленными нулевыми отсчетами остается тем же самым.

На рис. 7.2 показаны спектры процесса при интерполяции в три раза до, и после добавления нулевых отсчетов. $F_{Si} = 3F_S$ – новая частота дискретизации.

Для полного восстановления отсчетов интерполированного процесса необходимо отфильтровать “лишние” спектральные составляющие. АЧХ фильтра показана на рис. 7.2б пунктиром. При этом фильтр, для сохранения амплитуды восстановленного процесса, должен

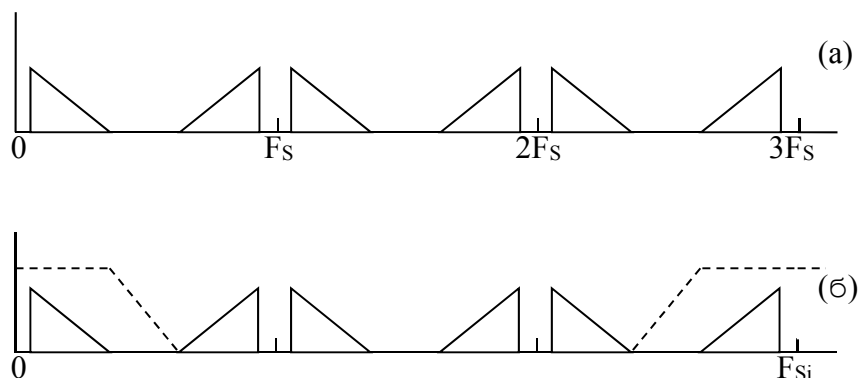


Рис. 7.

иметь коэффициент передачи в полосе пропускания, равный коэффициенту интерполяции. Выполняя указанные действия можно, например, перейти от процесса на рис. 7.1б к процессу на рис. 7.1а.

При втором способе интерполяции вместо добавления нулевых отсчетов повторяют k раз отсчеты исходного процесса. По сути это эквивалентно фильтрации исходного процесса КИХ-фильтром порядка k с единичными коэффициентами. Поэтому при втором способе на спектр рис. 7.2б накладывается АЧХ однородного фильтра вида: $\sin(k\omega t_s/2)/\sin(\omega t_s/2)$. Для полного восстановления формы процесса необходимо, как и в первом случае, отфильтровать ненужные спектральные составляющие. Масштабировать коэффициент передачи фильтра не требуется.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и

выше; Microsoft Office13 и выше, системы компьютерной математики Machcad и MATLAB R2017a и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

1. Сформируйте исходный процесс в виде аддитивной смеси ЧМ-сигнала и тонального сигнала с частотой дискретизации F_s .
2. Постройте графики спектров исходного процесса в реальных и относительных частотах.
3. Выполните децимацию процесса с коэффициентом децимации $k=2$. Постройте график спектра процесса после децимации.
4. Ограничьте спектр исходного процесса с помощью КИХ-фильтра таким образом, чтобы после децимации не было искажений ЧМ-сигнала. АЧХ фильтра постройте на графике спектра исходного процесса в относительных частотах. Вновь выполнить децимацию, постройте спектр и сравните его с предыдущим.
5. Выполните интерполяцию процесса, полученного в п.1.4 методом добавления нулевых отсчетов. Постройте спектр процесса.
6. Синтезируйте фильтр для подавления ненужных спектральных составляющих и постройте его АЧХ на одном графике со спектром.
7. Выполните фильтрацию процесса и постройте график спектра после фильтрации.
8. Выполните интерполяцию процесса методом повторения отсчетов и повторить п.п.1.5-1.8.

Порядок выполнения:

1. Сформировать исходный процесс в виде аддитивной смеси ЧМ-сигнала и тонального сигнала:

$$\text{process} = \sin(w_0 * t + (w_d/w_m) * \sin(w_m * t)) + 2 * \sin(w_i * t),$$

где $w_0 = 2 * \pi * (0.2 + \# * 0.02)$ МГц, $d = 2 * \pi * (0.15 + \# * 0.0025)$ МГц, $w_m = 2 * \pi * 0.005$ МГц, $w_i = 2 * \pi * (F_s/2 - 0.8 * f_0)$ МГц, $F_s = 4 * 1.2288$ МГц - частота дискретизации, # - номер варианта

2. Постройте графики спектров исходного процесса в реальных и относительных частотах.
3. Выполните децимацию процесса с коэффициентом децимации $k=2$, для чего из исходного процесса взять каждый второй отсчет. Постройте график спектра процесса после децимации.
4. Используя спектр исходного процесса, синтезируйте КИХ-фильтр и отфильтруйте частотные составляющие исходного процесса, выходящие за

половину новой частоты дискретизации. АЧХ фильтры постройте на графике спектра исходного процесса в относительных частотах. При этом, для соответствия масштабов АЧХ и спектра следует умножить отсчеты АЧХ на масштабирующий коэффициент. Отсчеты АЧХ рассчитать с помощью оператора `freqz` с параметром `'whole'`. Выполните децимацию фильтрованного процесса и постройте его спектр. Сравните спектр с предыдущим. Порядок фильтра следует выбирать минимальным, добиваясь при этом, чтобы в спектре ЧМ-сигнала после децимации не было видимых искажений и следов тонального сигнала.

5. Проведите интерполяцию процесса, полученного в предыдущем пункте методом добавления нулевых отсчетов с коэффициентом интерполяции $k=2$. Выполните это с помощью оператора цикла, присвоив нечетным отсчетам нулевые значения, а четным – значения отсчетов децимированного процесса. Постройте спектр процесса.

6. По виду полученного спектра синтезируйте КИХ-фильтр, отфильтровав ненужные спектральные составляющие и приблизьте, таким образом, спектр интерполированного процесса к виду исходного ЧМ сигнала. После фильтрации необходимо умножить отсчеты интерполированного процесса на $k=2$, для восстановления амплитуды исходного ЧМ сигнала. АЧХ фильтра постройте на одном графике со спектром из п. 2.5. Для синтеза фильтра применить процедуру `remez` или `firls`, определяя зону подавления в соответствии с частотами ненужных спектральных составляющих. Порядок фильтра должен быть минимальным.

6. Проведите интерполяцию процесса, полученного в п.2.4, методом повторения отсчетов. Как и в п.2.5 можно выполнить это в цикле. Затем повторить действия п.2.6.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Объясните, в каком случае и для чего нужна фильтрация перед выполнением децимации?

2. Какой должна быть частота среза фильтра при переходе от частоты дискретизации $FS1 = \#*100\text{кГц}$ до частоты дискретизации $FS2 = \#*25\text{кГц}$?

3. Докажите с помощью формул для ДПФ что при добавлении нулевых отсчетов при интерполяции форма спектра остается неизменной.

4. В чем заключаются преимущества и недостатки двух рассмотренных способов интерполяции?

ЛАБОРАТОРНАЯ РАБОТА 6

Спектральный анализ и синтез сигналов средствами MATLAB

Цель: практическое изучение методов спектрального анализа и синтеза сигналов средствами MATLAB

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

Основная задача математического моделирования процесса прохождения сигналов через линейные искажающие (или преобразующие) их устройства (фильтры, усилители, каналы связи и т. д.) может решаться различными методами. При спектральном подходе к ее решению она может быть представлена следующей диаграммой:

$$y_i(t) \rightarrow \text{ППФ} \rightarrow A_i(\omega), \varphi_i(\omega) \rightarrow \text{ИУ} \rightarrow A_o(\omega), \varphi_o(\omega) \rightarrow \rightarrow \text{ОПФ} \rightarrow y_o(t) \quad (11.1)$$

Исходный входной сигнал, представленный произвольной временной зависимостью $y_i(t)$ с помощью прямого преобразования Фурье (ППФ, или DFT), преобразуется в свое частотное представление. Если сигнал периодический, то он представляется суммой дискретных гармоник с амплитудами $A_i(\omega)$ и фазами $\varphi_i(\omega)$ где i - номер гармоники от 0 (постоянная составляющая) до N , и $\omega=2\pi f$ - круговая частота (f - частота в герцах). Первая гармоника имеет частоту повторения сигнала ω_1 , а высшие - частоты, кратные основной гармонике $i\omega_1$. Для непериодических сигналов спектр будет сплошным, понятие гармоник уже не существует, но можно говорить о плотности мощности сигнала в определенной полосе частот.

В общем виде ППФ записывается следующим образом:

$$A(\omega) = \int_{-\infty}^{\infty} y(t)e^{-i\omega t} dt \quad (11.2)$$

В результате ППФ мы получаем комплексную амплитуду на заданной круговой частоте ω . После тривиальных преобразований можно получить

модуль $A(\omega)$ - частотный спектр сигнала $A_i(\omega)$ и фазу $\varphi_i(\omega)$ - фазовый спектр сигнала.

Моделирование сигналов – одна из важнейших задач пакета Signal Processing Toolbox. Многие сигналы являются детерминированными, то есть их временная зависимость $y(t)$ аналитически определена и для любого момента времени t при неоднократном моделировании мы будем получать одни и те же значения $y(t)$. Детерминированные сигналы удобны для аналитического описания и анализа систем обработки и преобразования сигналов и широко используются в роли тестовых сигналов.

Однако на практике большинство реальных сигналов не являются детерминированными. Одной из причин этого является засоренность сигналов шумами и помехами, имеющими случайный характер с тем или иным законом распределения вероятности.

Моделирование зашумленных сигналов в пакете SignalProcessingToolbox обеспечивается с помощью генератора случайных чисел. Разумным компромиссом между детерминированными и недетерминированными сигналами являются детерминированные сигналы с повторяющейся при каждом запуске моделирования шумовой компонентой. Это позволяет обеспечить, с одной стороны, учет влияния шума, а с другой — обеспечить повторяющиеся в процессе многократного моделирования результаты.

Спектральный анализ заключается в определении частотного состава сигнала и оценивании его спектральных характеристик, являющихся функциями частоты.

С помощью спектрального анализа решаются задачи обнаружения, разрешения и оценки параметров сигналов, сжатия данных, идентификации объектов, распознавания образов – речи, изображений и т.д. Спектральный анализ случайных сигналов направлен на выявление скрытых периодичностей и статистических (корреляционных) связей.

Спектральный анализ сигналов в пакете Signal Processing Toolbox реализуется большим числом методов. Все они вычисляют спектральную плотность мощности (СПМ) сигналов и имеют одинаковые (как по смыслу, так и по обозначению) параметры.

Названия функций начинаются с буквы **p** (от слова «power» - мощность).

Входные параметры функций спектрального анализа следующие:

x– вектор значений сигнала;

p– порядок модели;

nfft– число отсчетов сигнала, используемого при быстром преобразовании Фурье – БПФ (минимальное значение по умолчанию равно 256);

f– частота (в Гц);

w– круговая частота;

fs– частота дискретизации;

'range'– параметр указания диапазона частот;

'squared'– параметр, отменяющий вывод графика СПМ в децибелах и дающий график в прямоугольной системе координат.

Некоторые входные параметры могут отсутствовать. Их следует задавать как пустую матрицу [], что служит указанием для задания соответствующего параметра по умолчанию.

Выходные параметры функций этого раздела следующие:

freq– вектор частот, для которых производится оценка СПМ;

Pxx– вектор-столбец оценки СПМ, размер которого при четном **nfft** равен $nfft/2+1$, а при нечетном $(nfft+1)/2$. Если **x** содержит комплексные данные, то СПМ оценивается для всех частот, и число элементов вектора **Pxx** равно **nfft**.

Все описанные ниже функции имеют внутреннее обращение к функции **plot(f, Pxx)** и строят графики зависимости СПМ от частоты, используя векторы частот **f** и СПМ на них **Pxx**.

В пакете Signal Processing Toolbox существуют следующие методы спектрального анализа: метод Бурга (pburg), ковариационный метод (pcov), модифицированный ковариационный метод (pmscov), многооконный метод (pmtm), метод Уэлча (pwelch) и метод периодограмм (periodogram)[1].

Таблица 11.1. Сравнение основных методов спектрального анализа

Метод	Функция	Достоинства	Недостатки
Периодограмм, непараметрический	periodogram	Хорошее разрешение по частоте	Изрезанность при длинных сигналах. Флюктуации при большом числе отсчетов
Уэлча, непараметрический	Pwelch	Хорошее сглаживание сигналов	Возможная потеря острых пиков спектра
Бурга, на основе авторегрессии	Pburg	Высокое разрешение и стабильность при анализе коротких сигналов. Хорошее устранение шума. Использование окон (весовых функций) и частично перекрывающихся сегментов сигнала. Дисперсия оценки спектра мощности меньше, чем при методе Бартлетта	Начальные фазы синусоид сильно влияют на положение спектральных пиков; появляется смещение спектральных пиков при анализе суммы синусоид с шумом; при большом порядке модели может наблюдаться расщепление спектральных пиков
Ковариационный	Pcov	Высокая разрешающая способность при спектральной оценке коротких сигналов. Минимизация погрешностей прямого предсказания. Хорошая оценка сигналов в виде суммы чистых синусоид	Рассчитанный формирующий фильтр может оказаться нестабильным. При анализе суммы синусоид с шумом возможно смещение спектральных пиков
Модифицированный ковариационный	Pmscov	Высокое разрешение при анализе коротких сигналов, отсутствие расщепления пиков	Зависимость положения спектральных линий от начальных фаз синусоид
Многооконный, на основе последовательностей Слепиана	pmtm	Высокое разрешение	Работа только с вещественными данными

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и

выше; Microsoft Office13 и выше, системы компьютерной математики Machcad и MATLAB R2017a и выше.

Указания по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 1. Вычислим СПМ методом Бурга.

Вычисление реализовано следующими вариантами записи функции pburg:

$P_{xx} = \text{pburg}(x, p)$

$[P_{xx}, w] = \text{pburg}(x, p)$

$[P_{xx}, w] = \text{pburg}(x, p, \text{nfft})$

$[P_{xx}, f] = \text{pburg}(x, p, \text{nfft}, fs)$

$[P_{xx}, f] = \text{pburg}(x, p, \text{nfft}, fs, 'range')$

$[P_{xx}, w] = \text{pburg}(x, p, \text{nfft}, 'range')$

Если в качестве выходного параметра указан только вектор P_{xx} , то вычисляется только СПМ. При указании в качестве выходных параметров списка $[P_{xx}, w]$ или $[P_{xx}, f]$ дополнительно вычисляются векторы частот — обычных f (в Гц) или круговых w , на которых выполнено вычисление P_{xx} . По умолчанию могут задаваться параметры $\text{nfft} = 256$, $f = 1$ Гц.

При использовании параметра `range` рекомендуется уточнять диапазон частот — f или w — по справке заданной функции. Обычно он выбирается разным для вещественных и комплексных данных и зависит от выбора частоты дискретизации fs .

Для иллюстрации оценки СКМ методом Бурга воспользуемся наглядным примером из справки по этой функции. Вначале построим АЧХ и ФЧХ автокорреляционной AR-системы:

```

a=[1 -2.2137 2.9403 -2.1697 0.9606]; %Коэффициенты AR-фильтра
freqz(1,a) % АЧХ и ФЧХ AR-фильтра
title('АЧХ AR-системы')

```

Их графики представлены на рисунке 1.

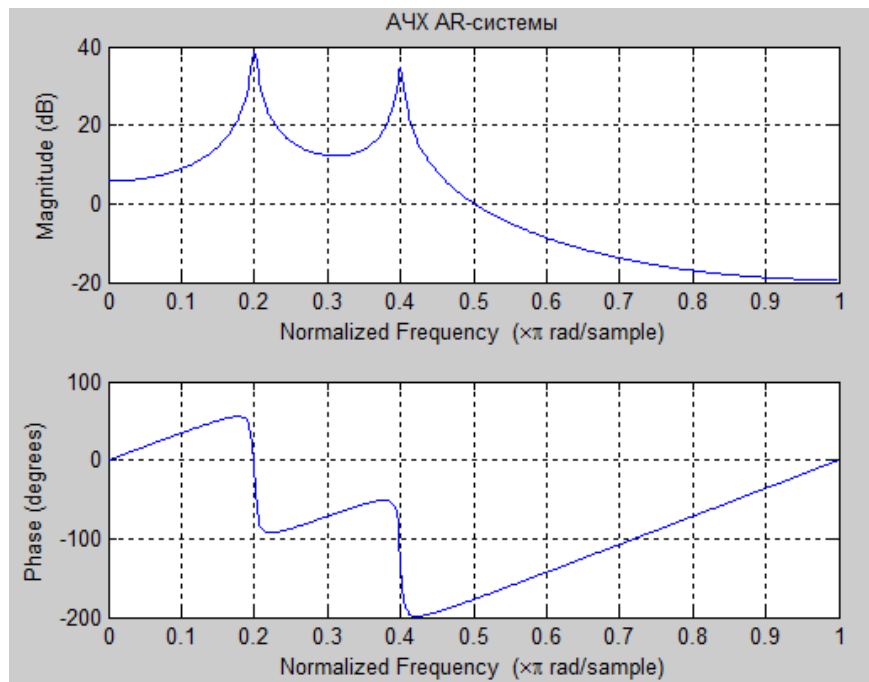


Рисунок 1. Графики АЧХ и ФЧХ AR-фильтра

Теперь построим АЧХ для сигнала из белого шума, используя AR-систему 4-го порядка:

```

randn('state',1);
x=filter(1,a,randn(256,1)); % Выход AR-системы
rburg(x,4) % Оценка СКМ 4-го порядка

```

Построенный график АЧХ для этого случая показан на рис. 11.2.

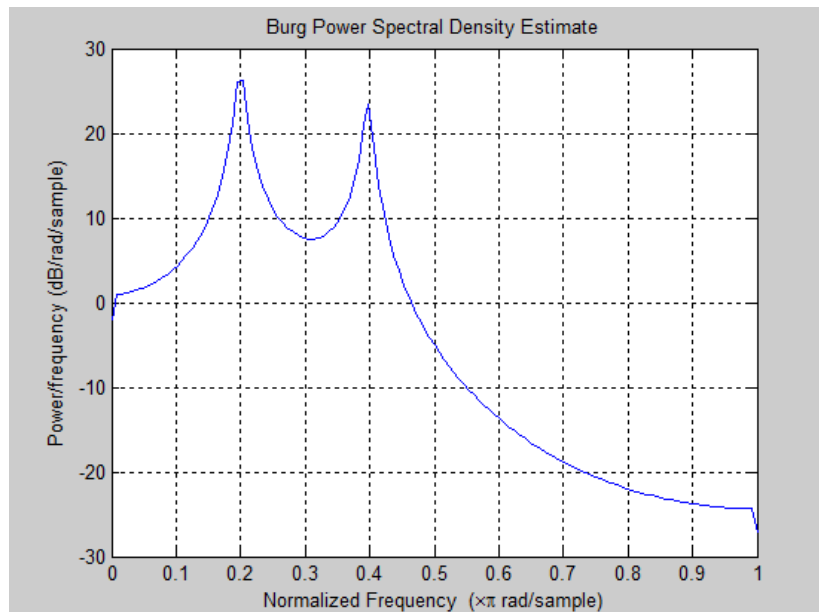


Рисунок 2. АЧХ белого шума, построенная с помощью функции `rburg`

Задание 2. Вычислим СПМ модифицированным ковариационным методом.

Ковариационный метод реализует функция `pcov`, имеющая следующие варианты записи:

`Pxx=pcov(x,p)`

`[Pxx,w]=pcov(x,p)`

`[Pxx,w]=pcov(x,p,nfft)`

`[Pxx,f]=pcov(x,p,nfft,fs)`

`[Pxx,f]=pcov(x,p,nfft,fs,'range')` `[Pxx,w]=pcov(x,p,nfft,'range')`

Назначение параметров этой функции уже описано. С ее работой можно ознакомиться, выполнив приведенный ниже пример:

```
a=[1 -2.2137 2.9403 -2.1697 0.9606];
freqz(1,a)
title('AR System Frequency Response')
randn('state',1);
x=filter(1,a,randn(256,1));
pcov(x,4)
```

Проверьте насколько графики, построенные при выполнении этого примера, отличаются от графиков предыдущего примера.

Задание 3. Вычислим СПМ ковариационным методом.

Модифицированный ковариационный метод реализован функцией `pmcov`:

```
Pxx=pmcov(x,p) [xx,w]=pmcov(x,p)
```

```
[Pxx,w]=pmcov(x,p,nfft)
```

```
[Pxx,f]=pmcov(x,p,nfft,fs)
```

```
[Pxx,f]=pmcov(x,p,nfft,fs,'range')
```

```
Pxx,w]=pmcov(x,p,nfft,'range')
```

Параметры и техника применения этой функции подобны уже описанным. Аналогично предыдущим примерам постройте график и сравните с уже построенными.

Задание 4. Вычислим СПМ многооконным методом.

Для реализации многооконного метода служит функция `pmtm`.

Варианты ее записи представлены ниже:

```
[Pxx,w]=pmtm(x,nw)
```

```
[Pxx,w]=pmtm(x,nw,nfft)
```

```
[Pxx,f]=pmtm(x,nw,nfft,fs)
```

```
[Pxx,Pxxc,f]=pmtm(x,nw,nfft,fs)
```

```
[Pxx,Pxxc,f]=pmtm(x,nw,nfft,fs,p)
```

```
[Pxx,Pxxc,f]=pmtm(x,e,v,nfft,fs,p)
```

```
[Pxx,Pxxc,f]=pmtm(x,dpss_params,nfft,fs,p)
```

```
[..]=pmtm(... 'method')
```

```
[..]=pmtm(... 'range')
```

Эта функция работает с вещественными данными и использует совокупность ортогональных окон. В качестве их используются дискретные сфероидальные последовательности Слепиана. Параметр `nw` задает спектрально-временное разрешение. По умолчанию `nw = 4`, рекомендуется

выбирать его значения равными 2, 5/2, 3, 7/2. Число последовательностей Слепиана равно $2 \cdot n_w - 1$.

Функция имеет параметр `method`, позволяющий задать метод вычисления СПМ:

adapt– адаптивный нелинейный алгоритм Томсона комбинации индивидуальных оценок (используется по умолчанию);

unity– линейная комбинация индивидуальных оценок с весами, равными 1;

eigen– линейная комбинация индивидуальных оценок с весами, задаваемыми собственными значениями.

Ниже представлен пример оценки СПМ синусоиды с частотой 250 Гц, на которую наложен небольшой шум.

```
randn('state', 0); fs=1000; t=0:1/fs:0.3;
x=cos(2*pi*t*250)+0.05*randn(size(t));
[Pxx, Pxxc, f]=pmtm( x, 3.5, 512, fs, 0.99);
hpsd = dspdata.psd([Pxx Pxxc], 'Fs', fs);
plot(hpsd)
```

Этот пример создает графики, представленные на рисунке3.

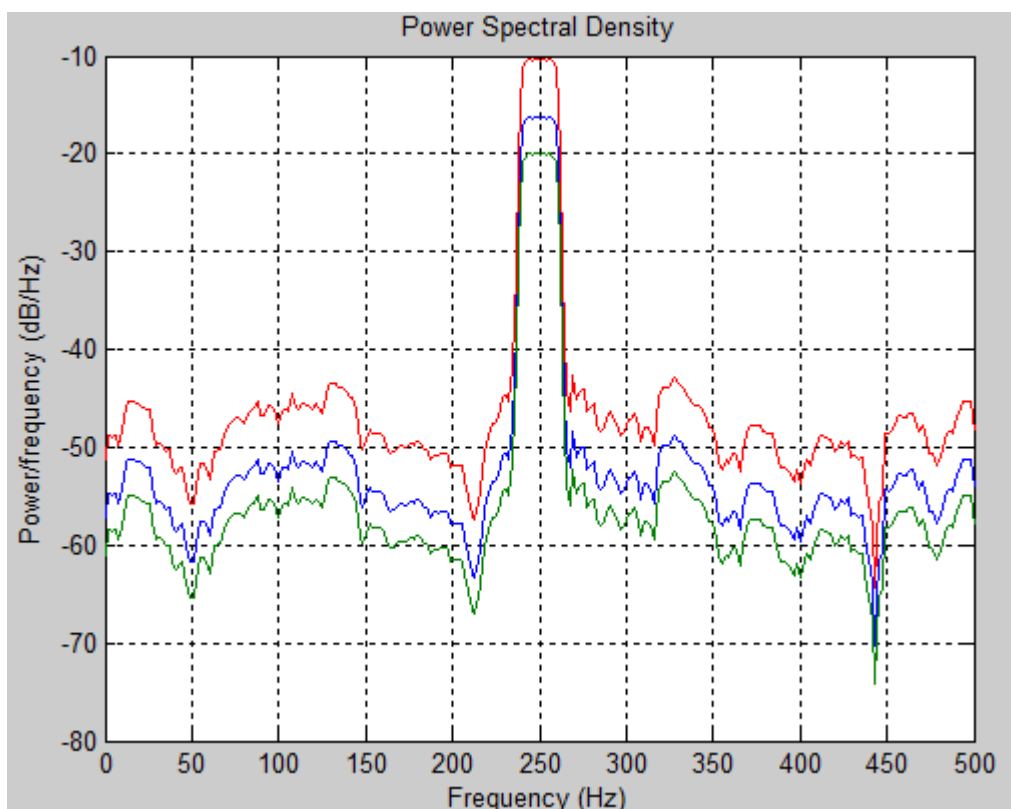


Рисунок 3. Пример оценки СПМ с помощью функции `pmtm`

Задание 5. Вычислим СПМ методом Уэлча.

Метод Уэлча вычисления СКМ реализуется следующей функцией:

```
[Pxx,w]=pwelch(x) [Pxx,w]=pwelch(x,nwin)
```

```
[Pxx,w]=pwelch(x,nwin, noverlap)
```

```
[Pxx,w]=pwelch(x,nwin,noverlap,nfft)
```

```
[Pxx,f]=pwelch(x,nwin,noverlap,nfft,fs)
```

```
[..]=pwelch(x, nwin, noverlap,...,'range')
```

Здесь целочисленное положительное значение параметра `nwin` задает длину окна Хэмминга. Если `nwin` задан как двухэлементный вектор, он задает размеры прямоугольного окна. Остальные параметры этой функции подобны уже описанным. Для расширения знакомства с этой функцией используйте ее демонстрационный пример:

```
randn('state',0); Fs=1000; t=0:1/Fs:.3;
x=cos(2*pi*t*200)+randn(size(t));
pwelch(x,33,32,[],Fs,'twosided')
```

Посмотрите получаемый при этом график (рисунок 4) и попробуйте модифицировать этот пример. Описание алгоритма реализации этого метода можно найти в справочной системе.

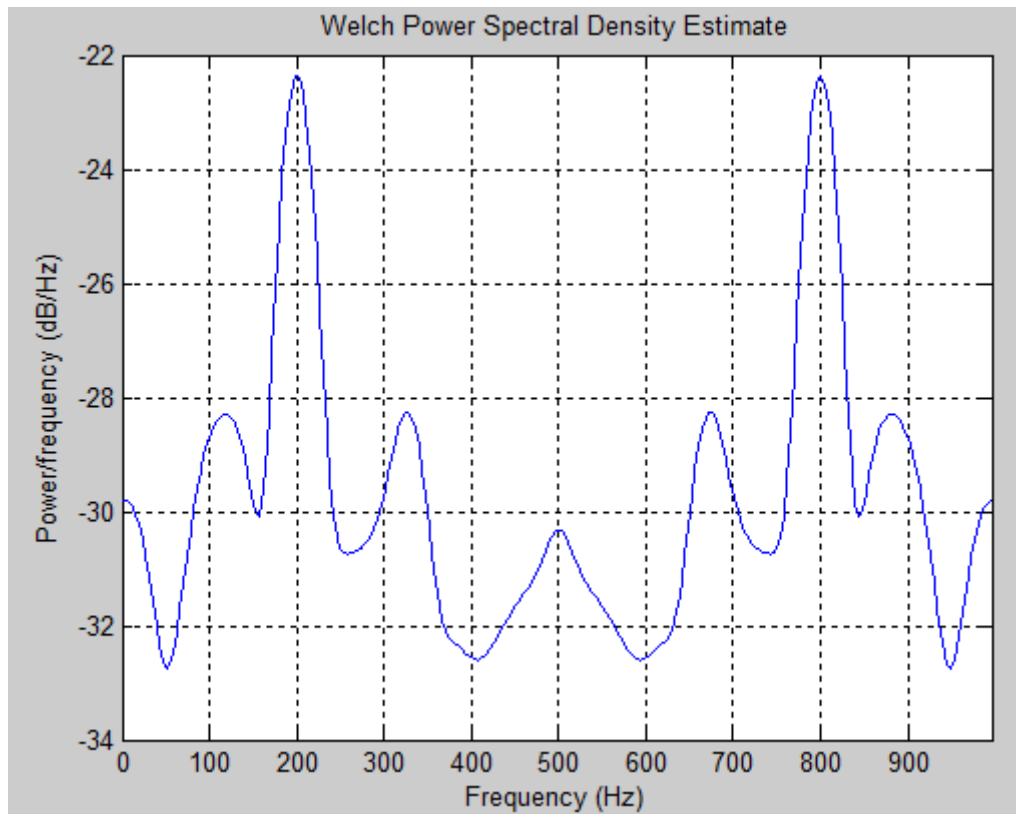


Рисунок 4. Пример оценки СПМ с помощью функции pwelch

Задание 6 Вычислим СПМ методом периодограмм.

Для построения периодограмм как зависимостей спектральной мощности в дБ/Вт от частоты служит функция periodogram:

`[Pxx,w]=periodogram(x)`

`[Pxx,w]=periodogram(x>window)`

`[Pxx,w]=periodogram(x>window,nfft)`

`[Pxx,f]=periodogram(x>window,nfft,fs)`

`[Pxx,...]=periodogram(x,...,'range')`

Частота может задаваться как угловая w или в герцах f . Параметр `range` может иметь два значения:

twosided – вычисляет двухполосную спектральную мощность в частотном диапазоне $[0, f_s)$. При задании вместо этой опции пустого вектора `[]` частотный диапазон определяется как $[0, 1)$. Если не используется спецификация f , то этот диапазон выбирается равным $[0, 2)$.

onesided – вычисляется однополосная спектральная мощность в диапазоне частот, заданном для вещественных компонент вектора x . Для x с вещественными компонентами эта опция используется по умолчанию.

Для построения периодограмм используется функция `periodogram`, что иллюстрирует следующий пример:

```
Fs=1000; t=0:1/Fs:.3;
x=sin(2*pi*t*200)+cos(2*pi*t*350)+0.01*randn(size(t));
periodogram(x,[],'onesided',512,Fs)
```

В результате получим график (рисунок 5).

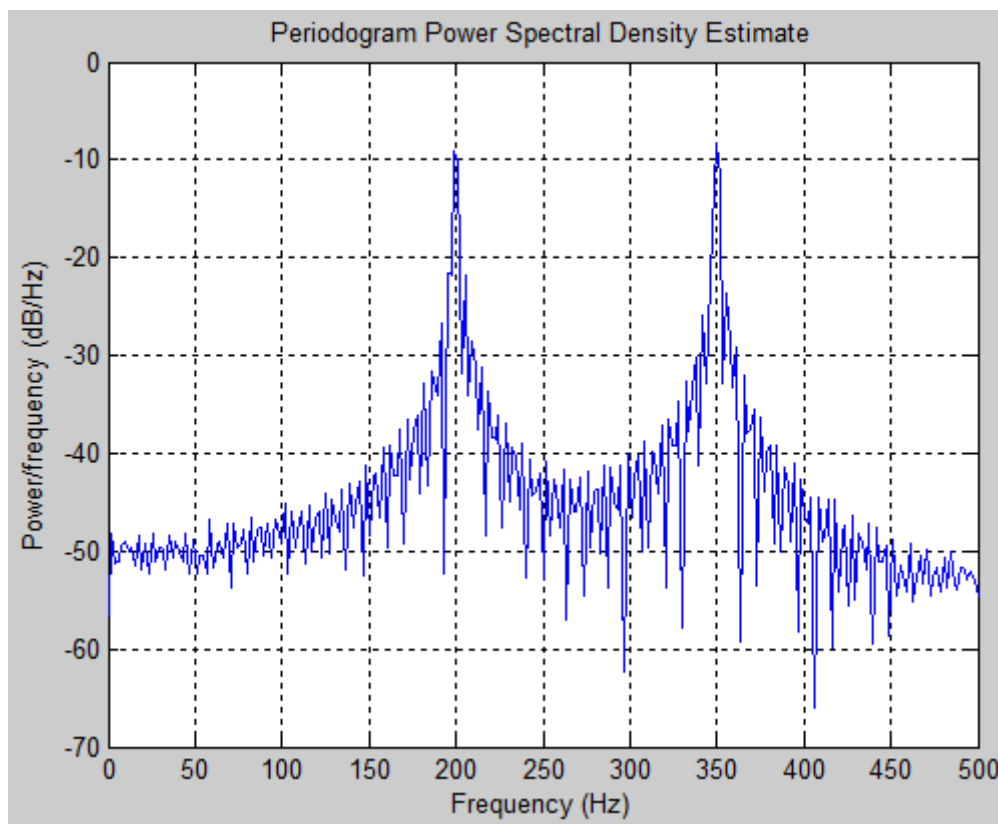


Рисунок 5. Пример оценки СПМ с помощью функции `periodogram`

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) условия выполненных заданий и их решение;
- 4) технологию решения задач с помощью прикладного программного обеспечения, выводы;

5) ответы на контрольные вопросы.

Вопросы для защиты работы

1. Дайте содержательную характеристику спектральному анализу
2. Какой пакет MATLAB содержит методы спектрального анализа сигналов?
3. Какие основные методы представлены в этом пакете?
4. Дайте краткую характеристику каждому из представленных методов.
5. Опишите достоинства и недостатки представленных методов.

ЛАБОРАТОРНАЯ РАБОТА 7

Вейвлет-анализ и синтез сигналов в MATLAB

Цель: получить основные навыки применения вейвлет-преобразования сигналов. Освоение GUI Wavemenu MATLAB.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Вейвлеты и их применение – новейшее научное направление, возникшее на стыке математики, информатики и техники связи. Технология вейвлетов базируется на обобщенном представлении сигналов $s(t)$ в векторном пространстве в виде взвешенной суммы базисных функций $\Psi_k(t)$, помноженных на коэффициенты C_k :

$$s(t) = \sum_k C_k \Psi_k(t) \quad (1)$$

Термин wavelet, введенный впервые Морле (J. Morlet), означает «короткая или маленькая волна».

Под вейвлет-анализом будем понимать декомпозицию сигнала при помощи прямого вейвлет-преобразования. Прямым вейвлет-преобразованием (ContinuousWaveletTransform — CWT) называют вычисление вейвлет-коэффициентов в области определения R:

$$C(a, b) = \int_{-\infty}^{\infty} s(t) a^{-\frac{1}{2}} \Psi\left(\frac{t-b}{a}\right) dt \quad (2)$$

Интегральное преобразование (2) называют непрерывным, если параметры масштаба a и сдвига b , в ходе обработки сигнала, принимают любые действительные значения. Оно избыточно по затратам времени на вычисления, но зато позволяет добиться наибольшей детализации при анализе сигналов.

Прямое непрерывное вейвлет-преобразование лежит в основе построения вейвлет-спектрограмм сигналов. В простейшем случае вейвлет-спектрограмма отражает уровень вейвлет-коэффициентов (градиациям и яркости) в плоскости масштаб-время. На рисунке 1 показан пример построения нестационарного сигнала (синусоида, на которую наложен прямоугольный импульс с изменяющимся периодом повторения). Перепады сигнала на спектрограмме, полученной в MATLAB, представлены в виде «зубцов». Уменьшение амплитуды перепадов в сотни-тысячи раз (при этом они сливаются с синусоидой) перепады тем не менее, отчетливо выявляются спектрограммой (подокно снизу). Способность вейвлет-спектрограмм обнаруживать артефакты сигналов не имеет прецедентов в технике спектрального анализа. Это полезно в сейсмографии, медицине и других областях применения вейвлет-анализа.

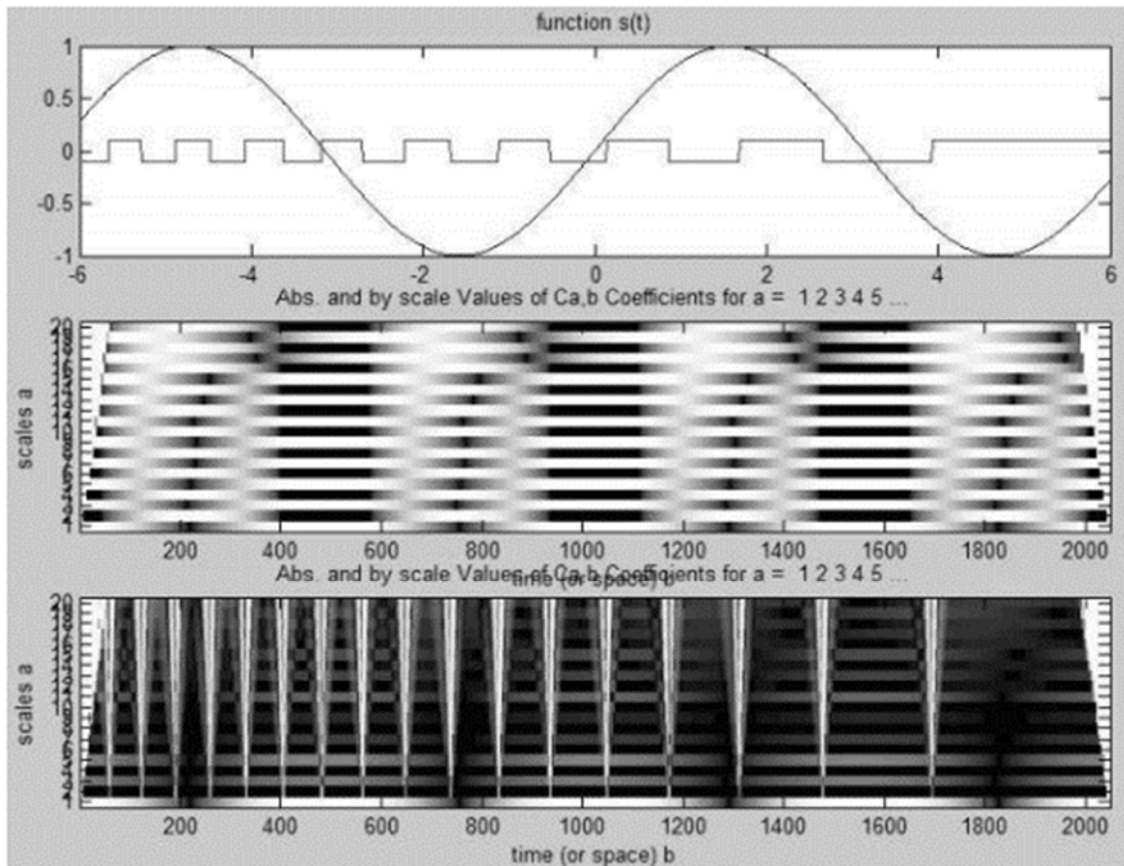


Рисунок 1. Вейвлет – спектрограмма сигнала синусоиды, на которую наложен прямоугольный импульс с изменяющимся периодом повторения

Пакет расширения WaveletToolbox системы MATLAB является одним из самых мощных средств для работы с вейвлетами. Эта работа возможна как с помощью окон графического интерфейса пользователя GUI (пример рисунок 2), так и из окна командного режима. Будучи установленный (вместе с базовой системой MATLAB) пакет может использоваться для обработки осциллограмм и спектров от современных цифровых измерительных приборов, расширяя области их применения.

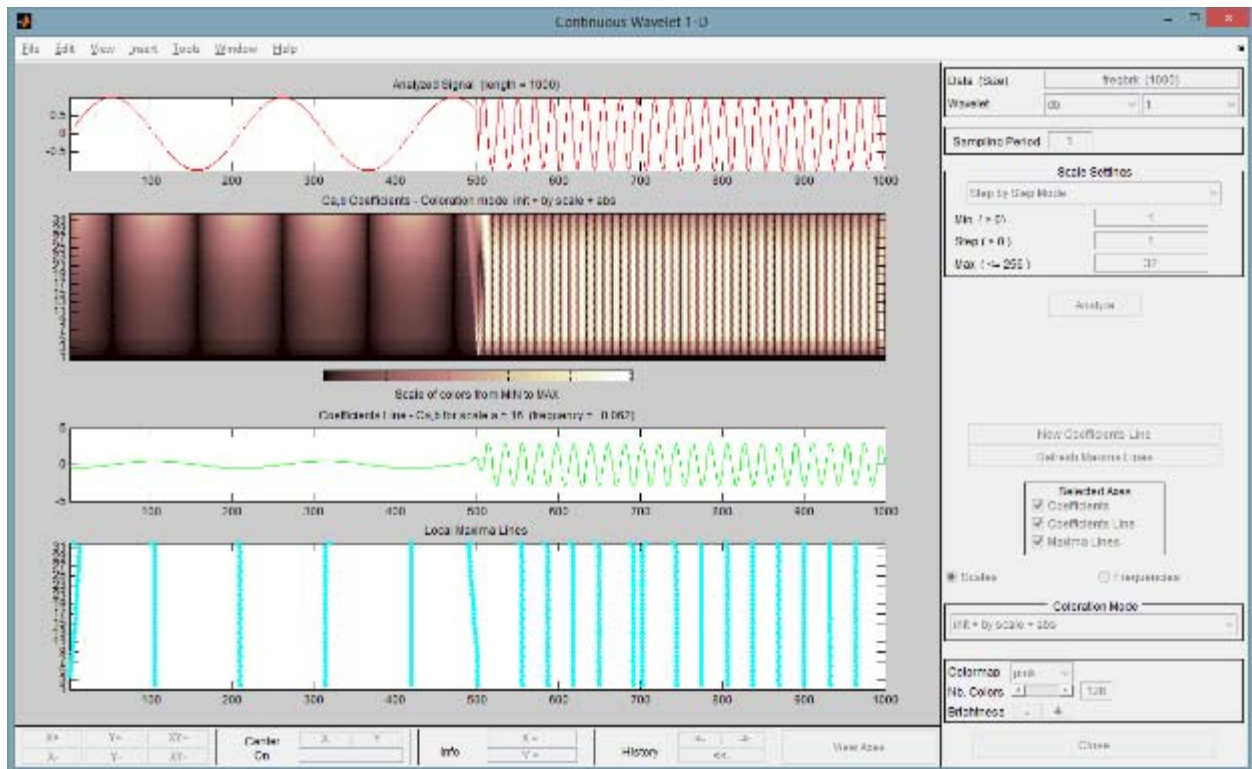


Рисунок 2. Вейвлет-спектрограмма двухчастного сигнала, полученная в окне GUI системы MATLAB

Вейвлет-анализ в системе MATLAB реализуется CWT функцией $\text{COEF} = \text{cwt}(\text{ydata}, \text{start}:\text{step}:\text{end}, 'W', 'opt')$, где ydata — вектор сигнала, start и end — начальный и конечный уровни разложения, W — тип вейвлета и opt — опция, задающая тип окраски вейвлет-спектрограммы, которую строит эта функция. Функция возвращает вектор вейвлет-коэффициентов COEF .

Вейвлет-синтез сигнала заключается в изменении вейвлет-коэффициентов исходного сигнала с целью в результате выполнения обратного вейвлет-преобразования получить сигнал с новыми свойствами. В зависимости от способа модификации коэффициентов достигаются различные цели вейвлет-синтеза, например, фильтрация или сжатие.

Обратное непрерывное вейвлет-преобразование осуществляется по формуле реконструкции во временной области:

$$f = C_{\psi}^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{dad b}{a^2} \langle f, \Psi^{a,b} \rangle \Psi^{a,b} \quad (3)$$

В пакете расширения MATLAB — WaveletToolbox используется следующая формула реконструкции сигнала:

$$s(t) = \frac{1}{K_{\psi}} \int_{R+} \int_{R+} C(a, b) a^{-\frac{1}{2}} \Psi\left(\frac{t-b}{a}\right) \frac{dad b}{a^2}, \quad (4)$$

где K_{ψ} — константа, определяемая функцией ψ .

Если a и b задавать дискретно, что обычно и делается, то можно реализовать дискретное кратно масштабное вейвлет-преобразование. Чаще всего кратность берется равной 2^L , что соответствует диадическому вейвлет-

преобразованию, где целое L есть уровень преобразования, задающий разрешение.

Средства вейвлет-преобразований пока не используются в серийных цифровых осциллографах и анализаторах спектров. Но они уже представлены в ряде пакетов расширений по вейвлетам СКМ Mathcad, Mathematica и MATLAB. При установке этих систем в приборы открытой архитектуры эти средства уже можно полноценно использовать в практике исследований и измерений. Это возможно так же при стыковке измерительных приборов с закрытой архитектурой с ПК, на которых установлены СКМ. Вейвлет-преобразования наиболее эффективны при решении следующих наиболее типовых задач:

- построение вейвлет-спектрограмм и анализ тонких особенностей сигналов;
- фильтрация сигнала путем ограничения числа используемых вейвлет-коэффициентов;
- очистка и сглаживание сигналов (в том числе нестационарных) и изображений от шума и помех;
- компрессия (сжатие) сигналов с малыми их искажениями;
- интерполяция, экстраполяция и аппроксимация сигналов [4];

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

1. Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Пакет расширения WaveletToolbox системы MATLAB позволяет использовать вейвлет анализ и преобразование данных в самых различных областях науки и техники.

Программное обеспечение пакета позволяет выполнять вейвлет-преобразования как в командном режиме (и готовить специализированные программы), так и в диалоговом режиме по интерфейсу GUI, который включается командой «wavemenu» (рисунок 3).

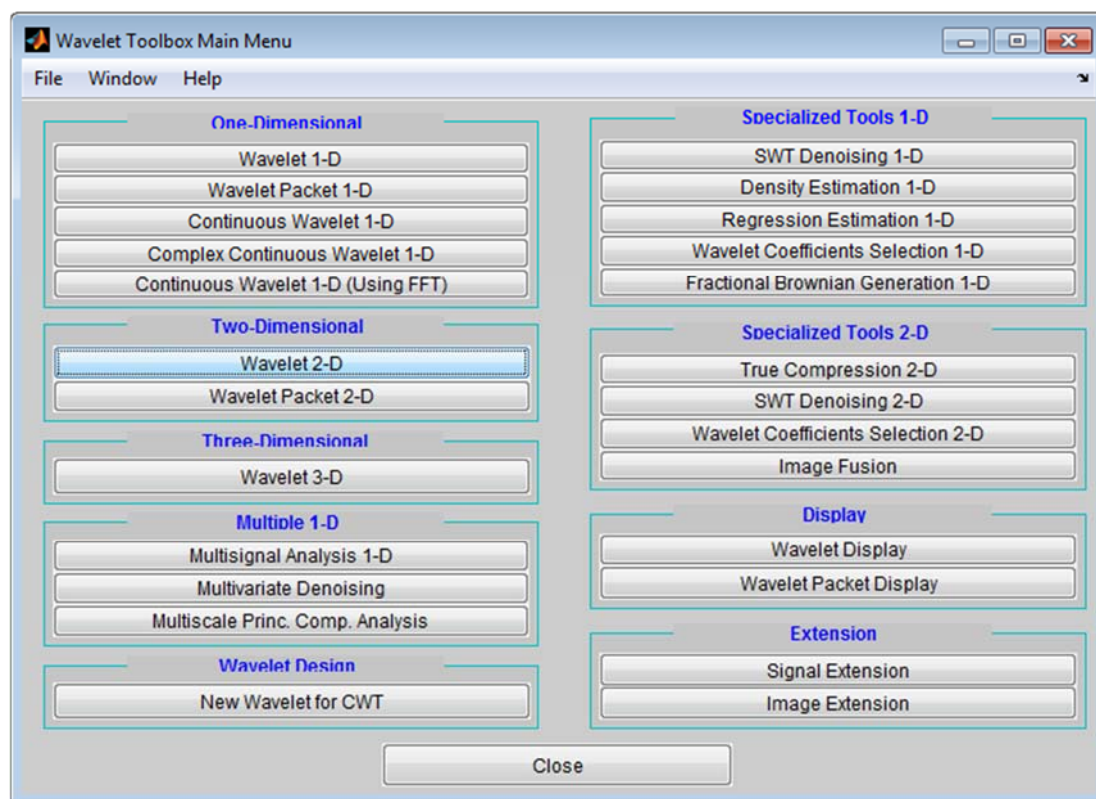


Рисунок 3. Интерфейс WaveletToolbox

Пакет имеет демонстрационные примеры вейвлет-преобразований, окно которых включаются командой «wavedemo» (рисунок 4).

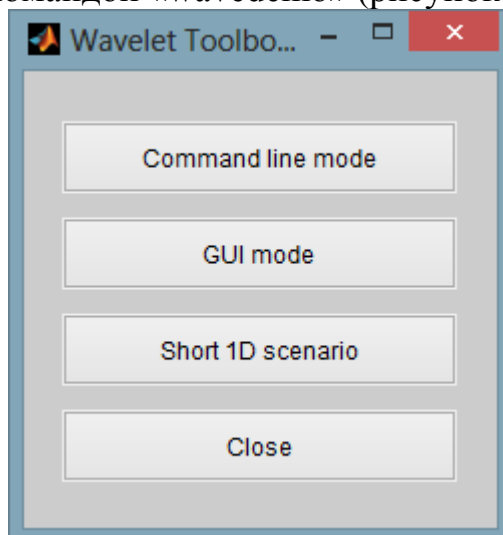


Рисунок 4. Интерфейс WaveletToolboxDemo

Первой кнопкой окна (Commandlinemode) включается довольно обширное меню примеров работы в командном режиме с одно- и двумерными вейвлетами всех типов в обычном и в пакетном исполнении (непрерывные и дискретные вейвлет-преобразования с декомпозицией и реконструкцией сигналов, сжатие сигналов, очистка от шумов и пр.). Слайды примеров сопровождаются соответствующими листингами программных фрагментов.

Аналогично второй и третьей кнопками включается доступ к демонстрационным примерам в интерфейсе GUI.

При изучении настоящей темы рекомендуется самостоятельно просматривать соответствующие демонстрационные примеры по ходу изложения материала.

Задание 1. Произвести анализ спектра непрерывного вейвлет-преобразования сигнала, параметры и аналитическое выражение которого, приведены ниже:

$$s(t) = \sin(2\pi \cdot 10^4 \cdot t + 45).$$

Построить вид рассматриваемого сигнала во временной области, спектр его непрерывного вейвлет-преобразования, тип вейвлет-функции, применяемой для анализа.

Алгоритм вычисления непрерывного вейвлет-преобразования сигнала является реализованным в пакете MATLAB с использованием функции `cwt`, листинг которой приведен ниже.

`COEFS = cwt(S, SCALES, 'wname', PLOTMODE, XLIM)`

```
% функция для вычисления непрерывного вейвлет-преобразования
% COEFS -- коэффициенты непрерывного вейвлет-преобразования
% S -- анализируемый сигнал
% SCALES -- параметр, изменяющий временной масштаб вейвлет-функции
% 'wname' -- тип вейвлет-функции, применяемый
% для вычисления спектра непрерывного вейвлет-преобразования
% PLOTMODE -- параметр, изменяющий настройки цвета, для отображения набора
% вейвлет-коэффициентов. PLOTMODE может принимать следующие значения:
% 'lvl' -- окраска шаг за шагом
% 'glb' -- окраска с учетом всех коэффициентов
% 'abslvl' или 'lvlabs' -- окраска шаг за шагом с использованием
% абсолютных значений вейвлет-коэффициентов
% 'absglb' или 'glbabs' -- окраска с масштабированием и использованием
% абсолютных значений вейвлет-коэффициентов
% Использование в настройках PLOTMODE значений
% '3Dlvl', '3Dglb', '3Dabslvl' ('3Dlvlabs'),
% '3Dabsglb' ('3Dglbabs'), взамен 'lvl', 'glb',
% 'abslvl' ('lvlabs'), 'absglb' ('glbabs'),
% соответственно, позволяет отображать спектр вейвлет-коэффициентов
% в виде поверхности взамен плоскости
% XLIM - параметр, изменяющий настройки цвета, для отображения набора
% вейвлет-коэффициентов.
% Представляет собой вектор [x1 x2], 1<=x1<x2<=length
% Программа для вычисления и построения спектра
% непрерывного вейвлет-преобразования
```

```

t = 0:0.000001:0.0004;
A = 1; F = 10000; phi = 45;
signal = A*sin(2*pi*F*t + phi);
subplot(211); plot(t,signal);
axis([0 0.0004 -1 1]);
title('Исходный сигнал'); grid
xlabel('Время, t'); ylabel('Значения сигнала')
subplot(212);
c = cwt(signal,1:1:16,'mexh','3Dglbabs',[0 15]);
title('Вейвлет-спектр');
xlabel('Временной сдвиг, b');
ylabel('Временной масштаб, a')
lb = -5; ub = 5; n = 1000;
[psi,x] = gauwavf(lb,ub,n,2);
figure; plot(x,psi);
title('Гауссов вейвлет'); grid
xlabel('t'); ylabel('\psi(t)')

```

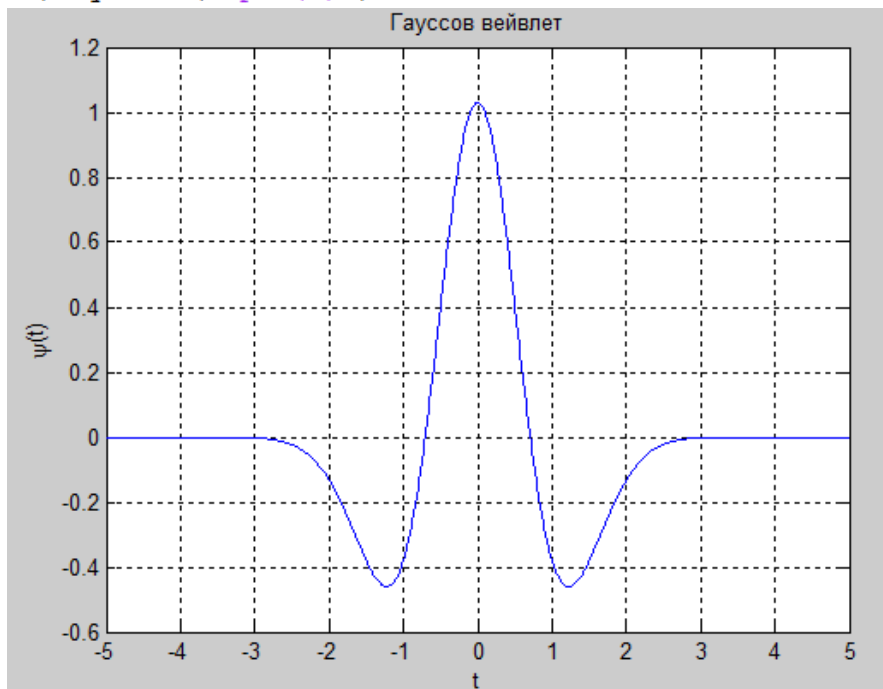


Рисунок 5 – Пример вейвлет-функции используемой для построения спектра приведенного на рисунке 6

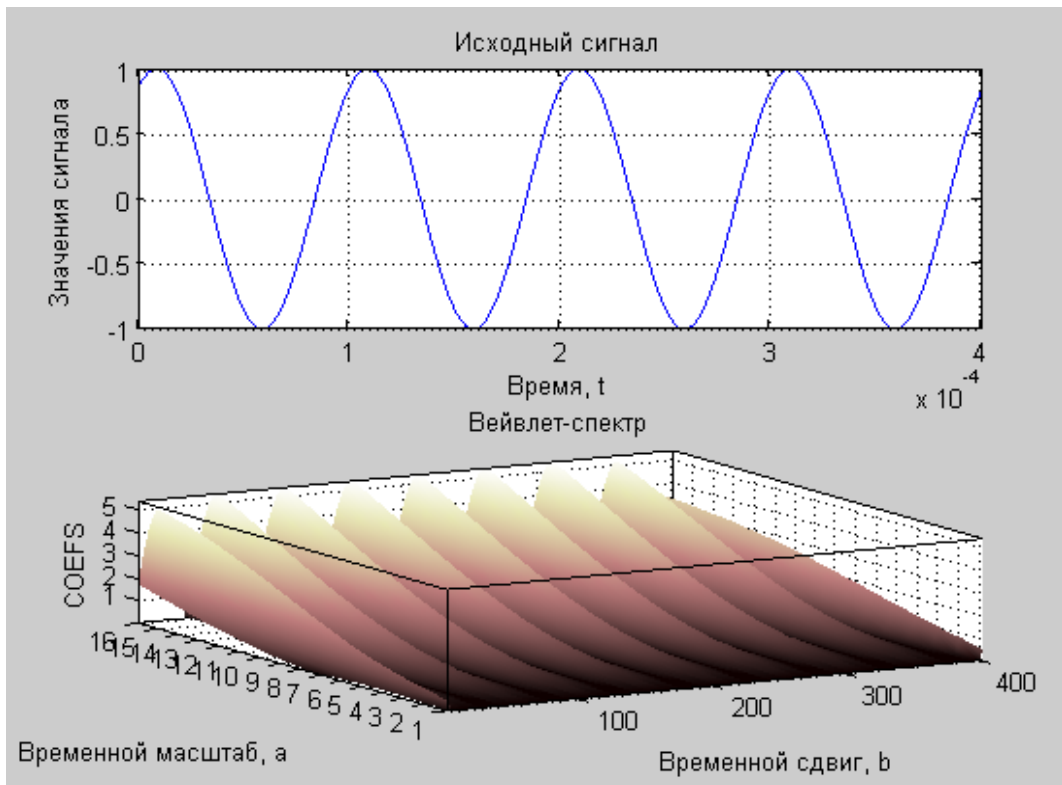


Рисунок 6 – Пример анализируемого сигнала и его непрерывного вейвлет-спектра

Задание 2. Выполнить реконструкцию сигнала используя коэффициенты детализации.

```
% Исходный сигнал
x = 0: pi/10: 10*pi;
l = length (x);
s = sin (x) + rand (1, l);
% Выбор вейвлета
wname = 'coif2';
% Дискретное преобразование
[sa, sd] = dwt (s, wname, 'mode', 'ppd');

% Графическая поддержка разложения
subplot (411)
plot (s), title ('Исходный сигнал')
subplot (412)
plot (sa), title ('Коэффициенты аппроксимации')
```

```

subplot (413)
plot (sd), title ('Коэффициенты детализации')
% Восстановление сигнала
s1 = idwt (sa, sd, wname, 'mode', 'ppd');
% Рисование восстановленного сигнала
subplot (414)
plot (s1), title ('Восстановленный сигнал')

```

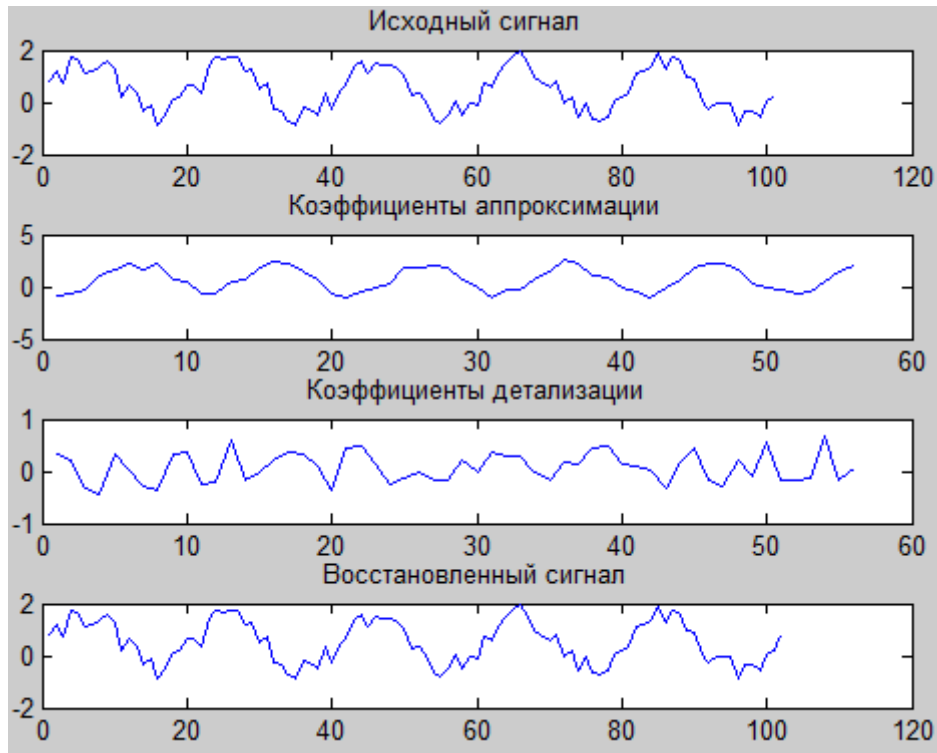


Рисунок 7 – Восстановление сигнала с использованием коэффициентов детализации

Задание 3. Выполнить реконструкцию сигнал без коэффициентов детализации.

```

% Исходная последовательность данных
x = 0: pi/10: 10*pi;
l = length (x);
s = sin (x) + rand (1, l);
% Вейвлет-функция как базис преобразования
wname = 'coif2';

% Дискретный анализ
[sa, sd] = dwt (s, wname, 'mode', 'ppd');
% Графическая поддержка
subplot (211)
plot (s), title ('Исходный сигнал')

```

```

% Восстановление без коэффициентов детализации
% играет роль низкочастотного фильтра
s1 = idwt (sa, [], wname, 1);
% "Восстановленный" сигнал
subplot (212)
plot (s1), title ('Восстановленный сигнал')

```

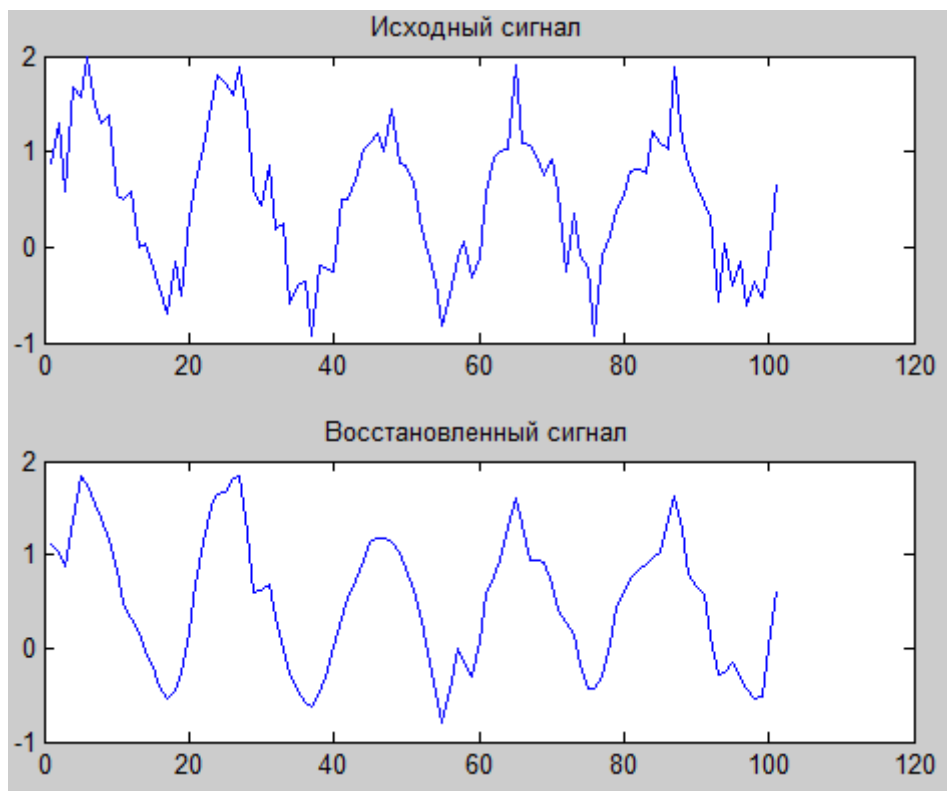


Рисунок 8 – Восстановление сигнала без использования коэффициентов детализации

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое вейвлет?

2. Что понимаю под вейвлет-синтезом?
3. Что такое непрерывное вейвлет-преобразование?
4. Какая функция MATLAB реализует вейвлет-преобразование?
5. В чем заключается вейвлет-синтез сигнала?

ЛАБОРАТОРНАЯ РАБОТА 8

Вейвлет-обработка и компрессия сигналов в MATLAB

Цель: получить основные навыки вейвлет-обработки и компрессии сигналов.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Важным применением вейвлетов стала очистка сигналов от шума и их сжатие (компрессия). Они основаны на ограничении уровня используемых коэффициентов.

В простейшей модели предполагается, что зашумленный сигнал имеет вид $s(n) = f(n) + \sigma e(n)$, где $f(n)$ – полезный сигнал, σ – уровень шума и $e(n)$ – гауссов белый шум, т. е. стационарная случайная последовательность с нулевым математическим ожиданием, абсолютно некоррелированная и с дисперсией, равной единице. Название «белый шум» отражает тот факт, что такой сигнал $e(n)$ имеет постоянный спектр на всех частотах по аналогии с белым светом, имеющим в видимой части равномерный сплошной спектр. Цель состоит в том, чтобы подавить шумовую часть $e(n)$ сигнала и восстановить f .

Основу методики удаления шумов из сигнала с использованием вейвлет-преобразования составляет использование пороговых функций различной формы, на основе которых происходит ограничение уровня детализирующих коэффициентов. Задав определенный порог для их уровня и «отсекая» коэффициенты ниже этого порога, можно значительно снизить уровень шума и сжать сигнал. Пример пороговых функций, рассматриваемых в данной работе и зачастую используемых в современных алгоритмах фильтрации, основанных на кратномасштабных схемах преобразования, представлен на рисунке 1.

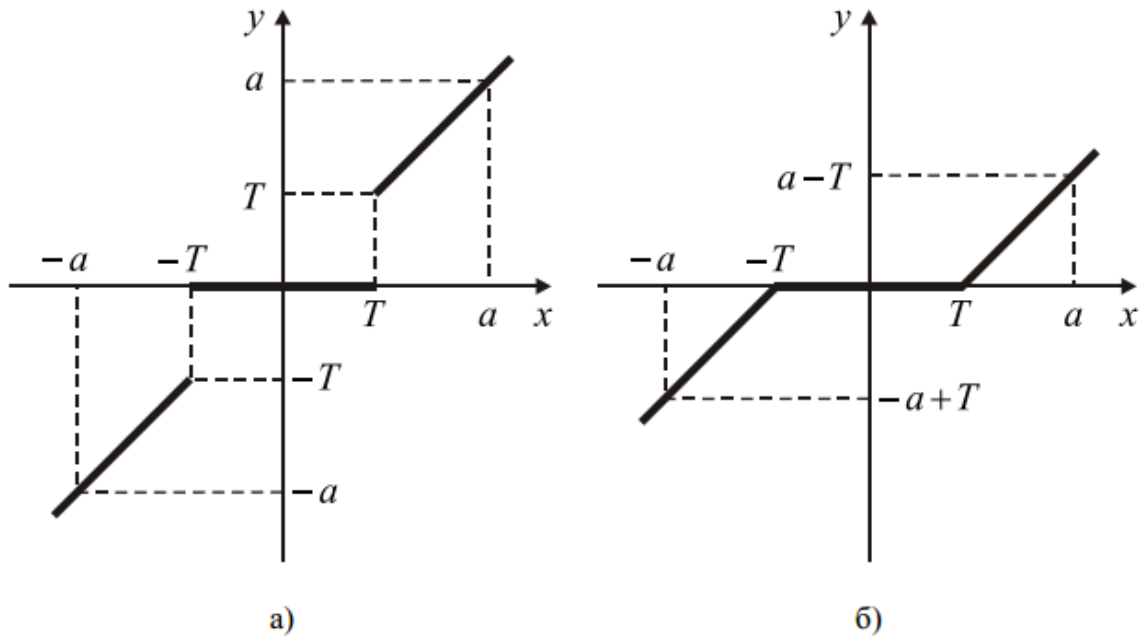


Рисунок 1. Пороговые функции обработки коэффициентов преобразования: а) жесткая пороговая функция; б) мягкая пороговая функция

На рисунке 1а представлена жесткая пороговая функция (жесткая пороговая оценка), описываемая следующим выражением

$$y(x) = \begin{cases} x, & \text{если } |x| \geq T, \\ 0, & \text{если } |x| < T \end{cases} \quad (1)$$

Здесь в качестве величины T выступает некоторое пороговое значение, о возможности определения которого будет сказано ниже, в качестве x выступают входное и выходное значения коэффициентов преобразования. На рисунке 1 б представлена мягкая пороговая функция (мягкая пороговая оценка), описываемая следующим выражением:

$$y(x) = \begin{cases} \text{sign}(x)(|x| - T), & \text{если } |x| \geq T, \\ 0, & \text{если } |x| < T \end{cases} \quad (2)$$

Обозначения, представленные в выражение (2) те же, что и описанные выше, в качестве функции $\text{sign}(x)$ определяющей знак коэффициента выступает функция вида:

$$\text{sign}(x) = \begin{cases} 1, & \text{если } x > 0, \\ 0, & \text{если } x = 0, \\ -1, & \text{если } x < 0. \end{cases} \quad (3)$$

Основное отличие мягкой пороговой функции от жесткой заключается в том, что первая из них (мягкая пороговая функция) не содержит разрыва в точке определяемой пороговым значением T . Иными словами мягкая пороговая функция, в отличие от жесткой, является непрерывной. Последнее обстоятельство, в случае мягкой пороговой обработки позволяет лучшим образом обрабатывать зашумленный сигнал (изображение) в окрестности точки его разрыва. Нужно отметить, что понижение значения коэффициентов разложения на величину порогового значения, в случае мягкой пороговой

обработки, в целом для большого количества сигналов оказывает неблагоприятное воздействие на итоговую оценку качества восстановленного сигнала. Поэтому в целом, как показывают эксперименты, более лучшей, с точки зрения численной оценки качества восстановленного сигнала, является жесткая пороговая оценка.

Как показывают эксперименты для того, чтобы в случае мягкой пороговой обработки численная оценка качества восстановленного сигнала приближалась к последней в случае жесткой пороговой обработки, необходимо выбирать значение порога T равным половине значения порога жесткой обработки сигнала. Основной проблематикой использования пороговых методов обработки является выбор порогового значения. В большом количестве научных работ выбор порога осуществлялся с использованием следующего выражения:

$$T = \sigma \sqrt{2 \ln(N)} \quad (4)$$

Формула (4) записана для случая одномерного сигнала. В формуле использованы следующие обозначения: N - общее количество отсчетов обрабатываемого сигнала, σ - среднеквадратическое отклонение шума, оцениваемое в данной работе с использованием следующего выражения

$$\sigma = \frac{M_X}{0.6745} \quad (5)$$

где M_X - медиана абсолютного значения множества коэффициентов $\{X\}$

Выражение (5) очень хорошо подходит для оценки среднеквадратического отклонения белого гауссова шума с нулевым математическим ожиданием, который является аддитивно подмешанным в некоторый полезный сигнал [5].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

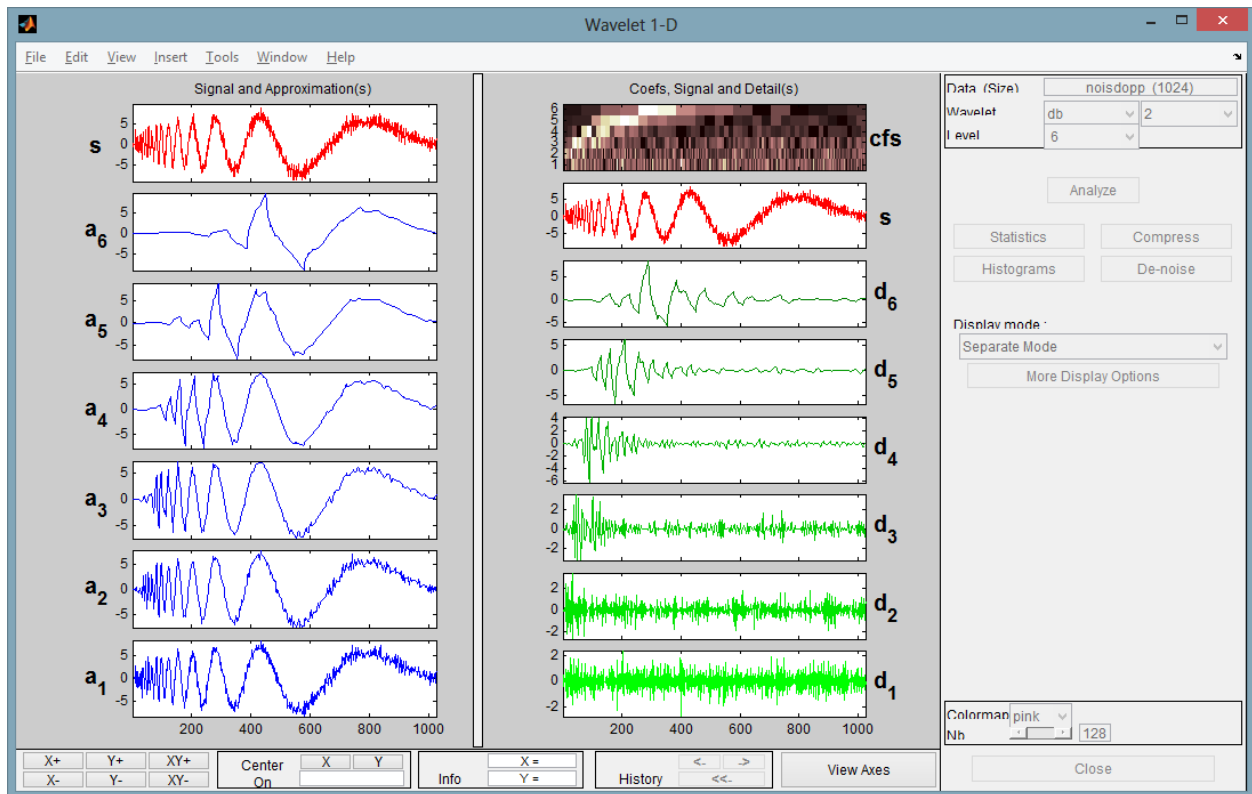


Рисунок 1. Окно преобразования в режиме графического вывода всех коэффициентов разложения (SeparateMode)

Окно преобразования включается из 'Wavelet Toolbox Main Menu' кнопкой "Wavelet 1-D" и имеет несколько большие функциональные возможности. Пример окна приведен на рисунке 1 в режиме графического вывода всех коэффициентов разложения (Separate Mode). Переключателем "Display mode" режимы вывода можно изменять, детали вывода можно устанавливать в отдельном подокне (кнопка "More Display Options").

Под кнопкой "Analyze" окно имеет 4 кнопки включения окон выполнения специальных операций над результатами разложения сигнала.

Окна "Statistics" и "Histograms" предназначены для анализа и графического вывода статистических характеристик сигнала и всех коэффициентов его разложения. В окнах "Compress" и "De-noise" устанавливаются режимы компрессии (сжатия) сигналов и очистки сигналов от шумов, и выполняются эти операции. Их применение будет рассмотрено в дальнейшем.

Подробности использования функций для проведения очистки сигнала от шума, а также компрессии можно найти в руководстве пользователя «De-noising» в Chapter 6, «Advanced Concepts» (Wavelet Toolbox\Using the Wavelet Toolbox\Advanced Concepts\Wavelet Applications\De-noising).

Основные функции очистки сигналов от шума:

- **thselect** – выбор порога для удаления шума 1-D;
- **wthresh** – мягкая или жесткая пороговая обработка (1-D, 2-D);
- **wthcoef** – пороговая обработка;
- **wthcoef2** – двумерная пороговая обработка;

- **wpthcoef** – пороговая обработка пакетных коэффициентов (1-D, 2-D);
- **wden** – автоматическое удаление шума 1-D;
- **wdencmp** – удаление шума и сжатие при помощи вейвлетов (1-D, 2-D);

Функция **thselect** – выбор порога для удаления шума 1D.

$THR = thselect(X, TPTR)$.

Находит значение порога для сигнала X , используя правило, определенное строкой $TPTR$. Этот параметр может принимать следующие значения:

- $TPTR = 'rigrsure'$ – адаптивный выбор порога, используя принцип Штейна несмещенной оценки риска;
- $TPTR = 'heursure'$ – эвристический вариант первого выбора;
- $TPTR = 'sqrtwolog'$ – $\text{poporsqrt}(2 * \log(\text{length}(X)))$;
- $TPTR = 'minimaxi'$ – минимаксный порог.

Правила выбора порога базируются на основной модели $y = f(t) + e$, где e является белым шумом $N(0,1)$. Если шум не масштабированный или небелый, можно его обработать, используя перемасштабирование выходного порога THR .

Функция **wden** – автоматическое удаление шума (1D). Функция **wden** автоматически удаляет шум одномерного сигнала X , используя вейвлеты. Применяется в виде:

```
[XD,CXD,LXD] = wden(X,TPTR,SORH,SCAL,N,'wname'),  
[XD,CXD,LXD] = wden(C,L,TPTR,SORH,SCAL,N,'wname').
```

Первая версия функции вычисляет очищенную от шума версию XD входного сигнала X , используя пороговую обработку вейвлет-коэффициентов. Дополнительно находит вейвлет-коэффициенты $[CXD,LXD]$ разложения очищенного сигнала XD .

Исходные параметры:

- X – сигнал;
- $TPTR$ – строка, определяющая выбор порога:
 - $'rigrsure'$ – адаптивный выбор порога, используя принцип Штейна несмещенной оценки риска (SURE);
 - $'heursure'$ – эвристический вариант первого выбора;
 - $'qtwolog'$ – универсальный порог $\text{sqrt}(2 * \log(\text{length}(X)))$;
 - $'minimaxi'$ – минимаксный порог;
- $SORH = 's'$ или $'h'$ – выбор мягкого или жесткого порога;
- $SCAL$ – строка, которая определяет мультипликативное пороговое перемасштабирование. Если шум вне пределов $[0,1]$ или небелый, то порог должен быть перемасштабирован с использованием оценки уровня шума. Возможны следующие варианты:
 - $'one'$ – без перемасштабирования, когда используется базовый метод;
 - $'sln'$ – для перемасштабирования порога с использованием оценки уровня шума на базе коэффициентов первого уровня;
 - $'mln'$ – для перемасштабирования порога с использованием оценки уровня шума, зависящей от уровня разложения;

- N – уровень вейвлет-разложения и 'wname' – имя ортогонального вейвлета.

Во втором случае функция `wden` производит то же самое, но использует уже полученное вейвлет-разложение $[C,L]$ исходного сигнала до уровня N и ортогональный вейвлет 'wname'.

Функция `wdencomp` – удаление шума и сжатие (компрессия) при помощи вейвлетов (1D, 2D).

```
[XC,CXC,LXC, PR0, PRL2] =
```

```
wdencomp('gbl',X,'wname',N,THR,SORH,KEEPAPP),
```

```
[XC,CXC,LXC, PR0, PRL2] = wdencomp('lvd',X,'wname',N,THR,SORH),
```

```
[XC,CXC,LXC, PR0, PRL2] = wdencomp('lvd',C,L,'wname',N,THR,SORH).
```

Описание. Случай параметра 'gbl'. Функция производит:

- XC – очищенную от шума версию входного сигнала X (1-D или 2-D);
- $[CXC,LXC]$ – структуру вейвлет-разложения сигнала XC ;
- $PR0$ – оценку числа обнуленных коэффициентов сигнала X в процентах;
- $PRL2$ – отношение L^2 -норм сжатого и первоначального сигналов в процентах. $PRL2 = 100 \cdot (\text{norm}(CXC) / \text{norm}(C))^2$, где $[C,L]$ – вейвлет-разложение X . Если X – одномерный сигнал и 'wname' – ортогональный вейвлет, то $PRL2 = 100 \cdot \|XC\|^2 / \|X\|^2$.

Функция использует:

- X – сигнал, или изображение;
- wname – имя вейвлета и N – уровень вейвлет-разложения;
- THR – значение порога;
- SORH = 's' или 'h' – выбор мягкого или жесткого порогового метода;
- KEEPAPP = 1, то коэффициенты аппроксимации не подвергаются пороговой обработке, иначе это возможно.

Функция `wdencomp('gbl',C,L,'wname',N,THR,SORH,KEEPAPP)` выполняет тоже самое, используя вейвлет-разложение $[C,L]$ исходного сигнала до уровня N .

В одномерном случае и опции 'lvd'. Функция выполняет то же самое, но используя зависимые от уровня пороги, указанные в векторе THR (THR должен иметь длину N). Кроме того, коэффициенты аппроксимации сохраняются.

Для двумерного случая и опции 'lvd'. В этом случае THR должен быть матрицей $3 \times N$, содержащей зависимые от уровня пороги в трех направлениях: по горизонтали, по диагонали и по вертикали.

Отметим, что сравнительно с `wden` (автоматическое удаление шума), `wdencomp` дает больше возможностей и можно осуществить свою стратегию удаления шума. Подробности можно найти в Chapter 2, «Using Wavelets».

Задание 1. Используя средства написания программного кода системы Matlab произвести построение и исследование алгоритма фильтрации аддитивного белого гауссова шума с нулевым математическим ожиданием и произвольно выбранным значением дисперсии из некоторого полезного сигнала на основе вейвлет-преобразования.

Основные этапы реализуемого алгоритма должны включать в себя:

1. Вычисление прямого вейвлет-преобразования сигнала (выбор типа вейвлет-функции и числа уровней вейвлет-разложения).

2. Изменение полученных коэффициентов преобразования по определенному правилу (выбор алгоритма нахождения порогового значения, выбор пороговой функции, выбор стратегии обработки детализирующий коэффициентов вейвлет-разложения).

3. Вычисление обратного вейвлет-преобразования от множества модифицированных (обработанных) вейвлет-коэффициентов.

Основные пункты исследования работы алгоритма должны включать:

1. Сравнение использования мягкой и жесткой пороговых функций внутри рассматриваемого алгоритма фильтрации. Подобную процедуру необходимо проводить при пяти различных степенях зашумления сигнала (дисперсии шума), произвольно выбранных настроек алгоритма, включающих в себя: тип вейвлет-функции, число уровней вейвлет-разложения, алгоритм нахождения порогового значения, стратегию обработки детализирующий коэффициентов вейвлет-разложения. Сравнение качества работы алгоритма осуществить на основе критерия среднеквадратической ошибки (СКО), вычисляемой с использованием следующего математического выражения:

$$\text{СКО} = \frac{1}{N} \sum_{k=0}^{N-1} (f(k) - \hat{f}(k))^2 \quad (6)$$

где $f(k)$ – эталонный сигнал, $\hat{f}(k)$ – восстановленный сигнал

2. Сравнение использования различных типов вейвлет-функций внутри рассматриваемого алгоритма фильтрации. В качестве вейвлет-функций рекомендуется выбирать вейвлеты из класса Добеши.

Подобную процедуру необходимо проводить при пяти различных степенях зашумления сигнала (дисперсии шума), трех различных типов вейвлетов, произвольно выбранных настроек алгоритма, включающих в себя: число уровней вейвлет-разложения, алгоритм нахождения порогового значения, тип пороговой функции, стратегию обработки детализирующий коэффициентов вейвлет-разложения. Сравнение качества работы алгоритма осуществить на основе критерия среднеквадратической ошибки (СКО).

3. Сравнение использования различного числа уровней вейвлет-разложения внутри рассматриваемого алгоритма фильтрации. Подобную процедуру необходимо проводить при пяти различных степенях зашумления сигнала (дисперсии шума), трех различных значений числа уровней вейвлет-разложения, произвольно выбранных настроек алгоритма, включающих в себя: тип вейвлет-функции, алгоритм нахождения порогового значения, тип пороговой функции, стратегию обработки детализирующий коэффициентов вейвлет-разложения. Сравнение качества работы алгоритма осуществить на основе критерия среднеквадратической ошибки (СКО).

Создадим файл MSE.mи добавим в него следующую функцию:

```

function mse = MSE(signal,rsignal)
% функция для оценки качества восстановленного
% сигнала по критерию среднеквадратической ошибки
%
% mse -- среднеквадратическая ошибка
% signal -- эталонный сигнал
% rsignal -- восстановленный сигнал
%
if isequal(signal,rsignal)
    disp('Сигналы идентичны; СКО равно нулю')
else
    N = length(signal);
    mse = sum((double(signal) - double(rsignal)).^ 2)/N;
    disp(sprintf('СКО = %5.4f',mse));
end

```

В дальнейшем мы будем вызывать эту функцию для оценки качества восстановленного сигнала.

```

% Программа для обработки (фильтрации) сигнала
% на основе дискретного вейвлет-преобразования
%
t = 0:0.000001:0.0004;
A = 1; F = 10000; phi = 45;
signal = A*sin(2*pi*F*t + phi);
subplot(311); plot(t,signal)
title('Исходный не зашумленный сигнал'); grid
xlabel('Время, t'); ylabel('Значения сигнала')
signal_noise = signal + ...
random('norm',0,0.2,[1 length(t)]);
subplot(312); plot(t,signal_noise)
title('Зашумленный сигнал'); grid
axis([min(t) max(t) min(signal_noise) max(signal_noise)])
xlabel('Время, t'); ylabel('Значения сигнала')
tptr = 'sqrtwolog';
sorr = 'h';
scal = 'one';
lev = 4;
wavefunc = 'db8';
rsignal = wden(signal_noise,tptr,sorr,scal,lev,wavefunc);
mse = MSE(signal,rsignal);
subplot(313); plot(t,rsignal)
title(['Восстановленный сигнал ' ',' ' ' 'СКО' '=' num2str(mse)]); grid
axis([min(t) max(t) min(rsignal) max(rsignal)])
xlabel('Время, t'); ylabel('Значения сигнала')

```

Результат выполнения программы представлен на рисунке 2.

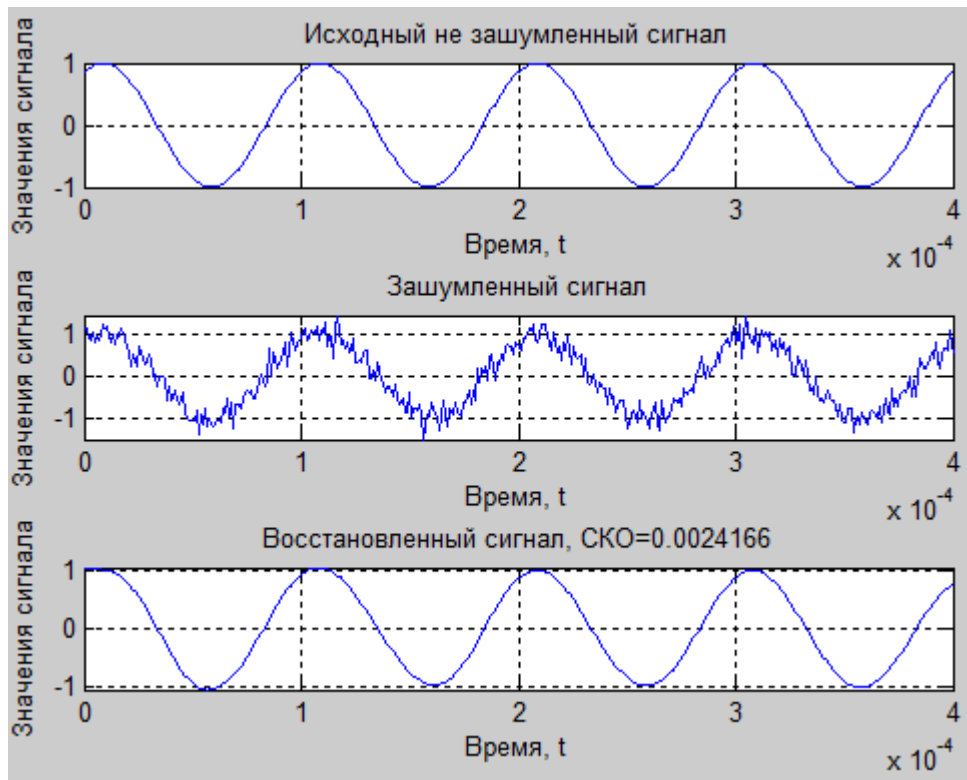


Рисунок 3. Окно обработки (фильтрации) сигнала на основе дискретного вейвлет-анализа

Задание 2. Удаление шума, компрессия Функции вейвлетанализа в MATLAB. Создать зашумленный сигнал функцией `wnoise`. Для удаления шума используем мягкий эвристический метод SURE. Разложение x берем до уровня 5 по вейвлету `sym8`.

```
snr = 3; init = 2055615866;
[xref,x] = wnoise(3,11,snr,init);
xd = wden(x,'heursure','s','one',5,'sym8');
subplot(311), plot(xref), axis([1 2048 -10 10]);
title('Original signal');
subplot(312), plot(x), axis([1 2048 -10 10]);
title('Noisy signal');
subplot(313), plot(xd), axis([1 2048 -10 10]);
title('De-noised signal - heuristic SURE')
```

Результат представлен на рисунке 3.

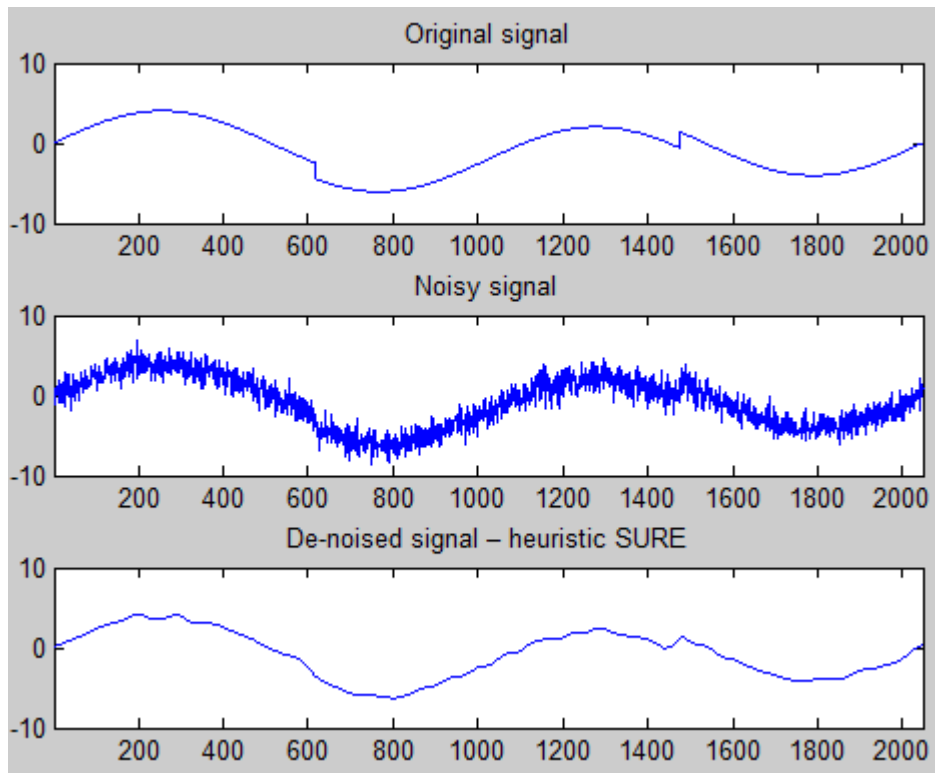


Рисунок 3. Окно преобразования в режиме графического вывода всех коэффициентов разложения (Separate Mode)

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. На чем основана идеология построения алгоритма фильтрации на основе вейвлет-преобразования?
2. Какие функции Matlab используются для создания зашумленного сигнала?
3. Назовите примерные этапы проведения алгоритма фильтрации зашумленного сигнала на основе вейвлет-преобразования.
4. Расскажите о функциях MATLAB, обеспечивающих удаление шума и сжатие (компрессию) при помощи вейвлетов.

ЛАБОРАТОРНАЯ РАБОТА 9

Основные приложения и команды по обработке изображений в Matlab

Цель: ознакомиться с основными утилитами и командами Matlab по обработке изображений.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

В Matlab R2015a можно обрабатывать изображения посредством разнообразных утилит. Утилиты можно посмотреть на вкладке APPS в разделе **Image Processing and Computer Vision**, рисунок 1.1. Разумеется в эти утилиты вынесены лишь самые общие функции и возможности среды. Для более детальной обработки придётся писать код.

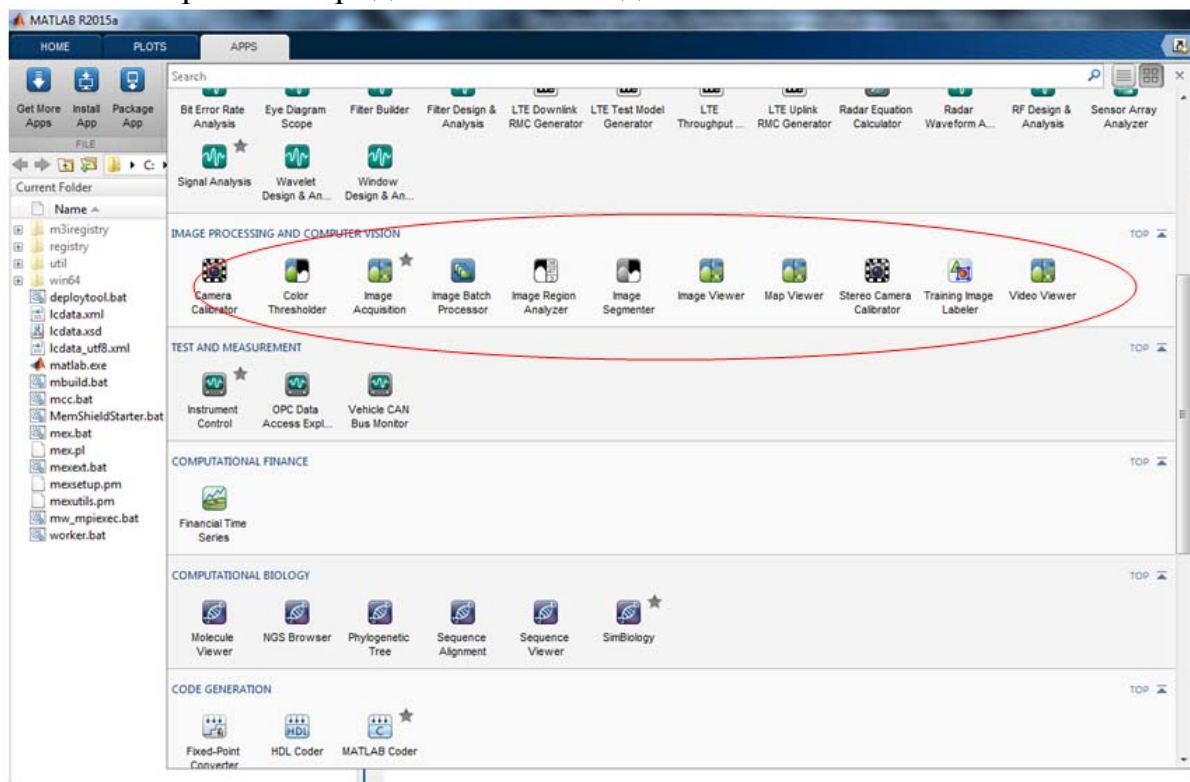


Рисунок 1.1 – Список утилит для обработки изображений и компьютерного зрения

Рассмотрим некоторые утилиты более подробно.

Утилита **Color Threshold** позволяет работать с цветом. Мы можем загрузить картинку, используя кнопку **Load Image**, приложение предложит выбрать тип цветовой кодировки, рисунок 1.2.

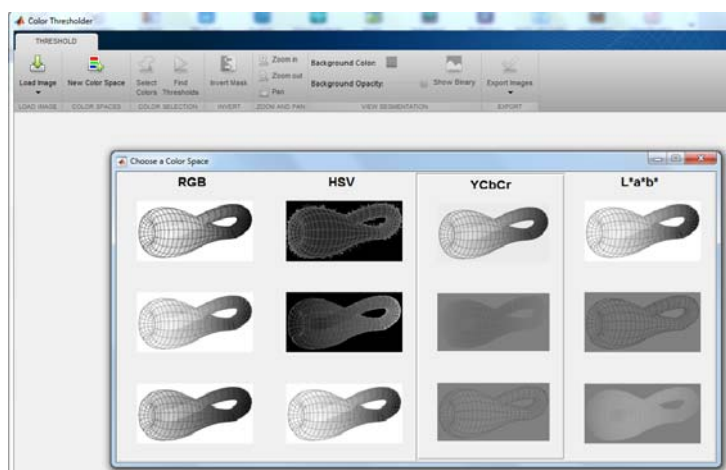


Рисунок 1.2 – Выбор типа цветовой кодировки для загруженного изображения

Для конкретной кодировки будут построены гистограммы по значениям того или иного цвета, рисунок 1.3

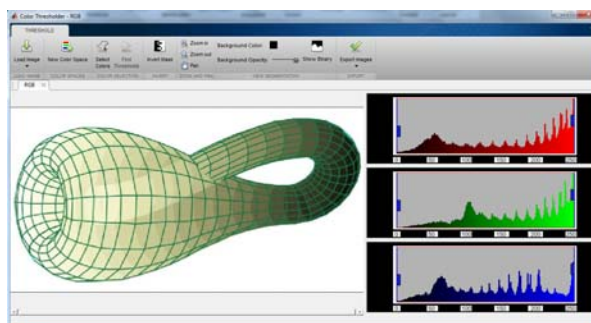


Рисунок 1.3 – Информация о цветных компонентах изображения

Возможно также выделить область и запустить алгоритм поиска цветового порога, как показано на рисунке 1.4.

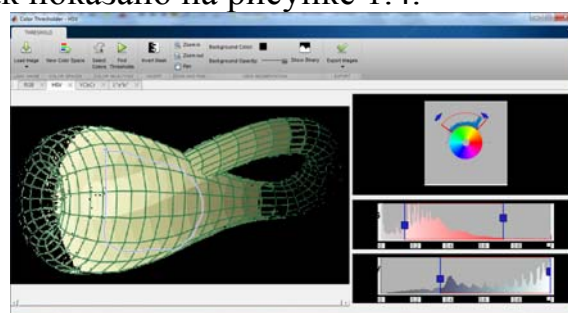


Рисунок 1.4 – Алгоритм поиска цветового порога

Рассмотрим утилиту **Image Batch Processor**, которая позволяет обрабатывать сразу несколько изображений (возможно даже в параллельном режиме), используя заранее определённую функцию. Общий вид показан на рисунке 1.5.

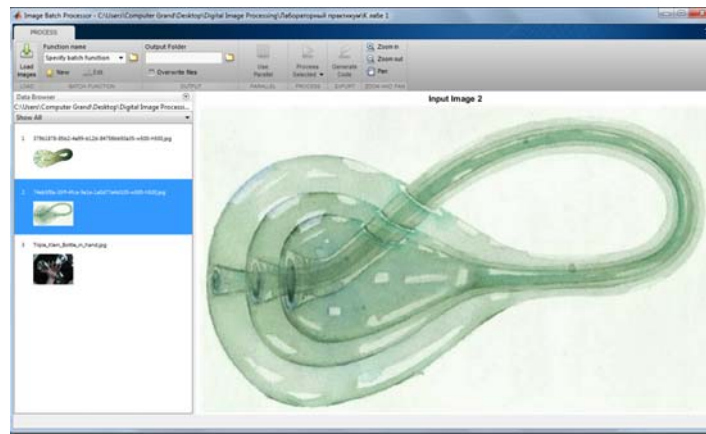


Рисунок 1.5 – Общий вид утилиты Image Batch Processor

В поле Function name можно ввести имя заранее запрограммированной функции. По щелчку правой кнопки мыши на картинке можно экспортировать её в рабочую среду, рисунок 1.6

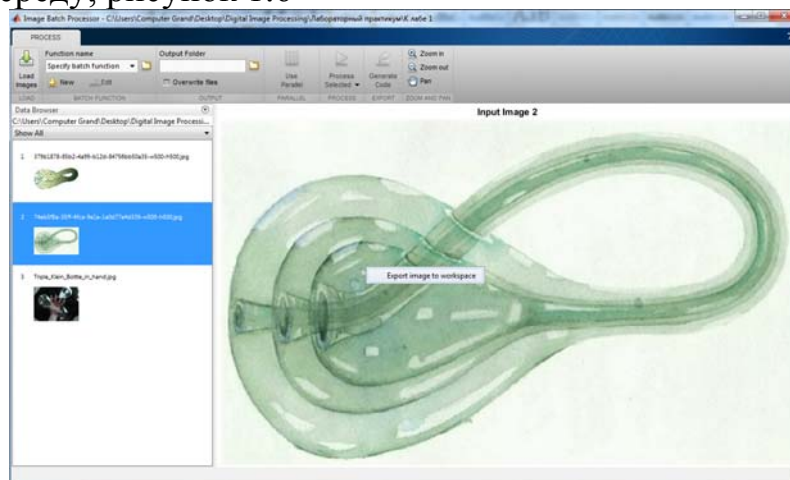


Рисунок 1.6 – Экспорт картинки в рабочую среду

В Matlab картинка будет отображаться в окне Workspace в качестве переменной-ресурса. Как показано на рисунке 1.7, если картинка не слишком большая, то значение её пикселей можно отобразить в окне просмотра значений переменных. Справа от имени переменной-картинке будут указаны её основные параметры: размерность её массива и тип для хранения каждого элемента этого массива. Из рисунка 1.7 видно, что картинка содержится в трёхмерном массиве $303 \times 500 \times 3$, где каждый элемент кодируется одним байтом (unit8).

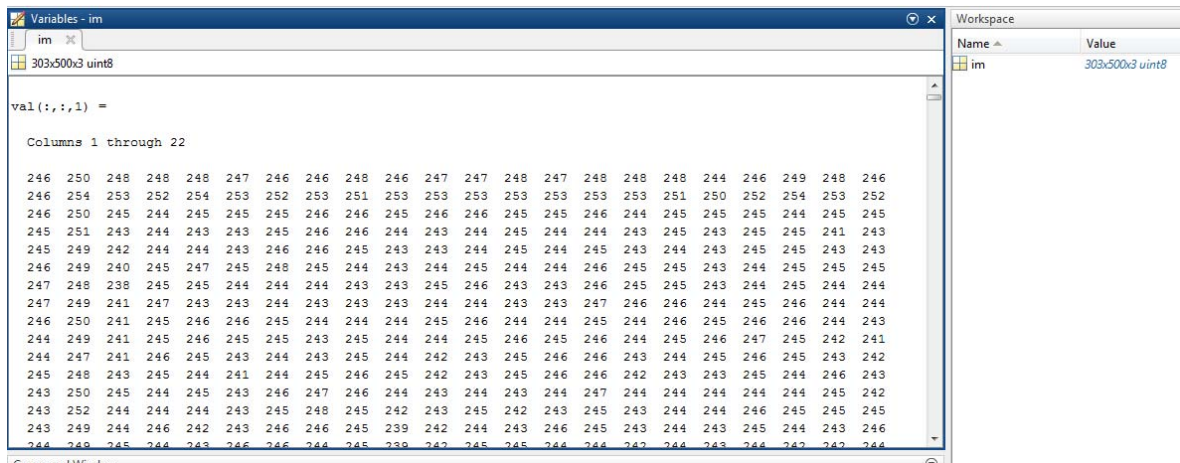


Рисунок 1.7 – Экспортированная в среду Matlab картинка

Следующая утилита – Image Segmenter, предназначена для сегментации изображения. Общий вид приложения показан на рисунке 1.8.

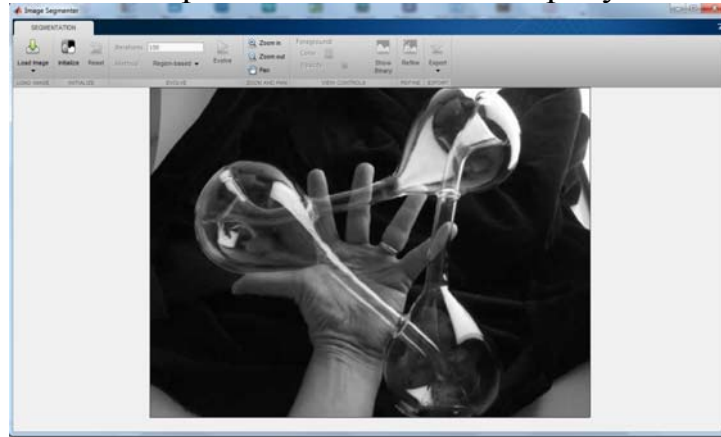


Рисунок 1.8 – Общий вид Image Segmenter

С помощью этой утилиты мы можем быстро настроить алгоритм по выделению некоторых частей картинки, которые имеют что-то общее между собой, т.е. выделение различных зон изображения. Результат работы алгоритма сегментации для произвольной картинке представлен на рисунке 1.9.

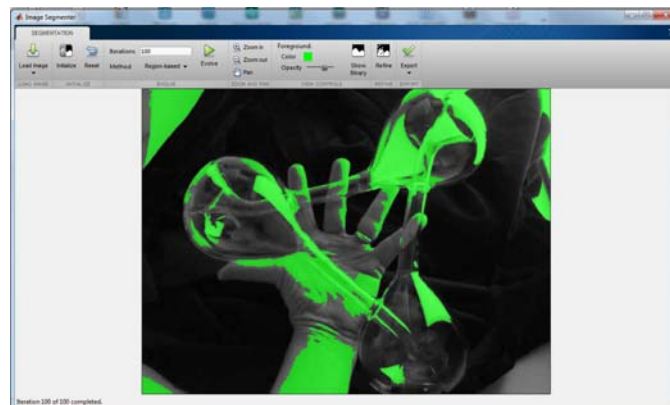


Рисунок 1.9 – Пример работы алгоритма сегментации картинки

Ещё одна утилита общего назначения – Image Viewer. Предназначена

для просмотра изображений, получения информации о них и ряда других второстепенных действий. На рисунке 1.10 показан вывод информации о картинке, загруженной слева. В отличие от обычных графических редакторов выводится и ряд специфичной информации.

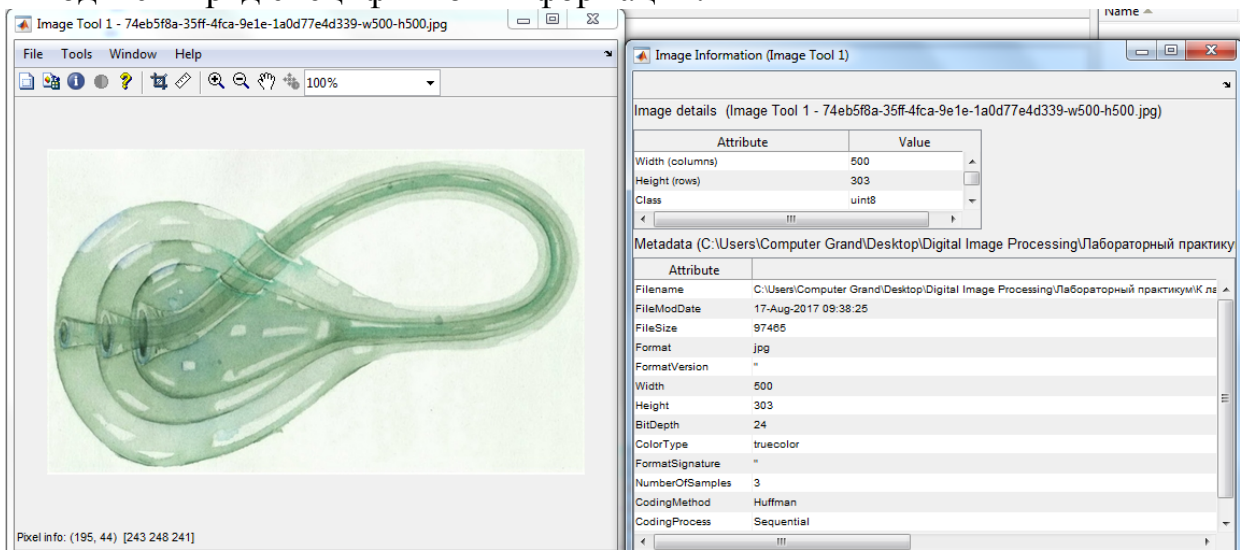


Рисунок 1.10 – Просмотр информации о загруженном изображении

У загруженной картинке можно посмотреть значения пикселей в выделенном месте, рисунок 1.11. Можно измерить длину картинки в дюймах, рисунок 1.12.

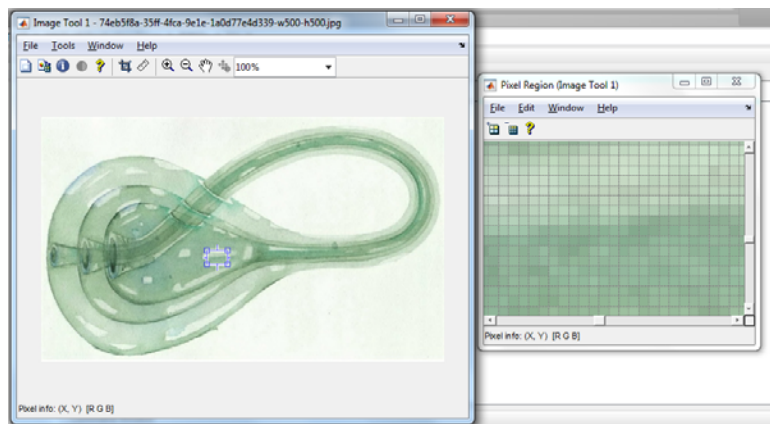


Рисунок 1.11 – Просмотр значений пикселей в выделенном месте

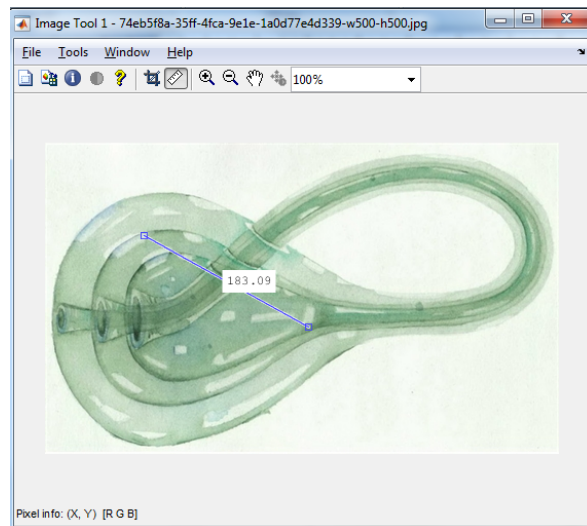


Рисунок 1.12 – Измерение в дюймах величины картинки

Для более подробной работы с изображениями необходимо подключать код. Рассмотрим примеры простого кода.

Загрузка, чтение изображений.

```
f=imread('C:\Users\Computer Grand\Desktop\Digital Image Processing\Лабораторный практикум\К лабе 1\odins-wilde-jagd.jpg');
```

```
% Получаем размеры изображения
```

```
size(f)
```

```
ans =
```

```
557 806 3
```

Вывод более подробной информации с помощью команды **whos**.

```
whos f
```

Name	Size	Bytes	Class	Attributes
------	------	-------	-------	------------

f	557x806x3	1346826	uint8	
---	-----------	---------	-------	--

```
% Просмотр изображения в отдельном окне
```

```
imshow(f, [])
```

Результат приведён на рисунке 1.13.

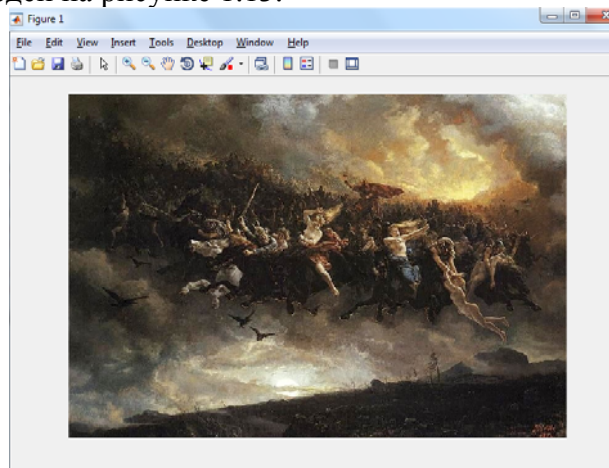


Рисунок 1.13 – Вывод загруженного изображения на экран

Можно загрузить несколько картинок и отразить их в разных окнах. Для этого `imshow()` нужно чередовать с `figure`, рисунок 1.14.

```
g=imread('C:\Users\Computer Grand\Desktop\Digital Image Processing\Лабораторный
практикум\К лабе 1\74eb5f8a-35ff-4fca-9e1e-1a0d77e4d339-w500-h500.jpg');
imshow(f, []), figure, imshow(g, [])
```



Рисунок 1.14 – Вывод двух картинок в двух разных окнах

Сохранить картинку можно с помощью команды `imwrite()` в нужном расширении. Получить информацию о ней можно с помощью команды `iminfo()`. Значения структуры переменной `info` в коде ниже можно посмотреть в среде Matlab, как показано на рисунке 1.15.

```
imwrite(f, 'C:\Users\Computer Grand\Desktop\Digital Image Processing\Лабораторный
практикум\К лабе 1\Picture', 'tif');
Info = iminfo('C:\Users\Computer Grand\Desktop\Digital Image
Processing\Лабораторный практикум\К лабе 1\74eb5f8a-35ff-4fca-9e1e-1a0d77e4d339-
w500-h500.jpg');
```

Variables - Info	
Info BitDepth	
1x1 struct with 14 fields	
Field	Value
Filename	'C:\Users\Computer ...'
FileModDate	'17-Aug-2017 09:38:25'
FileSize	97465
Format	'jpg'
FormatVersion	''
Width	500
Height	303
BitDepth	24
ColorType	'truecolor'
FormatSignature	''
NumberOfSamples	3
CodingMethod	'Huffman'
CodingProcess	'Sequential'
Comment	0x0 cell

Рисунок 1.15 – Информационные поля, содержащие данные о картинке

С помощью кода также можно уменьшить картинку или перевернуть её.

Для этого достаточно по-другому отразить элементы матрицы. Ниже в `fp` записываем матрицу, где строки перечисляем от последнего индекса (`end`) до первого с шагом в `-1`. Индексы столбцов и индекс номера квадратной матрицы (массив трёхмерный) остаются неизменными (`:`).

```
fp = f(end:-1:1, :, :);  
imshow(fp, []);
```

Перевернутая картинка показана на рисунке 1.16.



Рисунок 1.16 – Перевернутое изображение

Работая с индексами также можно довольно легко уменьшить изображение в два раза. Результат показан на рисунке 1.17.

Необходимо организовать цикл по матрице с шагом в два.

```
fs=f(1:2:end, 1:2:end, :);  
imshow(f, []), figure, imshow(fs, []);
```

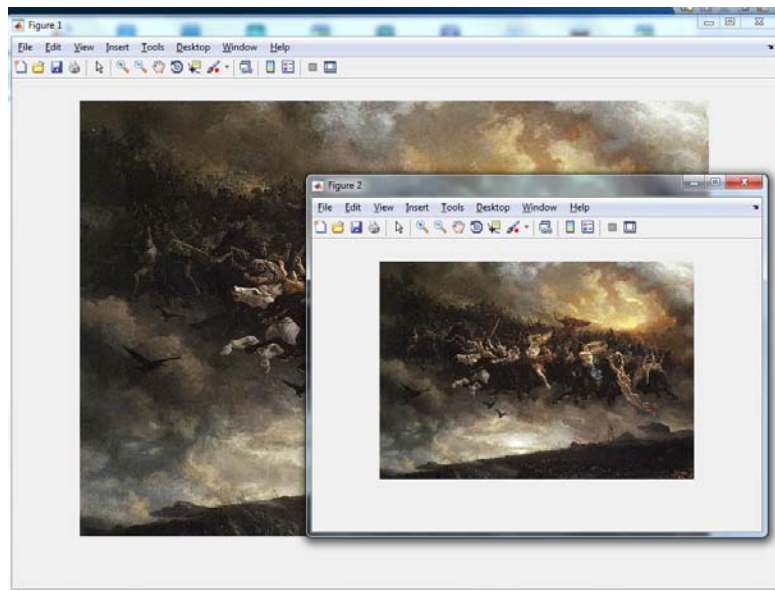


Рисунок 1.17 – Исходное изображение (на заднем фоне) и уменьшенное в два раза

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память

1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab: утилиты Patch Image Processor обработку изображений. Написать функцию для этой задачи на языке Matlab.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов по работе соответствующей утилиты Matlab.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

- 1) Какие существуют утилиты Matlab для обработки изображений?
- 2) Какие вы знаете основные команды Matlab для работы с обработкой изображений?
- 3) В каком виде изображения представляются в среде Matlab?

ЛАБОРАТОРНАЯ РАБОТА 10

Преобразование яркости изображений

Цель: научиться преобразовывать яркость изображения.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Как мы говорили в начале, результатом дискретизации и квантования изображения является матрица чисел. В результате цифровое изображение состоит из конечного числа элементов, каждый из которых расположен в конкретном месте и имеет определенное значение. Эти элементы принято называть пикселями.

Предположим, что изображение $f(x,y)$ после дискретизации представлено в виде матрицы, у которой имеется M строк и N столбцов. Тогда говорят, что изображение имеет размер $M \times N$. В литературе по обработке изображений за начало координат принимается верхний левый угол изображения, координатами которого служит пара $(0, 0)$. Следующая точка в первой строке изображения имеет координаты $(0,1)$. Это, однако, не обозначает настоящие физические координаты при регистрации соответствующего изображения. Фактически точки изображения мы задаем их индексами (i, j) , например, i – номер строки с отсчетом от нуля, а j – номер столбца. Однако мы будем придерживаться обозначения (x, y) .

Обозначения массивов, используемые в пакете IPT MatLab, отличается от описанного выше тем, что левый верхний пиксель имеет координаты $(1,1)$ а переменные номера строк и столбцов изменяются от 1 до M и от 1 до N . Т.о. цифровое изображение имеет естественное представление в виде матрицы, рисунок 2.1.

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

Рисунок 2.1 – Представление изображения в виде матрицы

Кроме того, мы предполагаем, что дискретные уровни яркости расположены с постоянным шагом (т.е. используем равномерное квантование). Интервал значений яркости, где – минимальное и максимальное значение матрицы f , называют динамическим диапазоном изображения. Если заметная доля пикселей занимает значительную часть всего диапазона уровней серого, то мы говорим, что изображение имеет высокий контраст. Наоборот, изображение с малым динамическим диапазоном обычно выглядит тусклым, размытым и серым. Увеличение динамического диапазона изображения делает его более контрастным.

Методы, которые мы будем рассматривать в этой главе, направлены на улучшение изображений. При этом следует иметь в виду, что визуальное оценивание качества изображений является достаточно субъективным.

Процедуры обработки пространственной области изображения оперируют напрямую с пикселями изображения. Методы обработки изображения, используемые в этой главе, можно описать преобразованием

$f(x,y)$ – входное изображение, $g(x,y)$ – выходное (обработанное) изображение, а T — некоторый оператор (преобразование) над f , который определен в некоторой окрестности точки (x, y) . В ряде случаев оператор T может обрабатывать несколько изображений, например, он может суммировать или усреднять N входных изображений.

Основа этих методов состоит в определении пространственной окрестности вокруг точки (x, y) , которая может быть квадратной, прямоугольной или дугой по форме областью с центром в точке (x, y) , как показано на рисунке.

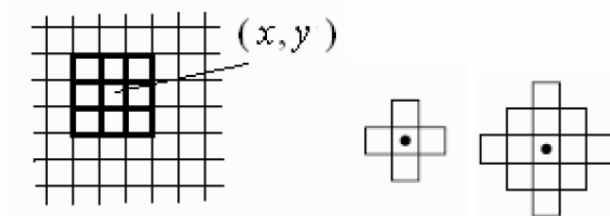


Рисунок 2.1 – Различные виды окрестности точки (x, y)

Центр заданной шаблонной подобласти перемещается от пиксела к пикселу, начиная, из верхнего левого угла, и на своем пути накрывает различные окрестности. Преобразование T применяется в каждой точке, давая в результате выходное (обработанное) значение g для данной точки. В процессе вычислений используются только пиксели внутри заданной окрестности с центром в (x, y) ,

Простейшая форма преобразования T получается, когда окрестность на рисунке имеет размер 1×1 (т.е. состоит из одного пиксела). В этом случае значение g в точке (x, y) зависит только от значения f в этой точке, и T становится функцией преобразования яркости.

Усиление контрастности. На следующем рисунке слева показано входное изображение f . Применяя к нему преобразования $101 + ff/g$ и $5222 + ff/g$, мы получаем изображения, приведенные в центре и справа.

```
f=imread('Bone2.tif');
imshow(f); % исходное изображение (слева)
g1=double(f)./(double(f)+10);
imshow(g1); % растяжения контрастности
g2=double(f).^2./(double(f).^2+5);
imshow(g2); % слишком контрастное изображение (справа)
```

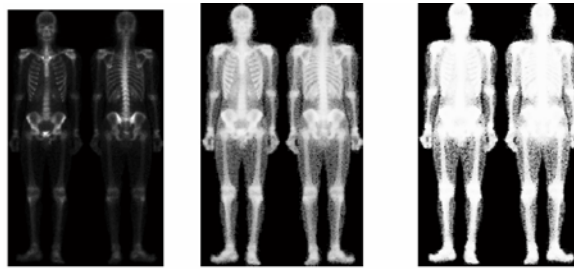


Рисунок 2.3 – Результат усиления контрастности

Кривые, используемые для получения среднего и правого изображения и выполняющие преобразование контрастности, имеют следующие графики

```
x=0:1:255; y=x./(x+10);plot(x,y,'LineWidth',2); grid on;
```

```
x=0:1:255; y=x.^2./(x.^2+5);plot(x,y,'LineWidth',2); grid on;
```

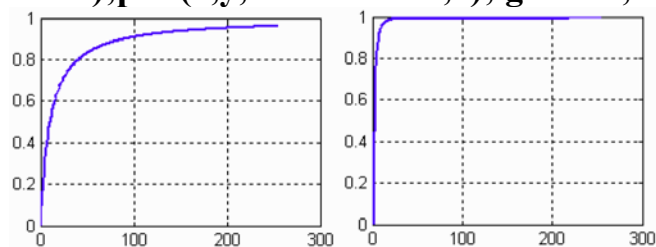


Рисунок 2.4 – Графики преобразования контрастности

Откладывая на горизонтальной оси значение яркости точки, которое в нашем примере лежит в диапазоне от 0 до 255, на вертикальной оси находим значение преобразованной яркости, которое для изображения класса double лежит в диапазоне от 0 до 1. Для определения максимального значения яркости преобразованного изображения можно использовать команду

```
max(max(g1))
```

```
ans =
```

```
0.9623
```

Функции, выполняющие преобразование могут выбираться по-разному. Часто используется логарифмическое преобразование, которое выполняется с помощью выражения

c — некоторая константа. Форма этой кривой при $c=1$ и $0 \leq f$ показана на следующем рисунке.

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае

обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab преобразование яркости изображений. Допустим, обработать обучающую и тестовую выборку для нейронной сети.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов по работе соответствующей утилиты Matlab.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое преобразование яркости изображения?
2. Для чего нужна операция преобразования яркости изображения?
3. Как в Matlab можно организовать преобразование яркости изображения?

ЛАБОРАТОРНАЯ РАБОТА 11

Пространственная фильтрация изображений

Цель: научиться проводить пространственную и частотную фильтрацию изображений.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Рассмотрим некоторые методы пространственной фильтрации. Большинство из них относится к локальным методам. Реализация этих методов состоит из четырех основных этапов:

1. формирование локальной окрестности и ее центральной точки;
2. проведение операции на пикселями данной локальной окрестности;
3. присвоение результата операции (п. 2) центральной точке окрестности;
4. повторение операций, описанных в п.п. 1 – 3, для каждой точки изображения.

Далее приведем примеры применения некоторых видов пространственных фильтров.

Сначала считаем исходное изображение

```
L=imread('leo1.bmp');
```

Для лучшего понимания сути приведем реализацию метода для одной цветовой составляющей.

```
L=L(:, :, 1);
```

```
figure, imshow(L);
```

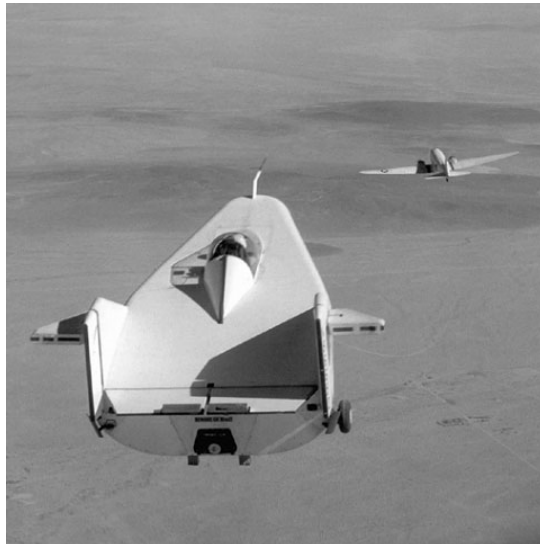


Рисунок 3.1 – Исходное изображение

Испортим исходное изображение шумом типа ”соль и перец”.

```
L=imnoise(L,'salt & pepper',0.01);
```

```
figure, imshow(L);
```

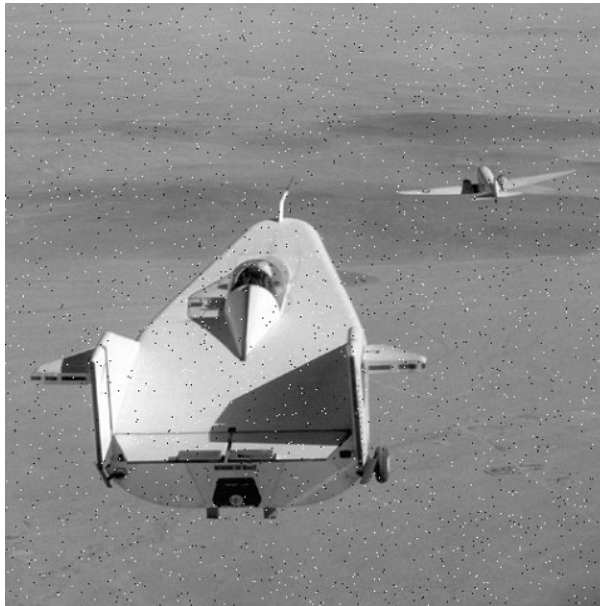


Рисунок 3.2 – Изображение с шумом «соль и перец»

Далее применим к этому зашумленному изображению различные виды пространственных фильтров и проанализируем результат обработки.

Прежде чем приступить к реализации локальных методов, необходимо расширить границы исходного изображения. Размеры расширения зависят от размеров маски фильтра. Если проводить фильтрацию с помощью встроенной функции `imfilter`, то параметры расширения можно задавать с помощью опций:

- ‘P’ – границы изображения расширяются значением P. По умолчанию значение $P=0$;
- ‘replicate’ – размер изображения увеличивается повторением величин на его боковых границах;
- ‘symmetric’ – размер изображения увеличивается путем зеркального отражения через границы;
- ‘circular’ – размер изображения увеличивается периодическим повторением двумерной функции;

В этом примере будем использовать не встроенную функцию `imfilter` и опции расширения, а приведем детальное описание работы нескольких пространственных фильтров. Размер изображения увеличим путем зеркального отражения через границы.

```
%n, m – размеры локальной окрестности
n1=fix(n/2);m1=fix(m/2);
a=L(1,1);b=L(1,M);c=L(N,1);d=L(N,M);
for i=1:n1;
    for j=1:m1;
        L1(i,j)=a;
        L3(i,j)=b;
        L6(i,j)=c;
        L8(i,j)=d;
    end;
end;
L2=L(1,1:M);
```

```

L02=L2;
for i=1:n1-1;
    L2=[L2;L02];
end;
L7=L(N,1:M);
L07=L7;
    for i=1:n1-1;
        L7=[L7;L07];
    end;
L4=L(1:N,1);
L4=L4';
L04=L4;
    for i=1:m1-1;
        L4=[L4;L04];
    end;
L4=L4';
L5=L(1:N,M);
L5=L5';
L05=L5;
    for i=1:m1-1;
        L5=[L5;L05];
    end;
    L5=L5'; L1=[L1;L4];
L1=[L1;L6]; L1=L1'; L2=[L2;L]; L2=[L2;L7];
    L2=L2'; L3=[L3;L5]; L3=[L3;L8];
L3=L3'; L1=[L1;L2]; L1=[L1;L3];
Lr=L1';

```

Приведем программную реализацию непосредственно самих методов

```

for i=1+n1:N+n1;
    disp(i);
    for j=1+m1:M+m1;

% Формирование локальной окрестности D
        if j==1+m1;
            D=0;
            for a=-n1:n1;
                for b=-m1:m1;
                    D(n1+1+a,m1+1+b)=Lr(i+a,j+b);
                end;
            end;
        end;
        if j>1+m1;
            for a=-n1:n1;
                D(n1+1+a,m+1)=Lr(i+a,j+m1);
            end;
            D=D(1:n,2:m+1);
        end;
    end;
end;

```

% Формирование результата обработки

% Контргармоническое среднее

```

    Lout1(i,j)=sum(sum(D.^(q+1)))/sum(sum(D.^q+eps));
% Арифметическое среднее
    Lout2(i,j)=sum(sum(D))./(m*n);
% Геометрическое среднее
    Lout3(i,j)=prod(prod(D)).^(1/(m*n));
% Гармоническое среднее
    Lout4(i,j)=(m*n)/sum(sum(1./(D+eps)));
% Медиана
    Lout5(i,j)=median(median(D));
% Максимум по окрестности
    Lout6(i,j)=max(D(:));
% Минимум по окрестности
    Lout7(i,j)=min(D(:));
% Срединная точка
    Lout8(i,j)=0.5*(max(D(:))+min(D(:)));
    end;
end;
% a-усеченное среднее
d=6;
    Dus=sort(D(:));
    Dus=Dus(d/2+1:end-d/2);
    Lout9(i,j)=mean(Dus);

```

Поскольку вначале проводилось расширение исходного изображения, то теперь все результирующие изображения необходимо привести к исходным размерам.

```

Lout1=Lout1(n1+1:N+n1,m1+1:M+m1);
Lout2=Lout2(n1+1:N+n1,m1+1:M+m1);
Lout3=Lout3(n1+1:N+n1,m1+1:M+m1);
Lout4=Lout4(n1+1:N+n1,m1+1:M+m1);
Lout5=Lout5(n1+1:N+n1,m1+1:M+m1);
Lout6=Lout6(n1+1:N+n1,m1+1:M+m1);
Lout7=Lout7(n1+1:N+n1,m1+1:M+m1);
Lout8=Lout8(n1+1:N+n1,m1+1:M+m1);
Lout9=Lout9(n1+1:N+n1,m1+1:M+m1);
Визуализируем полученные результаты.
figure, imshow(Lout1);
title('Контргармоническое среднее');

```

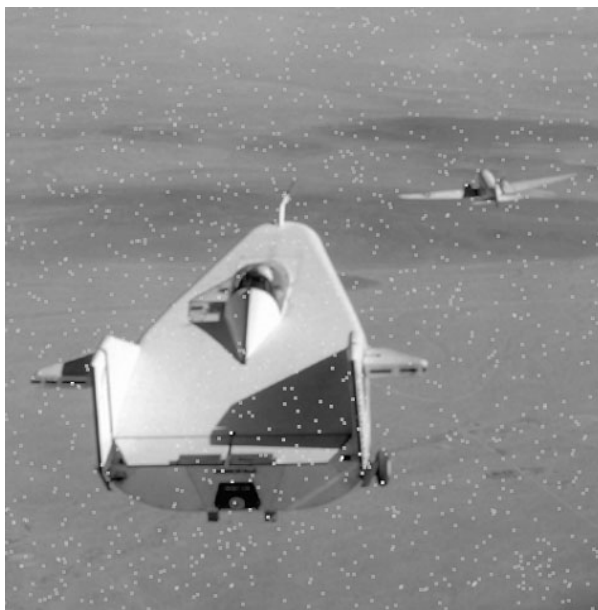


Рисунок 3.3 – Контргармоническое среднее при $q = 1.5$

Фильтр типа геометрическое среднее можно представить следующим выражением:

$$L_{out}(i, j) = \left[\prod_{(i, j) \in D} L(i, j) \right]^{\frac{1}{|D|}}. \quad (1)$$

На основе выражения (1) можно предположить, что результат обработки будет существенно зависеть от размеров локальной апертуры. Это подтверждают обработанные этим методом изображения, которые представлены на рис. 7 и 8.

```
figure, imshow(Lout4);  
title('Гармоническое среднее');
```

Гармоническое среднее

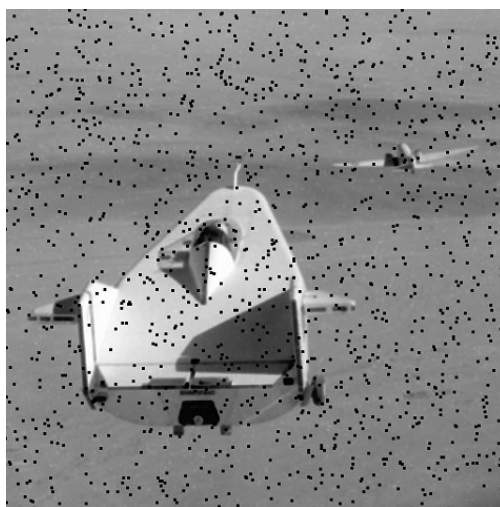


Рисунок 3.4 – Гармоническое среднее при размере локального окна 3×3 .

Гармоническое среднее

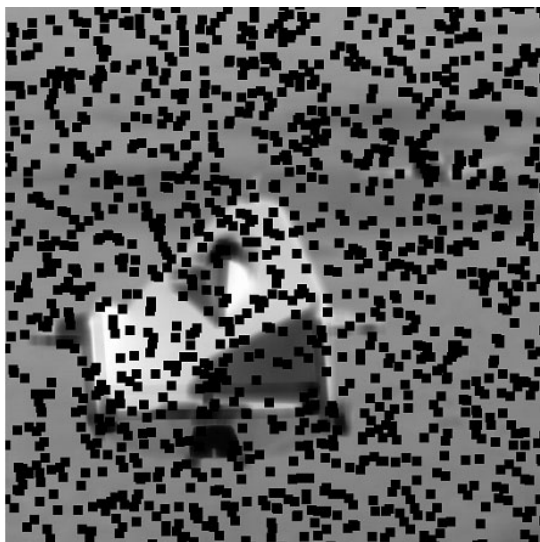


Рисунок 3.5 – Гармоническое среднее при размере локального окна 9×9.

Исходя из выражения фильтра гармонического среднего

$$Lout(i, j) = \frac{mn}{\sum_{(i,j) \in D} \frac{1}{L(i, j)}}, \quad (2)$$

можно предположить, что результат обработки, как и в предыдущем случае, будет существенно зависеть от размеров локальной апертуры. Подтверждением тому являются изображения, которые обработаны с использованием фильтра гармонического среднего при различных размерах локальной апертуры (см. рис. 3.4, 3.5).

```
figure, imshow(Lout5);  
title('Медиана');
```

Медиана изображения показана на рисунке 3.6.

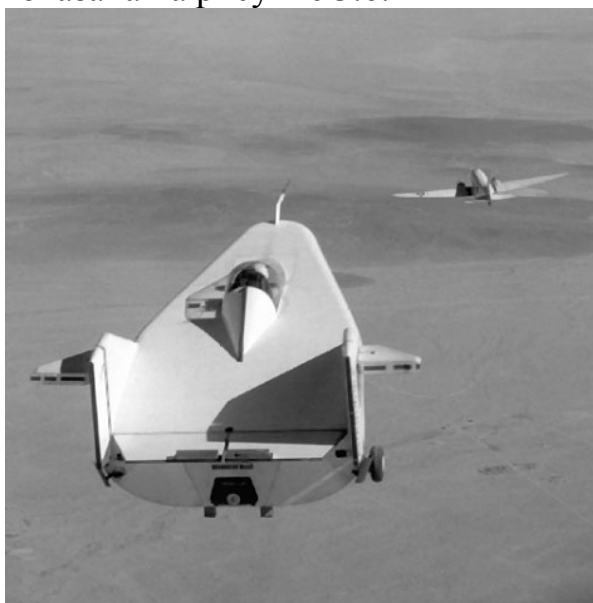


Рисунок 3.6 – Медиана при размере локального окна 3×3.

Из приведенных на рисунках 3.5 и 3.6 изображений видно, что фильтр

типа α –усеченное среднее является эффективным инструментом для удаления импульсных шумов. Конечно, нужно подобрать соответствующий размер локальной апертуры.

В данной работе приведены примеры работы нескольких наиболее известных методов пространственной фильтрации. Также в этой работе мы показали влияние изменения размеров локальной апертуры на результат обработки.

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab пространственную фильтрацию изображений.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов по работе с фильтрацией изображений в Matlab

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Чем пространственная фильтрация отличается от частотной?
2. Какие стандартные фильтры вы знаете? Для чего они применяются?
3. Как в Matlab можно организовать фильтрацию изображений?

4. Какие команды вы знаете для фильтрации изображений?

ЛАБОРАТОРНАЯ РАБОТА 12

Восстановление изображений

Цель: научиться восстанавливать изображения.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

При съемке движущегося объекта неподвижной камерой полученное изображение получается смазанным. Если знать параметры движения объекта, то можно построить ядро свертки, которое камера «применила» к снимаемому сигналу. Затем с помощью метода деконволюции можно в значительной степени устранить эффект размытия. Иногда при съемке камера может вносить в изображение интерференцию – периодический муар, накладываемый на изображение. Часто оказывается, что спектр этой интерференции состоит из одной – двух гармоник. В этом случае ее можно эффективно удалить с помощью фильтра, который подавляет заданные частоты (notch filter).

Какова цель разложения сигналов в ряд Фурье? Преобразование сигнала в его Фурье-спектр – это переход в другую модель представления информации, закодированной в сигнале. Часто эта другая модель оказывается более простой для понимания природы сигнала. Например, многие звуковые сигналы состоят из сумм колебаний, близких к синусоидальным. Поэтому после преобразования Фурье они выглядят, как несколько пиков в спектре, хотя их форма волны до преобразования Фурье могла быть очень сложной для анализа. Более того, сама природа нашего слуха такова, что ухо раскладывает поступающие в него звуки на отдельные частоты. Звуковая волна, поступающая в ухо, преобразуется в колебания базилярной мембраны. Базилярная мембрана имеет разную жесткость по своей длине, и соответственно – разные собственные частоты колебаний разных участков. Когда на мембрану поступает сложное колебание, оно возбуждает колебания тех участков мембраны, которым соответствуют отдельные гармонические составляющие сложного входного колебания. Таким образом, разложение звука на синусоиды (преобразование Фурье) близко по своей природе к механизму функционирования нашего уха. Рассмотрим, некоторые «недостатки» и «несоответствия» преобразования Фурье. 1. Преобразование Фурье раскладывает сигнал по кратным частотам. Если мы раскладываем звук, то базисные частоты получаются, например, такие: 10 Гц, 20 Гц, ..., 300 Гц, 310 Гц, ..., 4000 Гц, 4010 Гц, ... Это не очень хорошо согласуется с нашим восприятием высоты звука. Наше ухо чувствительно не к абсолютным изменениям высоты (на сколько-то герц), а к относительным (на сколько-то процентов). Поэтому при анализе звука с помощью преобразования Фурье часто оказывается, что частотное разрешение спектра недостаточно на низких частотах и избыточно на высоких. Казалось бы, что в этом случае можно

повысить частотное разрешение спектра (увеличив размер FFT), но при этом мы будем анализировать более длинный по времени отрезок сигнала, и полученный спектр будет отражать усредненные свойства исходного сигнала в течение всего отрезка FFT. А такое усреднение по времени не всегда желательно. 2. Базисные функции преобразование Фурье имеют одну и ту же протяженность, как для высоких частот, так и для низких. Это не всегда удобно, например, при сжатии изображений. Вспомним, как работает алгоритм JPEG. Он разбивает изображение на блоки 8x8 пикселей и выполняет на каждом блоке преобразование Фурье (точнее, его разновидность, ДКП). После этого некоторые полученные амплитуды обнуляются (откидываются при кодировании). Чаще всего отбрасываются верхние частоты, т.к. они обычно содержат меньше энергии (их амплитуды меньше). При восстановлении производится обратный процесс: обратное преобразование Фурье. При этом обнуленные амплитуды высоких частот вызывают эффект Гиббса: пульсации яркости декодированного изображения вблизи резких границ в изображении. Визуально такой эффект очень нежелателен. Хотелось бы сделать так, чтобы эти 46 пульсации не расплзались на весь блок 8x8, а были локализованы в пространстве. Другими словами, хотелось бы, чтобы базисные функции, соответствующие высоким частотам в преобразовании Фурье, были короче (лучшая пространственная локализация), чем для низких частот. Чтобы преодолеть эти недостатки, можно пользоваться, например, таким приемом. Для анализа высоких частот использовать FFT с более коротким окном (лучшая локализация в пространстве), а для анализа низких частот – с более длинным окном (лучшее разрешение по частоте). Для анализа это хорошо, но вот для обработки, сжатия и синтеза – не очень. Для решения таких задач существует вейвлетное преобразование, к рассмотрению которого мы и переходим. Сначала рассмотрим простой пример – частный случай вейвлетного преобразования. Пусть у нас имеется одномерный сигнал четной длины $x[n]$. Выполним следующее преобразование. Свернем сигнал $x[n]$ с сигналом $h1[n]$, где сигнал $h1[n]$ состоит из двух единичных отсчетов ($h1[0] = 1, h1[1] = 1$). Эта операция эквивалентна вычислению сигнала $y1[n]$, который состоит из сумм соседних элементов $x[n]$. Так если $x[n] = \{2, 2, 3, 4, 6, 1, 6, 6\}$, то получается $y1[n] = \{2, 4, 5, 7, 10, 7, 7, 12, 6\}$ (не забываем, что свертка может расширять сигнал с концов). Здесь индексы $y1[n]$ идут от 0 до 8 (см. уравнение свертки). Сигнал $y1[n]$ – это некоторая сглаженная копия сигнала $x[n]$, т.к. фильтр $h1[n]$ можно рассматривать как примитивный НЧ-фильтр. Теперь рассмотрим сигнал $y2[n]$, получающийся сверткой $x[n]$ с ядром $h2[n]$, где $h2[0] = 1, h2[1] = -1$. Очевидно, такой «разностный» сигнал будет равен $y2[n] = \{2, 0, 1, 1, 2, -5, 5, 0, -6\}$. Легко видеть, что исходный сигнал $x[n]$ можно вычислить как полусумму сигналов $y1[n]$ и $y2[n]$. Действительно, это следует из свойств свертки и из выбора функций $h1[n]$ и $h2[n]$:
$$y1[n] + y2[n] = x[n] * h1[n] + x[n] * h2[n] = x[n] * (h1[n] + h2[n]) = x[n] * \delta[n] = x[n]$$
 Таким образом, можно считать сигнал $y1[n]$ неким грубым низкочастотным приближением $x[n]$, а сигнал $y2[n]$ – «уточняющим» сигналом, содержащим те

детали, которые были выброшены из $x[n]$ при НЧ-фильтрации. Теперь совершим прореживание сигналов $y_1[n]$ и $y_2[n]$ в 2 раза. Получим сигналы $z_1[n] = \{4, 7, 7, 12\}$, $z_2[n] = \{0, 1, -5, 0\}$. Оказывается, что исходный сигнал $x[n]$ можно восстановить не только из $y_1[n]$ и $y_2[n]$, но и из их прореженных вариантов $z_1[n]$ и $z_2[n]$. Для этого нужно совершить обратные операции. Сначала проводим интерполяцию нулями сигналов $z_1[n]$ и $z_2[n]$. Получаем $u_1[n] = \{0, 4, 0, 7, 0, 7, 0, 12\}$, $u_2[n] = \{0, 0, 0, 1, 0, -5, 0, 0\}$. После этого сворачиваем полученные сигналы с фильтрами $g_1[n] = h_1[-n]$ и $g_2[n] = h_2[-n]$ соответственно и суммируем результирующие сигналы. Получаем $\{4, 4, 7, 7, 7, 7, 12, 12\} + \{0, 0, -1, 1, 5, -5, 0, 0\} = 2 \cdot \{2, 2, 3, 4, 6, 1, 6, 6\}$, т.е. в точности исходный сигнал $x[n]$. Подытожим схему этого преобразования (см. рис. 17).

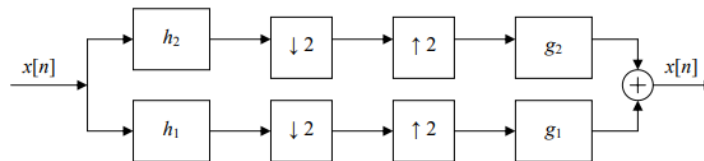


Рисунок 4.1 – Схема дискретного вейвлет-преобразования: разложение и синтез

Шаг 1: Считывание изображения.

На данном этапе приведем пример считывания яркостного полутонового изображения.

```
I=imread('cameraman.tif');  
figure;imshow(I); title('Исходное изображение');
```



Рисунок 4.2 – Исходное изображение

Шаг 2: Симуляция размытостей.

Проведем симулирование реалистических размытых изображений (например, в результате движения камеры или недостаточной фокусировки объектива). В этом примере симуляция размытостей на изображении

достигается в результате свертки гауссовского фильтра с исходным изображением. Для этого можно использовать также функцию `imfilter`. Гауссовский фильтр представляется функцией протяженности точки PSF.

```
PSF=fspecial('gaussian', 7, 10);
Blurred=imfilter(I, PSF, 'symmetric', 'conv');
figure; imshow(Blurred);title('Размытое изображение');
```



Рисунок 4.3 – Размытое изображение

Шаг 3: Восстановление размытого изображения с использованием функции протяженности точки с различными параметрами. Для иллюстрации того, насколько важно иметь информацию о параметрах истинной функции протяженности точки, приведем три варианта восстановления.

В первом варианте восстановления (J1 и P1) используется малоразмерный массив UNDERPSF. Иными словами, предполагается, что функция протяженности точки является именно такой. Размер массива UNDERPSF является на 4 пикселя меньше по каждой размерности относительно размеров истинной функции протяженности точки.

```
UNDERPSF=ones(size(PSF)-4);
[J1 P1]=deconvblind(Blurred, UNDERPSF);
figure; imshow(J1); title('Восстановление с малоразмерной PSF');
```



Рисунок 4.4 – Восстановление с малоразмерной PSF

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab восстановление изображений.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов по восстановлению изображения.

Отчет по лабораторной работе должен содержать:

- 1) название работы;

- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Для чего применяется восстановление изображений?
2. Для чего нужна операция преобразования яркости изображения?
3. Как в Matlab можно организовать преобразование яркости изображения?

ЛАБОРАТОРНАЯ РАБОТА 13

Обработка цветных изображений

Цель: научиться обрабатывать цветные изображения

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Существует очень много подходов к решению задачи пороговой обработки цветных изображений. В данном материале эту задачу будем решать через анализ интенсивности, насыщенности, цветового фона и нормированных цветовых составляющих.

Для этого сначала считаем и визуализируем некоторое цветное изображение.

```
L=imread('fabric.png');  
L=double(L)./255;  
figure, imshow(L);  
title('Исходное изображение');
```

Исходное изображение



Рисунок 5.1 – Исходное изображение

Теперь сформируем и визуализируем значения интенсивности исходного изображения и построим их гистограмму.

```
I=L(:,:,1)+L(:,:,2)+L(:,:,3);  
figure, subplot(211);imshow(I);title('Интенсивность');  
subplot(212); imhist(I);  
imwrite(I,'I.bmp');
```

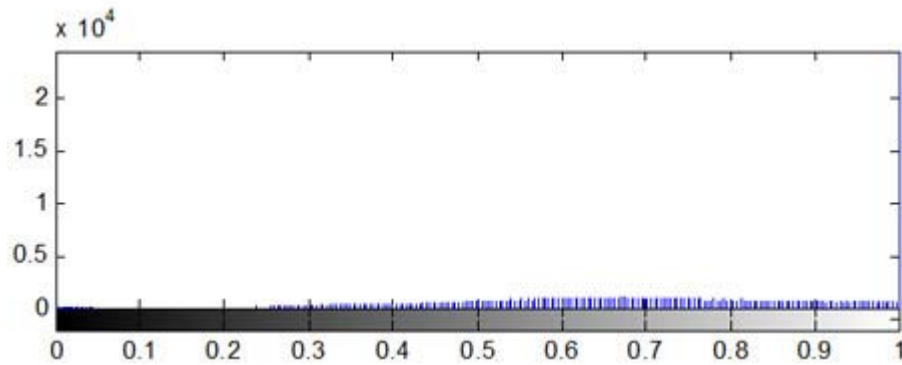
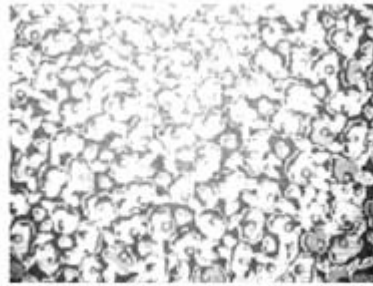
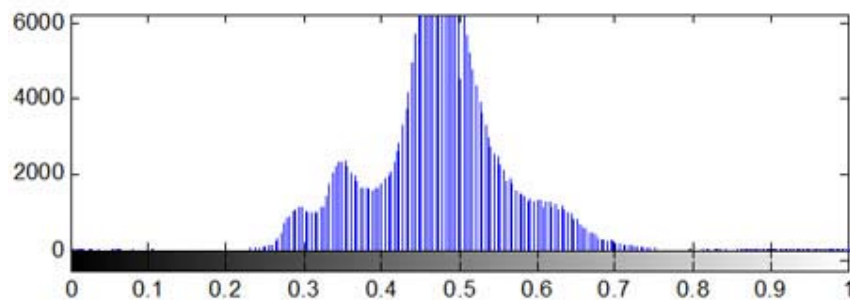
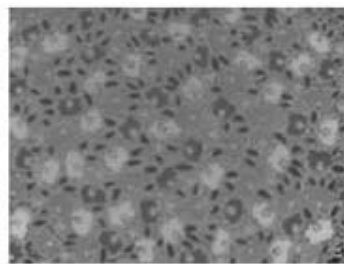


Рисунок 5.2 – Интенсивность изображения

Сформируем нормированное значение r составляющей изображения и визуализируем ее. Также построим гистограмму полученных данных.

```
r=L(:, :, 1)./(I+eps);  
figure, subplot(211);imshow(r);title('Составляющая r');  
subplot(212); imhist(r);  
imwrite(r,'r.bmp');
```

Выведем составляющие для цветного изображения
Составляющая r , рисунок 5.3.

Рисунок 5.3 – Составляющая r

Аналогично сформируем и визуализируем g цветовую составляющую исходного изображения и построим ее гистограмму.

```
g=L(:, :, 2)./(I+eps);  
figure, subplot(211);imshow(g);title('Составляющая g');  
subplot(212); imhist(g);  
imwrite(g,'g.bmp');
```

Составляющая g , рисунок 5.4.

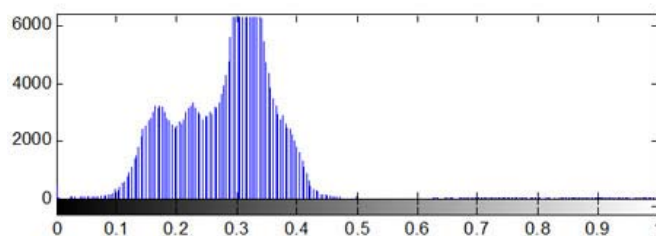
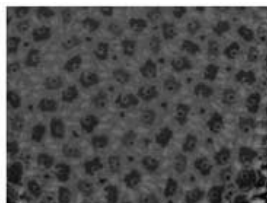


Рисунок 5.4 – Составляющая g

Далее сформируем и визуализируем b цветовую составляющую исходного изображения и построим ее гистограмму.

```
b=L(:, :, 3)./(I+eps);  
figure, subplot(211);imshow(b);title('Составляющая b');  
subplot(212); imhist(b);  
imwrite(b,'g.bmp');
```

Составляющая b

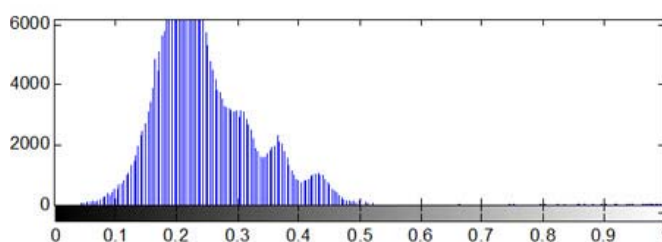
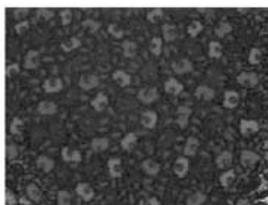


Рисунок 5.5 – Составляющая b

Далее на основе анализа гистограммы полученных данных выберем пороговое значение и проведем бинаризацию изображения:

1) Для изображения значений интенсивности исходного изображения. Пороговое значение равно среднеарифметическому значению уровней интенсивности.

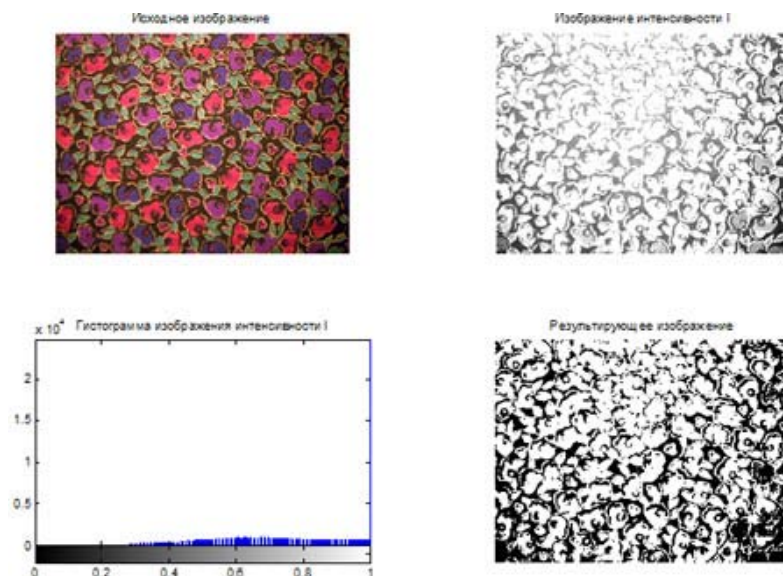


Рисунок 5.6 – Находим пороговое изображение

2) Для r цветовой составляющей изображения. Гистограмма r цветовой составляющей изображения имеет несколько ярко выраженных мод, которые и будут определять пороговые значения для обработки этого изображения. Это позволит выделить несколько типов объектов.

а) Пороговое значение определяется диапазоном $[0, 0.31)$.

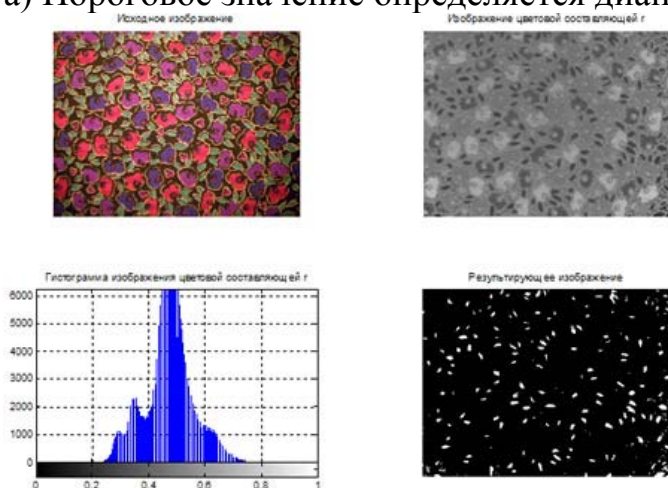


Рисунок 5.7 – Пороговое значение для другой компоненты

Бывают случаи, когда для выделения объектов на изображении необходимо использовать все или несколько параметров (значения цветовых составляющих, интенсивность, насыщенность и значения фона). Приведем

пример программы, когда для выделения объектов (например, зеленого листа) использовались все параметры.

```
clear;
L=imread('fabric.png');
L=double(L)./255;
figure,imshow(L);title('Исходное изображение');
R=imread('r.bmp');
R=R(:,:,1);
R=double(R)./255;
G=imread('g.bmp');
G=G(:,:,1);
G=double(G)./255;
B=imread('b.bmp');
B=B(:,:,1);
B=double(B)./255;
F=imread('F.bmp');
F=F(:,:,1);
F=double(F)./255;
I=imread('I.bmp');
I=I(:,:,1);
I=double(I)./255;
S=imread('S.bmp');
S=S(:,:,1);
S=double(S)./255;
[N M]=size(R);
%Lser=mean(mean(R));
for i=1:N;
    disp(i);
    for j=1:M;
        if ((R(i,j)-0)<.4)&((G(i,j)-0)>.35)&((B(i,j)-0)<.4)&
            ((F(i,j)-0)>.1)&((S(i,j)-0)>.1)&((I(i,j)-0)>.1);
            L(i,j,1)=0;L(i,j,2)=1;L(i,j,3)=0;
        end;
    end;
end;
figure,imshow(L);title('Результирующее изображение');
```

Результирующее изображение

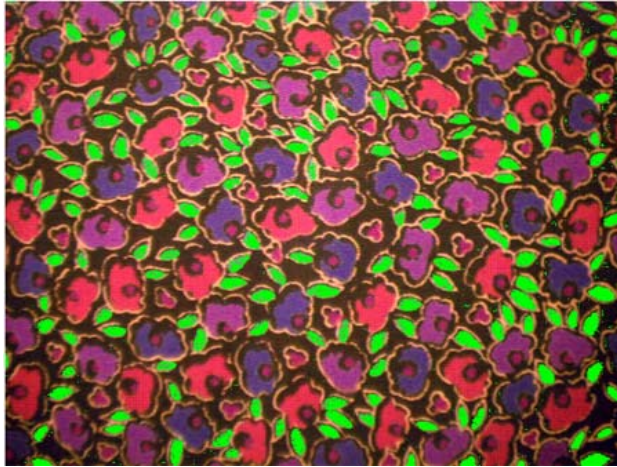


Рисунок 5.8 – Результирующее изображение

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab обработку цветных изображений.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов по обработке цветных изображений.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;

5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое преобразование яркости изображения?
2. Чем кодировка цветного изображения отличается от кодировки черно-белого изображения?
3. Какие команды Matlab для обработки цветных изображений вы знаете?

ЛАБОРАТОРНАЯ РАБОТА 14

Сжатие изображений

Цель: научиться с помощью Matlab сжимать изображения

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Преобразование Фурье

При использовании преобразований Фурье изображение представляется в виде суммы сложных показательных функций переменных амплитуды, частоты и фазы. Преобразование Фурье играет очень важную роль во многих областях обработки изображений, включая улучшение, анализ, восстановление и сжатие.

Далее рассмотрим следующие вопросы:

1. Основные определения преобразования Фурье
2. Дискретное преобразование Фурье, включая быстрое преобразование Фурье
3. Применение Фурье-преобразования (некоторые примеры практического применения преобразования Фурье)

Основные определения преобразования Фурье

Если $f(m, n)$ представляет собой функцию двух дискретных пространственных переменных m и n , тогда двумерное преобразование Фурье функции $f(m, n)$ может быть представлено следующим выражением

$$F(\omega_1, \omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) e^{-j\omega_1 m} e^{-j\omega_2 n} \quad (6.1)$$

Переменные ω_1 и ω_2 представляют собой угловые частоты. Таким образом, $F(\omega_1, \omega_2)$ представляет собой функцию $f(m, n)$ в частотной области. $F(\omega_1, \omega_2)$ является комплекснозначной функцией с соответствующими частотами ω_1 и ω_2 . Частоты находятся в пределах диапазона $-\pi \leq \omega_1, \omega_2 \leq \pi$. Отметим, что $F(0, 0)$ представляется в виде суммы всех переменных $f(m, n)$. По этой причине $F(0, 0)$ часто называют постоянной составляющей преобразования Фурье.

Обратное двумерное преобразование Фурье представляется выражением

$$f(m, n) = \frac{1}{4\pi^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} F(\omega_1, \omega_2) e^{j\omega_1 m} e^{j\omega_2 n} d\omega_1 d\omega_2 \quad (6.2)$$

Т.е. это выражение представляет $f(m, n)$ в виде суммы бесконечного числа сложных экспоненциальных функций (синусоид) с различными частотами. Амплитуда и фаза определяют вклад частот (ω_1, ω_2) в

представление $F(\omega_1, \omega_2)$.

Визуализация Фурье-преобразования

При иллюстрации Фурье-преобразования допустим, что функция $f(m, n)$ равна 1 и представлена в виде прямоугольника. Для упрощения диаграммы, функция $f(m, n)$ будет представляться непрерывной функцией двух дискретных переменных m и n .

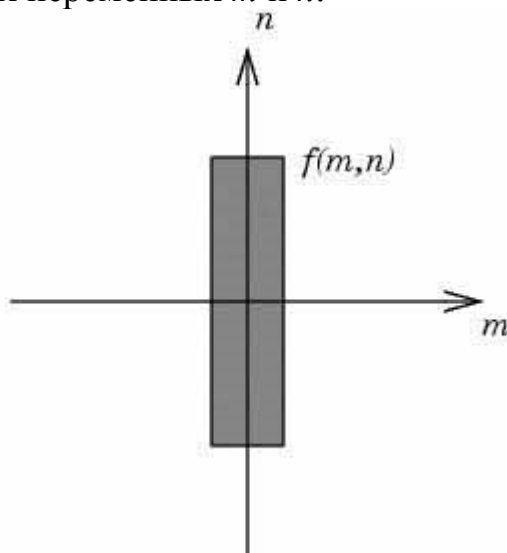


Рисунок 6.1 – Прямоугольная функция

На рисунке внизу с использованием функции mesh визуализировано значения амплитуд, которые получены при Фурье-преобразовании $F(\omega_1, \omega_2)$ прямоугольной функции, представленной на предыдущем рисунке. Визуализацию амплитуды еще называют визуализацией преобразований Фурье.

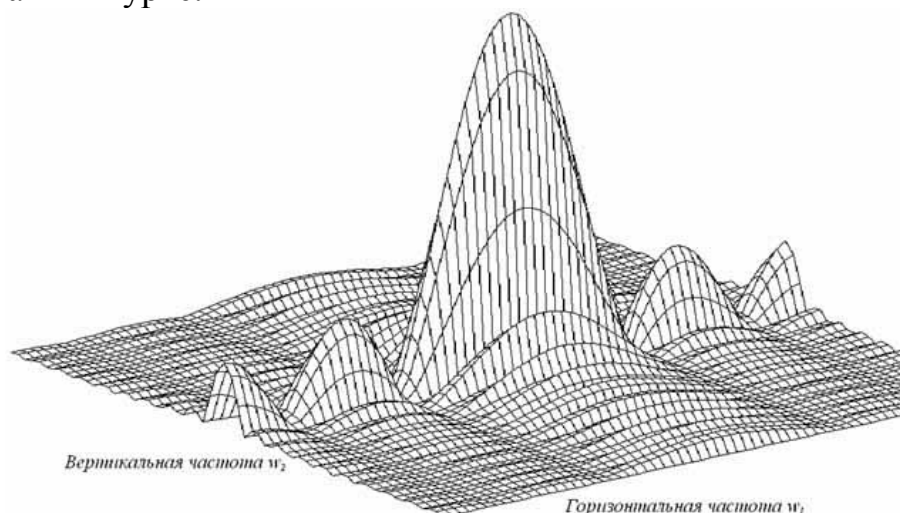


Рисунок 6.2 – Амплитуда изображения прямоугольной функции

Пик функции находится в центре и отображает значение $F(0,0)$, которое является суммой всех значений $f(m, n)$. Все остальные

составляющие $F(\omega_1, \omega_2)$ представляют собой распределение энергии по вертикальным и горизонтальным частотам.

Другой путь визуализации Фурье-преобразования заключается в отображении значений $\log |F(\omega_1, \omega_2)|$ в виде изображения.

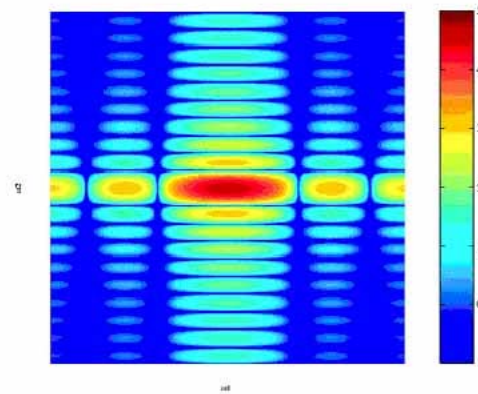


Рисунок 6.3 – Логарифмическое представление Фурье-преобразования прямоугольной функции

Рассмотрим пример вычисления двумерных дискретных косинусных преобразований в блоках с размерами 8×8 элементов исходного изображения. Далее при реконструкции изображения будем учитывать только 10 коэффициентов из каждого блока, остальные приравняем к нулю. При проведении описанных вычислений будет использоваться также матрица преобразований.

```
I = imread('cameraman.tif');
I = im2double(I);
T = dctmtx(8);
B = blkproc(I,[8 8],'P1*x*P2',T,T);
mask = [1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0];
B2 = blkproc(B,[8 8],'P1.*x',mask);
I2 = blkproc(B2,[8 8],'P1*x*P2',T',T);
imshow(I);
figure, imshow(I2)
```



Рисунок 6.4 – Реконструкция изображения после сжатия

На рисунке представлено два изображения – исходное и реконструированное. При реконструкции изображения использовалось только 15 % коэффициентов дискретных косинусных преобразований. Однако, следует отметить, что качество реконструированного изображения является довольно приемлемым. Для просмотра других свойств дискретного косинусного преобразования см. функцию `dctdemo`.

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab организовать сжатие изображений.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов по работе со сжатием изображений.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Для чего применяется сжатие изображений?
2. Какие проблемы возникают при сжатие изображений?
3. Какие команды и утилиты Matlab вы знаете для сжатия изображений?

ЛАБОРАТОРНАЯ РАБОТА 15

Сегментация изображений

Цель: научиться производить сегментацию изображений

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Шаг1: Считывание изображения.

Считаем файл hestain.png, который содержит изображение гематоксилина и эозина (H&E). Здесь применен метод окрашивания для детального анализа патологий.

```
he=imread('hestain.png');
imshow(he), title('H&E изображение');
text(size(he, 2),size(he, 1)+15,...
      'Image courtesy of Alan Partin, Johns Hopkins University', ...
      'FontSize',7,'HorizontalAlignment','right');
```

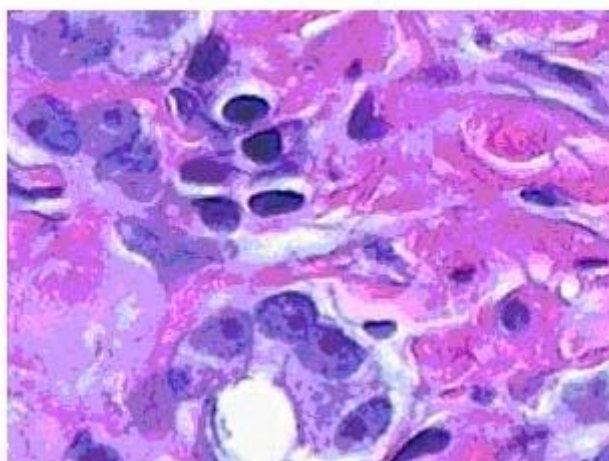


Рисунок 7.1 – Исходное изображение

Шаг 2: Преобразование изображения из цветовой системы RGB в цветовую систему $L^*a^*b^*$.

Какое количество цветов видно на изображении, когда не принимать во внимание возможность комбинации яркостей? На самом деле их три: белый, голубой и розовый. Следует отметить различия этих цветов между собой. Цветовое пространство $L^*a^*b^*$ (оно еще известно как CIELAB или CIE $L^*a^*b^*$) дает возможность различать эти визуальные различия.

Цветовое пространство $L^*a^*b^*$ получено на основе трехцветных значений CIE XYZ. Пространство $L^*a^*b^*$ включает информацию о значении интенсивности ' L^* ', значении цветности ' a^* ', которое показывает какой цвет выбран на красно-зеленой оси и значение цветности ' b^* ' показывает какой цвет выбран на голубо-желтой оси. Вся информация о цветах содержится в значениях ' a^* ' и ' b^* '. Оценить разницу между двумя цветами можно с

использованием евклидового расстояния.

Преобразуем изображение в цветное пространство $L^*a^*b^*$ с использованием функций `makecform` и `applycform`.

```
cform=makecform('srgb2lab');  
lab_he=applycform(he, cform);
```

Шаг 3: Классификация цветов в пространстве a^*b^* с использованием кластеризации (метод k-средних).

Кластеризация приводит до деления объектов на группы. Кластеризация методом k-средних приводит также к локализации объектов в пространстве. Поиск деления, т.е. какой объект к какому классу принадлежит, происходит на основе анализа метрического расстояния между объектами.

Далее на основании информации о цветах в пространстве a^*b^* , каждому пикселю объекта присваивается значение a^* и b^* . Используем кластеризацию методом k-средних для деления объектов на три кластера. Для этого используем евклидовую метрику.

```
ab=double(lab_he(:, :, 2:3));  
nrows=size(ab, 1);  
ncols=size(ab, 2);  
ab=reshape(ab, nrows*ncols, 2);  
nColors=3;  
[cluster_idx    cluster_center]=kmeans(ab,    nColors,    'distance',  
'sqEuclidean', ...  
                                     'Replicates',3);
```

Шаг 4: Присвоение меток каждому пикселю изображения на основе метода k-средних.

Для каждого объекта на исходном изображении метод k-средних возвращает индекс соответствующего кластера. Значение параметра `cluster_center`, которое получено в результате применения метода k-средних будет использовано при дальнейшей демонстрации метода. Отметим пиксели, которые содержатся в `cluster_index`.

```
pixel_labels=reshape(cluster_idx, nrows,ncols);  
imshow(pixel_labels, []), title('изображение, отмеченное кластерными  
индексами');
```

изображение, отмеченное кластерными индексами

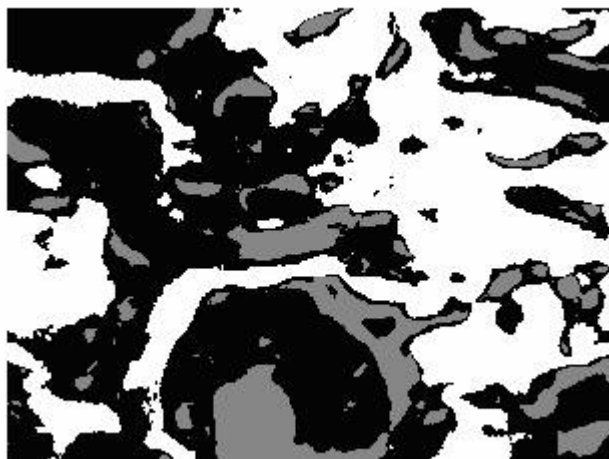


Рисунок 7.2 – Изображение, отмеченное кластерными индексами

Шаг 5: Создание сегментированного изображения на основе цветного.

Используя параметр `pixel_labels`, можно разделить объекты на изображении `hestain.png` по цветам.

```
segmented_images=cell(1, 3);
rgb_label=repmat(pixel_labels, [1 1 3]);
for k=1 : nColors
    color=he;
    color(rgb_label~=k)=0;
    segmented_images{k}=color;
end
imshow(segmented_images{1}), title('объекты в кластере 1');
```

объекты в кластере 1

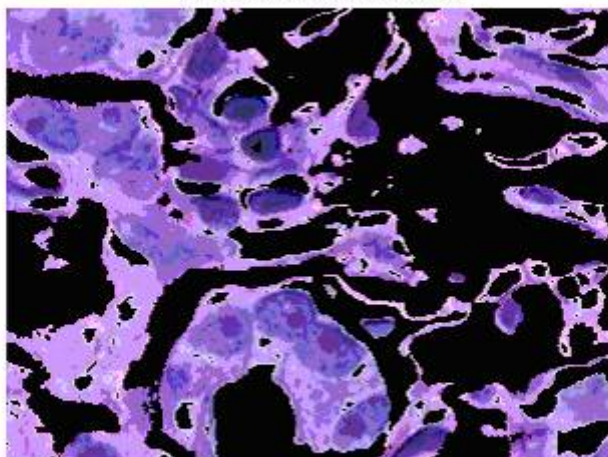


Рисунок 7.3 – Объекты в кластере 1

```
imshow(segmented_images{2}), title('объекты в кластере 2');
```

объекты в кластере 2

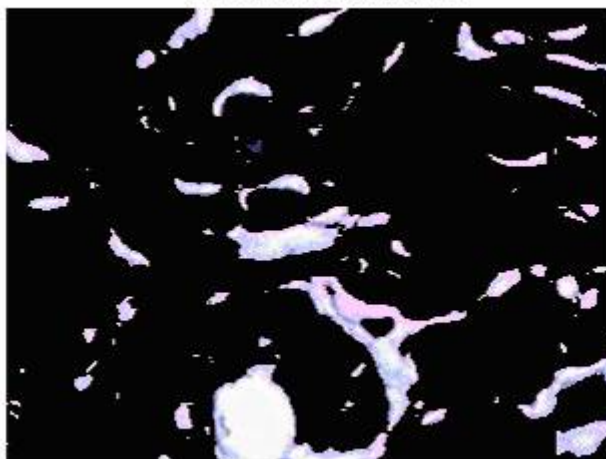


Рисунок 7.4 – Объекты в кластере 2

```
mshow(segmented_images{3}), title('объекты в кластере 3');
```

объекты в кластере 3

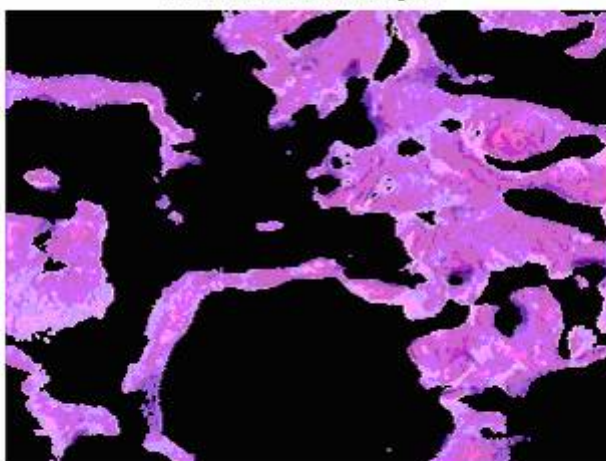


Рисунок 7.5 – Объекты в кластере 3

Шаг 6: Сегментация ядер на основании отдельного изображения.

Рассмотрим изображение, которое содержит синие объекты. Отметим, что они являются темно-синими и светло-синими. Используя значение 'L*' в цветовом пространстве $L^*a^*b^*$ можно отделить темно-синие объекты от светло-синих.

Напомним, что параметр 'L*' содержит значения интенсивностей для каждого цвета. Найдем кластеры, которые содержат синие объекты. Получим значения интенсивностей объектов в этом кластере и обработаем их пороговым методом с использованием функции `im2bw`.

Параметр `cluster_center` содержит среднее значение 'a*' и 'b*' для каждого кластера. Синий кластер имеет второе наибольшее значение `cluster_center`.

```
mean_cluster_val=zeros(3, 1);  
for k=1:nColors
```

```

    mean_cluster_val(k)=mean(cluster_center(k));
end
[mean_cluster_val,idx]=sort(mean_cluster_val);
blue_cluster_num=idx(2);
L=lab_he(:, :, 1);
blue_idx=find(pixel_labels==blue_cluster_num);
L_blue=L(blue_idx);
is_light_blue=im2bw(L_blue,graythresh(L_blue));

```

Используем маску is_light_blue для того, чтобы пометить те пиксели, которые являются частью синего ядра. Отообразим синие ядра на разделенном изображении.

```

nuclei_labels=repmat(uint8(0), [nrows ncols]);
nuclei_labels(blue_idx(is_light_blue==false))=1;
nuclei_labels=repmat(nuclei_labels, [1 1 3]);
blue_nuclei=he;
blue_nuclei(nuclei_labels~=1)=0;
imshow(blue_nuclei), title('синие ядра');

```

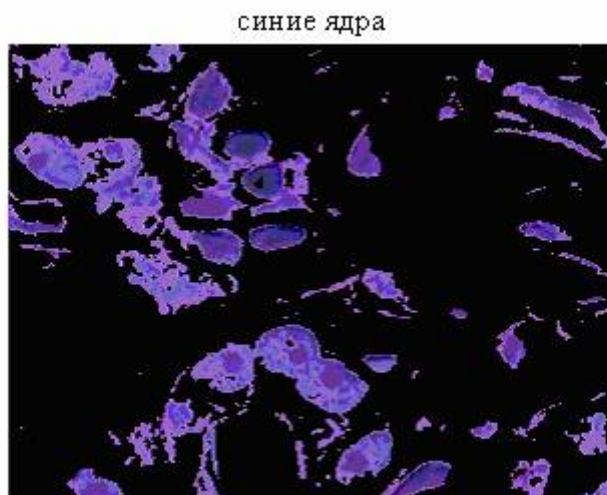


Рисунок 7.6 – Синие ядра

Аппаратура и материалы. 64-разрядный (x64) персональный компьютер, процессор с тактовой частотой 1 ГГц и выше, оперативная память 1 Гб и выше, свободное дисковое пространство не менее 1 Гб, графическое устройство DirectX 9. Программное обеспечение: операционная система Windows 7 и выше, Matlab (R2013) и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае

обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab работу по сегментации изображений.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен состоять из описания основных этапов сегментации изображений в Matlab.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое преобразование сегментация изображения?
2. Какие алгоритмы сегментации изображений вы знаете?
3. Для чего нужна сегментация изображений?
4. Как в Matlab через утилиту организовать сегментацию изображений?

ЛАБОРАТОРНАЯ РАБОТА 16

Коррекция неравномерной засветки изображения

Цель: научиться корректировать неравномерную засветку изображения.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Шаг 1: Считывание изображения

```
I=imread('rice.png');  
imshow(I)
```

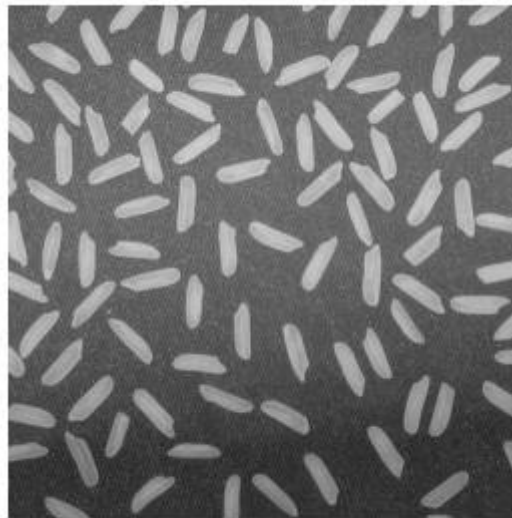


Рисунок 8.1 – Исходное изображение

Шаг 2: Использование морфологической операции раскрытия для оценки фона

Отметим, что яркость фона является различной в разных частях изображения. Используем функцию `IMOPEN` для анализа (оценки) яркости фона.

```
background=imopen(I, strel('disk', 15));  
% Отобразим поверхность фона  
figure, surf(double(background(1:8:end, 1:8:end))), zlim([0 255]);  
set(gca, 'ydir', 'reverse');
```

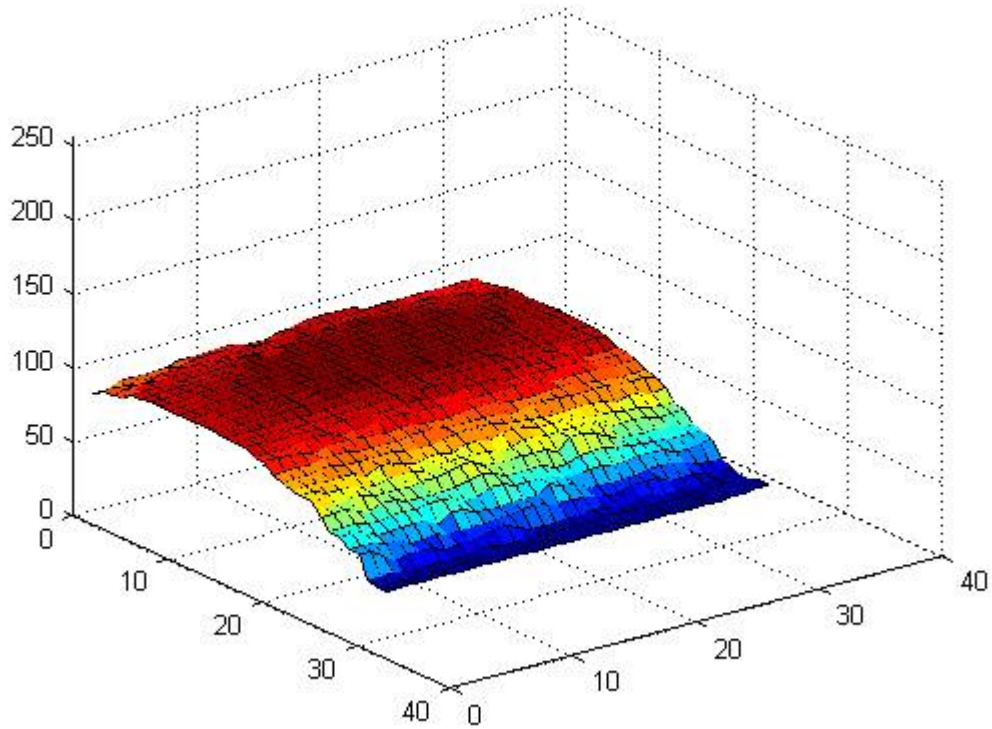


Рисунок 8.2 – Поверхность фона

Шаг 3: Извлечение фона из исходного изображения

Таким образом, у нас имеется изображение и его фон в формате uint8. Далее, используя функцию IMSUBTRACT, проведем операцию извлечения фона из изображения.

```
I2=imsubtract(I, background);  
imshow(I2)
```

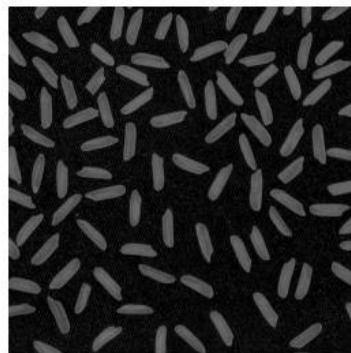


Рисунок 8.3 – Извлечение фона из изображения

Шаг 4: Увеличение контрастности изображения

```
I3=imadjust(I2);  
imshow(I3);
```



Рисунок 8.4 – Увеличение контрастности

Шаг 5: Пороговая обработка изображения

Создадим бинарное изображение на основе данного.

```
level=graythresh(I3);
bw=im2bw(I3, level);
imshow(bw)
```

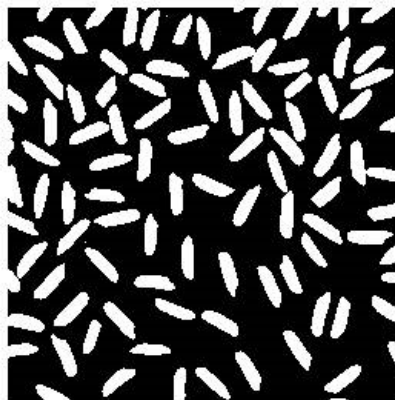


Рисунок 8.5 – Создание бинарного изображения

Шаг 6: Присвоение меток объектам изображения

Функция BWLABEL отмечает все связные компоненты на бинарном изображении. Точность результата зависит от размеров объектов, оценки фона. Проводится анализ отмеченных объектов, поскольку на изображении некоторые зерна риса соприкасаются.

```
[labeled, numObjects]=bwlabel(bw, 4); % 4-связные
numObjects % Число всех отдельных объектов на изображении.
numObjects=
```

101

Шаг 7: Исследование матрицы меток

Каждый отдельный объект является отмеченным некоторым числом. Например, большая часть зерен отмечена числом 50.

```
rect=[105 125 10 10];
```

```
grain=imcrop(labeled, rect)
```

```
grain=
```

0	0	50	50	50	50	50	50	50	50	0
0	0	50	50	50	50	50	50	50	50	0
0	0	50	50	50	50	50	50	50	50	0
0	0	50	50	50	50	50	50	50	50	0
0	0	50	50	50	50	50	50	50	50	0
0	0	50	50	50	50	50	50	50	50	0
0	0	50	50	50	50	50	50	50	50	0
0	0	0	50	50	50	50	50	50	50	0
0	0	0	50	50	50	50	50	50	0	0
0	0	0	50	50	50	50	50	50	0	0
0	0	0	0	50	50	50	0	0	0	0

Одним из путей просмотра матрицы меток является ее отображение в виде псевдоцветного индексного изображения. На этом изображении числа, которые идентифицируют каждый объект в матрице меток, отображаются различными цветами. Используем функцию LABEL2RGB для выбора палитры, цвета фона и цветов объектов в матрице меток.

```
RGB_label=label2rgb(labeled, @spring, 'c', 'shuffle');  
imshow(RGB_label)
```

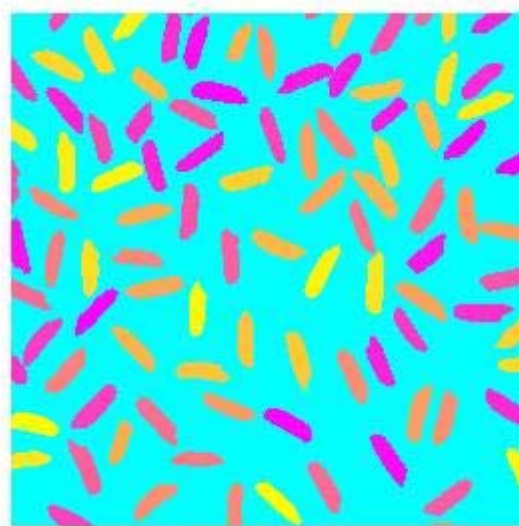


Рисунок 8.6 – Использование палитры для выбора объектов

Шаг 8: Измерение характеристик объектов на изображении

Функция REGIONPROPS проводит анализ объектов или характеристик

окрестности на изображении и возвращает результаты в виде структур массива. При обработке изображения с отмеченными компонентами, для каждого объекта создается свой структурный элемент. Используем функцию `regionprops` для создания массива структур, содержащего некоторые основные свойства отмеченных объектов изображения.

```
graindata=regionprops(labeled, 'basic')
```

```
graindata=
```

```
graindata(50).Area
```

```
ans =
```

```
203
```

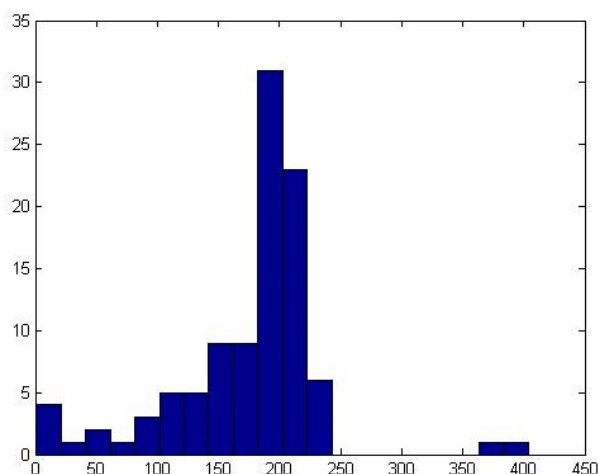


Рисунок 8.7 – Извлекаем отмеченные свойства объектов

Аппаратура и материалы Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Исходя из индивидуального задания аспиранта по кандидатской диссертации организовать с помощью среды Matlab организовать представление и описание изображений.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий. Отчет по лабораторной работе должен состоять из описания основных этапов представления и описания изображений в Matlab.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Для чего нужно представление и описание изображений?
2. Как в Matlab организовать представление и описание изображений?

ЛАБОРАТОРНАЯ РАБОТА 17

Формирование цифровых латентных изображений

Цель: изучить методику формирования цифровых латентных изображений, приобрести навыки создания цифровых латентных изображений в системе компьютерной математики MATLAB.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Латентные изображения это изображения, которые обладают свойством изменения видимости элементов изображения при изменении условий наблюдения или способа регистрации основного изображения.

В настоящее время латентные изображения широко применяются в различных сферах для защиты документов от копирования. Вместе с тем не существует универсального способа выявления латентных изображений, в связи, с чем затруднен и, как следствие, является дорогостоящим процесс отличия копии от подделки. Поэтому разработка и автоматизация методов контроля скрытой информации в латентных изображениях является актуальной задачей, решение которой позволит сделать более дешевыми, точными и качественными процессы контроля подлинности документов, использующих в качестве защиты латентные изображения[6].

К современным методам формирования цифровых латентных изображений относятся методики Шевелева А.А.[18] и Топчиева И.Н.[16]

Способ формирования латентного изображения, предложенный Шевелевым А.А., заключается в том, что основное изображение дважды растривается с использованием двух растровых структур. Из растриванных основных изображений удаляются части изображения с помощью позитивной и негативной масок, которые получают из внедряемого изображения. Затем фрагменты полученных изображений объединяют в единое целое. Для того чтобы качество латентного изображения, полученного этим способом, не отличалось от основного изображения необходимо, чтобы это изображение имело высокое разрешение, не менее 8000×6000 dpi.

Методика, предложенная Топчиевым И.Н., в отличие от способа Шевелева предполагает формировать латентное изображение путем растривания не основного изображения, а внедряемого:

$$\begin{cases} L(x, y) = k_1 \cdot I(x, y) + k_2 \cdot S_{adj}(x, y); \\ S_{adj}(x, y) = a \cdot S(x, y) + b; \\ S(x, y) = k_1 \cdot G_{inv}(x, y) + k_2 \cdot G_r(x, y); \\ G_{inv}(x, y) = (2^L - 1) - G(x, y); \\ G_r(x, y) = k_1 \cdot G(x, y) + k_2 \cdot R_i(x, y), \end{cases} \quad (19.1)$$

где $L(x, y)$, $I(x, y)$, $G(x, y)$ – числовые массивы, соответствующие латентному, основному и внедряемому изображениям соответственно;

$G_r(x, y)$, $G_{inv}(x, y)$ – числовые массивы растрированного внедряемого и инвертированного внедряемого изображений;

$S(x, y)$ – числовой массив, полученный в результате объединения инвертированного и растрированного изображений;

$S_{adj}(x, y)$ – числовой массив, полученный после яркостных преобразований изображения $S(x, y)$;

k_1, k_2 – неотрицательные константы; L – число разрядов квантования, $L = 8$; a и b – определяются желаемыми значениями яркости $S_{adj}(x, y)$:

$$a = (S_{adj}(x, y)_{\max} - S_{adj}(x, y)_{\min}) / (S(x, y)_{\max} - S(x, y)_{\min});$$

$$b = (S_{adj}(x, y)_{\min} \cdot S(x, y)_{\max} - S_{adj}(x, y)_{\max} \cdot S(x, y)_{\min}) / (S(x, y)_{\max} - S(x, y)_{\min})$$

и при $S_{adj}(x, y)_{\min} = 0$, $S_{adj}(x, y)_{\max} = 255$, $a = 2^L - 1 / S(x, y)_{\max} - S(x, y)_{\min}$ и $b = (1 - 2^L) \cdot S(x, y)_{\min} / S(x, y)_{\max} - S(x, y)_{\min}$.

Основное, скрываемое и изображение растровых структур – это числовые массивы одного класса и одинаковых размеров. Предложенная модель позволяет формировать латентные изображения с помощью стандартного программного обеспечения. Формирование латентных изображений с помощью модели (19.1) реализовано в системе компьютерной математики Matlab и графическом редакторе AdobePhotoshop. Методика формирования латентного изображения заключается в выполнении последовательности действий в соответствии с приведенной схемой (рисунок 3.3):



Рисунок 19.1 – Схема процесса формирования латентного изображения

- 1) Инвертирование скрываемого изображения.
- 2) Растривание дублированного скрываемого изображения растровыми структурами.
- 3) Объединение инвертированного скрываемого и растриванного изображений.
- 4) Яркостные преобразования изображения $S_{adj}(x, y)$.
- 5) Получение латентного изображения путем объединения основного изображения и изображения с измененным контрастом.

Рассмотрим способ реализации математической модели формирования латентного изображения с помощью системы компьютерной математики Matlab.

Для формирования латентного изображения необходимо выполнить следующую последовательность действий:

1. Импортировать в рабочую область Matlab необходимые для формирования латентного изображения данные: « I » – основное изображение $I(x, y)$; « G » – скрываемое изображение $G(x, y)$; « R_i » – растровые структуры $R_i(x, y)$, где $i = \overline{1, n}$, с помощью функции: `imread('filename')`, где: «*filename*» – строка символов имени загружаемого изображения (рисунок 19.2). В рассматриваемом примере скрываемое изображение растривается только одной растровой структурой $R_1(x, y)$, которая изображена на рисунке 19.2в. Основное и скрываемое изображение – это числовые массивы одного класса и одинакового размера. К общему размеру изображения можно привести с помощью функции `imresize(I, m, method)`, которая создает новое изображение, с размерами в m раз отличающихся от размеров исходного изображения « I ». Параметр «*method*» предназначен для изменения размеров по одному из предопределенных методов интерполяции (билинейная или бикубическая).

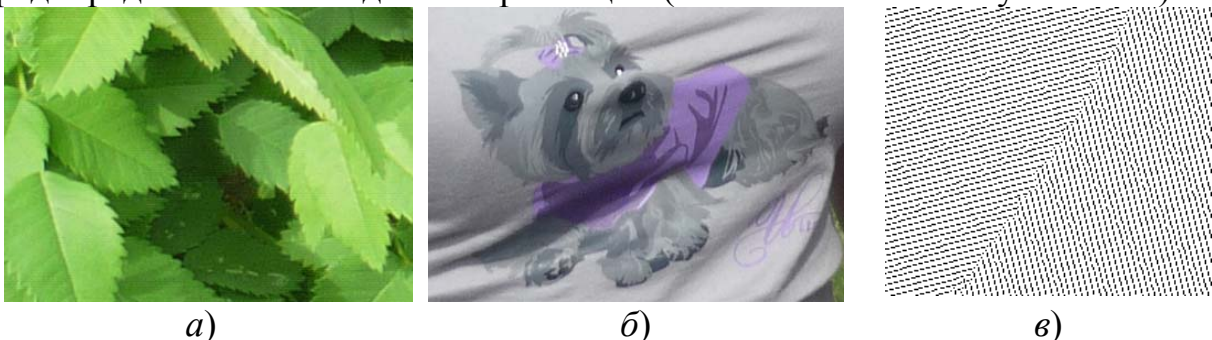


Рисунок 19.2 – Исходные изображения: а) – основное « I », б) – скрываемое « G », в) – растровая структура « R_i »

2. Сформировать на основе скрываемого изображения « G » его дубликат, присвоив новому массиву дублированного изображения « $GCopу$ » значения массива скрываемого изображения $G(x, y)$, с помощью операции присваивания $GCopу = G$.

3. Сформировать на основе массива скрываемого изображения « G » новый числовой массив, соответствующий его инвертированной копии « $GInvert$ », с помощью операции: $GInvert = 255 - G$.

4. Провести растривание дублированного скрываемого изображения « $GCopy$ » растровой структурой $R_1 = R_1(x, y)$ с помощью функции вычисления линейной комбинации « $imlincomb$ », для получения числового массива растриванного дублированного скрываемого изображения « $GCopy_R$ » = $G_r(x, y)$, с помощью операции: $GCopy_R = imlincomb(0.5, GCopy, 0.5, R_1)$.

5. Объединить числовые матрицы « $GInvert$ » и « $GCopy_R$ », соответствующие инвертированной копии и растриванному изображению в массив « SCR_SN » = $S(x, y)$, с помощью функции вычисления линейной комбинации изображений: $SCR_SN = imlincomb(0.5, GCopy_R, 0.5, GInvert)$.

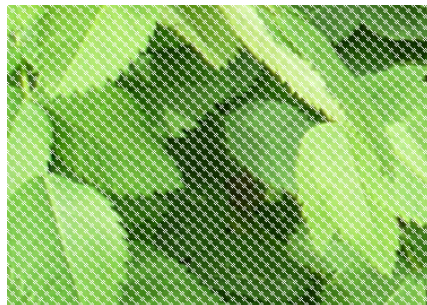
6. Изменить параметры яркости изображения « $SCR_SN_Lightness$ » = $S_{adj}(x, y)$, с помощью функции преобразования яркости изображения $SCR_SN_Lightness = imadjust(SCR_SN, [0.4 \ 0.5], [0.4 \ 0.5])$, которая формирует новое изображение, со значениями яркости в интервале 0,4 – 0,5.

7. Объединить основное изображение « I » и изображение с измененными значениями контраста « $SCR_SN_Lightness$ » в новый массив, соответствующий изображению « $Latent$ » = $L(x, y)$ с помощью функции линейной комбинации изображений: $Latent = imlincomb(0.25, SCR_SN_Lightness, 1, I)$.

На рисунке 19.3 представлено полученное изображение $L(x, y)$, содержащее скрытую информацию.



а)



б)

Рисунок 19.3 – Полученные латентные изображения с помощью методики Топчиева (а) и методики Шевелева (б)

Как видно из представленного рисунка 19.3 изображения характеризуются незаметностью внедренной информации.

Для оценки визуального качества латентного изображения использован известный эмпирический подход, согласно которому в качестве параметров изображений использовались: среднеарифметическое значение $\bar{L}$ яркостей, полнота использования градаций яркостей, резкость изображения и его обобщенный контраст. Проведенный анализ изображений показал, что для оценки визуального качества достаточными являются два параметра оценки

уровня адаптации зрительной системы и оценка степени использования возможных градаций яркостей, поэтому расчет интегрального критерия визуального качества осуществлялся с помощью выражения:

$$Q = k \cdot LQ \cdot KQ, \quad (19.2)$$

где k – нормирующий коэффициент;
 LQ – оценка уровня адаптации зрительной системы;
 KQ – аналитическая оценка степени использования возможных градаций яркостей.

$$LQ = \frac{\bar{L} - LMAX / 2}{LMAX / 2}, \quad (19.3)$$

где $LMAX / 2$ – половина максимально возможного диапазона яркостей.

$$KQ = \frac{S}{LMAX}, \quad (19.4)$$

где S – количество уровней яркостей, для каждого из которых на данном изображении присутствуют больше чем $b \times N \times M$ количество элементов с данной яркостью (N и M – размеры изображения, b – некоторая константа).

Полученные результаты расчета интегрального критерия для исходного, а также для сформированных латентных изображений известным способом и предложенной методикой соответственно равен: 0,79; 0,89; 0,77, что позволяет сделать вывод о том, что качество полученного латентного изображения с помощью методики Топчиева лучше, чем качество изображения, полученного с помощью способа Шевелева.

Для того чтобы качество латентного изображения, полученного способом Шевелева, не отличалось от оригинала необходимо использовать основное изображение высокого разрешения: 8000×6000 dpi, в то время как для формирования латентного изображения на основе модели Топчиева достаточно разрешения 400×300 dpi.

Предложенный способ позволяет формировать латентные изображения с высокой степенью защиты от подделки с помощью стандартного, промышленно используемого программного обеспечения. Высокая степень защиты обусловлена тем, что часть кодированной информации латентного изображения теряется при сканировании и обработке (изменение геометрических параметров изображений, например, разрешения, размера)[16].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Machcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 19.1. Сформировать латентное изображение, выбрав изображения самостоятельно.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Какие современные методы формирования цифровых латентных изображений существуют?
2. В чем заключается методика предложенная Шевелевым А.А.?
3. В чем заключается методика предложенная Топчиевым И.Н.?

ЛАБОРАТОРНАЯ РАБОТА 18

Методы выявления скрытой информации в латентных изображениях

Цель: изучить методику выявления скрытой информации в латентных изображениях, приобрести навыки выявления скрытого изображения в системе компьютерной математики MATLAB.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Для выявления латентного изображения используются оптические методы неразрушающего контроля, основанные на способности скрытого изображения, становится видимым в косопадющем свете, ультрафиолетовых, инфракрасных, лазерных лучах, а также методы, основанные на цифровой фильтрации. Выявление латентного изображения на основе Фурье-преобразования и фильтрации, дает положительные результаты, если в обрабатываемом изображении сохранилась растровая структура, для чего необходимо иметь разрешение изображения не менее 1200 dpi. При таком разрешении изображения операции прямого и обратного преобразования Фурье требуют больших вычислительных ресурсов компьютера.

Одним из перспективных методов выявления скрытого изображения в латентном является вейвлет-разложение.

В основе вейвлет-анализа сигнала лежит идея разложения сигнала на низкочастотную и высокочастотную составляющие, поэтому вейвлет-преобразование латентного изображения можно использовать для выявления скрытой информации.

Средства среды Matlab, представленные пакетом WaveletToolbox, позволяют анализировать изображения с помощью вейвлетов Хаара (haar), Добеши (db), Симлета (sym), Коифлета (coif), биортогональных вейвлетов (bior) и обратно-биортогональных (rbio) и вейвлетов Мейера в дискретном варианте (dmey).

Вейвлет-разложение латентного изображения и его анализ удобней всего производить графической среде **Wavelet 2-D** (рисунок 1). Эта среда предназначена для проведения двумерного дискретного вейвлет-анализа изображений, которые представляют собой двумерный сигнал, задаваемый матрицей. Латентное изображение загружается с помощью меню File/LoadImage. После загрузки изображения в поле **Original Image** необходимо выбрать вид вейвлета в поле **Wavelet** (1, рисунок 1) и уровень вейвлет-разложения (2).

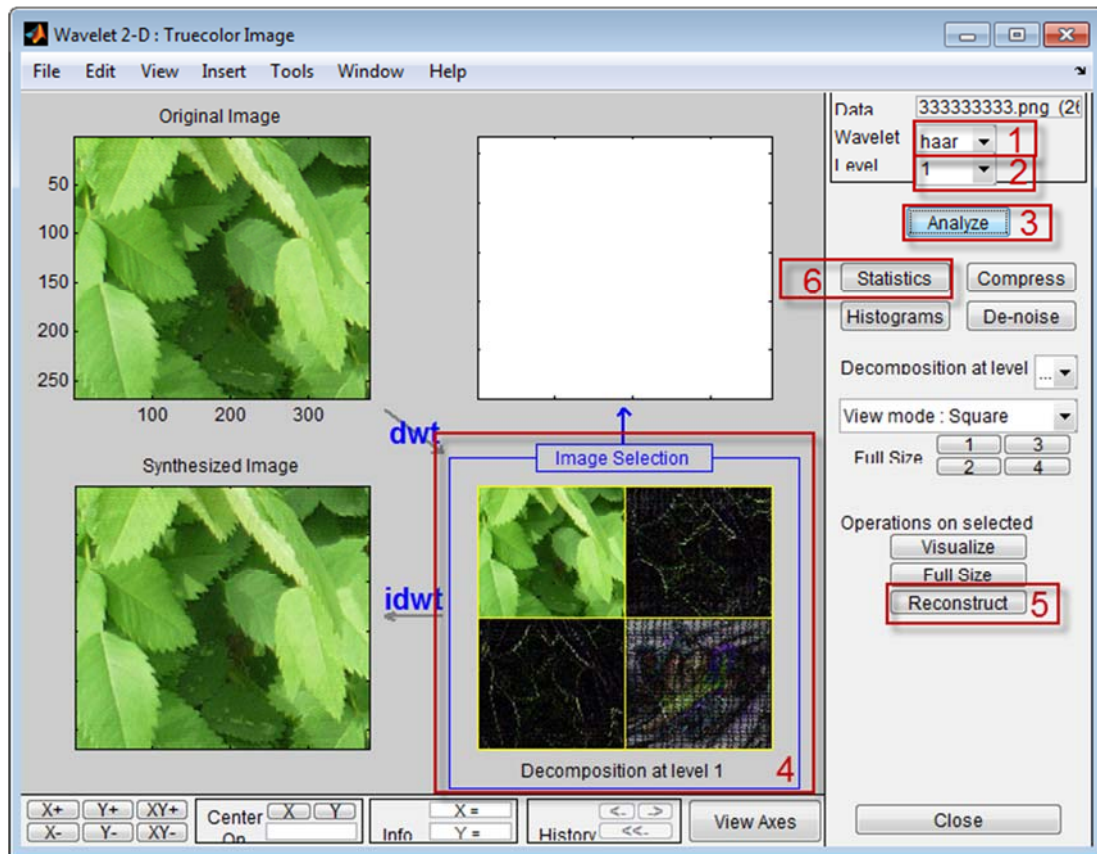


Рисунок 1. Графическая среда по анализу изображений

Для выполнения вейвлет-разложения необходимо нажать кнопку **Analyze**(3). Результаты разложения будут выведены в виде блочной матрицы (4) в поле **Decomposition at level 1** графического окна. В верхнем левом блоке представлены аппроксимирующие коэффициенты, в правом верхнем – горизонтальные детализирующие коэффициенты, в левом нижнем – вертикальные детализирующие коэффициенты, в правом нижнем – диагональные детализирующие коэффициенты. Все полученные коэффициенты образуют матрицы размером в два раза меньше, чем исходная матрица изображения. На рисунке 1 видно, что диагональные коэффициенты (правый нижний угол) выявили скрытое изображение.

Восстановление изображения осуществляется по коэффициентам выбранного блока после нажатия кнопки **Reconstruct** (5). Результат выводится в поле **SynthesizedImage**.

Для получения статистических характеристик используется кнопка **Statistic**(6, рисунок 1). В окне статистики **Wavelet 2-DStatistics** (рисунок 2) содержатся статистические данные о сигнале (исходном, синтезированном) или коэффициентах разложения. По умолчанию выводятся статистические данные об исходном сигнале. Для указания коэффициентов аппроксимации или детализирующих коэффициентов необходимо с помощью переключателя выбрать **Approximation** или **Detal** соответственно.

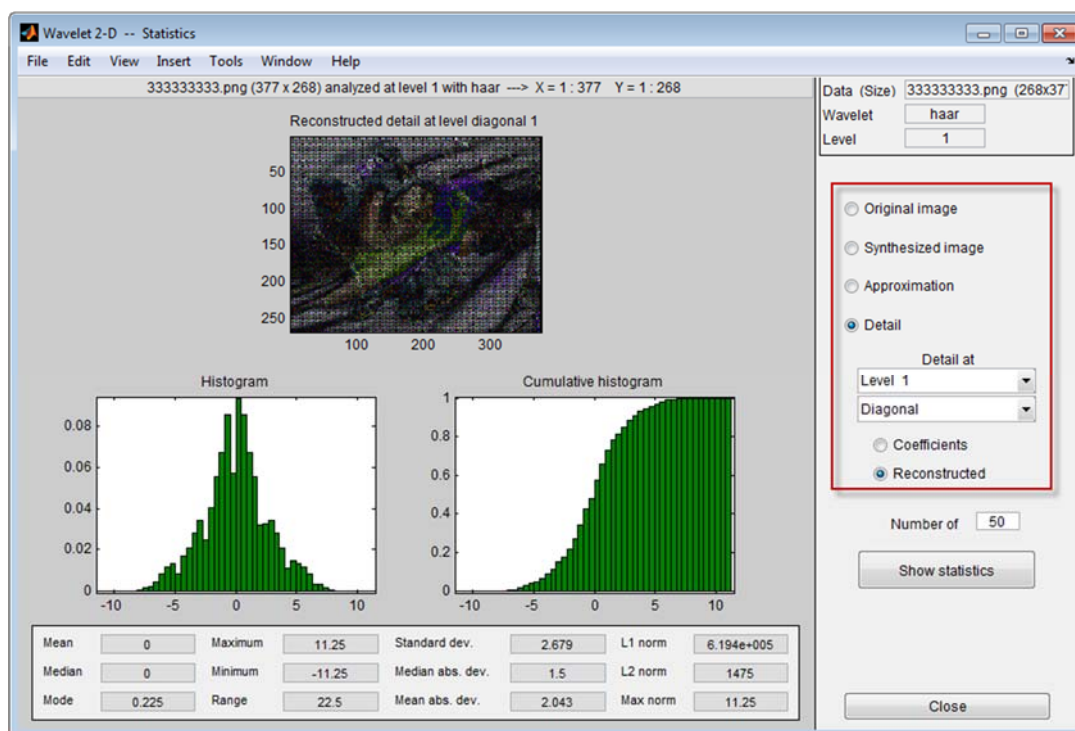


Рисунок 2. Окно статистики

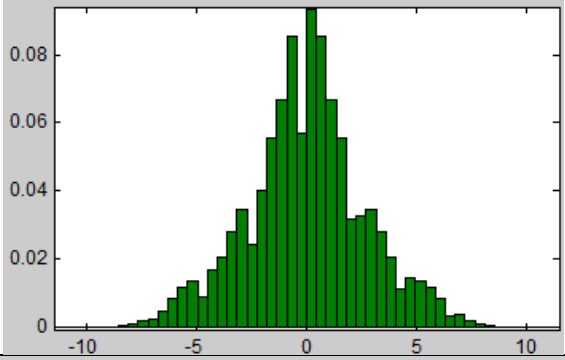
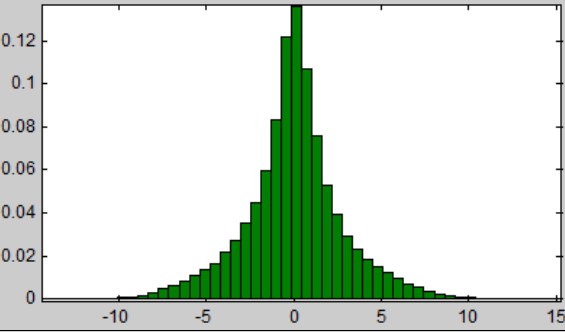
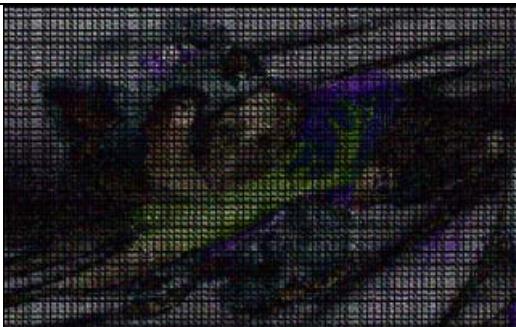
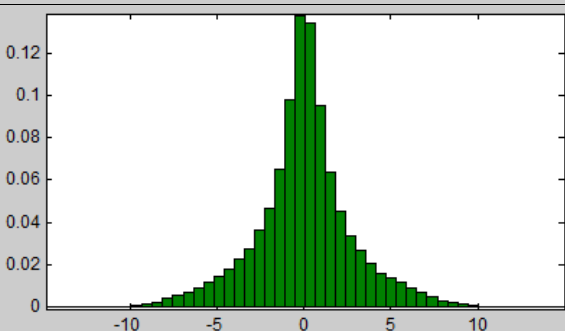
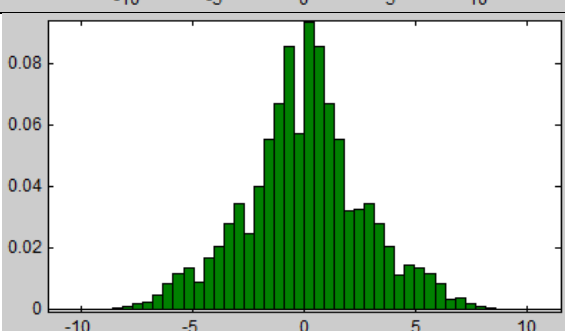
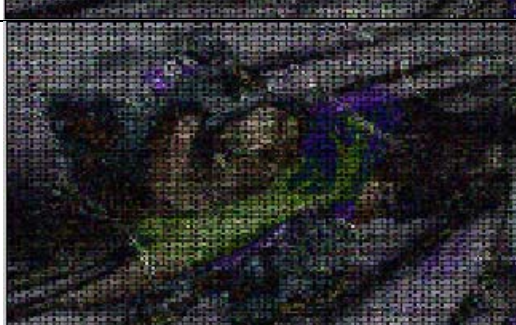
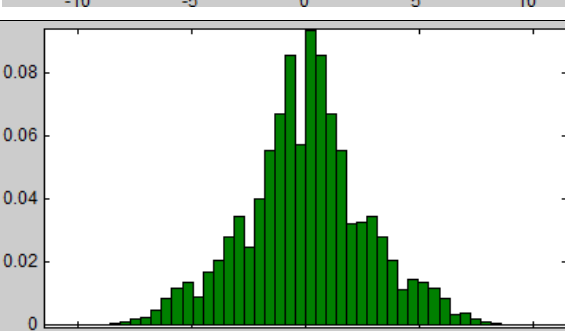
Если выбраны детализирующие коэффициенты, то необходимо указать уровень разложения, например, **Level 1** и для каких коэффициентов (горизонтальных, вертикальных или диагональных) необходимо рассчитать статистические характеристики.

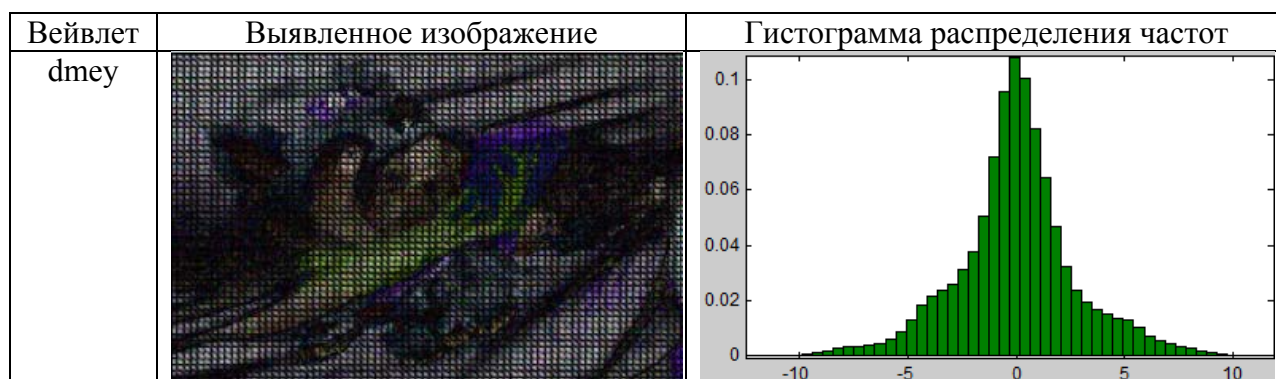
Как видно из рисунка 2 система Matlab позволяет получить статистические характеристики для детализирующих коэффициентов.

В таблице 1 представлены скрытые изображения и гистограммы относительных частот. Вид гистограммы отражает плотность распределения двумерной случайной величины, характеризующей выявленное изображение. Визуально установлено, что лучший результат был получен с использованием вейвлета Мейера. Неплохие результаты были получены с помощью симлетов и коифлетов.

Таблица 1. Результаты анализа

Вейвлет	Выявленное изображение	Гистограмма распределения частот
Haar		

Вейвлет	Выявленное изображение	Гистограмма распределения частот
Db		
Sym		
Coif		
Bior		
Rbio		



В таблице 2 приведены статистические характеристики анализируемых изображений.

Таблица 2. Статистические характеристики изображений

	<i>haar</i>	<i>db</i>	<i>sym</i>	<i>coif</i>	<i>bior</i>	<i>rbio</i>	<i>dmey</i>
Медиана	0	0	0.016	0.016	0	0	0.009
Мода	0.225	0.225	0.144	-0.236	0.225	0.225	-0.074
Макс.	11.25	11.25	15.03	14.68	11.25	11.25	11.61
Мин.	-11.25	-11.25	-14.16	-14.57	-11.25	-11.25	-12.24
Диапазон	22.5	22.5	29.19	29.25	22.5	22.5	23.85

Из таблицы 2 видно, что медиана у более качественных изображений (0,009; 0,016; 0,016) отлична от нуля, а такая числовая характеристика как мода меньше (-0,074; -0,236; 0,144), чем у остальных (0,225), в тоже время частотный диапазон таких изображений шире. Таким образом, по виду гистограммы и полученным статистическим характеристикам можно сделать вывод о качестве выявленного изображения [6].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Выполните предложенные задания, предварительно ознакомившись с теоретической частью.

Задание 1. С помощью вейвлет-разложения выявите скрытую информацию из изображения, сформированного в лабораторной работе №19, с применением различных вейвлетов.

Задание 2. Соберите статистические данные по диагональным коэффициентам, полученным при вейвлет-разложении изображения различными вейвлетами.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Какие виды вейвлетов доступны для вейвлет-анализа изображений в системе matlab?
2. Какие статистические характеристики полученных коэффициентов можно получить в системе matlab?
3. На чем основаны методики выявления латентных изображений?

ЛАБОРАТОРНАЯ РАБОТА 19

Кластеризация данных алгоритмом FOREL

Цель: изучить алгоритм FOREL. Построение гистограмм функций плотности распределения

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

Forel (Формальный Элемент) – алгоритм кластеризации, основанный на идее объединения в один кластер объектов в областях их наибольшего сгущения. Данный алгоритм предложен Загоруйко и Ёлкиной в 1967 году при решении одной прикладной задачи в области палеонтологии. 1.5.1 Алгоритм Forel Алгоритм имеет многочисленные вариации, но в основе всех этих вариаций лежит следующая базовая процедура.

Таксоны, получаемые этим алгоритмом, имеют сферическую форму. Количество таксонов зависит от радиуса сфер: чем меньше радиус, тем больше получается таксонов. Вначале признаки объектов нормируются так, чтобы их значения находились в диапазоне от нуля до единицы. Затем строится гиперсфера минимального радиуса R_0 , которая охватывает все пространство точек. Если бы был нужен один таксон, то он был бы представлен именно этой начальной сферой. Для получения большего числа таксонов радиус сфер постепенно уменьшается. Берется радиус $0,9 * R_0$ и центр сферы помещается в любую из имеющихся точек. Находятся точки, расстояние до которых меньше радиуса, и вычисляются координаты центра тяжести этих «внутренних» точек. Затем центр сферы переносится в этот центр тяжести и снова находятся внутренние точки. Сфера как бы плавает в сторону локального сгущения точек.

Такая процедура определения внутренних точек и переноса центра сферы продолжается до тех пор, пока сфера не остановится, т.е. пока на очередном шагесостав внутренних точек, а, следовательно, и их центр тяжести, не изменится. Это значит, что сфера остановилась в области локального максимума плотности точек в признаковом пространстве.

Точки, оказавшиеся внутри остановившейся сферы, объявляют принадлежащими таксону первому таксону и исключают их из дальнейшего рассмотрения. Для оставшихся точек описанная выше процедура повторяется до тех пор, пока все точки не окажутся включенными в таксоны [6].

Алгоритм Forel-2 – это модификация базового алгоритма, которая предназначена для получения таксономии с точно заданным числом таксонов k .

Здесь радиус сферы по мере надобности увеличивается или уменьшается на величину R , которая на каждой очередной итерации становится все меньше,

например, уменьшается вдвое. Этот метод последовательных приближений 15 позволяет быстро подойти к заданному числу таксонов при минимально возможном радиусе сфер [6].

1.5.3 Преимущества и недостатки алгоритма

К преимуществам алгоритма Forel можно отнести:

- точность минимизации функционала качества (при удачном подборе параметра R , варьируя этот параметр, можно получать кластеризации различной степени детальности);
- наглядность визуализации кластеризации;
- сходимости алгоритма;
- возможность операций над центрами кластеров – они известны в процессе работы алгоритма;
- возможность подсчета промежуточных функционалов качества, например, длины цепочки локальных сгущений;
- возможность проверки гипотез схожести и компактности в процессе работы алгоритма. Недостатками данного алгоритма являются:
- относительно низкая производительность;
- плохая применимость алгоритма при плохой делимости выборки на кластеры;
- неустойчивость алгоритма (зависимость от выбора начального объекта);
- произвольное по количеству разбиение на кластеры;
- необходимость априорных знаний о ширине (диаметре) кластеров [13].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Повторить реализованный пример.

1. Считывание точечных данных из файла в матрицу A и формирование массива точек (X, Y)

$A := \text{READBMR}('D:p3')$



A

A =

	0	1	2	3
0	255	255	255	255
1	255	255	255	255
2	255	255	255	255
3	255	255	255	255
4	255	255	255	255
5	255	255	255	...

```

points(A) :=
  k ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if Ai,j = 0
        | Xk ← i
        | Yk ← j
        | k ← k + 1
  for k ∈ 0..length(X) - 2
    if (Xk+1 - Xk)2 + (Yk - Yk+1)2 < 3
      | Xk ← -1
      | Yk ← -1
  s ← 0
  for k ∈ 0..length(X) - 1
    if Xk ≠ -1
      | X1s ← Xk
      | Y1s ← Yk
      | s ← s + 1
  (X1
   Y1)

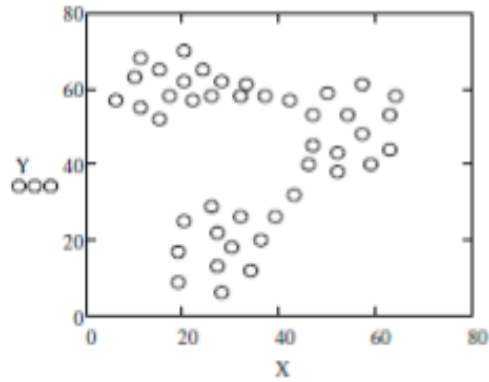
```

функция формирования из
матрицы изображения то-
чечных данных A массива
точек (X,Y)

$$X := \text{points}(A)_0 \quad X^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 6 & 10 & 11 & 11 & 15 & 15 & 17 & 19 & 19 & \dots \\ \hline \end{array}$$

$$Y := \text{points}(A)_1 \quad Y^T = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline 0 & 57 & 63 & 55 & 68 & 52 & 65 & 58 & 9 & 17 & \dots \\ \hline \end{array}$$

2. Функции алгоритма FOREL



$$d(C, S) := (C_0 - S_0)^2 + (C_1 - S_1)^2$$

$$\text{cent}(F, X, Y, r) := \begin{array}{|l} C1 \leftarrow (0 \ 0)^T \\ n \leftarrow 0 \\ \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad C_i \leftarrow (X_i \ Y_i)^T \\ \quad \text{if } d(C_i, F) < r^2 \\ \quad \quad n \leftarrow n + 1 \\ \quad \quad C1 \leftarrow C1 + C_i \\ \frac{C1}{n} \end{array}$$

функция определения центра масс точек массива (X, Y) , содержащихся внутри круга с центром в точке F и радиусом r

$$\text{ext}(F, X, Y, r) := \begin{array}{|l} \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad C_i \leftarrow (X_i \ Y_i)^T \\ \quad \text{if } d(C_i, F) > r^2 \\ \quad \quad C1 \leftarrow C_i \\ \quad \quad \text{break} \\ \text{for } i \in 0.. \text{length}(X) - 1 \\ \quad C_i \leftarrow (X_i \ Y_i)^T \\ \quad C1 \leftarrow \text{augmen}(C1, C_i) \text{ if } d(C_i, F) > r^2 \wedge C_i \neq C1 \\ C1 \end{array}$$

функция удаления из массива (X, Y) точек, содержащихся внутри круга с центром в точке F и радиусом r

$\text{fore}(X, Y, r) :=$

$$\begin{array}{l}
 F0 \leftarrow (X_0 \ Y_0)^T \\
 F \leftarrow F0 \\
 k \leftarrow \text{length}(X) \\
 \text{while } k > 0 \\
 \quad F1 \leftarrow \text{cent}(F0, X, Y, r) \\
 \quad \text{while } d(F0, F1) > 0.1 \\
 \quad \quad F0 \leftarrow F1 \\
 \quad \quad F1 \leftarrow \text{cent}(F1, X, Y, r) \\
 \quad F \leftarrow \text{augment}(F, F1) \\
 \quad X1 \leftarrow (\text{ext}(F1, X, Y, r)^T)^{(0)} \quad \text{if } \text{ext}(F1, X, Y, r) \neq 0 \\
 \quad \text{break otherwise} \\
 \quad Y1 \leftarrow (\text{ext}(F1, X, Y, r)^T)^{(1)} \quad \text{if } \text{ext}(F1, X, Y, r) \neq 0 \\
 \quad X \leftarrow X1 \\
 \quad Y \leftarrow Y1 \\
 \quad F0 \leftarrow (X_0 \ Y_0)^T \\
 \quad k \leftarrow \text{length}(X) \\
 F \leftarrow \text{submatrix}(F, 0, 1, 1, \text{cols}(F) - 1) \\
 F
 \end{array}$$

функция вычисления формальных элементов массива точек (X, Y) алгоритмом FOREL с параметром r

3. Результаты работы алгоритма FOREL

$r := 10$

$F := \text{fore}(X, Y, r)$

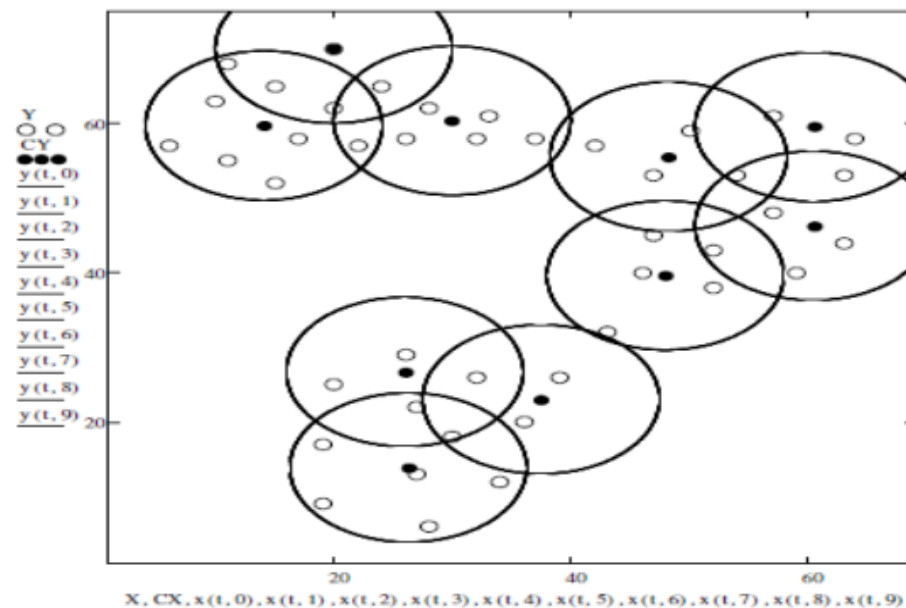
	0	1	2	3	4
0	14.111	26.286	26	30	20
1	59.667	13.857	26.667	60.333	...

$CX := (F^T)^{(0)}$

$x(t, i) := F_{0,i} + r \cdot \cos(t)$

$CY := (F^T)^{(1)}$

$y(t, i) := F_{1,i} + r \cdot \sin(t)$



Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) условия выполненных заданий и их решение;
- 4) технологию решения задач с помощью прикладного программного обеспечения, выводы;
- 5) ответы на контрольные вопросы.

Вопросы для защиты работы

1. Дайте определение алгоритму Forel.
2. Дайте определение таксону?

ЛАБОРАТОРНАЯ РАБОТА 20

Нахождения дискриминантной функции по прецедентам методом потенциальных функций

Цель: научиться определять множество прецедентов.

Формируемые компетенции ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

Forel (Формальный Элемент) – алгоритм кластеризации, основанный на идее объединения в один кластер объектов в областях их наибольшего сгущения. Данный алгоритм предложен Загоруйко и Ёлкиной в 1967 году при решении одной прикладной задачи в области палеонтологии. 1.5.1 Алгоритм Forel Алгоритм имеет многочисленные вариации, но в основе всех этих вариаций лежит следующая базовая процедура.

Таксоны, получаемые этим алгоритмом, имеют сферическую форму. Количество таксонов зависит от радиуса сфер: чем меньше радиус, тем больше получается таксонов. Вначале признаки объектов нормируются так, чтобы их значения находились в диапазоне от нуля до единицы. Затем строится гиперсфера минимального радиуса R_0 , которая охватывает все пространство точек. Если бы был нужен один таксон, то он был бы представлен именно этой начальной сферой. Для получения большего числа таксонов радиус сфер постепенно уменьшается. Берется радиус $0,9 * R_0$ и центр сферы помещается в любую из имеющихся точек. Находятся точки, расстояние до которых меньше радиуса, и вычисляются координаты центра тяжести этих «внутренних» точек. Затем центр сферы переносится в этот центр тяжести и снова находятся внутренние точки. Сфера как бы плывет в сторону локального сгущения точек.

Такая процедура определения внутренних точек и переноса центра сферы продолжается до тех пор, пока сфера не остановится, т.е. пока на очередном шагесостав внутренних точек, а, следовательно, и их центр тяжести, не изменится. Это значит, что сфера остановилась в области локального максимума плотности точек в признаковом пространстве.

Точки, оказавшиеся внутри остановившейся сферы, объявляют принадлежащими таксону первому таксону и исключают их из дальнейшего рассмотрения. Для оставшихся точек описанная выше процедура повторяется до тех пор, пока все точки не окажутся включенными в таксоны [6].

Алгоритм Forel-2 – это модификация базового алгоритма, которая предназначена для получения таксономии с точно заданным числом таксонов k .

Здесь радиус сферы по мере надобности увеличивается или уменьшается на величину R , которая на каждой очередной итерации становится все меньше, например, уменьшается вдвое. Этот метод последовательных приближений 15

позволяет быстро подойти к заданному числу таксонов при минимально возможном радиусе сфер [6].

1.5.3 Преимущества и недостатки алгоритма

К преимуществам алгоритма Forel можно отнести:

- точность минимизации функционала качества (при удачном подборе параметра R , варьируя этот параметр, можно получать кластеризации различной степени детальности);
- наглядность визуализации кластеризации;
- сходимость алгоритма;
- возможность операций над центрами кластеров – они известны в процессе работы алгоритма;
- возможность подсчета промежуточных функционалов качества, например, длины цепочки локальных сгущений;
- возможность проверки гипотез схожести и компактности в процессе работы алгоритма. Недостатками данного алгоритма являются:
- относительно низкая производительность;
- плохая применимость алгоритма при плохой разделимости выборки на кластеры;
- неустойчивость алгоритма (зависимость от выбора начального объекта);
- произвольное по количеству разбиение на кластеры;
- необходимость априорных знаний о ширине (диаметре) кластеров [13].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Machcad и MATLAB R2017a и выше.

Указания по технике безопасности. Самостоятельно не производить установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Повторить реализованный пример

1. Формирование множества по изображению точечных данных

1.1 Считывание точечных данных из файла

$A := \text{READBMP}('h:\text{выборка_пот_2}')$

1.2 Выделение точек из матрицы изображения, формирование массива точечных данных X

$$\text{quan}(A, a, b) \approx \left| \begin{array}{l} s \leftarrow 0 \\ \text{for } i \in 0.. \text{rows}(A) - 1 \\ \quad \text{for } j \in 0.. \text{cols}(A) - 1 \\ \quad \quad \text{if } (a \leq A_{i,j}) \cdot (A_{i,j} \leq b) \\ \quad \quad \quad \left| \begin{array}{l} X^{(s)} \leftarrow (i \ j)^T \\ s \leftarrow s + 1 \end{array} \right. \end{array} \right| X$$

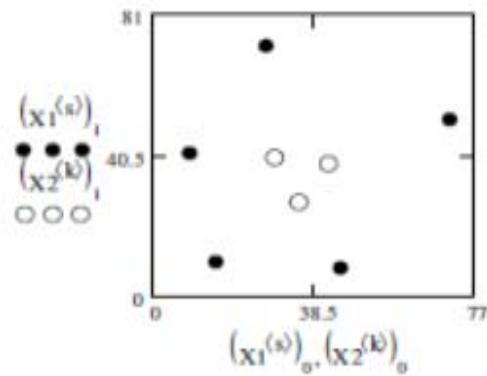
$X1 := \text{quan}(A, 0, 50)$ $X1 = \begin{pmatrix} 9 & 15 & 27 & 45 & 71 \\ 41 & 10 & 72 & 8 & 51 \end{pmatrix}$

- множество векторов
первого класса

$X2 := \text{quan}(A, 50, 200)$ $X2 = \begin{pmatrix} 29 & 35 & 42 \\ 40 & 27 & 38 \end{pmatrix}$

- множество векторов
второго класса

$s := 0.. \text{cols}(X1) - 1$ $k := 0.. \text{cols}(X2) - 1$



- изображение
точечных данных
массива X

1.3 Определение множества прецедентов

```

prec(A, b) :=
  s ← 0
  for i ∈ 0..rows(A) - 1
    for j ∈ 0..cols(A) - 1
      if Ai,j ≠ 255
        X(s) ← (i j 1)T if Ai,j ≤ b
        X(s) ← (i j -1)T otherwise
        s ← s + 1
        continue
  X

X := prec(A, 40)
X =  $\begin{pmatrix} 9 & 15 & 27 & 29 & 35 & 42 & 45 & 71 \\ 41 & 10 & 72 & 40 & 27 & 38 & 8 & 51 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \end{pmatrix}$ 

```

- множество прецедентов

Содержание отчета и его форма

Подготовьте отчет, в котором опишите технологию решения задачи. Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) условия выполненных заданий и их решение;
- 4) выводы;
- 5) ответы на контрольные вопросы.

Вопросы для защиты работы:

1. Дайте определение алгоритму Forel-2.
2. Назовите преимущества алгоритма Forel?
3. Назовите недостатки алгоритма Forel?

ЛАБОРАТОРНАЯ РАБОТА 21

Вычисление дискриминантной функции методом потенциальных функций

Цель и содержание: Научиться вычислять дискриминантную функцию методом потенциальных функций.

Организационная форма занятий: решение проблемных задач, разбор конкретных ситуаций.

Вопросы для обсуждения на лабораторном занятии: вычисление дискриминантной функции методом потенциальных функций в MATLAB.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме, используя материал, изложенный ниже.

Алгоритм FOREL.

Этот алгоритм был предложен Н.Г. Загоруйко и В.Н. Ёлкиной в 1967 году. В алгоритме FOREL (FORmal ELement), так же как и в алгоритме k -means, вычисляются центры тяжести кластеров. Но, в отличие от k -means, в качестве кластера рассматриваются не все ближайшие к данному центру элементы, а все элементы, находящиеся внутри сферы заданного радиуса r с центром в данной точке. Для фиксированного значения $r > 0$ и некоторого элемента $\mathbf{x} = \mathbf{e}^{(1)}$ обучающей выборки вычисляется формальный элемент $\mathbf{e}^{(2)}$ – центр тяжести всех векторов обучающей выборки Ξ , находящихся внутри круга $B_r(\mathbf{e}^{(1)})$ с центром в точке $\mathbf{x}_1$ и радиусом r . Затем вычисляется центр тяжести $\mathbf{e}^{(3)}$ всех элементов множества $B_r(\mathbf{e}^{(2)}) \cap \Xi$ и т.д. Можно показать (докажите!), что таким образом построенная последовательность формальных элементов $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots$ является сходящейся. Предел $\mathbf{e}$ этой последовательности объявляется центром первого кластера. Далее из обучающей выборки Ξ удаляются все элементы, находящиеся внутри сферы $B_r(\mathbf{e})$ и аналогично находится центр следующего кластера и т.д.

Алгоритм FOREL

1. Выбирается некоторый элемент $\mathbf{x} \in \Xi$ обучающей выборки, $\mathbf{e}^{(1)} := \mathbf{x}$.

2. Вычисляется центр тяжести $\mathbf{e}^{(2)} = \frac{1}{|B_r(\mathbf{e}^{(1)}) \cap \Xi|} \sum_{\mathbf{x} \in B_r(\mathbf{e}^{(1)}) \cap \Xi} \mathbf{x}$. Выполнение пункта 2 повторяется до тех пор, пока последовательность $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots$ не стабилизируется в точке $\mathbf{e}$.

3. Полагаем $\Xi := \Xi \setminus B_r(\mathbf{e})$ и переходим к пункту 1. Условием останова алгоритма является $\Xi = \emptyset$.

Результаты работы алгоритма FOREL для трех разных значений r показаны на рис. 4.3. Здесь незакрашенные кружочки – элементы обучающей выборки, черные точки – формальные элементы.

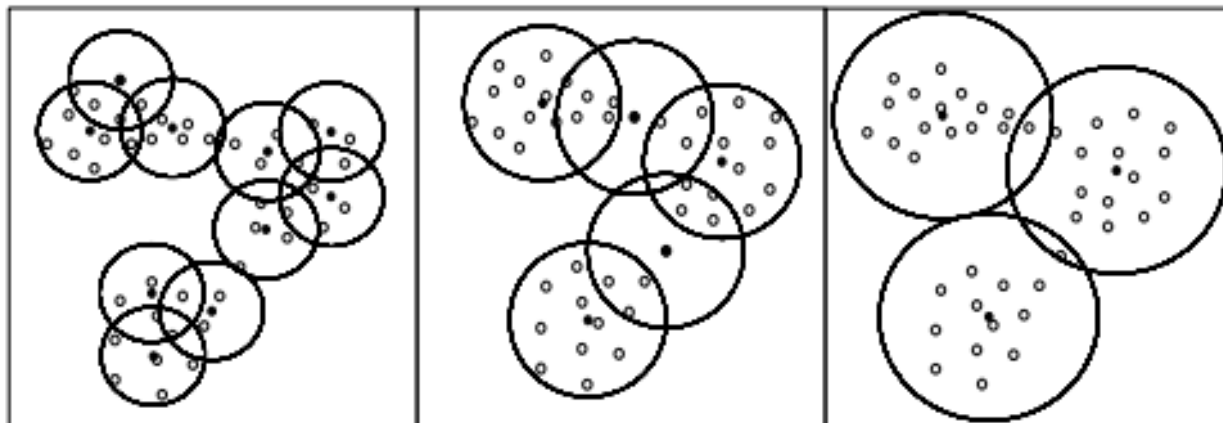


Рис. 4.3

В Приложении 1 приведен расчет кластеризации данных алгоритмом FOREL, выполненный с помощью пакета MathCad.

Результат работы алгоритма FOREL может существенно зависеть от выбора начальных точек – формальных элементов. Единственный параметр алгоритма – величина r подбирается исходя из задач кластеризации: если необходимо получить большие кластеры, то r следует увеличить, если же нужно описать структуру самих кластеров, то r следует уменьшить и кластеризовать «мелкие» кластеры. В этом случае алгоритм FOREL можно рассматривать как алгоритм предварительной кластеризации. Различные модификации алгоритма FOREL и его применение подробно рассмотрены в [11].

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Самостоятельно не производить установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Выполните предложенные задания

Задание метрики и потенциальной функции

$$d(x, y) := \sqrt{(x_0 - y_0)^2 + (x_1 - y_1)^2} \quad u(x, y) := \frac{1}{1 + d(x, y)^2}$$

Вычисление дискриминантной функции

```

potential(X) :=
  p ← 0
  for j ∈ 0..cols(X) - 1
    aj ← 0
  while p < cols(X)
    p ← 0
    for k ∈ -1..cols(X) - 2
      if  $\left[ X_{2,k+1} \cdot \sum_{j=0}^{\text{cols}(X)-1} a_j u \left[ \begin{pmatrix} X_{0,k+1} \\ X_{1,k+1} \end{pmatrix}, \begin{pmatrix} X_{0,j} \\ X_{1,j} \end{pmatrix} \right] \right] \leq 0$ 
        ak+1 ← ak+1 + 1 · X2,k+1
        p ← 0
      p ← p + 1 otherwise
  a

```

$a := \text{potential}(X)$

$a^T = (1 \ 0 \ 0 \ -1 \ 0 \ 0 \ 1 \ 0)$ - вектор коэффициентов решающей функции

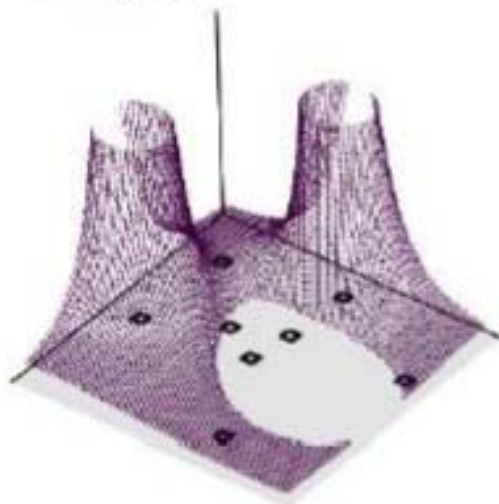
$D(y) := \sum_{j=0}^{\text{cols}(X)-1} a_j u \left[y, \begin{pmatrix} X_{0,j} \\ X_{1,j} \end{pmatrix} \right]$ - решающая функция

Визуализация решающей функции и прецедентов

$$i := 0..cols(A) - 1 \quad j := 0..rows(A) - 1$$

$$D_{i,j} := \begin{cases} D\left(\begin{pmatrix} i \\ j \end{pmatrix}\right) & \text{if } D\left(\begin{pmatrix} i \\ j \end{pmatrix}\right) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,j} := \begin{cases} 0.000001 & \text{if } \sum_{s=0}^{cols(X)-1} [(X_{0,s}-i)^2 + (X_{1,s}-j)^2] \leq 3 \\ 0 & \text{otherwise} \end{cases}$$



D, B

Содержание отчета и его форма

Подготовьте отчет, в котором опишите технологию решения задач/
Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) условия выполненных заданий и их решение;
- 4) выводы;
- 5) ответы на контрольные вопросы.

Вопросы для защиты работы:

1. Расскажите краткую историю возникновения алгоритма Forel.
2. Опишите этапы алгоритма Forel..

ЛАБОРАТОРНАЯ РАБОТА 22

Вычисление дискриминантной функции путем представления потенциальной с помощью системы базисных функций

Цель: Научиться вычислению дискриминантной функции путем представления потенциальной с помощью системы базисных функций.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

FOREL (от formal element, формальный элемент) — алгоритм кластеризации, основанный на идее объединения в кластер объектов в областях их наибольшего сгущения. Он предназначен для компьютерной реализации.

Цель кластеризации — разбить выборку на такое (заранее неизвестное) число кластеров, чтобы сумма расстояний от объектов кластеров до центров кластеров была минимальной по всем кластерам (т.е. задача алгоритма — выделить группы максимально близких друг к другу объектов, которые в силу гипотезы схожести и будут образовывать кластеры).

Минимизируемый алгоритмом функционал качества

$$F = \sum_{i=1}^k \sum_{x \in S_i} \rho(x, m_i),$$

где первое суммирование ведется по всем кластерам выборки, а второе — по всем объектам x , принадлежащим текущему кластеру S_i ; m_i — центр текущего кластера; $\rho(x, m_i)$ — расстояние между объектами.

Предположение, лежащее в основе алгоритма, — выполнение гипотезы компактности, предполагающей, что близкие друг к другу объекты с большой вероятностью принадлежат к одному кластеру.

Кластеризуемая выборка может быть задана описаниями объектов с помощью признаков, например, координатами линейного пространства либо матрицей попарных расстояний между объектами. В реальных задачах необходимые данные собираются в процессе кластеризации. Среди исходных данных важен параметр R — радиус поиска локальных сгущений. Его можно как задавать из априорных соображений (знание о диаметре кластеров), так и настраивать скользящим контролем. В модификациях алгоритма возможно введение параметра k — количества кластеров.

Выходные данные алгоритма — кластеризация на заранее неизвестное число кластеров.

Принцип работы — рекуррентный. На каждом шаге случайным образом выбирается объект из выборки, описывается вокруг него сфера радиуса R , внутри этой сферы выбирается центр тяжести, и он делается центром новой сферы (т.е. на каждом шаге сфера двигается в сторону локального сгущения).

объектов выборки, что соответствует стремлению захватить как можно больше объектов выборки сферой фиксированного радиуса). После того как центр сферы стабилизируется, все объекты внутри сферы с этим центром помечаются как кластеризованные и удаляются из выборки. Этот процесс повторяется до тех пор, пока вся выборка не будет кластеризована.

Структура алгоритма следующая.

1. Случайно выбирается текущий объект из выборки.
2. Помечаются объекты выборки, находящиеся на расстоянии менее чем R от текущего объекта.
3. Вычисляется их центр тяжести, этот центр помечается как новый текущий объект.
4. Повторяются шаги 2—3, пока новый текущий объект не совпадет с прежним.

5. Помечаются объекты внутри сферы радиуса R вокруг текущего объекта как кластеризованные, затем они удаляются из выборки.

6. Повторяются шаги 1—5, пока не будет кластеризована вся выборка.

Эвристические способы выбора центра тяжести:

- в линейном пространстве — центр масс;
- в метрическом пространстве — объект, сумма расстояний до которого минимальна среди всех внутри сферы;
- объект, который внутри сферы радиуса R содержит максимальное количество других объектов из всей выборки;
- объект, который внутри сферы малого радиуса r ($r < R$) содержит максимальное количество объектов из сферы радиуса R .

Алгоритм можно охарактеризовать следующими положениями.

Доказана сходимость алгоритма за конечное число шагов.

В линейном пространстве центром тяжести может выступать произвольная точка пространства, в метрическом — только объект выборки.

Чем меньше R , тем больше кластеров.

В линейном пространстве поиск центра происходит за время $O(p)$, в метрическом — $O(p^2)$, где p — объем кластеризуемой выборки.

Наилучших результатов алгоритм достигает на выборках с хорошим выполнением условий компактности.

При повторении итераций возможно уменьшение параметра R для скорейшей сходимости.

Кластеризация сильно зависит от начального приближения (выбора объекта на первом шаге).

Рекомендуется повторная прогонка алгоритма для исключения ситуации «плохой» кластеризации по причине неудачного выбора начальных объектов.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и

выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса.

Методика и порядок выполнения работы

Задание системы базисных функций

$$H(x, i, j) := \text{Leg}(i, x_0) \cdot \text{Leg}(j, x_1)$$

- многочлены Лежандра

3.2. Нормирование исходных данных

$$x_0 := (x^T)^{(0)} \quad x_1 := (x^T)^{(1)}$$

$m := 3$

$c := \text{potential}(XX, m)$

$$c^T = \begin{pmatrix} 3 & -2.056 & 3.065 \\ -2.023 & -1.773 & 0.464 \\ 1.756 & 0.09 & -0.478 \end{pmatrix}$$

- матрица
коэффициентов
решающей функции

$$DN(y) := \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} c_{i,j} H(y, i, j)$$

- нормированная
решающая функция

$$D2(y) := DN \left[\begin{pmatrix} \frac{y_0 - \min 0}{\max 0 - \min 0} \\ \frac{y_1 - \min 1}{(\max 1 - \min 1)} \end{pmatrix} \right]$$

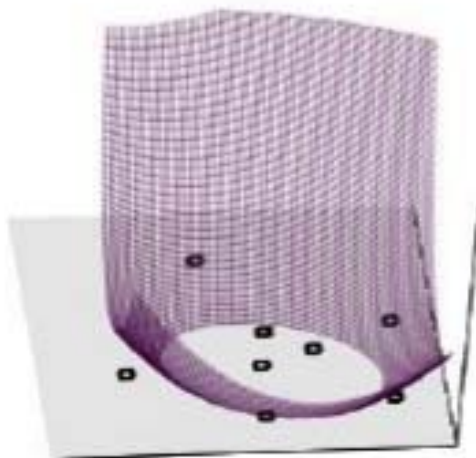
- денормированная
решающая функция

3.4. Визуализация решающей функции и пределентов

$i := 0.. \text{cols}(A) - 1 \quad j := 0.. \text{rows}(A) - 1$

$$D2_{i,j} := \begin{cases} D2 \left(\begin{pmatrix} i \\ j \end{pmatrix} \right) & \text{if } D2 \left(\begin{pmatrix} i \\ j \end{pmatrix} \right) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,j} := \begin{cases} 0.000001 & \text{if } \sum_{s=0}^{\text{cols}(X)-1} [(X_{0,s} - i)^2 + (X_{1,s} - j)^2] \leq 3 \\ 0 & \text{otherwise} \end{cases}$$



D2, B

Содержание отчета и его форма

Подготовьте отчет. Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) условия выполненных заданий и их решение;
- 4) выводы;
- 5) ответы на контрольные вопросы.

Вопросы для защиты работы:

1. Какова цель кластеризации?
2. Опишите структуру алгоритма Forel.

ЛАБОРАТОРНАЯ РАБОТА 23

Простейшая гистограмма с регулярными ячейками

Цель: Научиться построению простейшей гистограммы с регулярными ячейками.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

FOREL (от formal element, формальный элемент) — алгоритм кластеризации, основанный на идее объединения в кластер объектов в областях их наибольшего сгущения. Он предназначен для компьютерной реализации.

Цель кластеризации — разбить выборку на такое (заранее неизвестное) число кластеров, чтобы сумма расстояний от объектов кластеров до центров кластеров была минимальной по всем кластерам (т.е. задача алгоритма — выделить группы максимально близких друг к другу объектов, которые в силу гипотезы схожести и будут образовывать кластеры).

Минимизируемый алгоритмом функционал качества

$$F = \sum_{i=1}^k \sum_{x \in S_i} \rho(x, m_i),$$

где первое суммирование ведется по всем кластерам выборки, а второе — по всем объектам x , принадлежащим текущему кластеру S_i ; m_i — центр текущего кластера; $\rho(x, m_i)$ — расстояние между объектами.

Предположение, лежащее в основе алгоритма, — выполнение гипотезы компактности, предполагающей, что близкие друг к другу объекты с большой вероятностью принадлежат к одному кластеру.

Кластеризуемая выборка может быть задана описаниями объектов с помощью признаков, например, координатами линейного пространства либо матрицей попарных расстояний между объектами. В реальных задачах необходимые данные собираются в процессе кластеризации. Среди исходных данных важен параметр R — радиус поиска локальных сгущений. Его можно как задавать из априорных соображений (знание о диаметре кластеров), так и настраивать скользящим контролем. В модификациях алгоритма возможно введение параметра k — количества кластеров.

Выходные данные алгоритма — кластеризация на заранее неизвестное число кластеров.

Принцип работы — рекуррентный. На каждом шаге случайным образом выбирается объект из выборки, описывается вокруг него сфера радиуса R , внутри этой сферы выбирается центр тяжести, и он делается центром новой сферы (т.е. на каждом шаге сфера двигается в сторону локального сгущения объектов выборки, что соответствует стремлению захватить как можно

больше объектов выборки сферой фиксированного радиуса). После того как центр сферы стабилизируется, все объекты внутри сферы с этим центром помечаются как кластеризованные и удаляются из выборки. Этот процесс повторяется до тех пор, пока вся выборка не будет кластеризована.

Структура алгоритма следующая.

1. Случайно выбирается текущий объект из выборки.
2. Помечаются объекты выборки, находящиеся на расстоянии менее чем R от текущего объекта.
3. Вычисляется их центр тяжести, этот центр помечается как новый текущий объект.
4. Повторяются шаги 2—3, пока новый текущий объект не совпадет с прежним.
5. Помечаются объекты внутри сферы радиуса R вокруг текущего объекта как кластеризованные, затем они удаляются из выборки.
6. Повторяются шаги 1—5, пока не будет кластеризована вся выборка.

Эвристические способы выбора центра тяжести:

- в линейном пространстве — центр масс;
- в метрическом пространстве — объект, сумма расстояний до которого минимальна среди всех внутри сферы;
- объект, который внутри сферы радиуса R содержит максимальное количество других объектов из всей выборки;
- объект, который внутри сферы малого радиуса r ($r < R$) содержит максимальное количество объектов из сферы радиуса R .

Алгоритм можно охарактеризовать следующими положениями.

Доказана сходимость алгоритма за конечное число шагов.

В линейном пространстве центром тяжести может выступать произвольная точка пространства, в метрическом — только объект выборки.

Чем меньше R , тем больше кластеров.

В линейном пространстве поиск центра происходит за время $O(n)$, в метрическом — $O(n^2)$, где n — объем кластеризуемой выборки.

Наилучших результатов алгоритм достигает на выборках с хорошим выполнением условий компактности.

При повторении итераций возможно уменьшение параметра R для скорейшей сходимости.

Кластеризация сильно зависит от начального приближения (выбора объекта на первом шаге).

Рекомендуется повторная прогонка алгоритма для исключения ситуации «плохой» кластеризации по причине неудачного выбора начальных объектов.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Самостоятельно не производить установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

1.1. Считывание точечных данных из файла и формирование вектора данных X

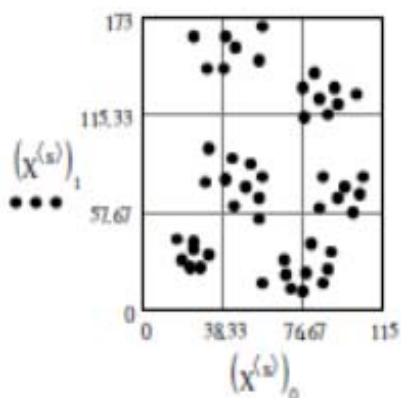
$A := \text{READBMP}('D:/\text{выборка}')$

$\text{quant}(A) :=$

$$\begin{array}{l} s \leftarrow 0 \\ \text{for } i \in 0.. \text{rows}(A) - 1 \\ \quad \text{for } j \in 0.. \text{cols}(A) - 1 \\ \quad \quad \text{if } A_{i,j} = 0 \\ \quad \quad \quad \left| \begin{array}{l} X^{(s)} \leftarrow (i \ j)^T \\ s \leftarrow s + 1 \end{array} \right. \end{array}$$
 X

$X \simeq \text{quant}(A)$

$s := 0.. \text{cols}(X) - 1$



$$X^T =$$

	0	1
0	16	42
1	18	30
2	22	25
3	24	36
4	24	41
5	24	162
6	27	25
7	29	75
8	30	142
9	31	33
10	31	96
11	38	142
12	39	77
13	39	162
14	42	89
15	43	61
16	44	155
17	49	73
18	51	86
19	55	54

1.2. Вычисление гистограммы с $m \times n$ регулярными ячейками

```

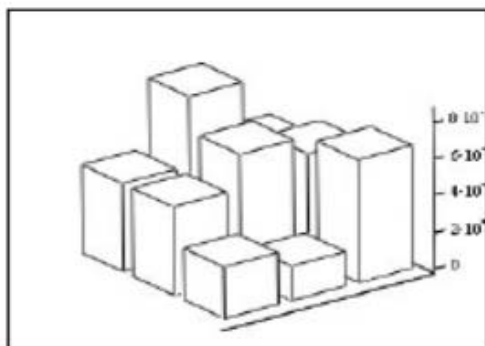
hist(X, m, n) :=
  for i ∈ 0..m-1
    for j ∈ 0..n-1
      Hi,j ← 0
    for i ∈ 0..m-1
      for j ∈ 0..n-1
        for s ∈ 0..cols(X)-1
          
$$R1 \leftarrow \left[ \frac{\text{rows}(A) \cdot i}{m} \leq (X^{(s)})_0 \right] \cdot \left[ (X^{(s)})_0 < \frac{\text{rows}(A) \cdot (i+1)}{m} \right]$$

          
$$R2 \leftarrow \left[ \frac{\text{cols}(A) \cdot j}{n} \leq (X^{(s)})_1 \right] \cdot \left[ (X^{(s)})_1 < \frac{\text{cols}(A) \cdot (j+1)}{n} \right]$$

          Hi,j ← Hi,j + 1 if R1 · R2
        
$$H = \frac{m \cdot n}{(\text{cols}(X) - 1) \cdot (\text{rows}(A) - 1) \cdot (\text{cols}(A) - 1)}$$


```

H := hist(X, 3, 3)



H

$$H = \begin{pmatrix} 6.694 \times 10^{-5} & 1.912 \times 10^{-5} & 2.869 \times 10^{-5} \\ 5.737 \times 10^{-5} & 6.694 \times 10^{-5} & 4.781 \times 10^{-5} \\ 4.781 \times 10^{-5} & 8.606 \times 10^{-5} & 4.781 \times 10^{-5} \end{pmatrix}$$

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;

5) приложение – файлы выполненных заданий.

Вопросы для защиты работы:

1. Опишите основную цель кластеризации.
2. Как задать кластеризуемую выборку??
3. Что такое выходные данные алгоритма?

ЛАБОРАТОРНАЯ РАБОТА 24

Вычисление гистограммы с нерегулярными ячейками

Цель: Научиться вычислять гистограммы с нерегулярными ячейками.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

FOREL (от formal element, формальный элемент) — алгоритм кластеризации, основанный на идее объединения в кластер объектов в областях их наибольшего сгущения. Он предназначен для компьютерной реализации.

Цель кластеризации — разбить выборку на такое (заранее неизвестное) число кластеров, чтобы сумма расстояний от объектов кластеров до центров кластеров была минимальной по всем кластерам (т.е. задача алгоритма — выделить группы максимально близких друг к другу объектов, которые в силу гипотезы схожести и будут образовывать кластеры).

Минимизируемый алгоритмом функционал качества

$$F = \sum_{i=1}^k \sum_{x \in S_i} \rho(x, m_i),$$

где первое суммирование ведется по всем кластерам выборки, а второе — по всем объектам x , принадлежащим текущему кластеру S_i ; m_i — центр текущего кластера; $\rho(x, m_i)$ — расстояние между объектами.

Предположение, лежащее в основе алгоритма, — выполнение гипотезы компактности, предполагающей, что близкие друг к другу объекты с большой вероятностью принадлежат к одному кластеру.

Кластеризуемая выборка может быть задана описаниями объектов с помощью признаков, например, координатами линейного пространства либо матрицей попарных расстояний между объектами. В реальных задачах необходимые данные собираются в процессе кластеризации. Среди исходных данных важен параметр R — радиус поиска локальных сгущений. Его можно как задавать из априорных соображений (знание о диаметре кластеров), так и настраивать скользящим контролем. В модификациях алгоритма возможно введение параметра k — количества кластеров.

Выходные данные алгоритма — кластеризация на заранее неизвестное число кластеров.

Принцип работы — рекуррентный. На каждом шаге случайным образом выбирается объект из выборки, описывается вокруг него сфера радиуса R , внутри этой сферы выбирается центр тяжести, и он делается центром новой сферы (т.е. на каждом шаге сфера двигается в сторону локального сгущения объектов выборки, что соответствует стремлению захватить как можно больше объектов выборки сферой фиксированного радиуса). После того как

центр сферы стабилизируется, все объекты внутри сферы с этим центром помечаются как кластеризованные и удаляются из выборки. Этот процесс повторяется до тех пор, пока вся выборка не будет кластеризована.

Структура алгоритма следующая.

1. Случайно выбирается текущий объект из выборки.
2. Помечаются объекты выборки, находящиеся на расстоянии менее чем R от текущего объекта.
3. Вычисляется их центр тяжести, этот центр помечается как новый текущий объект.
4. Повторяются шаги 2—3, пока новый текущий объект не совпадет с прежним.
5. Помечаются объекты внутри сферы радиуса R вокруг текущего объекта как кластеризованные, затем они удаляются из выборки.
6. Повторяются шаги 1—5, пока не будет кластеризована вся выборка.

Эвристические способы выбора центра тяжести:

- в линейном пространстве — центр масс;
- в метрическом пространстве — объект, сумма расстояний до которого минимальна среди всех внутри сферы;
- объект, который внутри сферы радиуса R содержит максимальное количество других объектов из всей выборки;
- объект, который внутри сферы малого радиуса r ($r < R$) содержит максимальное количество объектов из сферы радиуса R .

Алгоритм можно охарактеризовать следующими положениями.

Доказана сходимость алгоритма за конечное число шагов.

В линейном пространстве центром тяжести может выступать произвольная точка пространства, в метрическом — только объект выборки.

Чем меньше R , тем больше кластеров.

В линейном пространстве поиск центра происходит за время $O(n)$, в метрическом — $O(n^2)$, где n — объем кластеризуемой выборки.

Наилучших результатов алгоритм достигает на выборках с хорошим выполнением условий компактности.

При повторении итераций возможно уменьшение параметра R для скорейшей сходимости.

Кластеризация сильно зависит от начального приближения (выбора объекта на первом шаге).

Рекомендуется повторная прогонка алгоритма для исключения ситуации «плохой» кластеризации по причине неудачного выбора начальных объектов.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционная система Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Определение взвешенной евклидовой d и евклидовой dc метрик

$$d(x, B, k) := \frac{(x_0 - B_{0,k})^2}{B_{2,k}} + \frac{(x_1 - B_{1,k})^2}{B_{3,k}} \quad dc(x, y) = (x_0 - y_0)^2 + (x_1 - y_1)^2$$

Определение квадрата расстояния от точки x до границы матрицы изображения A

$$db(x, A) := \left(\min(x_0 \quad x_1 \quad \text{rows}(A) - x_0 \quad \text{cols}(A) - x_1) \right)^2$$

Функция распределения основного массива данных по ячейкам

Входные данные

массив X - в каждом i -м столбце - координаты i -го вектора данных;
 $s0$ - значение минимальной дисперсии;
 $h1$ - максимальное расстояние во взвешенной метрике от текущего центра до вектора данных, при котором он будет отнесен к данному классу;
 $h2$ - минимальное расстояние во взвешенной метрике от вектора данных до всех центров классов, при котором вектор станет центром нового класса.

Выходные данные

массив B - массив классов, в каждом k -м столбце записаны координаты центра k -го класса, покоординатные дисперсии и количество элементов в классе;
 массив X - в каждом i -м столбце - координаты i -го вектора данных и номер класса, к которому он относится.

```

distr(X,s0,h1,h2) :=
  X2,0 ← 1
  B(0) ← (X0,0 X1,0 s0 s0 1)T
  for i ∈ 1..cols(X) - 1
    s ← 0
    p ← 0
    while s ≤ cols(B) - 1
      (m0 m1 s1 s2 N) ← B(s)T
      if d(X(i), B, s) < h1
        X2,i ← s + 1
        B0,s ←  $\frac{1}{N+1} \cdot (N \cdot m0 + X_{0,i})$ 
        B1,s ←  $\frac{1}{N+1} \cdot (N \cdot m1 + X_{1,i})$ 
        B2,s ←  $\max \left[ s0 \frac{(N-1) \cdot s1}{N} + \frac{1}{N+1} \cdot (m0 - X_{0,i})^2 \right]$ 
        B3,s ←  $\max \left[ s0 \frac{(N-1) \cdot s2}{N} + \frac{1}{N+1} \cdot (m1 - X_{1,i})^2 \right]$ 
        B4,s ← N + 1
        break
      p ← p + 1 if d(X(i), B, s) > h2
      s ← s + 1
    if p = cols(B)
      B ← augment(B, (X0,i X1,i s0 s0 1)T)
      X2,i ← s + 1
  (B X)T

```

DISTR := distr(X,30,10,20)

B := DISTR₀ X := DISTR₁

$$B = \begin{pmatrix} 23.143 & 43.8 & 46.667 & 75.556 & 89.167 & 99.4 \\ 33.143 & 158.8 & 73.333 & 20.667 & 122.333 & 68.8 \\ 37.343 & 178.7 & 84.5 & 123.028 & 72.867 & 33.133 \\ 48.476 & 62.2 & 129.75 & 61.25 & 45.467 & 58.075 \\ 7 & 5 & 9 & 9 & 6 & 5 \end{pmatrix}$$
 - массив центров
 кластеров, дисперсий и
 количества элементов в
 кластерах

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Опишите принцип работы алгоритма.
2. Назовите основные положения, характеризующие алгоритм?

ЛАБОРАТОРНАЯ РАБОТА 25

Вычисление гистограммы с нерегулярными ячейками

Цель: Научиться вычислять гистограммы с нерегулярными ячейками.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Изучите теоретический материал по данной теме.

FOREL (от formal element, формальный элемент) — алгоритм кластеризации, основанный на идее объединения в кластер объектов в областях их наибольшего сгущения. Он предназначен для компьютерной реализации.

Цель кластеризации — разбить выборку на такое (заранее неизвестное) число кластеров, чтобы сумма расстояний от объектов кластеров до центров кластеров была минимальной по всем кластерам (т.е. задача алгоритма — выделить группы максимально близких друг к другу объектов, которые в силу гипотезы схожести и будут образовывать кластеры).

Минимизируемый алгоритмом функционал качества

$$F = \sum_{i=1}^k \sum_{x \in S_i} \rho(x, m_i),$$

где первое суммирование ведется по всем кластерам выборки, а второе — по всем объектам x , принадлежащим текущему кластеру S_i ; m_i — центр текущего кластера; $\rho(x, m_i)$ — расстояние между объектами.

Предположение, лежащее в основе алгоритма, — выполнение гипотезы компактности, предполагающей, что близкие друг к другу объекты с большой вероятностью принадлежат к одному кластеру.

Кластеризуемая выборка может быть задана описаниями объектов с помощью признаков, например, координатами линейного пространства либо матрицей попарных расстояний между объектами. В реальных задачах необходимые данные собираются в процессе кластеризации. Среди исходных данных важен параметр R — радиус поиска локальных сгущений. Его можно как задавать из априорных соображений (знание о диаметре кластеров), так и настраивать скользящим контролем. В модификациях алгоритма возможно введение параметра k — количества кластеров.

Выходные данные алгоритма — кластеризация на заранее неизвестное число кластеров.

Принцип работы — рекуррентный. На каждом шаге случайным образом выбирается объект из выборки, описывается вокруг него сфера радиуса R , внутри этой сферы выбирается центр тяжести, и он делается центром новой сферы (т.е. на каждом шаге сфера двигается в сторону локального сгущения объектов выборки, что соответствует стремлению захватить как можно больше объектов выборки сферой фиксированного радиуса). После того как

центр сферы стабилизируется, все объекты внутри сферы с этим центром помечаются как кластеризованные и удаляются из выборки. Этот процесс повторяется до тех пор, пока вся выборка не будет кластеризована.

Структура алгоритма следующая.

1. Случайно выбирается текущий объект из выборки.
2. Помечаются объекты выборки, находящиеся на расстоянии менее чем R от текущего объекта.
3. Вычисляется их центр тяжести, этот центр помечается как новый текущий объект.
4. Повторяются шаги 2—3, пока новый текущий объект не совпадет с прежним.
5. Помечаются объекты внутри сферы радиуса R вокруг текущего объекта как кластеризованные, затем они удаляются из выборки.
6. Повторяются шаги 1—5, пока не будет кластеризована вся выборка.

Эвристические способы выбора центра тяжести:

- в линейном пространстве — центр масс;
- в метрическом пространстве — объект, сумма расстояний до которого минимальна среди всех внутри сферы;
- объект, который внутри сферы радиуса R содержит максимальное количество других объектов из всей выборки;
- объект, который внутри сферы малого радиуса r ($r < R$) содержит максимальное количество объектов из сферы радиуса R .

Алгоритм можно охарактеризовать следующими положениями.

Доказана сходимость алгоритма за конечное число шагов.

В линейном пространстве центром тяжести может выступать произвольная точка пространства, в метрическом — только объект выборки.

Чем меньше R , тем больше кластеров.

В линейном пространстве поиск центра происходит за время $O(n)$, в метрическом — $O(n^2)$, где n — объем кластеризуемой выборки.

Наилучших результатов алгоритм достигает на выборках с хорошим выполнением условий компактности.

При повторении итераций возможно уменьшение параметра R для скорейшей сходимости.

Кластеризация сильно зависит от начального приближения (выбора объекта на первом шаге).

Рекомендуется повторная прогонка алгоритма для исключения ситуации «плохой» кластеризации по причине неудачного выбора начальных объектов.

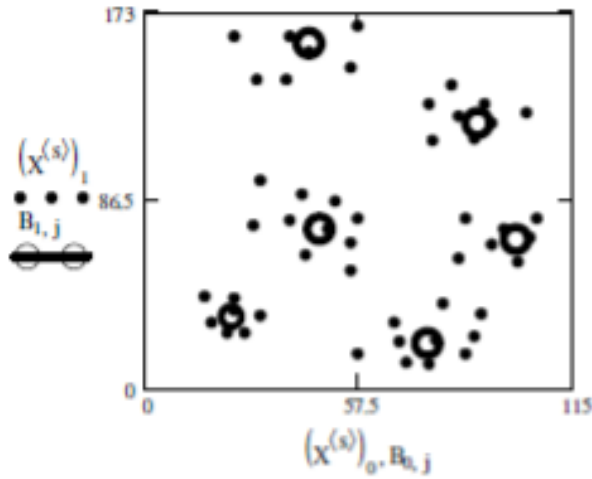
Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указание по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

$$X =$$

	0	1	2	3	4	5	6	7	8	9
0	16	18	22	24	24	24	27	29	30	31
1	42	30	25	36	41	162	25	75	142	33
2	1	1	1	1	1	2	1	3	0	1



2.4. Функция распределения оставшихся нераспределенных элементов по классам методом ближайшего соседа

```

dopdist(B, X) =
  for s ∈ 0..cols(X) - 1
    if X2,s = 0
      for j ∈ 0..cols(B) - 1
        mdj ← 100000000
        for i ∈ 0..cols(X) - 1
          R ← (mdj > d(X(i), X(s))) · (X2,i = j + 1)
          mdj ← d(X(i), X(s)) if R
        mn ← 100000000
        for r ∈ 0..cols(B) - 1
          if mdr < mn
            mn ← mdr
            k ← r
        X2,s ← k + 1
        B4,k ← B4,k + 1
  (B)
  (X)

```

$D := \text{dopdistr}(B, X)$

$B := D_0 \quad X := D_1$

$B = \begin{pmatrix} 23.143 & 43.8 & 46.667 & 75.556 & 89.167 & 99.4 \\ 33.143 & 158.8 & 73.333 & 20.667 & 122.333 & 68.8 \\ 37.343 & 178.7 & 84.5 & 123.028 & 72.867 & 33.133 \\ 48.476 & 62.2 & 129.75 & 61.25 & 45.467 & 58.075 \\ 7 & 7 & 10 & 10 & 8 & 7 \end{pmatrix}$

- итоговый массив
центров кластеров,
дисперсий и
количества элементов в
кластерах

$X = \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 0 & 16 & 18 & 22 & 24 & 24 & 24 & 27 & 29 \\ \hline 1 & 42 & 30 & 25 & 36 & 41 & 162 & 25 & 75 \\ \hline 2 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 3 \\ \hline \end{array}$

- массив элементов с
указанием номера
кластера, к которому
элемент относится

2.5. Функция построения областей Дирихле и вычисления их площадей

Входные данные:

массив X - массив векторов данных,

массив B - массив классов,

массив A - массив с исходной картинкой

Выходные данные:

массив B - массив классов, в который добавлена строка с площадями областей Дирихле, соответствующих своим классам,

массив AA - массив, в каждой точке которого записан номер класса, к которому эта точка принадлежит.

```

square(A, B, X) :=
  for k ∈ 0..cols(B) - 1
    B5, k ← 0
    for i ∈ 0..rows(A) - 1
      for j ∈ 0..cols(A) - 1
        for k ∈ 0..cols(B) - 1
          mnk ← 10000000
          for s ∈ 0..cols(X) - 1
            R ←  $\left[ \text{db} \left[ X^{(s)}, \begin{pmatrix} i \\ j \end{pmatrix} \right] < m_{nk} \right] \cdot (X_{2, s} = k + 1)$ 
            mnk ←  $\text{db} \left[ X^{(s)}, \begin{pmatrix} i \\ j \end{pmatrix} \right]$  if R
          md ← 10000000
          for k ∈ 0..cols(B) - 1
            if mnk < md
              md ← mnk
              p ← k + 1
          AAi, j ← 0 if  $\text{db} \left[ \begin{pmatrix} i \\ j \end{pmatrix}, A \right] < md$ 
          otherwise
            AAi, j ← p
            B5, p-1 ← B5, p-1 + 1
        (B AA)T

```

$G := \text{square}(A, B, X) \quad B := G_0 \quad AA := G_1$

Итоговый массив центров кластеров с добавленной строкой площадей областей Дирихле

$$B = \begin{pmatrix} 23.143 & 43.8 & 46.667 & 75.556 & 89.167 & 99.4 \\ 33.143 & 158.8 & 73.333 & 20.667 & 122.333 & 68.8 \\ 37.343 & 178.7 & 84.5 & 123.028 & 72.867 & 33.133 \\ 48.476 & 62.2 & 129.75 & 61.25 & 45.467 & 58.075 \\ 7 & 7 & 10 & 10 & 8 & 7 \\ 1.403 \times 10^3 & 2.314 \times 10^3 & 3.54 \times 10^3 & 2.067 \times 10^3 & 2.484 \times 10^3 & 1.704 \times 10^3 \end{pmatrix}$$

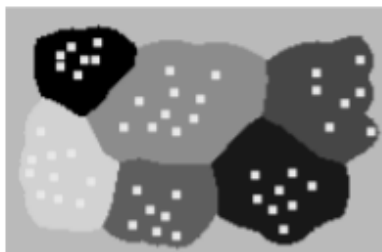
2.6. Функция "внесения" точек на изображение областей Диракле

```

point(AA,X,p) :=
  for s ∈ 0..cols(X) - 1
    for k ∈ 0..2·p - 1
      for l ∈ 0..2·p - 1
        AAX0,s-p+k,X1,s-p+l ← 253
  AA

```

AX := point(AA,X,2)



(AX + 253)·70

2.7. Функция вычисления адаптивной гистограммы

```

hist(AA,B,X) :=
  for k ∈ 0..cols(B) - 1
    for i ∈ 0..rows(A) - 1
      for j ∈ 0..cols(A) - 1
        AAi,j ←  $\frac{B_{i,k}}{\text{cols}(X) \cdot B_{j,k}}$  if AAi,j = k + 1
  AA

```

AA := hist(AA,B,X)



AA

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Приведите пример признаков, с помощью которых может быть задана кластеризуемая выборка?
2. Перечислите эвристические способы выбора центра тяжести?

ЛАБОРАТОРНАЯ РАБОТА 26

Подгонка кривых

Цель: Научиться работать в пакете CurveFitting системы MATLAB для подгонки кривых.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Обработка результатов измерений, содержащих случайные погрешности, включает 3 этапа:

1. Выбор вида аппроксимирующей зависимости.
2. Расчет коэффициентов аппроксимирующей зависимости.
3. Проверку адекватности полученной зависимости экспериментальным данным.

Наиболее совершенным программным средством для решения данной задачи является пакет программ MATLAB. MATLAB является средой разработки программ обработки данных и, одновременно, содержит большое количество готовых программ, в частности, программу подгонки кривых CurveFitting.

Программа подгонки кривых CurveFitting предоставляет следующие виды функций для аппроксимации и интерполяции одномерных массивов данных:

- а) экспоненциальную;

$$y = ae^{bx}$$

$$y = ae^{bx} + ce^{dx}$$

- б) гауссиан

$$y = \sum_{i=1}^n a_i e^{-\left(\frac{x-b_i}{c_i}\right)^2}$$

$n=1,2,\dots,10$

в) Фурье

$$y = a_0 + \sum_{i=1}^n a_i \cos(nwx) + b_i \sin(nwx)$$

$n=1,2,\dots,10$

г) полиномиальную

$$y = \sum_{i=1}^{n+1} p_i x^{n+1-i}$$

$n=1,2,\dots,10$

д) показательную

$$y = ax^b$$

$$y = a + bx^c$$

е) рациональную

$$y = \frac{\sum_{i=1}^{n+1} p_i x^{n+1-i}}{x^m + \sum_{i=1}^m q_i x^{m-i}}$$

$n, m=1,2,\dots,10$

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Освоить технологию работы с программой CurveFitting пакета Matlab.

Указания к выполнению

1. При выполнении п.1. Программы изучите теоретические основы подгонки кривых, положенные в основу программы Curve Fitting и описание библиотеки аппроксимирующих функций Fitting Library в MATLAB.

2. При выполнении п.2 выполните подгонку кривых, выберите наилучшую функциональную зависимость и обоснуйте сделанный выбор для зашумленного массива данных: файл censuN.mat

3. При работе с программой CurveFitting подгонку кривых производите в следующей последовательности:

а) Запустите MATLAB;

б) откройте окно Workspace (View/Workspace) и загрузите в это окно mat-файл данных по заданию преподавателя;

в) откройте панель CurveFitting для чего выберите Start/ Toolboxes/ CurveFitting в окне MATLAB;

г) на панели CurveFittingTool активизируйте кнопку Data. В окнах Xdata, Ydata станут доступными составляющие cdate (X) и pop (Y) из файла census.mat;

д) произведите выбор cdate и pop, затем активизируйте кнопку CreateDataSet;

е) активизируйте кнопку Fitting;

ж) выберите вид аппроксимирующей функции и активизируйте кнопку Apply;

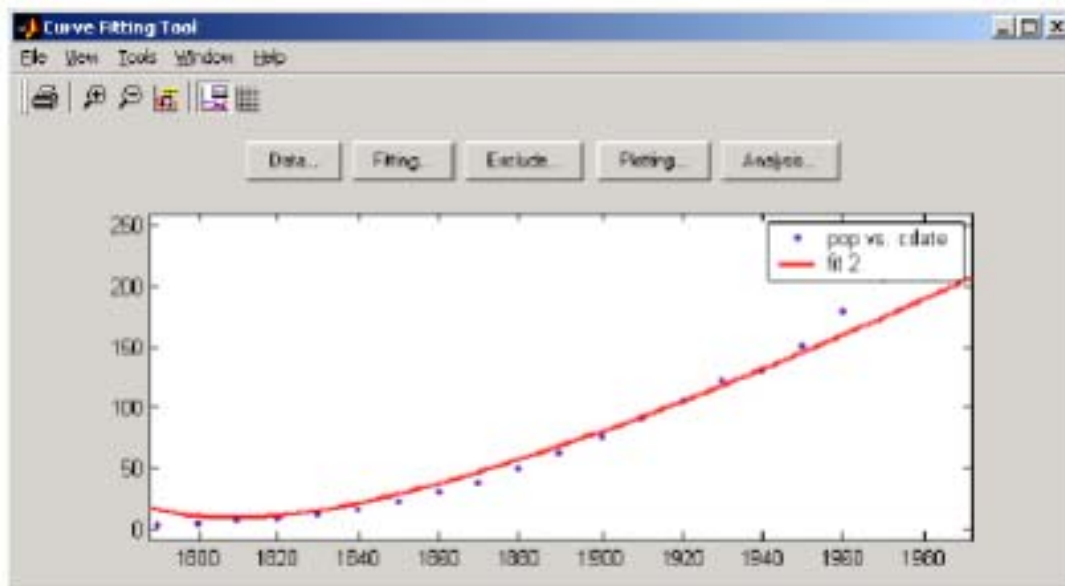


Рис. 6.1

Результаты аппроксимации будут представлены в графической форме в окне графического вывода на панели CurveFittingTool (рис.6.1) и в численной форме в окне Results на панели Tables of Fits (рис. 6.2).

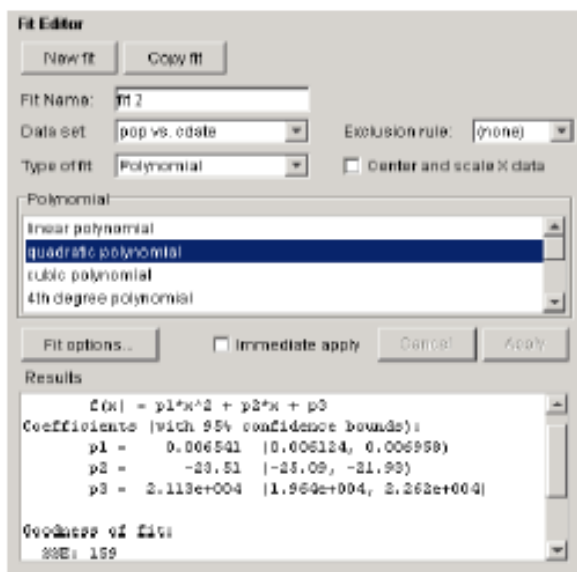


Рис. 6.2

з) выберите пункт меню View/Residuals/Line или View/Residuals/Scatter. В результате в окне графического вывода на панели CurveFittingTool будут одновременно выведен график ошибки интерполяции (см. пример на рис. 6.3).

Полученные в процессе аппроксимации результаты занесите в таблицу:

Табл. 6.1.

Вид модели	Порядок модели	Качество подгонки (+ / -)	Диапазон погрешностей	R-square	Adjusted R-square	RMSE

Исследуйте таким образом все возможные виды моделей, порядок модели - от 1 до 5.

е) по полученным и отраженным в таблице данным выберите три наилучших варианта аппроксимирующих функций. Для этих вариантов сохраните график аппроксимирующей функции и график погрешности аппроксимации, аналитические выражения с численными значениями коэффициентов для аппроксимирующей зависимости.

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;

- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Для чего используется пакет CurveFitting в MATLAB?
2. Какие виды функций предоставляет пакет CurveFitting в MATLAB?

ЛАБОРАТОРНАЯ РАБОТА 27

Оценка качества подгонки кривых

Цель: Научиться производить оценку качества подгонки кривых.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Критерием качества подгонки являются вычисляемые программой коэффициенты множественной детерминации (R-square и Adjusted R-square). Коэффициент R-square показывает насколько разброс данных относительно аппроксимирующей кривой может быть объяснен наличием случайных погрешностей в данных. Максимально возможное значение R-square равно 1.

Коэффициент Adjusted R-square характеризует то же самое, но учитывает количество степеней свободы дисперсий воспроизводимости и адекватности. Поэтому Adjusted R-square лучше характеризует степень соответствия экспериментальных данных аппроксимирующей кривой в том случае, если мы увеличиваем степень аппроксимирующего полинома и хотим проверить, происходит ли при этом улучшение качества подгонки. Максимально возможное значение Adjusted R-square также равно 1.

$$SSE = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2 \quad \text{SSE – sum square error}$$

$$SST = \sum_{i=1}^n w_i (y_i - \bar{y})^2 \quad \text{SST – sum square total,}$$

$$w_i = \frac{1}{\sigma_i^2} \quad \text{- коэффициент, выравнивающий степень разброса данных в}$$

опытах

$\hat{y}$ – значение полученное по аппроксимирующей зависимости

$$R - square = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$Adjusted R - square = 1 - \frac{SSE(n-1)}{SST(n-m)}$$

n – количество результатов измерений;

m – количество членов в уравнении регрессии;

$n-1$ – количество степеней свободы дисперсии SSE;

$n-m$ – количество степеней свободы дисперсии SST.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Оценку качества подгонки кривых и окончательный выбор аппроксимирующей зависимости производите последовательно:

1. Выберите «хорошие» модели.

1.1. Визуально, по степени согласованности графика полученной в результате подгонки аппроксимирующей кривой с отображенными в этом же окне графического вывода значениями y_i . Качество подгонки считается удовлетворительным, если значения y_i «вытянуты» вдоль аппроксимирующей кривой. На рис. 6.3 приведены результаты подгонки, которые можно считать неудовлетворительными (А) и удовлетворительными (Б).

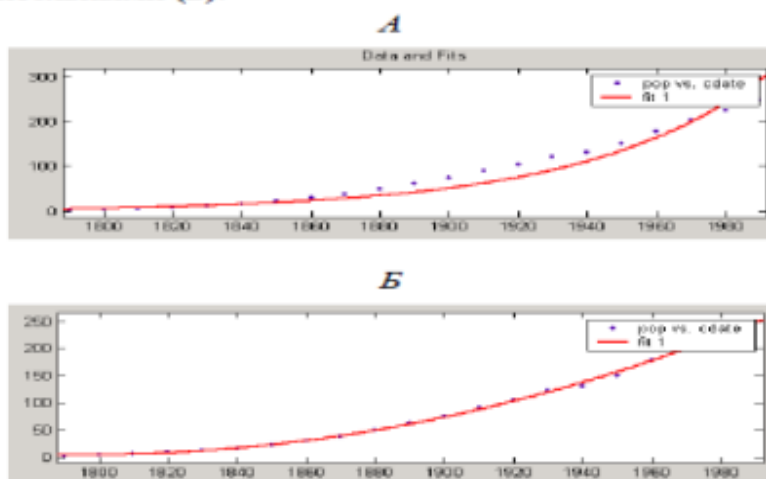


Рис. 6.3.

2. Выберите лучшую кривую в каждом классе аппроксимирующих зависимостей – по показателю Adjusted R-square. Лучшей кривой в классе является та, которой соответствует максимальное значение Adjusted R-square. Если максимальное значение соответствует двум или более кривым – выбирается кривая с наиболее простым аналитическим описанием.

3. Окончательный выбор самой лучшей аппроксимирующей зависимости – выбрать самую простую модель из лучших с условием примерно тех же значений диапазона погрешностей, R-square и RMSE (Root Mean Squared Error).

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что является критерием качества подгонки кривых?
2. Что характеризует коэффициент Adjusted R-square?

ЛАБОРАТОРНАЯ РАБОТА 28

Частотно-временной анализ нестационарных сигналов методом Гильберта-Хуанга

Цель: изучение методики частотно-временного анализа нестационарных сигналов на основе эмпирического метода декомпозиции Хуанга и частотно-временного анализа на основе преобразования Гильберта.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Под преобразованием Гильберта-Хуанга (Hilbert-Huang transform – ННТ) понимается эмпирический метод декомпозиции (EMD) нелинейных и нестационарных процессов и Гильбертов спектральный анализ (HSA). ННТ представляет собой частотно-временной анализ данных (сигналов) и не требует априорного функционального базиса. Функции базиса получаются адаптивно непосредственно из данных процедурами отсеивания функций «эмпирических мод». Мгновенные частоты вычисляются от производных фазовых функций Гильбертовым преобразованием функций базиса. Заключительный результат представляется в частотновременном пространстве EMD-HSA был предложен Норденом Хуангом в 1995 в США (NASA) для изучения поверхностных волн тайфунов, с обобщением на анализ произвольных временных рядов коллективом соавторов в 1998 г. В последующие годы, по мере расширения применения EMDHSA для других отраслей науки и техники, вместо термина EMD-HSA был принят более короткий термин преобразования: ННТ.

EMD (Empirical Mode Decomposition) - метод разложения сигналов на функции, которые получили название «эмпирических мод». Метод представляет собой итерационную вычислительную процедуру разложения исходных данных (непрерывных или дискретных сигналов) на эмпирические моды или внутренние колебания (intrinsic mode functions, IMF). IMF представляет собой колебательный режим, как часть простой гармонической функции, но вместо постоянной амплитуды и частоты, как в простой гармонике, у IMF могут быть переменная амплитуда и частота, как функции времени. IMF, определенные вышеприведенным способом, допускают вычисление физически значимых мгновенных частот, что дает возможность создать частотно-временное представление сигнала на основе преобразования Гильберта. Преобразование Гильберта позволяет разложить исходный процесс на две составляющие: амплитудную и фазовую.

Амплитуда не меняется во времени, а фаза линейно растет с коэффициентом пропорциональности, который носит название частоты. Частота определяет число периодов (повторений значений функции) в единицу времени и является постоянной во времени величиной. Амплитуда и фаза гармонической функции отражают различную информацию: амплитуда

описывает энергию, а фаза характеризует повторяемость процесса во времени, и в этом смысле они могут рассматриваться как независимые характеристики гармонической функции. Поскольку фаза зависит от времени произвольным образом, то вводится понятие мгновенной частоты.

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Имеется модельный сигнал, представляющий собой зашумленный гармонический сигнал с частотой, изменяющейся по линейному, квадратичному, кубическому закону или сумму двух зашумленных гармонических сигналов с изменяющейся частотой. Требуется исследовать условия их адекватного частотно-временного представления.

1. Для изучения эмпирической декомпозиции Хуанга возьмите за основу базовую программу (см. Приложение 1) и выполните декомпозицию для трех видов сигналов: суммы двух стационарных гармонических сигналов; нестационарного гармонического сигнала; зашумленного гармонического сигнала.

В первом случае представляет интерес возможность разделения гармонических составляющих сигналов в функциях IMF. Во втором и третьем случае представляет интерес определение зависимости оптимального количества удаляемых функций IMF для очистки сигнала от шума в зависимости от уровня шума в исходном сигнале. 2. Для изучения преобразования Гильберта возьмите за основу базовую программу (см. Приложение 2)

Приложение 1. Текст базовой программы модовой декомпозиции Хуанга
 %Эмпирическая модовая декомпозиция Хуанга.
 %Используется для очистки сигналов от шумов
 clc;%очистка Command Window
 N=2401;%количество точек
 d=0.005;%шаг изменения переменной time

```

M=(N-1)*d;%
time = 0:d:M;
Q=0.2;%уровень шума в долях СКО
NF=3;%кол-во функций разложения, содержащих шум (NF=1..3)
KP1=5;%кол-во периодов первого сигнала
KP2=7;%кол-во периодов второго сигнала
%ПРОГРАММНАЯ ГЕНЕРАЦИЯ СИГНАЛА
noise=rand(N);%генерация массива нормально распределенного шума
% noise=wgn(kt,1,0);%генерация массива белого Гауссова шума
for k=1:N % генерация гармонического сигнала
s(k) = sin(2*pi*KP1*k/N)+ sin(2*pi*KP2*k/N);%1-й вид сигнала
f(k)=s(k)+ Q*noise(k); % суммирование сигнала и шума
end
%Генерация нестационарного сигнала
s = 4*sin(10*time.*time./100.0);
f = s(1:N)+ Q*noise(1:N);
% Чтение сигнала из файла
% filename = sensor_signal.xls';%3-й вид сигнала
% x = xlsread(filename);%3-й вид сигнала
% f = x(1:N);
% f(N) = f(N-1);
% f = zeros(1,1+M/d);
% i=1:1+M/d;
% f(i)=x(i);
figure
plot(f);
title ('Исходный сигнал')
%Обнуление матрицы остатков
r = zeros(10,1+M/d);
r(1,:) = f;%Операция создания 1-й строки матрицы остатков r из массива f
%первая строка матрицы остатков равна самому модельному сигналу
%ПОЛУЧЕНИЕ ФУНКЦИЙ РАЗЛОЖЕНИЯ
for j = 1:10 %10 - максимальное количество функций разложения IMF
h = r(j,:);%r(j,:)-операция выделения строки j из матрицы r
maxLastCount = -10;
for i = 1:N % N - максимальное количество экстремумов
[ymax,imax,ymin,imin] = extrema(h);%нахождение массивов локальных
максимумов
%и минимумов с помощью стандартной функции extrema, имеющейся в
MATLAB
if((length(imax) < 2) || (length(imin) < 2))
break;
end;

```

```

    kUp = spline(time(imax),ymax,time);%сплайн-интерполяция локальных
максимумов
    kDown = spline(time(imin),ymin,time);%сплайн-интерполяция локальных
минимумов
    kMean = (kUp + kDown)/2;
    h = h - kMean;
    [one, maxCount] = size(ymax);
    if(abs(maxCount - maxLastCount) < 4)
    break;
    end;
    maxLastCount = maxCount;
    end;
    c(j,:) = h;
    %Формирование искусственных экстремумов в начале и в конце массива
    c(j,1)=c(j,2);%1-ый элемент массива Cj устанавливается равным 2-му
    c(j,2)=c(j,3)+0.1*abs(c(j,3));%2-ой элемент массива Cj равен 3-му + 10% от
модуля 3-го
    c(j,3)=c(j,4)-0.1*abs(c(j,4));%3-ий элемент массива Cj равен 4-му минус
10% от модуля 4-го
    c(j,N)=c(j,N-1); %N-й элемент массива Cj устанавливается равным N-1-му
    c(j,N-2)=c(j,N-3)+0.1*abs(c(j,N-3)); %N-2-й элемент массива Cj равен N-3-
му +10% от модуля N-3-
    го
    c(j,N-3)=c(j,N-4)-0.1*abs(c(j,N-4)) ; %N-3-й элемент массива Cj N-4-му
минус 10% от модуля N-4-
    го
    r(j + 1,:) = r(j,:) - c(j,:);
    end;
    %Вывод функций разложения IMF
    figure; plot(c(1,:));title('IMF_1');
    figure;plot(c(2,:)); title('IMF_2');
    figure; plot(c(3,:));title('IMF_3');
    figure;plot(c(4,:));title('IMF_4');
    figure; plot(c(5,:));title('IMF_5');
    figure;plot(c(6,:));title('IMF_6');
    figure;plot(c(7,:));title('IMF_7');
    %Вывод остатков
    figure; plot(r(1,:)); title('Остаток_1');
    figure;plot(r(2,:));title('Остаток_2');
    figure;plot(r(3,:));title('Остаток_3');
    figure;plot(r(4,:));title('Остаток_4');
    figure;plot(r(5,:));title('Остаток_5');
    figure;plot(r(6,:));title('Остаток_6');
    figure;plot(r(7,:));title('Остаток_7');

```

```

C1 = zeros(1,1+M/d);
%Суммирование функций разложения, содержащих только шум
for i = 1:Nf
C1=C1+c(i,:);%накопление шумовых функций IMF
end
i=1:N;
f1(i)=f(i)-C1(i);%исходный зашумленный сигнал за вычетом шумовых
IMF
%f1(i)=r(kki,i);
figure;plot(i,f(i),'r-');title('Исходный зашумленный сигнал');
hold on;
%figure
plot(i,f1(i));title('Отфильтрованный сигнал');
hold off;
% Следующий блок программы комментируется, если зашумленные
% данные берутся из файла (в этом случае)
% чистый незашумленный сигнал s неизвестен)
for i=1:N
DZ(i)=f1(i)-s(i);%уровень зашумления в сигнале после
%вычитания функций разложения, содержащих только шум
end
SKO=std(DZ)%СКО полной погрешности
KSKO=Q/SKO %коэффициент подавления шума
pause;
close all; %заккрытие всех окон графического вывода
clear; %очистка Workspace
Приложение 2.Текст базовой программы преобразования Гильберта
%Частотно-временной анализ нестационарного сигнала
%с помощью преобразования Гильберта
n=1000; % количество отсчетов
k=5 ; % количество периодов
h=zeros(1,n);
for i=1:n
h(i)= sin(3*pi*k*i*i/n/n);%линейная частотная модуляция (ЛЧМ)
% h(i)=sin(2*pi*k*i*i*i/n/n/n); %квадратичная частотная модуляция
end
figure
plot(h)
title('Исходный сигнал')
xlabel('Номер отсчета');
hx=hilbert(h); %преобразование Гильберта
phi = angle(hx); %фаза
phi2 = unwrap(phi);
p = polyfit(0:n-1,phi2,2); %аппроксимация полиномом второй степени

```

```

dp = polyder(p); %производная полинома
f = polyval(dp, 0:n-1); %Вычисление значений полинома
figure
plot(f)
title('Частотно-временное представление сигнала')
xlabel('Номер отсчета');
ylabel('Мгновенная частота');
pause
close all;
clear;

```

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что понимается под преобразованием Гильберта-Хуанга?
2. Что такое EMD (Empirical Mode Decomposition)?
3. Что подразумевается под мгновенной частотой??

ЛАБОРАТОРНАЯ РАБОТА 29

Алгоритм эмпирической декомпозиции сигнала

Цель: освоение методики получения адекватного аналитического описания зависимости $Y=f(X)$, по данным, содержащим случайные погрешности.

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Под преобразованием Гильберта-Хуанга (Hilbert-Huang transform – HHT) понимается эмпирический метод декомпозиции (EMD) нелинейных и нестационарных процессов и Гильбертов спектральный анализ (HSA). HHT представляет собой частотно-временной анализ данных (сигналов) и не требует априорного функционального базиса. Функции базиса получаются адаптивно непосредственно из данных процедурами отсеивания функций «эмпирических мод». Мгновенные частоты вычисляются от производных фазовых функций Гильбертовым преобразованием функций базиса. Заключительный результат представляется в частотновременном пространстве.

EMD-HSA был предложен Норденом Хуангом в 1995 в США (NASA) для изучения поверхностных волн тайфунов, с обобщением на анализ произвольных временных рядов коллективом соавторов в 1998 г. В последующие годы, по мере расширения применения EMDHSA для других отраслей науки и техники, вместо термина EMD-HSA был принят более короткий термин преобразования: HHT.

EMD (Empirical Mode Decomposition) - метод разложения сигналов на функции, которые получили название «эмпирических мод». Метод представляет собой итерационную вычислительную процедуру разложения исходных данных (непрерывных или дискретных сигналов) на эмпирические моды или внутренние колебания (intrinsic mode functions, IMF). IMF представляет собой колебательный режим, как часть простой гармонической функции, но вместо постоянной амплитуды и частоты, как в простой гармонике, у IMF могут быть переменная амплитуда и частота, как функции времени.

IMF, определенные вышеприведенным способом, допускают вычисление физически значимых мгновенных частот, что дает возможность создать частотно-временное представление сигнала на основе преобразования Гильберта. Преобразование Гильберта позволяет разложить исходный процесс на две составляющие: амплитудную и фазовую. Амплитуда не меняется во времени, а фаза линейно растет с коэффициентом пропорциональности, который носит название частоты. Частота определяет число периодов (повторений значений функции) в единицу времени и является постоянной во времени величиной.

Амплитуда и фаза гармонической функции отражают различную информацию: амплитуда описывает энергию, а фаза характеризует повторяемость процесса во времени, и в этом смысле они могут рассматриваться как независимые характеристики гармонической функции. Поскольку фаза зависит от времени произвольным образом, то вводится понятие мгновенной частоты.

Где $\varphi(t)$ — это фаза процесса.

Преобразование Гильберта действительной функции $x(t)$, определенной во всей временной области $-\infty < t < \infty$, есть действительная функция $\tilde{x}(t)$, также определенная на всей временной оси и задаваемая формулой:

$$\tilde{x}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau$$

Применение преобразования Гильберта позволяет выполнять квадратурную модуляцию сигналов, в каждой текущей координате модулированных сигналов производить определение огибающей и мгновенной фазы и частоты сигналов, выполнять анализ каузальных систем обработки сигналов.

Декомпозиция Хуанга

Декомпозиция основана на предположении, что любые данные состоят из различных внутренних колебаний. В любой момент времени данные могут иметь различные сосуществующие внутренние колебания, нанесенные одно на другое. Каждое колебание, линейное или нелинейное, представляет собой модовую функцию, которая имеет экстремумы и нулевые пересечения. Кроме того, колебание будет в определенной степени «симметрично» относительно локального среднего значения. Результат – конечные сложные данные.

Допустим, что имеется произвольный сигнал $x(t)$. Сущность метода EMD заключается в последовательном вычислении функций эмпирических мод $c_j(t)$ и остатков $r_j(t) = r_{j-1}(t) - c_j(t)$, где $j = 1, 2, 3, \dots, n$ при $r_0 = x(t)$. Результатом разложения будет представление сигнала в виде суммы модовых функций и конечного остатка.

Входной сигнал $x(t)$ раскладывается по базису, который удовлетворяет всем традиционным требованиям базиса. На основании проверки на модельных и опытных данных он является:

- законченным,
- сходящимся,
- ортогональным,
- единственным.

Также он является адаптивным, так как получен непосредственно из анализируемых данных эмпирическим методом.

Алгоритм эмпирической декомпозиции сигнала⁴

Операция 1. Идентифицируем по координатам и амплитудам все локальные экстремумы (максимумы и минимумы) сигнала. Группируем отдельно массивы векторов координат (номеров отсчетов) $x_{\max}(k)$ и соответствующих амплитудных значений $y_{\max}(k)$ максимумов, и аналогичные массивы векторов $x_{\min}(k)$ и $y_{\min}(k)$ минимумов всех выделенных экстремумов.

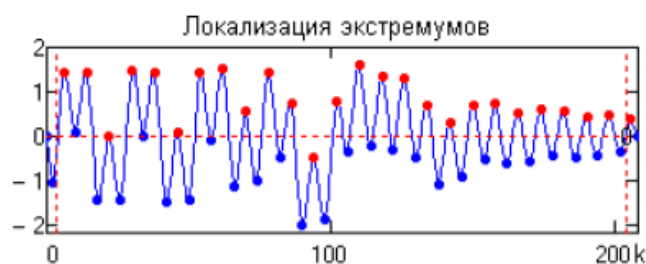


Рис. 7.1

Операция 2. Кубическим (или каким либо другим) сплайном вычисляем верхнюю и нижнюю огибающие сигнала по выделенным максимумам и минимумам, как это показано на рисунке (красный и синий цвет соответственно). Определяем функцию средних значений $m_1(k)$ между огибающими (черный цвет) и находим первое приближение к первой функции IMF:

$$h_1(k) = y(k) - m_1(k).$$

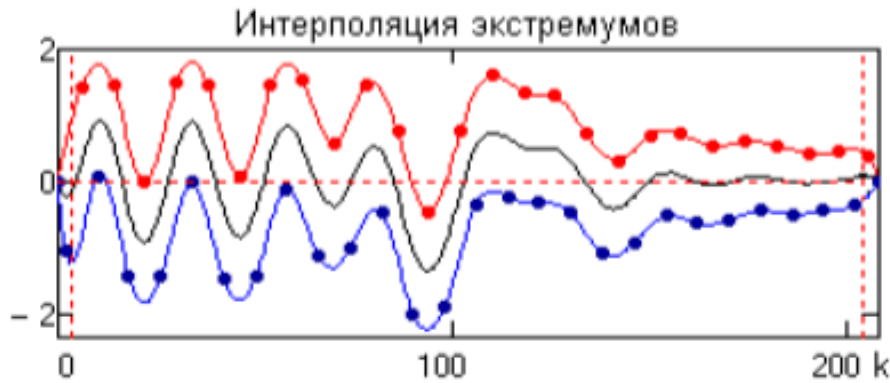


Рис. 7.2.

Операция 3. Повторяем операции 1 и 2, принимая вместо $y(k)$ функцию $h_1(k)$, и находим второе приближение к первой функции IMF – функцию $h_2(k)$.

$$h_2(k) = h_1(k) - m_2(k).$$

Аналогично находим третье и последующие приближения к первой функции IMF. По мере увеличения количества итераций функция $m_i(k)$, равно как и функция $h_i(k)$, стремится к неизменяемой форме. С учетом этого, естественным критерием останова итераций является задание определенного предела по нормализованной квадратичной разности между двумя последовательными операциями приближения, определяемой как

$$d = \frac{\sum_k |h_i(k) - h_{i-1}(k)|}{\sum_k h_{i-1}^2(k)}$$

$$\sum_k^{n_{i-1}(\infty)}$$

Результат разложения

Последнее значение $h_i(k)$ итераций принимается за наиболее высокочастотную функцию $c_1(k) = h_i(k)$ семейства IMF, которая непосредственно входит в состав исходного сигнала $y(k)$. Это позволяет вычлест $c_1(k)$ из состава сигнала и оставить в нем более низкочастотные составляющие:

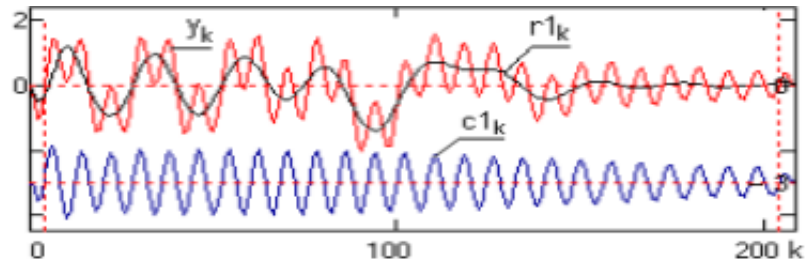


Рис. 7.3.

$$r_1(k) = y(k) - c_1(k).$$

Функция $r_1(k)$ обрабатывается как новые данные по аналогичной методике с нахождением второй функции IMF – $c_2(k)$, после чего процесс продолжается:

$$r_2(k) = r_1(k) - c_2(k), \text{ и т.д.}$$

Таким образом, достигается декомпозиция сигнала в n – эмпирическом приближении: $y(t) = c_n(t) + r_n(t)$.

Критерии останова процесса декомпозиции

1. Остаток $rn(k)$ во всем интервале задания сигнала становятся несущественными по своим значениям по сравнению с сигналом.
2. Остаток $rn(k)$ становится монотонной функцией, из которой больше не может быть извлечено функций IMF.
3. Так как в конечном итоге суммирование всех функций IMF (реконструкция сигнала) должно давать исходный сигнал, то можно останавливать разложение заданием относительной погрешности среднеквадратической реконструкции (без учета остатка $rn(k)$).
4. По мере увеличения количества функций IMF относительная среднеквадратическая погрешность реконструкции достаточно сложных и протяженных сигналов уменьшается, но, как правило, имеет определенный минимум. По-видимому, это определяется попытками алгоритма разложить остаток на функции, частично компенсирующие друг друга. Соответственно, останов программы может выполняться, если следующая выделенная функция IMF увеличивает погрешность реконструкции.

Практический критерий останова процесса декомпозиции

Другими словами, остановка декомпозиции сигнала должна происходить при максимальном «выпрямлении» остатка, т.е. превращения его в тренд сигнала по интервалу задания с числом экстремумов не более 3. Даже для данных с нулевым средним значением конечный остаток может отличаться от

нуля. Чтобы применять метод EMD, центрирования данных не требуется, метод нуждается только в локализациях экстремумов. Нулевая линия для каждого компонента декомпозиции будет сформирована процессом отсеивания. Очистка сигналов от шумов с использованием разложения Хуанга. Особенностью сигналов, получаемых при физических экспериментах, является то, что они, как правило, нестационарные и значительно зашумлены. Применение традиционных способов очистки сигналов от шумов с помощью частотных фильтров (сглаживающих, медианных, Баттерворта и др.) приводят к значительному искажению формы сигнала, что может приводить к ошибочной физической интерпретации изучаемого процесса.

Известен другой способ очистки сигналов от шумов, основанный на использовании эмпирической модовой декомпозиции Хуанга [7,8]. Способ заключается в разложении исходного сигнала на составляющие (моды), затем в исключении первых составляющих (от 1 до 3), которые обычно представляют собой шум, содержащийся в исходном сигнале.

Преобразование Гильберта

Одним из достоинств преобразования Гильберта является то, что с помощью него можно $\varphi(t) = \arctan \left[\frac{Hs(t)}{s(t)} \right]$ вычислить мгновенную частоту сигнала.

Для этого используется понятие аналитического сигнала. Аналитический сигнал – комплексный сигнал $z(t) = s(t) + jHs(t)$, где $s(t)$ – исходный сигнал, который представляется в качестве действительной части комплексного сигнала, $Hs(t)$ – мнимая часть и является преобразованием Гильберта исходного сигнала $s(t)$. Тогда амплитуда сигнала $a(t)$ и фаза $\varphi(t)$ вычисляются по формулам:

$$a(t) = \sqrt{s(t)^2 + Hs(t)^2}$$

Мгновенная частота находится путем вычисления производной от фазы по времени. На рис.7.4 приведен пример частотно-временного представления сигнала с линейной частотной модуляцией.

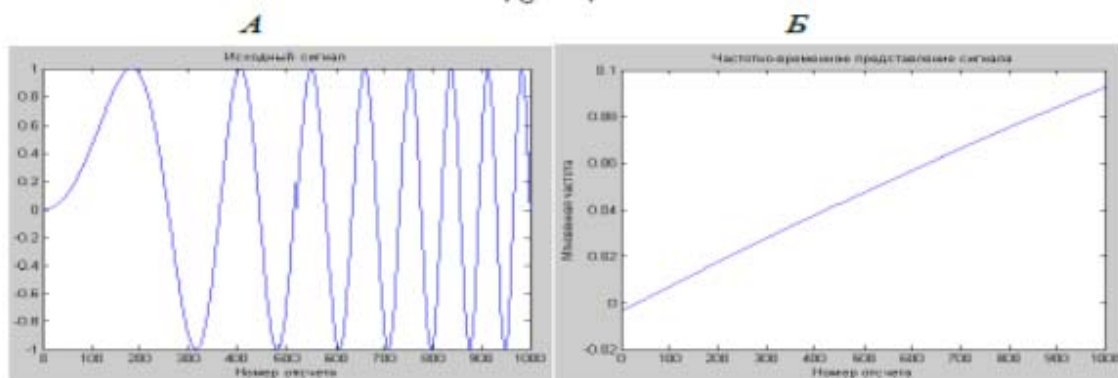


Рис. 7.4. Сигнал с линейной частотной модуляцией (А) и его частотно-временное представление (Б).

Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и

выше; Microsoft Office13 и выше, системы компьютерной математики Machcad и MATLAB R2017a и выше.

Указания по технике безопасности. Студенты должны следовать общепринятой технике безопасности для пользователей персональных компьютеров. Не следует самостоятельно производить ремонт технических средств, установку и удаление программного обеспечения. В случае обнаружения неисправностей необходимо сообщить об этом администратору компьютерного класса (обслуживающему персоналу лаборатории).

Методика и порядок выполнения работы

Имеется модельный сигнал, представляющий собой зашумленный гармонический сигнал с частотой, изменяющейся по линейному, квадратичному, кубическому закону или сумму двух зашумленных гармонических сигналов с изменяющейся частотой. Требуется исследовать условия их адекватного частотно-временного представления.

1. Для изучения эмпирической декомпозиции Хуанга возьмите за основу базовую программу (см. Приложение 1) и выполните декомпозицию для трех видов сигналов: суммы двух стационарных гармонических сигналов; нестационарного гармонического сигнала; зашумленного гармонического сигнала.

В первом случае представляет интерес возможность разделения гармонических составляющих сигналов в функциях IMF. Во втором и третьем случае представляет интерес определение зависимости оптимального количества удаляемых функций IMF для очистки сигнала от шума в зависимости от уровня шума в исходном сигнале.

Приложение 1. Текст базовой программы модовой декомпозиции Хуанга

%Эмпирическая модовая декомпозиция Хуанга.

%Используется для очистки сигналов от шумов

clc;%очистка Command Window

N=2401;%количество точек

d=0.005;%шаг изменения переменной time

M=(N-1)*d;%

time = 0:d:M;

Q=0.2;%уровень шума в долях СКО

NF=3;%кол-во функций разложения, содержащих шум (NF=1..3)

KP1=5;%кол-во периодов первого сигнала

KP2=7;%кол-во периодов второго сигнала

%ПРОГРАММНАЯ ГЕНЕРАЦИЯ СИГНАЛА

noise=rand(N);%генерация массива нормально распределенного шума

% noise=wgn(kt,1,0);%генерация массива белого Гауссова шума

for k=1:N % генерация гармонического сигнала

s(k) = sin(2*pi*KP1*k/N)+ sin(2*pi*KP2*k/N);%1-й вид сигнала

```

f(k)=s(k)+ Q*noise(k); % суммирование сигнала и шума
end
%Генерация нестационарного сигнала
s = 4*sin(10*time.*time./100.0);
f = s(1:N)+ Q*noise(1:N);
% Чтение сигнала из файла
% filename = sensor_signal.xls';%3-й вид сигнала
% x = xlsread(filename);%3-й вид сигнала
% f = x(1:N);
% f(N) = f(N-1);
% f = zeros(1,1+M/d);
% i=1:1+M/d;
% f(i)=x(i);
figure
plot(f);
title ('Исходный сигнал')
%Обнуление матрицы остатков
r = zeros(10,1+M/d);
r(1,:) = f;%Операция создания 1-й строки матрицы остатков r из массива f
%первая строка матрицы остатков равна самому модельному сигналу
%ПОЛУЧЕНИЕ ФУНКЦИЙ РАЗЛОЖЕНИЯ
for j = 1:10 %10 - максимальное количество функций разложения IMF
h = r(j,:);%r(j,:)-операция выделения строки j из матрицы r
maxLastCount = -10;
for i = 1:N % N - максимальное количество экстремумов
[ymax,imax,ymin,imin] = extrema(h);%нахождение массивов локальных
максимумов
%и минимумов с помощью стандартной функции extrema, имеющейся в
MATLAB
if((length(imax) < 2) || (length(imin) < 2))
break;
end;
kUp = spline(time(imax),ymax,time);%сплайн-интерполяция локальных
максимумов
kDown = spline(time(imin),ymin,time);%сплайн-интерполяция локальных
минимумов
kMean = (kUp + kDown)/2;
h = h - kMean;
[one, maxCount] = size(ymax);
if(abs(maxCount - maxLastCount) < 4)
break;
end;
maxLastCount = maxCount;
end;

```

```

c(j,:) = h;
%Формирование искусственных экстремумов в начале и в конце массива
c(j,1)=c(j,2);%1-ый элемент массива Cj устанавливается равным 2-му
c(j,2)=c(j,3)+0.1*abs(c(j,3));%2-ой элемент массива Cj равен 3-му + 10% от
модуля 3-го
c(j,3)=c(j,4)-0.1*abs(c(j,4));%3-ий элемент массива Cj равен 4-му минус
10% от модуля 4-го
c(j,N)=c(j,N-1); %N-й элемент массива Cj устанавливается равным N-1-му
c(j,N-2)=c(j,N-3)+0.1*abs(c(j,N-3)); %N-2-й элемент массива Cj равен N-3-
му +10% от модуля N-3-
го
c(j,N-3)=c(j,N-4)-0.1*abs(c(j,N-4)) ; %N-3-й элемент массива Cj N-4-му
минус 10% от модуля N-4-
го
r(j + 1,:) = r(j,:) - c(j,:);
end;
%Вывод функций разложения IMF
figure; plot(c(1,:));title('IMF_1');
figure;plot(c(2,:)); title('IMF_2');
figure; plot(c(3,:));title('IMF_3');
figure;plot(c(4,:));title('IMF_4');
figure; plot(c(5,:));title('IMF_5');
figure;plot(c(6,:));title('IMF_6');
figure;plot(c(7,:));title('IMF_7');
%Вывод остатков
figure; plot(r(1,:)); title('Остаток_1');
figure;plot(r(2,:));title('Остаток_2');
figure;plot(r(3,:));title('Остаток_3');
figure;plot(r(4,:));title('Остаток_4');
figure;plot(r(5,:));title('Остаток_5');
figure;plot(r(6,:));title('Остаток_6');
figure;plot(r(7,:));title('Остаток_7');
C1 = zeros(1,1+M/d);
%Суммирование функций разложения, содержащих только шум
for i = 1:Nf
C1=C1+c(i,:);%накопление шумовых функций IMF
end
i=1:N;
f1(i)=f(i)-C1(i);%исходный зашумленный сигнал за вычетом шумовых
IMF
%f1(i)=r(kki,i);
figure;plot(i,f(i),'r-');title('Исходный зашумленный сигнал');
hold on;
%figure

```

```

plot(i,f1(i));title('Отфильтрованный сигнал');
hold off;
% Следующий блок программы комментируется, если зашумленные
% данные берутся из файла (в этом случае
% чистый незашумленный сигнал s неизвестен)
for i=1:N
DZ(i)=f1(i)-s(i);%уровень зашумления в сигнале после
%вычитания функций разложения, содержащих только шум
end
SKO=std(DZ)%СКО полной погрешности
KSKO=Q/SKO %коэффициент подавления шума
pause;
close all; %заккрытие всех окон графического вывода
clear; %очистка Workspace
Приложение 2.Текст базовой программы преобразования Гильберта
%Частотно-временной анализ нестационарного сигнала
%с помощью преобразования Гильберта
n=1000; % количество отсчетов
k=5 ; % количество периодов
h=zeros(1,n);
for i=1:n
h(i)= sin(3*pi*k*i*i/n/n);%линейная частотная модуляция (ЛЧМ)
% h(i)=sin(2*pi*k*i*i*i/n/n/n); %квадратичная частотная модуляция
end
figure
plot(h)
title('Исходный сигнал')
xlabel('Номер отсчета');
hx=hilbert(h); %преобразование Гильберта
phi = angle(hx); %фаза
phi2 = unwrap(phi);
p = polyfit(0:n-1,phi2,2); %аппроксимация полиномом второй степени
dp = polyder(p); %производная полинома
f = polyval(dp, 0:n-1); %Вычисление значений полинома
figure
plot(f)
title('Частотно-временное представление сигнала')
xlabel('Номер отсчета');
ylabel('Мгновенная частота');
pause
close all;
clear;

```

Содержание отчета и его форма

Подготовьте отчет, в котором приведите технологию выполнения заданий.

Отчет по лабораторной работе должен содержать:

- 1) название работы;
- 2) цель лабораторной работы;
- 3) формулировку задания и технологию его выполнения;
- 4) ответы на контрольные вопросы;
- 5) приложение – файлы выполненных заданий.

Вопросы для защиты работы

1. Что такое декомпозиция Хуанга?
2. Опишите алгоритм эмпирической декомпозиции сигнала?
3. Перечислите критерии останова процесса декомпозиции?
4. Укажите одно важное достоинство преобразования Гильберта?

ЛАБОРАТОРНАЯ РАБОТА 30

Построение Байесовского классификатора по выборке двумерных нормально распределенных векторов. Построение Байесовского классификатора по прецедентам

Цель: изучение Байесовского классификатора

Формируемые компетенции: ПК-8 ПК-10 ПК-12

Теоретическое обоснование

Построение байесовского классификатора по выборке двумерных нормально распределенных векторов

1. Априорные вероятности появления классов

$$p1 \approx 0.4$$

$$p2 \approx 0.6$$

2. Генерирование двумерных обучающих векторов, распределенных по нормальному закону

2.1. Задание параметров нормального сферического распределения

$$mx1 \approx 0 \quad my1 \approx 0 \quad \sigma x1 \approx 2 \quad \sigma y1 \approx 1 \quad \text{- параметры нормального распределения в первом классе}$$

$$mx2 \approx 4 \quad my2 \approx 3 \quad \sigma x2 \approx 2 \quad \sigma y2 \approx 1 \quad \text{- параметры нормального распределения во втором классе}$$

2.2. Генерирование центрированных нормально распределенных значений - координат случайных векторов

$$N \approx 50 \quad \text{- размер выборки}$$

$$X1 \approx \text{rnorm}(N, 0, \sigma x1) \quad X2 \approx \text{rnorm}(N, 0, \sigma x2)$$

$$Y1 \approx \text{rnorm}(N, 0, \sigma y1) \quad Y2 \approx \text{rnorm}(N, 0, \sigma y2)$$

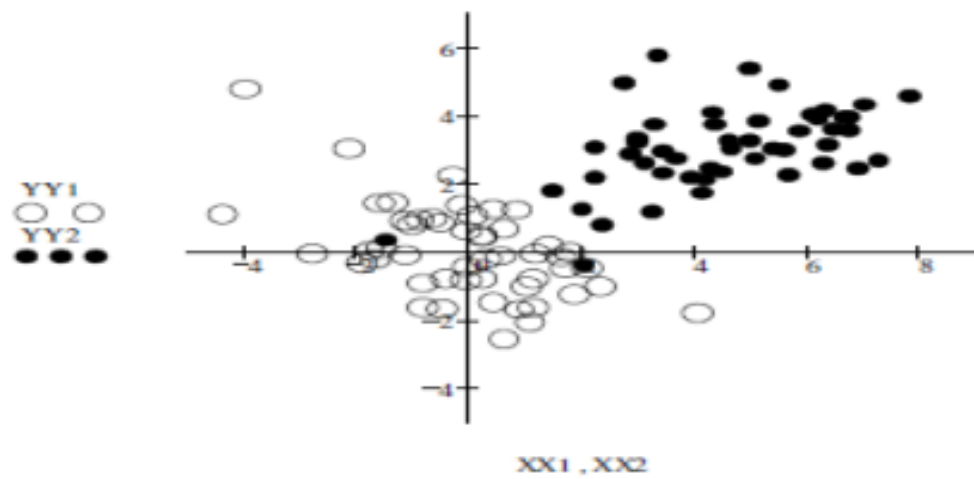
2.3. Поворот и смещение нормально распределенных векторов

$$a1 \approx \frac{\pi}{6} \quad a2 \approx \frac{5 \cdot \pi}{6} \quad \text{- углы поворота эллипсов рассеяния}$$

$$i \approx 0..N-1$$

$$XX1_i \approx X1_i \cdot \cos(a1) + Y1_i \cdot \sin(a1) + mx1 \quad XX2_i \approx X2_i \cdot \cos(a2) + Y2_i \cdot \sin(a2) + mx2$$

$$YY1_i \approx -X1_i \cdot \sin(a1) + Y1_i \cdot \cos(a1) + my1 \quad YY2_i \approx -X2_i \cdot \sin(a2) + Y2_i \cdot \cos(a2) + my2$$



3. Вычисление оценок числовых характеристик распределений векторов в классах

3.1. Вычисление оценки центра рассеяния в первом классе

$$MX1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} XX1_i \quad MY1 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} YY1_i$$

$$MX1 = -0.083 \quad MY1 = 0.019 \quad M1 := \begin{pmatrix} MX1 \\ MY1 \end{pmatrix}$$

3.2. Вычисление оценки центра рассеяния во втором классе

$$MX2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} XX2_i \quad MY2 := \frac{1}{N} \cdot \sum_{i=0}^{N-1} YY2_i$$

$$MX2 = 4.58 \quad MY2 = 3.037 \quad M2 := \begin{pmatrix} MX2 \\ MY2 \end{pmatrix}$$

3.3. Вычисление оценок элементов ковариационной матрицы в первом классе

$$K_{XX1} := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (X_{X1i} - M_{X1})^2 \quad K_{YY1} := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (Y_{Y1i} - M_{Y1})^2$$

$$K_{XY1} := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (X_{X1i} - M_{X1}) \cdot (Y_{Y1i} - M_{Y1})$$

$$S1 := \begin{pmatrix} K_{XX1} & K_{XY1} \\ K_{XY1} & K_{YY1} \end{pmatrix} \quad S1 = \begin{pmatrix} 2.388 & -1.122 \\ -1.122 & 1.762 \end{pmatrix}$$

3.4. Вычисление оценок элементов ковариационной матрицы во втором классе

$$K_{XX2} := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (X_{X2i} - M_{X2})^2 \quad K_{YY2} := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (Y_{Y2i} - M_{Y2})^2$$

$$K_{XY2} := \frac{1}{N} \cdot \sum_{i=0}^{N-1} (X_{X2i} - M_{X2}) \cdot (Y_{Y2i} - M_{Y2})$$

$$S2 := \begin{pmatrix} K_{XX2} & K_{XY2} \\ K_{XY2} & K_{YY2} \end{pmatrix} \quad S2 = \begin{pmatrix} 3.878 & 1.235 \\ 1.235 & 1.445 \end{pmatrix}$$

- оценки ковариационных матриц

$$S1^{-1} = \begin{pmatrix} 0.598 & 0.381 \\ 0.381 & 0.81 \end{pmatrix} \quad S2^{-1} = \begin{pmatrix} 0.354 & -0.303 \\ -0.303 & 0.951 \end{pmatrix}$$

- обратные к ковариационным матрицы

4. Вычисление решающих функций, определяемых байесовским классификатором

4.1. Определение метрик Махалоубиса в каждом классе

$$Mach_1(x, y) := (S1^{-1})_{0,0} \cdot x^2 + 2 \cdot (S1^{-1})_{0,1} \cdot x \cdot y + (S1^{-1})_{1,1} \cdot y^2$$

$$Mach_2(x, y) := (S2^{-1})_{0,0} \cdot x^2 + 2 \cdot (S2^{-1})_{0,1} \cdot x \cdot y + (S2^{-1})_{1,1} \cdot y^2$$

4.2. Определение решающих функций

$$dk1(x, y) := \ln(p1) - 0.5 \cdot \ln(|S1|) - 0.5 \cdot Mach_1(x - M_{X1}, y - M_{Y1})$$

$$dk2(x, y) := \ln(p2) - 0.5 \cdot \ln(|S2|) - 0.5 \cdot Mach_2(x - M_{X2}, y - M_{Y2})$$

$$d(x, y) := dk1(x, y) - dk2(x, y)$$

5. Прорисовка линии разделения двух классов

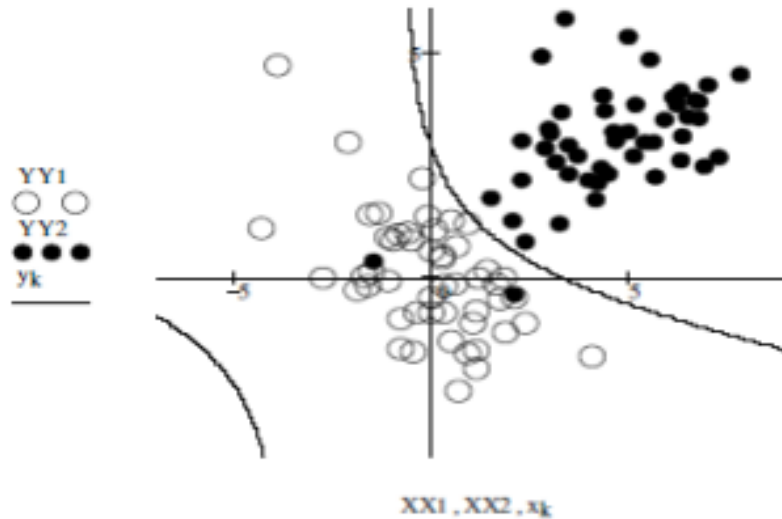
$$n := 10000$$

$$k := 0..n-1$$

$$a := -7 \quad b := 9$$

$$x_k := a + \frac{k \cdot (b - a)}{n}$$

$$y0 := -4 \quad y_k := \text{root}(d(x_k, y0), y0)$$



Приложение 5. Построение байесовского классификатора по прецедентам

1. Формирование обучающей выборки по изображению точечных данных

1.1. Считывание точечных данных из файла

$A := \text{READBMP}("d\backslash\text{выборка_2}")$

1.2. Выделение точек из матрицы изображения, формирование массива точечных данных X

quant(A, a, b) := $\begin{cases} s \leftarrow 0 \\ \text{for } i \in 0.. \text{rows}(A) - 1 \\ \quad \text{for } j \in 0.. \text{cols}(A) - 1 \\ \quad \quad \text{if } (a \leq A_{i,j}) \cdot (A_{i,j} \leq b) \\ \quad \quad \quad \left| \begin{array}{l} X^{(s)} \leftarrow \begin{pmatrix} i \\ j \end{pmatrix} \\ s \leftarrow s + 1 \end{array} \right. \\ \end{cases}$
X

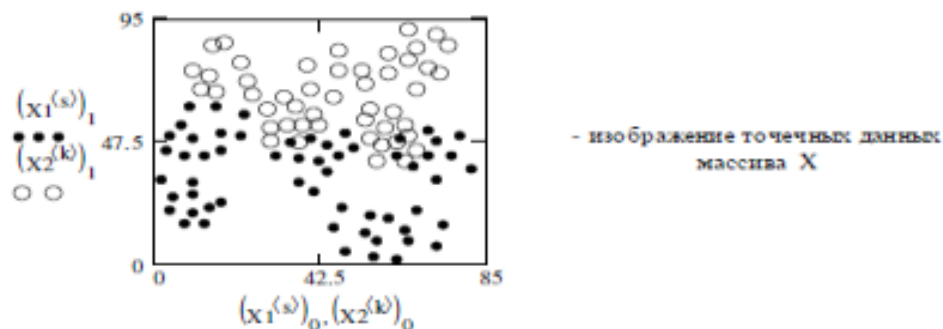
$X1 := \text{quant}(A, 0, 50)$ $X1 =$

	0	1	2	3	4	5	6	7	8	9
0	2	3	4	4	5	7	8	8	9	10
1	33	44	21	50	26	54	16	42	61	20

$X2 := \text{quant}(A, 50, 200)$ $X2 =$

	0	1	2	3	4	5	6	7	8	9
0	10	12	14	15	16	18	22	24	25	29
1	75	68	73	85	67	86	78	71	66	60

$s := 0.. \text{cols}(X1) - 1$ $k := 0.. \text{cols}(X2) - 1$



2. Вычисление гистограмм с нерегулярными ячейками

2.1. Распределение основного массива данных по ячейкам в соответствии со взвешенной метрикой, включение центров классов и дисперсий (см. Приложение 2)

$$\text{DISTR1} := \text{distr}(X1, 10, 8, 25) \quad B1 := \text{DISTR1}_0 \quad X1 := \text{DISTR1}_1$$

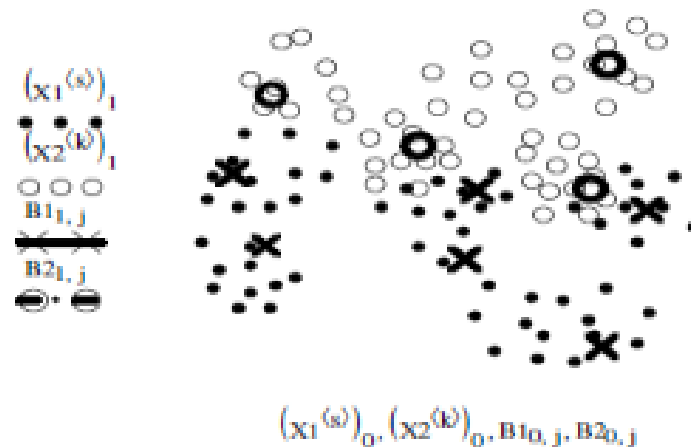
$$B1 = \begin{pmatrix} 12.5 & 7 & 46.2 & 44.333 & 66.333 & 73.571 \\ 32.25 & 51 & 46.6 & 28.667 & 6 & 41.429 \\ 31.318 & 11.75 & 19.2 & 15.083 & 29.083 & 26.952 \\ 141.295 & 10 & 13.675 & 49.333 & 13 & 36.952 \\ 12 & 3 & 5 & 3 & 3 & 7 \end{pmatrix}$$

- массив центров кластеров, дисперсий и количества элементов в кластерах для первого класса

$$\text{DISTR2} := \text{distr}(X2, 10, 8, 25) \quad B2 := \text{DISTR2}_0 \quad X2 := \text{DISTR2}_1$$

$$B2 = \begin{pmatrix} 13 & 36.909 & 67.143 & 64.4 \\ 70.75 & 58.091 & 79.143 & 47 \\ 10.667 & 28.391 & 36.81 & 10 \\ 14.917 & 40.491 & 21.476 & 29 \\ 4 & 11 & 7 & 5 \end{pmatrix}$$

- массив центров кластеров, дисперсий и количества элементов в кластерах для второго класса



2.3. Распределение оставшихся нераспределенных элементов по кластерам методом ближайшего соседа (см. Приложение 2)

$$D1 := \text{dopdistr}(B1, X1) \quad B1 := D1_0 \quad X1 := D1_1$$

$$B1 = \begin{pmatrix} 12.5 & 7 & 46.2 & 44.333 & 66.333 & 73.571 \\ 32.25 & 51 & 46.6 & 28.667 & 6 & 41.429 \\ 31.318 & 11.75 & 19.2 & 15.083 & 29.083 & 26.952 \\ 141.295 & 10 & 13.675 & 49.333 & 13 & 36.952 \\ 15 & 7 & 8 & 12 & 6 & 10 \end{pmatrix}$$

- итоговый массив центров кластеров, дисперсий и количества элементов в кластерах для первого класса

$$X1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 2 & 3 & 4 & 4 & 5 & 7 & 8 & 8 & 9 & 10 \\ 1 & 39 & 44 & 21 & 50 & 26 & 54 & 16 & 42 & 61 & 20 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 1 & 2 & 1 \end{pmatrix}$$

- массив элементов первого класса с указанием номера кластера, к которому элемент относится

$$D2 := \text{dopdistr}(B2, X2) \quad B2 := D2_0 \quad X2 := D2_1$$

$$B2 = \begin{pmatrix} 13 & 36.909 & 67.143 & 64.4 \\ 70.75 & 58.091 & 79.143 & 47 \\ 10.667 & 28.391 & 36.81 & 10 \\ 14.917 & 40.491 & 21.476 & 29 \\ 9 & 17 & 10 & 12 \end{pmatrix}$$

- итоговый массив центров кластеров, дисперсий и количества элементов в кластерах для второго класса

$$X2 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 10 & 12 & 14 & 15 & 16 & 18 & 22 & 24 & 25 & 29 \\ 1 & 75 & 68 & 73 & 85 & 67 & 86 & 78 & 71 & 66 & 60 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \end{pmatrix}$$

- массив элементов второго класса с указанием номера кластера, к которому элемент относится

2.4. Построения областей Дириссле и вычисления их площадей (см. Приложение 2)

$$G1 := \text{square}(A, B1, X1, 8) \quad B1 := G1_0 \quad AA1 := G1_1$$

$$B1 = \begin{pmatrix} 12.5 & 7 & 46.2 & 44.333 & 66.333 & 73.571 \\ 32.25 & 51 & 46.6 & 28.667 & 6 & 41.429 \\ 31.318 & 11.75 & 19.2 & 15.083 & 29.083 & 26.952 \\ 141.295 & 10 & 13.675 & 49.333 & 13 & 36.952 \\ 15 & 7 & 8 & 12 & 6 & 10 \\ 920 & 534 & 563 & 925 & 430 & 804 \end{pmatrix}$$

- итоговый массив центров кластеров первого класса с добавленной строкой площадей областей Дириссле

$$G2 := \text{square}(A, B2, X2, 8) \quad B2 := G2_0 \quad AA2 := G2_1$$

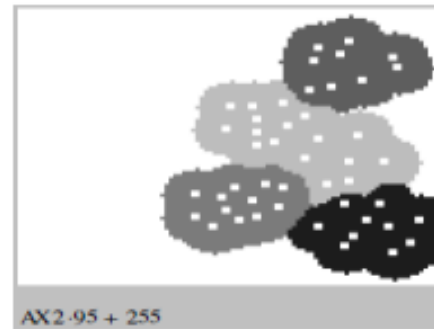
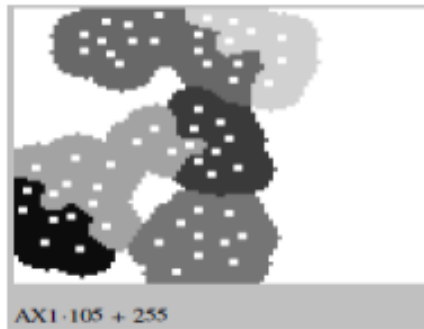
$$B2 = \begin{pmatrix} 13 & 36.909 & 67.143 & 64.4 \\ 70.75 & 58.091 & 79.143 & 47 \\ 10.667 & 28.391 & 36.81 & 10 \\ 14.917 & 40.491 & 21.476 & 29 \\ 9 & 17 & 10 & 12 \\ 751 & 1.17 \times 10^3 & 737 & 728 \end{pmatrix}$$

- итоговый массив центров кластеров второго класса с добавленной строкой площадей областей Дириссле

2.5. "Нанесение" точек на изображения областей Дириссле

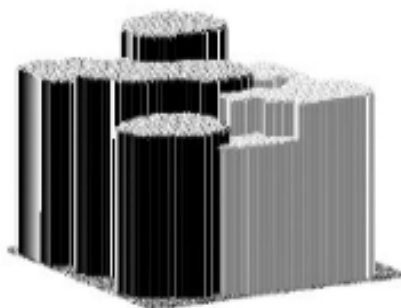
$$AX1 := \text{point}(AA1, X1, 1) \quad AX2 := \text{point}(AA2, X2, 0)$$

Области Дириссле первого и второго классов



2.6. Вычисления адаптивных гистограмм в классах (см. Приложение 2)

$$AA1 := \text{his}(AA1, B1, X1) \quad AA2 := \text{his}(AA2, B2, X2)$$



- адаптивные гистограммы плотностей распределения признаков в классах

AA1, AA2

3. Построение байесовского классификатора по гистограммам распределения признаков в классах

3.1. Задание априорных вероятностей появления классов p и $1-p$

$p := 0.5$

3.2. Функция вычисления областей предпочтения байесовской классификации

```

bayes(A1,A2,p) :=
  for i ∈ 0..rows(A1) - 1
    for j ∈ 0..cols(A1) - 1
      |
      | AAi,j ← 100 if A2i,j = 0
      | AAi,j ← 100 if  $\frac{A1_{i,j}}{A2_{i,j}} > \frac{1-p}{p}$  otherwise
      |
    AA

```

$BB := \text{bayes}(AA1, AA2, 0.5)$

3.3. Функция нахождения границы разделения областей предпочтения

```

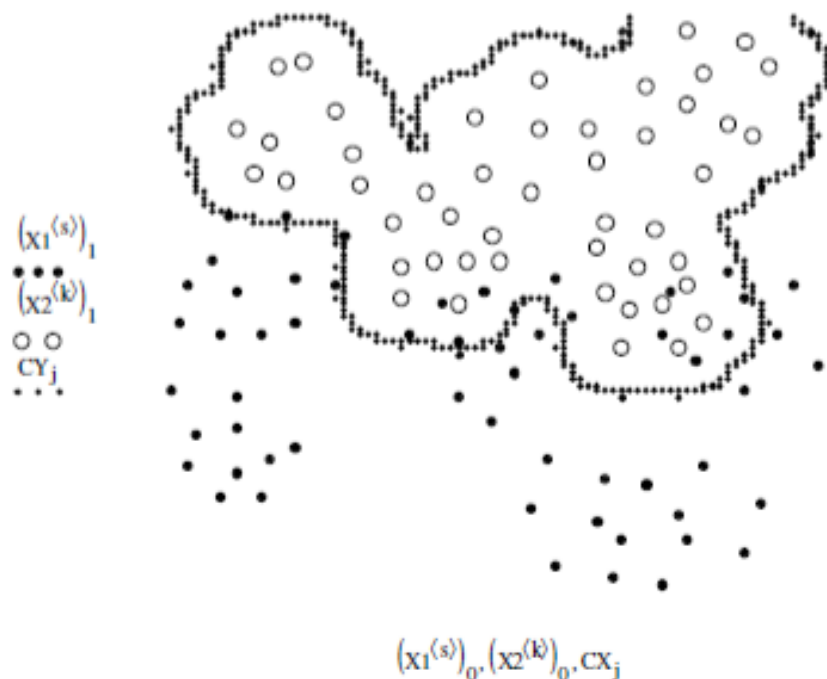
solve_curve(BB) :=
  k ← 0
  for i ∈ 1..rows(BB) - 2
    for j ∈ 1..cols(BB) - 2
      |
      | R1 ← (BBi-1,j-1 ≠ 0) + (BBi-1,j ≠ 0) + (BBi-1,j+1 ≠ 0)
      | R2 ← (BBi,j-1 ≠ 0) + (BBi,j+1 ≠ 0) + (BBi+1,j-1 ≠ 0)
      | R3 ← (BBi+1,j ≠ 0) + (BBi+1,j+1 ≠ 0)
      | if [(BBi,j = 0) · (R1 + R2 + R3)]
      |   |
      |   | CXk ← i
      |   | CYk ← j
      |   | k ← k + 1
      |
    (CX CY)T

```

$C := \text{solve_curve}(BB) \quad CX := C_0 \quad CY := C_1$

$s := 0..cols(X1) - 1 \quad k := 0..cols(X2) - 1 \quad j := 0..rows(CX) - 1$

Граница разделения областей предпочтения



Аппаратура и материалы. Для выполнения лабораторной работы необходимо использовать следующее: аппаратное обеспечение: персональный компьютер и программное обеспечение: операционную систему Windows 10 и выше; Microsoft Office 13 и выше, системы компьютерной математики Mathcad и MATLAB R2017a и выше.

Указания по технике безопасности. Самостоятельно не производить: установку и удаление программного обеспечения; ремонт персонального компьютера. Соблюдать правила технической эксплуатации и техники безопасности при работе с электрооборудованием.

Методика и порядок выполнения работы

Повторить реализованный пример в разделе теоретическое обоснование.

Содержание отчета и его форма

Отчет по лабораторной работе должен содержать:

- 6) название работы;
- 7) цель лабораторной работы;
- 8) условия выполненных заданий и их решение;
- 9) технологию решения задач с помощью прикладного программного обеспечения, выводы;
- 10) ответы на контрольные вопросы.

Вопросы для защиты работы

3. Что понимают под Байесовским классификатором?
4. В каких сферах применяют Байесовский классификатор?

ЛИТЕРАТУРА И ИСТОЧНИКИ

Основная литература

1. Иванова, В.Е., Тяжев, А.И. Цифровая обработка сигналов и сигнальные процессоры: учебное пособие: Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017 ЭБС

2. Селянкин, В.В. Компьютерное зрение. Анализ и обработка изображений: учебное пособие: Санкт-Петербург: Лань, 2019 ЭБС

Дополнительная литература

1. Капитонова, Т.А. Нейросетевое моделирование в распознавании образов: философско-методические аспекты: Минск: Белорусская наука, 2009 ЭБС

2. Шостак, А.С. Прием и обработка сигналов: курс лекций: Томск: Томский государственный университет систем управления и радиоэлектроники, 2012 ЭБС

3. Гонсалес, Р., Вудс, Р. Цифровая обработка изображений: практические советы: Москва: Техносфера, 2012 ЭБС

4. Оппенгейм, А., Шафер, Р. Цифровая обработка сигналов: Москва: Техносфера, 2012 ЭБС

5. Щетинин Ю.И. Анализ и обработка сигналов в среде MATLAB: Новосибирск: Новосибирский государственный технический университет (НГТУ), 2011 ЭБС

8.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю)

1. Руководство для преподавателей по организации и планированию различных видов занятий и самостоятельной работы обучающихся в Донском государственном техническом университете: метод. указания. Ростов н/Д.: ИЦ ДГТУ, 2018 ЭБС

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

1. <http://biblioclub.ru> – ЭБС «Университетская библиотека on-line», <http://old.exponenta.ru/> – Образовательный математический сайт по математике и программированию

2. <http://MATLAB.exponenta.ru/Simulink/default.php> – сайт, посвященный интерактивному инструменту для моделирования Simulink,

3. <https://www.intuit.ru/> – Национальный Открытый Университет «ИНТУИТ»

4. <http://window.edu.ru/> – Информационная система "Единое окно доступа к образовательным ресурсам"

5. <http://artspb.com/> – Образовательный портал: математика, кибернетика и программирование дополнительного образования