В.Н. КОПЕНКОВ, В.В. СЕРГЕЕВ

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ОБРАБОТКИ ДАННЫХ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

2007

СГАУ

САМАРА

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

В.Н. Копенков, В.В. Сергеев

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ОБРАБОТКИ ДАННЫХ

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Утверждено Редакционно-издательским советом университета в качестве учебного пособия

С А М А Р А
Издательство СГАУ
2007

УДК 681.3
ББК 22.343
К 65

Инновационная образовательная программа "Развитие центра компетенции и подготовка специалистов мирового уровня в области аэрокосмических и геоинформационных технологий"

Рецензенты: д-р техн. наук, проф. А. И. Х р а м о в,
д-р техн. наук, проф. В. Г. К а р т а ш е в с к и й

Копенков В.Н.

К 65 **Современные информационные технологии анализа и обработки данных. Лабораторный практикум:** учеб. пособие / *В.Н. Копенков, В.В.Сергеев* – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2007. – 96 с.: ил.

ISBN 978-5-7883-0577-6

Учебное пособие представляет собой сборник методических указаний к лабораторным работам, для каждой из которых приводятся описание теоретических основ, задание и список контрольных вопросов. Разделы теоретических основ содержат описания наиболее распространенных современных методов обработки и анализа данных и изображений: методов шифрования и дешифрования данных, построения геометрических и алгебраических фракталов, использования кратно-масштабного анализа и теории вейвлет-преобразования для анализа изображений и методов построения, обучения и применения нейронных сетей.
Предназначено для студентов факультета информатики, обучающихся по специальности "Прикладная математика и информатика" по курсу “Современные информационные технологии анализа изображений” и для специалистов, проходящих курсы повышения квалификации.

УДК 681.3
ББК 22.343

ISBN 978-5-7883-0577-6

Содержание

1. Лабораторная работа № 1

Использование частотного анализа для декодирования текста

Цель работы – изучение методов декодирования шифра простой замены на основе частотного анализа; получение навыков работы с шифрами.

Содержание работы:

1.1. Теоретические основы лабораторной работы

1.1.1. Простейшие коды подстановки

Криптография — тайнопись. Термин ввел *Д. Валлис.* Потребность шифровать и передавать шифрованные сообщения возникла очень давно. Так, еще в V-IV вв. до н. э. *греки* применяли специальное шифрующее устройство. По описанию *Плутарха* оно состояло из двух палок одинаковой длины и толщины. Одну оставляли себе, а другую отдавали отъезжающему. Эти палки называли *скиталами.* Когда правителям нужно было сообщить какую-нибудь важную тайну, они вырезали длинную и узкую, вроде ремня, полосу папируса, наматывали ее на свою скиталу, не оставляя на ней никакого промежутка, так чтобы вся поверхность палки была охвачена этой полосой. Затем, оставляя папирус на скитале в том виде, как он есть, писали на нем все, что нужно, а написав, снимали полосу и без палки отправляли адресату. Так как буквы на ней разбросаны в беспорядке, то прочитать написанное он мог, только взяв свою скиталу и намотав на нее без пропусков эту полосу.

Аристотелю принадлежит способ дешифрования этого шифра. Надо изготовить длинный конус и, начиная с основания, обертывать его лентой с шифрованным сообщением, постепенно сдвигая ее к вершине. В какой-то момент начнут просматриваться куски сообщения. Так можно определить диаметр скиталы.

A	B	C	D	E
F	G	H	I, J	K
L	M	N	O	P
Q	R	S	T	U
V	W	X	Y	Z

В *Древней Греции* (П в. до н. э.) был известен шифр, называемый *"квадрат Полибия"*. Это устройство представляло собой квадрат 5 x 5, столбцы и строки которого нумеровали цифрами от 1 до 5. В каждую клетку этого квадрата записывалась одна буква. В греческом варианте одна клетка оставалась пустой, в латинском – в одну клетку помещали две буквы i и j. В результате каждой букве отвечала пара чисел, и шифрованное сообщение превращалось в последовательность пар чисел.

В I в. н.э. *Ю. Цезарь* во время войны с *галлами,* переписываясь со своими друзьями в *Риме,* заменял в сообщении первую букву латинского алфавита (A) на четвертую (D), вторую (B) - на пятую (E), наконец, последнюю - на третью:

⇑ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
⇓ D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Сообщение об одержанной им победе выглядело так:

YHQL YLGL YLFL

Император Август (I в. н. э.) в своей переписке заменял первую букву на вторую, вторую - на третью и т. д., наконец, последнюю - на первую:

⇑ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
⇓ B C D E F G H I J K L M N O P Q R S T U V W X Y Z A

Его любимое изречение было:

"GFTUJOB MFOUF"

Квадрат *Полибия,* шифр *Цезаря* входят в класс шифров, называемых *"подстановка"* или *"простая замена"*. Это такой шифр, в котором каждой букве алфавита соответствует буква, цифра, символ или какая-нибудь их комбинация.

1.1.2. Простейшие коды перестановки

В другом классе шифров *"перестановка"* – буквы сообщения каким-нибудь способом переставляются между собой. К этому классу принадлежит шифр скитала.

К классу *"перестановка"* относится шифр *"маршрутная транспозиция"* и его вариант *"постолбцовая транспозиция"*. В каждом из них в данный прямоугольник [n x m] сообщение вписывается заранее обусловленным способом, а столбцы нумеруются или обычным порядком следования, или в порядке следования букв *ключа* - буквенного ключевого слова. Так, ниже в первом прямоугольнике столбцы нумеруются в обычном порядке следования - слева направо, а во втором - в порядке следования букв слова *"Петербург"*.

Используя расположение букв этого ключа в алфавите, получим набор чисел [5 3 8 4 6 1 9 7 2]:

1	2	3	4	5	6	7	8	9
п	р	и	л	е	п	л	я	я
р	д	у	м	е	р	п	я	с
у	м	п	р	е	м	у	д	р
в	б	а	ь	ш	е	д	у	б

5	3	8	4	6	1	9	7	2
п	р	и	л	е	п	л	я	я
с	я	п	р	е	м	у	д	р
у	м	п	р	е	м	у	д	р
б	у	д	е	ш	ь	а	б	в

В первом случае шифрованный текст найдем, если будем выписывать буквы очередного столбца в порядке следования столбцов (прямом или обратном), во втором - если будем выписывать буквы столбца в порядке следования букв ключа. Таким образом, будем иметь:

1) прувр дмбиу палмр ьееешпрмел пудяя дуясрб;
2) пммья ррвря мулрр епсуб ееешя ддбип пдлууа;

(Пример Из послания Даниила Заточенаго к великому князю Ярославу Всеволодтю)

К классу *"перестановка"* принадлежит и шифр, называемый *"решетка Кардано"*. Это прямоугольная карточка с отверстиями, чаще всего квадратная, которая при наложении на лист бумаги оставляет открытыми лишь некоторые его части. Число строк и столбцов в карточке четно. Карточка сделана так, что при ее последовательном использовании (поворачивании) каждая клетка лежащего под ней листа окажется занятой. Карточку с прорезями сначала заполняют, потом поворачивают вдоль вертикальной оси симметрии на 180°, опять заполняют, а затем поворачивают вдоль горизонтальной оси также на 180°. И вновь повторяют ту же процедуру (рис 1.1.).

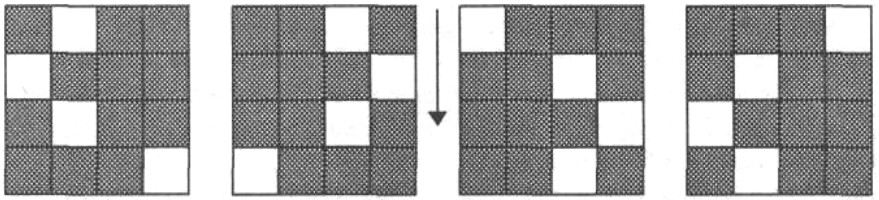


Рис 1.1. Решетка Кардано

Если решетка Кардано – квадрат, то возможен второй вариант самосовмещений фигуры, а именно, последовательные повороты вокруг центра квадрата на 90° (рис. 1.2).

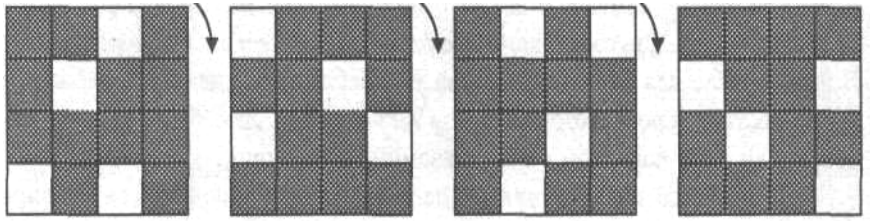


Рис 1.2. Поворот квадратной решетки Кардано

Рассмотрим еще несколько примеров решетки Кардано (рис. 1.3).

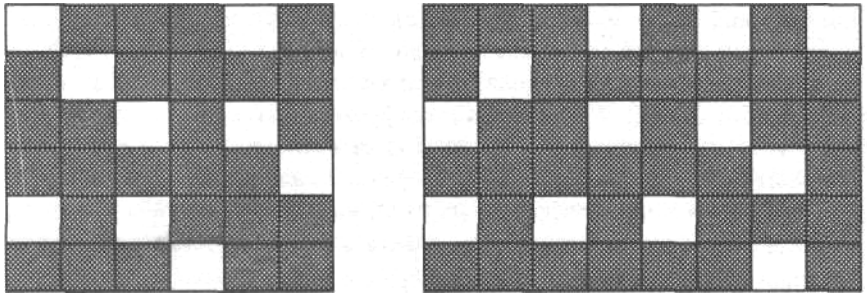


Рис 1.3. Примеры решеток Кардано.

Легко прочесть зашифрованное квадратной решеткой Кардано сообщение:

1) вавочс муноти мыжрое ьухсой мдосто яаснтв

2) ачшдеалб еымтяовн лыриелбм оянгеаюш дтинрент еоеыпрни,

также нетрудно расшифровать, пользуясь прямоугольной решеткой.

1.1.3. Раскрытие кода подстановки

Термин "шифр" арабского происхождения. В начале XV в. арабы опубликовали энциклопедию *"Шауба Алъ-Аща",* в которой есть специальный раздел о шифрах. В этой энциклопедии указан способ раскрытия шифра простой замены. Он основан на различной частоте повторяемости букв в тексте. В этом разделе есть перечень букв в порядке их повторяемости на основе изучения текста Корана. Заметим, что в русском тексте чаще всего встречается буква "О", затем буква "Е" и на третьем месте стоят буквы "И" и "А". Более точно: на 1000 букв русского текста в среднем приходится 90 букв "О", 72 буквы "Е" или "Ё" и по 60 букв "И" и "А" и т.д.

Неудобство шифров типа *"подстановка"* ("простая замена") в случае использования стандартного алфавита очевидно. Таблица частот встречаемости букв алфавита позволяет определить один или несколько символов, а этого иногда достаточно для дешифрования всего сообщения *("Пляшущие человечки" Конан Дойля* или *"Золотой жук" Эдгара По).*

1.1.4. Усложненные коды подстановки

Для усложнения раскрываемости шифра простой подстановки цели используют *многобуквенную систему шифрования* - систему, в которой одному символу отвечает одна или несколько комбинаций двух и более символов.

Другой прием - *использование нескольких алфавитов.* В этом случае для каждого символа употребляют тот или иной алфавит в зависимости от ключа, который связан каким-нибудь способом с самим символом или с его порядком в передаваемом сообщении.

М	О	Н	А	С	Т	Ы	Р	Ь	М	О	Н	А	С	Т	Ы	Р	Ь	М	О	Н
Р	А	С	К	И	Н	У	Л	О	С	Ь	М	О	Р	Е	Ш	И	Р	О	К	О
Э	О	Я	К	Щ	А	П	Ы	Й	Ю	Й	Щ	О	В	Ч	Ф	Ш	Л	Ь	Ш	Ы

Таблица Виженера

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я
Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А
В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б
Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В
Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г
Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д
Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е
З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж
И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ь	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь
Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы
Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э
Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю

Рис. 1.4 Таблица Виженера

В процессе шифрования (и дешифрования) используется таблица *("таблица Виженера")*, которая устроена следующим образом: в первой строке выписывается весь алфавит, в каждой следующей осуществляется циклический сдвиг на одну букву. Так получается квадратная таблица, число строк которой равно числу столбцов и равно числу букв в алфавите. На рисунке

1.4 представлена таблица, составленная из 31 буквы русского алфавита (без букв Ё и Ъ). Чтобы зашифровать какое-нибудь сообщение, поступают следующим образом. Выбирается слово - лозунг (например, "монастырь") и подписывается с повторением над буквами сообщения.

Чтобы получить шифрованный текст, находят очередной знак лозунга, начиная с первого в вертикальном алфавите, а ему соответствующий знак сообщения – в горизонтальном. В данном примере сначала находим столбец, отвечающий букве "м" лозунга, а затем строку, соответствующую букве "р" открытого текста. На пересечении выделенных столбца и строки находим букву "э". Так продолжая дальше, находим шифрованный текст полностью:

Наконец, к сообщению можно применять несколько систем шифрования.

Аббат Тритемиус – автор первой печатной книги о тайнописи (1518 г.) - предложил несколько шифров и среди них шифр, который можно считать усовершенствованием шифра Цезаря. Этот шифр устроен так. Все буквы алфавита нумеруют по порядку (от 1 до 33 в русском варианте). Затем выбирают какое-нибудь слово, называемое *"ключом",* например *"Вологда",* и подписывают под сообщением с повторением, как показано ниже:

о п е р а ц и я н а ч и н а е т с я в в о с к р е с е н ь е
в о л о г д а в о л о г д а в о л о г д а в о л о г д а в о

Чтобы получить шифрованный текст, складывают номер очередной буквы с номером соответствующей буквы ключа. Если полученная сумма больше 33, то из нее вычитают 33. В результате получают последовательность чисел от 1 до 33. Вновь заменяя числа этой последовательности соответствующими буквами, получают шифрованный текст. Разбивая этот текст на группы одной длины (например, по 5), получают шифрованное сообщение.

Если под ключом шифра понимать однобуквенное слово "В" (в русском варианте), то мы получим шифр Цезаря.

Появившийся в XVIII в. шифр *"по книге"* можно рассматривать как дальнейшее усовершенствование шифра *Ю. Цезаря.* Чтобы воспользоваться этим шифром, два корреспондента договариваются об определенной книге, имеющейся у каждого из них. Например, *Гашек Я.* Похождения бравого солдата Швейка. М., 1977. В качестве *ключа* каждый из них может выбрать "слово" той же длины, что и передаваемое сообщение. Этот ключ кодируется парой чисел, а именно номером страницы и номером строки на ней, и передается вместе с шифрованным сообщением. Например, (287,2) определяет "слово", т. е. текст избранной книги: *"Внимательно прочитав эту страницу, офицеры ничего не поняли..."* Этому ключу отвечает последовательность чисел:

03 15 10 14 01 20 06 13 30 15 16 17 18 16 25 10 20 01
03 31 20 21 19 20 18 01 15 10 21 24

Зная этот ключ, можно легко расшифровать сообщение:

РОНЮП ЕЧХВШ РХЩЮЩ ХУШРМ ШВЧФА

(Заметим, что в названной книге на указанной странице описывается вариант шифра *"по книге".)*

1.2. Применение частотного анализа для дешифрования сообщений

Пусть есть сообщение:

Д ЖТЦ БЦТ ЧКЙ ХТЖЙФЬЙССТ ХЙОФЙЦСТЙ ХТТЕЭЙСМЙ СДР УФМЫПТХа ХИЙПДЦа ЙЗТ ИТХЦДЦТЫСТ ИПМССЯРОЫЦТЕЯ РТКСТ ЕЯПТ УФТИЙРТСХЦФМФТЖДЦа ФДЕТЦЧ РЙЦТИДЫДХЦТЦСТЗТ ДСДПМЛД

Индекс частоты появления букв в стандартном тексте(похожей структуры):

О == 0.0886740955078085
И == 0.0653614890516941
Е == 0.065094707463728
Т == 0.0601900305772743
А == 0.0570296948429067
С == 0.0461326930575222
Н == 0.0453323482936239
В == 0.0381292454185393
Р == 0.032177963840834
Л == 0.0320343122165446
М == 0.0311929241314207
К == 0.0240719078987872
Д == 0.0231484331712122
. . .

Индекс частоты появления букв в закодированном тексте:

Т == 0.152866
Ц == 0.082805
С == 0.076433
Й == 0.076433
Д == 0.070063
Х == 0.050956
Ф == 0.044589
М == 0.031815
Р == 0.031815
П == 0.031815

И == 0.031815
Е == 0.025477
Ж == 0.019108
Ы == 0.019108
Я == 0.019108
а == 0.019108
Ч == 0.012739
З == 0.012739
Ь == 0.012739
У == 0.012739
К == 0.012739
Л == 0.003669
О == 0.003669
Э == 0.003669

Попробуем заменить самую частую букву в шифротексте «Т» на «О» – самую частую букву в русском языке по собранным ранее сведениям.

Дешифруем и получим сообщение:

Д ЖОЦ БЦО ЧКЙ ХОЖЙФЬЙССО ХЙОФЙЦСТЙ ХООЕЭЙСМЙ СДР УФМЫПОХа ХИЙПДЦа ЙЗО ИОХЦДЦОЫСО ИПМССЯРОЫЦОЕЯ РОКСО ЕЯПО УФОИЙРОСХЦФМФОЖДЦа ФДЕОЦЧ РЙЦОИДЫДХЦОЦСОЗО ДСДПМЛД

Теперь можно попробовать сменить «Д» на «И», но, посмотрев в таблицу, мы увидим, что вероятность «Д» в шифротексте несколько меньше и «Д» стоит на пятом месте вместо положенного второго, а вот «А» стоит как раз на пятом месте.

Попробуем заменить:

А ЖОЦ БЦО ЧКЙ ХОЖЙФЬЙССО ХЙОФЙЦСТЙ ХООЕЭЙСМЙ САР УФМЫПОХа ХИЙПАЦа ЙЗО ИОХЦАЦОЫСО ИПМССЯРОЫЦОЕЯ РОКСО ЕЯПО УФОИЙРОСХЦФМФОЖАЦа ФАЕОЦЧ РЙЦОИАЫАХЦОЦСОЗО АСАПМЛА

Фрагмент «САР» в тексте очень похож на «ТАК». Тем не менее, сверившись с таблицей вероятностей, принимаем решение не делать такой замены –

буквы слишком далеко отстоят друг от друга. Фрагмент «ЖОЦ» в самом начале похож на «ВОТ»; пришлось отвергнуть по той же причине. Остается попробовать все известные союзы и предлоги, *пробуя делать* соответствующие замены. В результате фрагмент «ЖОЦ», замененный на «ВОТ», оказался верным решением. Аналогично «БЦО» получился замененным на «ЭТО» и так далее, пока не получилось вот это:

А ВОТ ЭТО УЖЕ ХОВЕФЬЕННО ХЕОФЕТНОЕ ХООЕЭЕНМЕ НАМ УФМЬПОХа ХИЕПАТа ЕЗО ИОХТАТОЫНО ИПМННЯМОЫТОЕЯ МОЖНО ЕЯПО УФОИЕМОНХТФМФОВАТа ФАЕОТУ МЕТОИАЫАХТОТНОЗО АНАПМЛА

Слова «ХОВЕФЬЕННО ХЕОФЕТНОЕ» есть не что иное как «СОВЕРШЕННО СЕКРЕТНОЕ». Осуществив замены новых, найденных букв, получим почти все сообщение. Продолжив, таким образом, дальнейшее нахождение и замену букв, получим сообщение:

А ВОТ ЭТО УЖЕ СОВЕРШЕННО СЕКРЕТНОЕ СООБЩЕНИЕ НАМ ПРИШЛОСЬ СДЕЛАТЬ ЕГО ДОСТАТАЧНО ДЛИННЫМ ЧТОБЫ МОЖНО БЫЛО ПРОДЕМОНСТРИРОВАТЬ РАБОТУ МЕТОДА ЧАСТОТНОГО АНАЛИЗА

Надо заметить, что если бы у нас были большие словари, в которых бы находились все словоформы большинства русских букв на определенные тематики, мы могли бы подбирать слова для «отгадки» автоматически, проверяя всевозможные слова и выбирая наиболее «близкие» к словам с дешефрованными фрагментами.

Для этого можно отобразить одинаковые слова по следующему принципу:

- слова из словаря и дешифрованного фрагмента должны быть одной длины;
- слова одной длины сравнивать по количеству букв совпадающих в соответствующих местах дешифрованных букв.

Например, для фрагмента «ХОВЕФЬЕННО» из нашего примера алгоритм мог бы выдать следующую статистику:

ХОВЕФЬЕННО – СОВЕРШЕННО = 7
ХОВЕФЬЕННО – НЕСЕРЬЕЗНО = 4
ХОВЕФЬЕННО – РУГАТЕЛЬНО = 2

Руководствуясь ею, мы можем подбирать и угадывать наиболее вероятные слова гораздо быстрее. Создать же такую программу очень просто. Следует отметить, что если алгоритм реализован итерационно (замена букв происходит поочередно, а не одновременно), то необходимо предусмотреть механизм, отличающий прошедшую замену букву от исходной буквы (например «О», получившуюся из «Т», от «О» из исходного текста).

1.3. Выполнение лабораторной работы

1.3.1. Общий план выполнения работы

1. Изучить метод частотного анализа.
2. Получить от преподавателя номер варианта задания (у преподавателя также можно получить варианты текстов для шифрования и дешифрования в электронном виде).
3. Написать программу шифровки первой части задания шифром простой подстановки (моноалфавитная замена).
4. Написать программу дешифровки второй части задания, зашифрованной шифром простой подстановки (моноалфавитная замена).
5. Составить отчет о выполненной работе.
6. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

1.3.2. Порядок проведения обработки.

1. Кодирование текста.

Придумать ключ подстановки и зашифровать им исходное сообщение. Сохранить в виде файлов исходное, закодированное сообщение и ключ шифрования.

2. Декодирование текста.

Используя метод частотного анализа, расшифровать закодированное сообщение, воспользовавшись примером из части 2. Сохранить в виде файлов исходное, декодированное сообщение и найденный ключ шифрования.

1.3.3. Содержание отчета

1. Результат выполнения первой части задания:
   а) исходный текст;
   б) зашифрованный текст;
   в) последовательность ключа кодирования текста.
2. Результат выполнения второй части задания:
   а) исходный текст;
   б) дешифрованный текст;
   в) найденный ключ кодирования текста.

В результате проделанной работы необходимо получить:

4 текстовых последовательности: 2 текста на русском языке, 2 текста закодированных шифром простой подстановки;

2 ключа простой подстановки (моноалфавитная замена);

2 программных модуля – кодирование и декодирование текста.

1.4. Контрольные вопросы

1. Шифр простой подстановки – принцип работы.
2. Шифр простой подстановки – достоинства недостатки.
3. Шифр перестановки – принцип работы.
4. Усложнение шифра простой подстановки – примеры.
5. Метод частотного анализа – описание метода.

1.5. Данные для выполнения лабораторной работы

1.5.1. Общие данные

Алфавит: русский, все буквы большие, 33 символа (без Ё с пробелом):

АБВГДЕЖЗИЙКЛМОПРСТУФХЦЧШЩЪЫЬЭЮЯ (пробел)

Индекс частот появления букв русского алфавита:

(пробел) = 0.128675
О = 0.096456
И = 0.075312
Е = 0.072292
А = 0.064841
Н = 0.061820
Т = 0.061619
С = 0.051953
Р = 0.040677
В = 0.039267
М = 0.029803
Л = 0.029400
Д = 0.026983
Я = 0.026379
К = 0.025977
П = 0.024768
З = 0.015908
Ы = 0.015707
Ь = 0.015103
У = 0.013290
Ч = 0.011679
Ж = 0.010673
Г = 0.009867
Х = 0.008659
Ф = 0.007249
Й = 0.006847
Ю = 0.006847

Б = 0.006645
Ц = 0.005034
Ш = 0.004229
Щ = 0.003625
Э = 0.002416
Ъ = 0.000000

1.5.2. Задание

Задание 1

Закодировать любой текст (не менее 500 символов), любым произвольным шифром простой подстановки (моноалфавитная замена).

Результат представить в виде 3-х файлов:

1) исходный текст;
2) зашифрованный текст;
3) ключ шифрования.

Задание 2

Расшифровать текст, представленный во второй части задания (**Дешифровка**), закодированный шифром простой подстановки (моноалфавитная замена).

Алфавит открытого (исходного) текста – русский, все буквы большие, 33 символа без буквы Ё, но с символом пробел.

Результат представить в виде 2-х файлов:

1) дешифрованный текст;
2) найденный ключ шифрования.

Написать отчет по результатам лабораторной работы.

1.5.3. Варианты заданий

Вариант №1

1) Зашифровать текст

2) Дешифровка: cod1.txt

r2Я Ь9К>>К2ЙЬБt ЬrДЫt12<Д>>tr82tM<КЫ1ХО>12>t<rЬЧ82ЯК
КБrtАДХО>12ЯЬ2ЙЬББrЬ<02Я Ь>О
ДЫ>Or02ЬЕК>ЯКatrД12r2Е0Б0cК<2Ф..КЙОtrЫЬК2ЯЬ2r К<КЫt2БКЙЬБt
ЬrДЫtК2>ЛДОЬАЬ2ОКЙ>ОД2ЫЬ2r<К>ОК2>2tМ<КЫКЫtК<2
Д>ЯЬЧЬЛКЫt12ЬО КМЙЬr2>ЬЬОrКО>Or0Xct32>t<rЬЧД<2r2ЙЬББrЬ<2Я
Ь>О ДЫ>ОrК2tМ<КЫ1КО>12ОДЙЛК2t2Я
КБ>ОДrЧКЫt12>ЛДО832>t<rЬЧЬr2ОЬ2К>О,2r2ЫДbt32ОК <tЫД32bt.
ЬЕЬМЫДаКЫt1?2ЫДЯ t<К 2ЯК
КБrtЫ0r2>t<rЬЧ2r2tМ2ЙЬЫ9Д2ЙЬББrЬАЬ2Я Ь>О
ДЫ>ОrД2r2ЫДаДЧЬ2ЙЬБt
0ХсКК2МЫДаКЫtК2<ЬЛКО2r8АЧ1БКО,2ЫК2ЙДЙ2МДЯЧДЫt
ЬrДЫЫЬК2tМЫДаДЧ,ЫЬ2Д2ЯЬБ
0АЬ<02>ЬЬОrКО>ОrКЫЫЬ2<КЫ1ХО>12t2EtО82ЙЬБt 0XctК2ЫДbК2
КМ0Ч,Ot 0ХсКК2МЫДаКЫtК2rБЬЕДrЬЙ2ДБДЯОtrЫЬ<2ДЧАЬ
tО<К2r>К2<КЫ1КО>12БtЫД<taК>Йt2ЯЬ>ЧК2tМ<КЫКЫt12ЙДЛБЬАЬ2>t<r
ЬЧД

Вариант №2

1) Зашифровать текст

2) Дешифровка: cod2.txt

>Р>rОХ2Х>ДЕ2rА2Х>РОЕ7О1Аа>Х29аФ7Е2РО17аХ> РФ аАr
>2Ма1УО>РаЕ2 А12rА7>Д2 РФа17 >УОb9232>r7ХР2ФО>АО0Ф7МО>РаЕ2
А12rАа8>Д2 РФа17 >УОb9232>r7ХР2ФО>2r12РО11О
>1О>Д29rtаАа>tОrА2А>7Л>РrАЕatОаХ2rА7>1Ос<РОаАr
>Х29аФЫЯ>ДаЕР232>ДЕ70Ф7bа17 >7Ф7>аЙа>ДаЕР232>Д2Е
9УО>АО0Ф7МО>Бt7А<РОЯЙО
>02Фаа>rФ2b1<а>cОР7r7Х2rА7>1ОДЕ7ХаЕ>Д2
РФа17а>291232>r7ХР2ФО>Д2rФа>УОУ232А2>9ЕБ3232>7ХаЯА>r22АРаАrА
Ра112>РА2Е28>7>02Фаа>Д2Е 92У>r>Д2Х2ЙЫЯ>ЬА7Л>РаЕ2

A12rAa8>X2baA>0<АЫ>P<t7rФa1O>Ь1АЕ2Д7
>7>Д2rАЕ2a1>92rАОА2t12>ЬККаУА7Р1<8>У29>Д2cР2Ф
ЯЙ78>rbОАЫ>АаУгА>1О7ФБтИО >rEa91
>9Ф71О>У29О>7>r22АРаArАРa112>rАаДa1Ы>rbОА7 >92rА73ОаAr >X29aФ
X7>P>У2А2Е<Л>2Ma1У7>РаЕ2 A12rA7>УОУ>X2b12>02Фaa>A2t1<

Вариант №3

1) Зашифровать текст

2) Дешифровка: cod3.txt

7ОУ8cr8ЛБ8ЧХОДХЛМtcbЛrAc<МАcPcМАЕc<ХЛД8cXcБАc5МА1cБКХ<Xr
8cКД8cr8cБК87ЛМОРДb8МЛbcАБМХЕОД4r
ЕcОД>АКХМЕАЕcЛУОМХb?cБАЛ2АД42tcArcr8cХЛБАД4Фt8McrX2O2A1c
XrИАКЕОЧХХcАcЛУХЕО8ЕАЕcМ82ЛМ8?c<МАП
cБArbМ4c2O2cБКАХЛЙА7ХМcЛУОМХ8cХЛБАД4ФtaЬ88cФrOrXbcAcM82Л
М8cР8Kr8ЕЛbc2cБArbМХbЕc5rМКАБХХcXcXrИАКЕОМХРrАЛМХcМ82ЛМ
ОccЛРbФ4cЕ8У7tcР8КАbМrАЛМbЕХcXc2А7ОЕХcХФt<О8МЛbcРcОД>8ПК
ОХ<8Л2А1cМ8АКХХc2А7ХКАРОrXbcАЛrАРrА1cМ8АКХ81c2АМАКА1cbР
Дb8МЛbcrX2МАcXrA1c2O2c2ДА7c5ДРt7cЫ8rrArc8>AcМ8АК8Е
c2А7ХКАРОrXbcХЛМА<rX2OcХЛБАД4ФtaМЛbcА<8r4cO2МХРrАcXcФ78Л
4

Вариант №4

1) Зашифровать текст

2) Дешифровка: cod4.txt

X4МЕОb1cХЛЫХ>7МcОХАМЕtЬОЕratKrМЕ1rX>МД4РУ><Х
ЙМ4Уb1Д>rФ1aМЕП4r>ЛМ5ОРМ21rОДАМЕМ4c42r>aX
Ф>М>МЕr4r><ОЕ8>Ф>МФ1cОУЛФ>М2ДОcЕr4aУОХ>ЛМc4XX
ЙМ>ЙМaЕОМФ1ПХ1МД4Р5>rАМХ4МХО2ОДОЕО84КЬ>ОЕЛМ8У4ЕЕ
М>У>МЕОФО7Era4М21МЕ21Е15tМ81c>Д1a4X>ЛМc4XX
ЙМ84Пc1ОМЕОФО7Era1М2Д>Pa4X1МЕП4rАМrО8ЕrМ12ДОcОУОХХ17МЕ
rДt8rtД МЕМ12ДОcОУОХХ17МЕrО2ОХАКМБИИО8r>aX1Er>M>M5
ЕrД1cО7Era>ЛМaМР4a>Е>Ф1Er>М1rMr1b1M<r1Ma4ПХООМаМ2Д>У1ПОХ
>>МЕ81Д1ЕrАМ>У>М81БИИ>Ч>ОХrМЕП4r>Л

Вариант №5

1) Зашифровать текст

2) Дешифровка: cod5.txt

И7У24>2 >МР4ДД >М2ЕПЧЙМД48
О4ЙАИЛМrЕt48ДЕ2ЧММИЙtЕХ4МШ4Ф1МЙ>ХИЙМУ1УМРОЕУ<Д
>МР4ДД >МУ1УМЕЪУШtЕО4ДД ФМИ-
УПД41МИМЙЕ<ХУМ8t>ДУЛМЙ>ЕtУУМrt>РrЕ14П4>ЙИЛМ<ЙЕМР4ДД
>МrtЕУ8ОЕРЛЙИЛМУИЙЕ<ДУХЕ2МУМrt>РЕИЙ4О1ЛКЙИЛМХЕ2rt>ИИЕ
tЧМОМОУР>МИУ2ОЕ14МД4РМД>ХЕЙЕt
2М41Ш4ОУЙЕ2Мt4ДАЫ>МО>ИАМrtЕЪ>ИИМИ74ЙУЛМД48
О41УМХЕРУtЕО4ДУ>2МУИЙЕ<ДУХ4МrЕИХЕ1АХЧМЕДЕМrtУ8О4ДЕМЧ
Р41УЙАМУ85 ЙЕ<ДЕИЙАМОМР4ДД
ЩМД4МЕИДЕО>МУЩМrt>РИХ48Ч>2ЕИЙУМrЧЙ>2МУДЕПЕМrt>РИЙ4О
1>ДУЛМР4ДД
ЩММЙЕМ>ИЙАМУЩМХЕРУtЕО4ДУЛМ84МРО4МrЕИ1>РДУЩМР>ИЛЙ
У1>ЙУЛМХ4tЙУД4МД>ИХЕ1АХЕМУ82>ДУ14ИАМr>tОЕФМ14ИЙЕ<ХЕФ
МИЙ414МУР>ЛМt48Р>1УЙАМrtЕЪ>ИИМИ74ЙУЛМД4МРО4МО84У2ЕИО
Л84ДД
ЩМrtЕЪ>ИИ4МХЕРУtЕО4ДУ>МД>rЕИt>РИЙО>ДДЕМОЕИrtЕУ8ОЕРЛЬ>>
МИ74Й ФМrЕЙЕХ-
МИУ2ОЕ1ЕОМУМ2ЕР>1УtЕО4ДУ>Mrt>РЕИЙ4О1ЛКЬ>>МОИКМД>Е5ЩЕ
РУ2ЧКМР1ЛМХЕРУtЕО4ДУЛМУДШЕt24ЪУК

Вариант №6

1) Зашифровать текст

2) Дешифровка: cod6.txt

tАЪЧХАЫ151Ъ1ХЯ Ч<tЫФБ7Х1ЪЕЬ ЧБ<ХЩ ЧЯБЬЕ 1ЛЧ5tАЩЬЕЬХЯ tЬ2
1ЙЬД1ЫЧФХr1ЫЫ>МХБЬХЬЩ1ИtБАФХ5ББХ1ЪЕЬ
ЧБ<ХАИ1БЧФХЫtХ<ЬИtБХ8ЛЛtЩБЧДЫЬХ 12ЬБ1Б7ХЫ1ХОЧЛ
ЬБtЩАБ1МХМЬ ЬОЧМХЩ ЧЯБЬ1ЪЕЬ ЧБ<ЬДХАИ1Б7ХОЧЛ
ЬБtЩАБХДtА7<1ХБФИtЪЬХЯЬХБЬШХЯ ЬАБЬШХЯ Ч5ЧЫtХ5ББХЩ ЧЯ-
ББЕ
1ЛЧ5tАЩ1ФХАЧАБt<1ХrЪЪИЫ1ХЪУ2><ЧХrЬАБКЯЫ><ЧХАЯЬАЬ21<ЧХ
ЧЙ21ДЪФБ7ХОЧЛ ЬБtЩАБХЬБХЬ5tДЧrЫ>МХДЫКБ
tЫЫЧМХЙ1ДЧАЧ<ЬАБtШХ1ХЯЬББ<КХtЕЬХ8ЫБ

ЬЯЧФХ2КrtБХДtА7<1ХД>АЬЩЬШХЧХКИХБЬ5ЫЬХЫ1<ЫЬЕЬХД>OtX5t<
ХКХЬБЩ >ББЕЬХБtЩАБ1ХЯ trАБ1ДЧБ7ХАt2tХ1ЪЕЬ
ЧБ<ХАИ1БЧФХЩЬББЬ >ШХ2>ХМЬ ЬОЬХА-
ИЧ<1ЪХЙ1ОК<ЪtЫЫ>tXr1ЫЫ>tXrЬДЬЪ7ЫЬХБФИtЪЬХДtr7Xr1ИtХЧЫБК
ЧБЧДЫЬХЯЬЫФБЫЬХ5ББХАИ1БЧtХДАt4tЪЬХ21ЙЧ
КtБАФХЫ1ХЧЙ2>ББ5ЫЬАБЧХЧЫЛЬ <14ЧЧХДХБtЩАБtХОЧЛ
ЬД1ЫЧtХ8БКХЧЙ2>ББ5ЫЬАБ7ХКАБ 1ЫФtБХ1ХД<tАБtХАХЫtШХКАБ
1ЫФtБХЧХДЬЙ<ЬИЫЬАБ7ХАИ1Б7ХБtЩАБ

Вариант №7

1) Зашифровать текст

2) Дешифровка: cod7.txt

КwЧ5Д>ЫХЧ1ЪEt Й2>ХИЬЧЙ ФХ 1 ХБЧБХЫПЫХЪЕЩЕtФЙХБЕ2rtЫИИ
ХrЕЯЩЕ1ФУЙХДЫХЙЕ17БЕХ8ЛЛЫБЙ ЩДЕХМtЧД Й7ХБЕДЛ wЫД4
Ч17Д>ЫХwЧДД>ЫХДЕХ ХДЧrt 2ЫtХЯДЧ5 ЙЫ17ДЕХК2ЫД7О
Й7XtЧЯ2ЫtXrtЕЪtЧ22>ХБЕЙЕtКУХЯЧХЕw ДХrt Ы2Х2ЕЬДЕХЯЧЪtКЯ
Й7ХЩХrЧ2ФЙ7Х Х ИrЕ1Д Й7ХИКПЫИЙЩКЫЙХДЫХ2ЫД7ОЫХwУЬ
Д>ХtЧЯ1 5Д>МХКrЧБЕЩП БЕЩХ ИrЕ1Д 2>МХЛЧА1ЕЩХДЫБЕЙЕt>ЫХ
ЯХД МХИЕwЫtЬЧЙХtЫЧ1 ЯЧ4 ХЧ1ЪEt Й2ЕЩХО ЛtЕЩЧД
ФХИХ4Ы17УХКИ1ЕЬД Й7ХЬ ЯД7ХrЕЙЫД4 Ч17Д>2ХМЧБЫtЧ2ХДЕХД
ХЩХЕwДЕАХ ЯХД МХЧ1ЪEt Й2ХО ЛtЕЩЧД ФХД БЧБХДЫХИЩФЯЧД-
ХИХЧ1ЪEt Й2Е2ХИЬЧЙ ФХЕД ХtЫЧ1
ЯЕЩЧД>ХБЧБХЕЙwЫ17Д>ЫХЧ1ЪEt Й2>Х Х
ИrЕ17ЯКУЙИФХtЧЯwЫ17ДЕ

Вариант №8

1) Зашифровать текст

2) Дешифровка: cod8.txt

БЯОД КЪМ4ИЯЕtrМtЯЛМ41tДЯr 2
ХtЯМДЯЬБКЬЫРПЬЯПР4КЪЯ5ОtЛtРtБМЩЛtЯЙ ЙЯМ 5ОtЛДОЯr 2 Х Яr
ЦtТО4Б МtЬЯ42М414ЯТ ИК Я4ХДМЪЯХ
ПР4ЯРОДА>ДРПЬЯМДЯР4КЪЙ4ЯtЯМДПР4КЪЙ4Яr ЦtТО4Б РЪЯ2
ММЩДЯМ4ЯДЧДЯtЯП2ДК РЪЯtУЯУО МДМtДЯtЯ5ДОД2 Х>ЯЙ
ЙЯЛ4ЕМ4ЯА4КДДЯwТТДЙРtБМДДЯБЯ5О41О
ЛЛМЩУЯ5ОtК4ЕДМtЬУЯБ4БПЫЯtП54КЪr>ЫРПЬЯ К14ОtРЛЩЯПЕ РtЬЯ

Я5ДОtТДОtИМЩДЯ>ПРО4ИПРБ ЯМ Я 55 О РМ4ЛЯ>О4БМДЯ5422ДОЕtБ
ЫРЯО А4Р>ЯЛДР424БЯ4АДП5ДХДМtЬЯФДК4ПРМ4ПРtЯУО
МtЛЩУЯtЯ5ДОД2 Б ДЛЩУЯ2 ММЩУ?

Вариант №9

1) Зашифровать текст

2) Дешифровка: cod9.txt

36-w0n/e\s/e-ren\05w2t-50n`i2\0i21-im07e86w2`-/-2wpemq6f2ewwe1-
703er6nweni2-noc0ni80wwe-23q0w2\2ns-5e-w6k6\6-d2me/e9e-2nre\s3e86w2`-
68ieq6i232me86ww;t-n2ni0q-e7m67ei/2-56ww;t-703er6nwenis-2wpemq6f22-
5eni296\6ns-2n/\zk2i0\swe-p232k0n/2q2-2-65q2w2nim6i28w;q2-q0m6q2-n-
re`8\0w20q-/eqrszi0me8-ni6\6-ek0825we1-w0e7te52qenis-2nre\s3e86w2`-
68ieq6i2k0n/2t-nm05ni8-36c2i;-p61\e8-56ww;t-2-rme9m6qqwe1-nm05;-
n\05ozc21-xi6r-m6382i2`-68ieq6i2k0n/2t-nm05ni8-36c2i;-n8`36w-n-re`8\0w20q-
m6nrm050\0ww;t-n2ni0q-e7m67ei/2-56ww;t-2-/eqrszi0mw;t-n0i01-8-/eiem;t-
nm05ni86-n0i08e1-703er6nweni2-2nre\s3ozin`-8-r0m8oz-ek0m05s-5\`-36c2i;-
r0m056860q;t-re-n0i`q-56ww;t-8-w627e\00-re\we1-im6/ie8/0-re5-nm05ni86q2-
n0i08e1-703er6nweni2-q;-7o50q-2q0is-8-825o-q0m;-rm05ei8m6c0w2`-
w6mod0w21-703er6nweni2-/eiem;0-8e3w2/6zi-rm2-r0m056k0-2wpemq6f22-re-
n0i`q-6-i6/40-q0m;-re38e\`zc20-erm050\`is-kie-i6/20-w6mod0w2`-703er6nweni2-
2q0\2-q0nie?

Вариант №10

1) Зашифровать текст

2) Дешифровка: cod10.txt

m291tPE2/69xz-
tPzt/0OP193`z1Pt2Q\OP3t`zd962PR/t`z9z1Q2QW/R/t`zd962PR/OO;Qz3PP\cQO9
xzRP8O9mN/zPkQO`zW/ROPzQcQzWPzO/dQ0z-
2;zE2Qm9z129IQOxN9z31Q49/N`OPQzd962fscQQzf3t?2P03tRPz1PzP193/O9sz
1Nft/25/zPOPz3P3tPxNPz98zWRf5z1/NPmzPW9O/mPRP0zWN9O;z9ztPNc9O;z
PWOfzP3t/RNxN9z3Q\Qz/zW2fEfszPtW/R/N9zPtvQ87/scQIfz-
t9z1/Nm9zO/8;R/N9z3m9t/N/I9zmPEW/z12/R9tQNxIzOf7OPz\;NPz3PP\c9t`zm/
mfs?O9\fW`zR/7Ofszt/0OfzPO9zR;2Q8/N9zWN9OOfsz9zf8mfszR2PWQz2QIOx
z1PNP?3fz1/192f3/zO/I/t;R/N9zQQzO/z3RPsz3m9t/NfzOQzP3t/RNxxzO/zOQ0z
O9m/mPEPz12PIQ?7ftm/zt/mzktP\;zR3xz1PRQ25OP3t`z1/Nm9z\;N/zP5R/kQO/z

-
tP0z1PNP3P0z8/tQIzP3?t/RNxxz1/192f3zO/z3m9t/NQzRztPIzR9WQzm/mzPOzQ3t`z193/N9zO/zOQIzR3QzktPzOf7OPz/zO/193/Rz3O9I/N9z1PNP3fz9z\Q8z1/Nm9zPt12/RNxN9z/W2Q3/tfzt/mzm/mz\fm?R;zO/zOQ0z2/8\2P3/O;zRz\Q31P2xWmQztPz12Pk9t/t`zO/193/OOPQzPOzIPEztPN`mPzR8xRz3RPsz3m9t/Nfz9zO/IPt/RzO/zOQQz\Q8z12P1f3mPRz-tfz1PNP3f?

Примечание. У преподавателя можно получить варианты текстов для дешифрования в электронном виде.

1.6. Список рекомендуемой литературы

1. Введение в криптографию/ под общ. ред. Ященко В.В. – М.: МЦНМО, «ЧеРо», 1998. – 272 с.

2. Основы современной криптографии: учеб. курс/ [Баричев С.Г. и др.]; 2-е изд., пер. и доп. – М.: Горячая линия - Телеком, 2002.

3. Нечаев, В.И. Элементы криптографии (Основы теории защиты информации): учеб. пособие для ун-тов и вузов/ В.И. Нечаев. – М.: Высшая школа, 1999.

2. Лабораторная работа № 2
Построение алгебраических фракталов

Цель работы – изучение методов построения простейших алгебраических фракталов; получение навыков программного моделирования и создания фракталов.

Содержание работы:

2.1. Теоретические основы лабораторной работы

2.1.1. Понятие фрактала

Понятия **фрактал** и **фрактальная геометрия**, появившиеся в конце 70-х – с середины 80-х гг. прочно вошли в обиход математиков и программистов. Слово **фрактал** образовано от латинского **fractus** и в переводе означает *состоящий из фрагментов*. Оно было предложено Бенуа Мандельбротом в 1975 году для обозначения нерегулярных, но самоподобных структур, которыми он занимался.

Роль фракталов в машинной графике сегодня достаточно велика. Они приходят на помощь, например, когда требуется с помощью нескольких коэффициентов задать линии и поверхности очень сложной формы. С точки зрения машинной графики фрактальная геометрия незаменима при генерации искусственных облаков, гор, поверхности моря. Фактически найден способ легкого представления сложных неевклидовых объектов, образы которых весьма похожи на природные.

Одним из основных свойств фракталов является самоподобие. В самом простом случае небольшая часть фрактала содержит информацию о всем фрактале.

Определение фрактала, данное Мандельбротом, звучит так: *"Фракталом называется структура, состоящая из частей, которые в каком-то смысле подобны целому"*.

Заметим, что данное определение в строгом смысле не математическое. Больше наталкивает на рассуждение об объектах, чем об их определении.

На данный момент, в большинстве случаев, под фракталами понимается визуализация некоторой динамической системы Определение динамической системы:

- Пусть M – множество (множество состояний) и S: R*M -> M (закон эволюции).
- Динамической системой называется пара <S, M>.

Мандельброт дал также строгое математическое определение фрактала как множества, *хаусдорфова размерность* которого строго больше топологической размерности.

Размерность Хаусдорфа (dimA) множества A определяется формулами:

$$H_\alpha(A) = \lim_{\delta\to 0}\inf\left\{\sum_i (\text{diam } E_i)^\alpha \mid A\subseteq\bigcup_i E_i, \text{diam } E_i < \delta\right\},$$

$$\dim A = \inf\{\alpha \mid H_\alpha(A)\geq 0\}.$$

Кубическая размерность (box dimension) является несколько более простым понятием. Если ***A*** – некоторое компактное множество и ***N***(***r***) есть минимальное число шаров радиуса ***r***, покрывающих ***A***, и если существует предел

$$\dim A = \lim_{r\to 0}\left(\frac{\log(N(r))}{\log(1/r)}\right)$$

при r, стремящемся к нулю, то этот предел называется кубической размерностью множества A. Известно, что Хаусдорфова размерность не превосходит кубическую, а для *самоподобных фракталов* они совпадают.

Строгое определение самоподобных множеств было дано Дж.Хатчинсоном. Множество *F* Хатчинсон назвал самоподобным, если оно состоит из нескольких компонент, подобных *F* (т.е компонент, получаемых поворотом, сжатием и отражением множества *F*). Стоит оговориться, что оригинальное определение Хатчинсона немного сложнее, но чаще пользуются именно этим.

Ключевая теорема Хатчинсона выглядит следующим образом:

Теорема: Если $S_1,..,S_N$ - набор сжимающих отображений полного метрического пространства X на себя, то найдется единственное замкнутое множество F, такое что $F=S_1(F)$U..U$S_N(F)$. Более того, множество F - компакт.

Множество *F* из теоремы называют инвариантным множеством системы сжимающих отображений или IFS-множеством (Iterated Function System) и обозначают F=IFS(X;S_1,..,S_N). В качестве пространства *X* обычно рассматривают n-мерное пространство R_n. Если все отображения S_i являются аффинными сжатиями, то соответствующее инвариантное множество называют самоаффинным.

2.1.2. Наиболее известные фракталы

Геометрические фракталы

Фракталы этого класса самые наглядные. В двухмерном случае их получают с помощью некоторой ломаной (или поверхности в трехмерном случае), называемой *генератором*. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломаную-генератор в соответствующем масштабе. В результате бесконечного повторения этой процедуры получается геометрический фрактал.

Рассмотрим один из таких фрактальных объектов - триадную кривую Кох. Построение кривой начинается с отрезка единичной длины (рис.2.1) - это 0-е поколение кривой Кох.

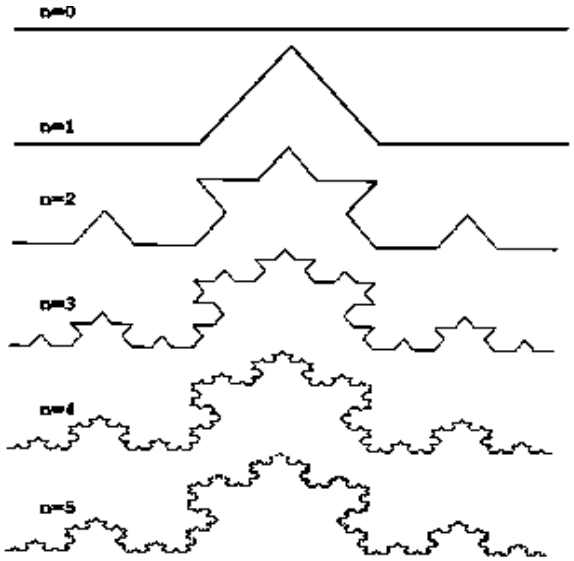


Рис 2.1. Построение триадной кривой Кох

Далее каждое звено (в нулевом поколении один отрезок) заменяется на *образующий элемент*, обозначенный на рис. 2.1 через **n=1**. В результате такой замены получается следующее поколение кривой Кох. В 1-м поколении - это кривая из четырех прямолинейных звеньев, каждое длиной по **1/3**. Для получения 3-го поколения проделываются те же действия - каждое звено заменя-

ется на уменьшенный образующий элемент. Итак, для получения каждого последующего поколения все звенья предыдущего поколения необходимо заменить уменьшенным образующим элементом. Кривая **n**-го поколения при любом конечном **n** называется *предфракталом*.

На рис.2.1 представлены пять поколений кривой. При **n**, стремящемся к бесконечности кривая Кох становится фрактальным обьектом [3].

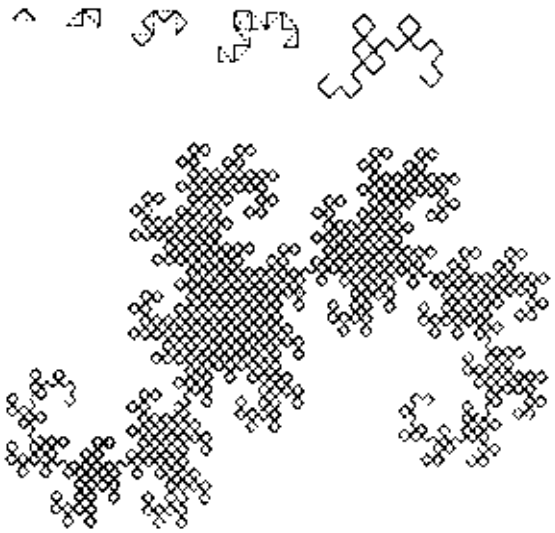


Рис 2.2. Построение "дракона" Хартера-Хейтуэя

Для получения другого фрактального объекта нужно изменить правила построения. Пусть образующим элементом будут два равных отрезка, соединенных под прямым углом. В нулевом поколении заменим единичный отрезок на этот образующий элемент так, чтобы угол был сверху. Можно сказать, что при такой замене происходит смещение середины звена. При построении следующих поколений выполняется правило: самое первое слева звено заменяется на образующий элемент так, чтобы середина звена смещалась влево от направления движения, а при замене следующих звеньев направления смещения середин отрезков должны чередоваться. На рис.2.2 представлены несколько первых поколений и 11-е поколение кривой, которая построена по

вышеописанному принципу. Предельная фрактальная кривая (при **n**, стремящемся к бесконечности) называется *драконом Хартера-Хейтуэя*.

Другими примерами геометрических фракталов являются:

- Канторовское множество.
- Фракталы Серпинского.
- Кривая Минковского.
- H-дерево.
- Кривая Гильберта.

В машинной графике использование геометрических фракталов необходимо при получении изображений деревьев, кустов, береговой линии. Двухмерные геометрические фракталы используются для создания объемных текстур.

В случае работы с 2 мерными изображениями получаются еще более интересные картины, которые часто используются при моделировании рельефа местности, создании береговой линии, воды, горных вершин и т.д. в большинстве современных компьютерных игр и фильмов. Например, один из хорошо известных фракталов, называющийся «ковер Серпинского», строится следующим образом.

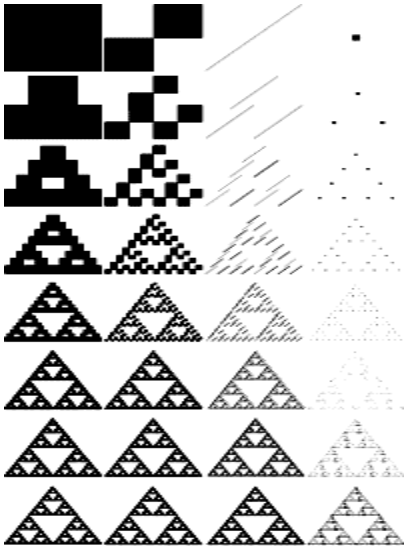


Рис 2.3. Построение ковра Серпинского

Сначала берется произвольный многоугольник – например квадрат. После чего мы можем задать функцию f(X), которая отображает любую точку нашего квадрата в другую его точку, например так:

$$f(X) = (0{,}5 \cdot x;\ 0{,}5 \cdot y).$$

Применяя описанную выше функцию к исходному единичному квадрату (квадрат со стороной единица), получим квадрат поменьше — с длиной стороны 0,5 (в последующем — 0,25 и т. д.). В конце концов наш квадрат «сожмется» в точку.

Рассмотрим, например, следующую систему функций:

$$f1(X) = (0{,}5 \cdot x;\ 0{,}5 \cdot y);$$

$$f2(X) = (0{,}5 \cdot x + 0{,}5;\ 0{,}5 \cdot y);$$

$$f3(X) = (0{,}5 \cdot x + 0{,}25;\ 0{,}5 \cdot y + \sqrt{3}/4).$$

В результате получим фрактал, представленный на рис 2.3.

Алгебраические фракталы

Это самая крупная группа фракталов. Получают их с помощью нелинейных процессов в **n**-мерных пространствах. Наиболее изучены двухмерные процессы. Интерпретируя нелинейный итерационный процесс как дискретную динамическую систему, можно пользоваться терминологией теории этих систем: *фазовый портрет*, *установившийся процесс*, *аттрактор* и т.д.

Известно, что нелинейные динамические системы обладают несколькими устойчивыми состояниями. То состояние, в котором оказалась динамическая система после некоторого числа итераций, зависит от ее начального состояния. Поэтому каждое устойчивое состояние (или, как говорят, аттрактор) обладает некоторой областью начальных состояний, из которых система обязательно попадет в рассматриваемые конечные состояния. Таким образом, фазовое пространство системы разбивается на *области притяжения* аттракторов. Если фазовым является двухмерное пространство, то, окрашивая области притяжения различными цветами, можно получить *цветовой фазовый портрет* этой системы (итерационного процесса). Меняя алгоритм выбора цвета, можно получить сложные фрактальные картины с причудливыми многоцветными узорами. Неожиданностью для математиков стала возможность с помощью примитивных алгоритмов порождать очень сложные нетривиальные структуры.

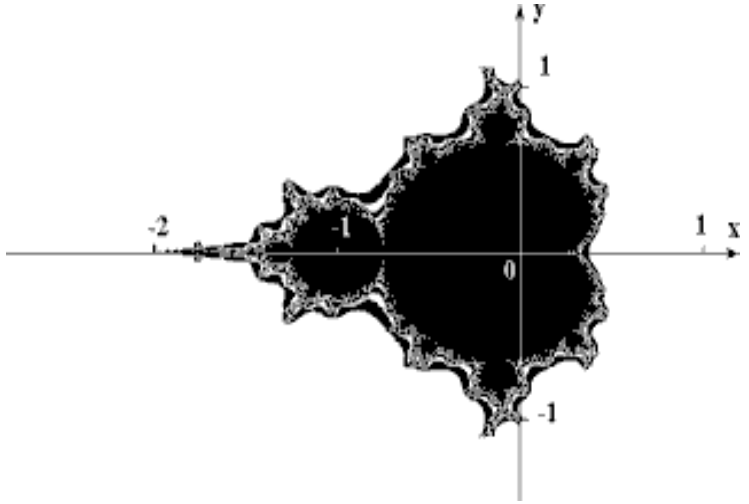


Рис 2.4. Множество Мандельброта

В качестве примера рассмотрим множество Мандельброта (рис.2.4 и рис.2.5). Алгоритм его построения достаточно прост и основан на простом итеративном выражении:

$$\mathbf{Z[i+1] = Z[i] * Z[i] + C,} \qquad (2.1)$$

где **Z**[i] и **C** - комплексные переменные.

Итерации выполняются для каждой стартовой точки **C** прямоугольной или квадратной области - подмножестве комплексной плоскости. Итерационный процесс продолжается до тех пор, пока **Z**[i] не выйдет за пределы окружности радиуса 2, центр которой лежит в точке (0,0) (это означает, что аттрактор динамической системы находится в бесконечности), или после достаточно большого числа итераций (например, 200-500 итераций) **Z**[i] сойдется к какой-нибудь точке окружности.

В зависимости от количества итераций, в течение которых **Z**[i] оставалась внутри окружности, можно установить определенный цвет точки **C** в зависимости от шага итерационного процесса, на котором эта точка «вылетела». При этом, если **Z**[i] остается внутри окружности в течение достаточно большого количества итераций, итерационный процесс прекращается и эта точка растра окрашивается в черный цвет, если точка вылетела на первых шагах итерационного процесса – то цвет будет белый.

Рис 2.5. Участок границы множества Мандельброта, увеличенный в 200 раз

Вышеописанный алгоритм дает приближение к так называемому множеству Мандельброта. Множеству Мандельброта принадлежат точки, которые в течение *бесконечного* числа итераций не уходят в бесконечность (точки, имеющие черный цвет). Точки, принадлежащие границе множества (именно там возникают сложные структуры), уходят в бесконечность за конечное число итераций, а точки, лежащие за пределами множества, уходят в бесконечность через несколько итераций (белый фон).

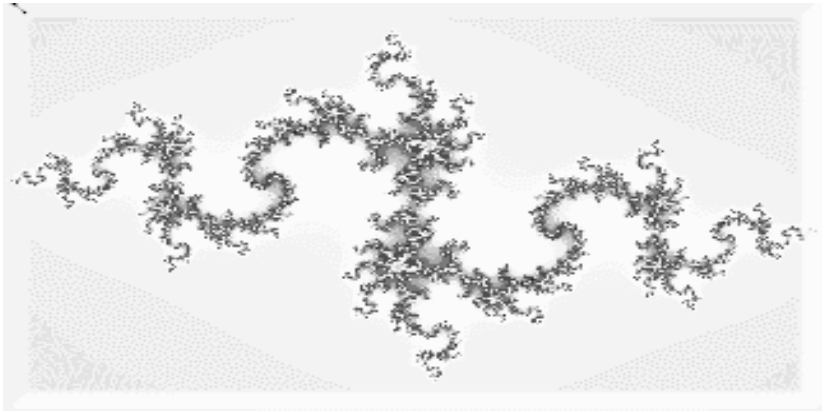


Рис 2.6. Одно из множеств Жулиа

Множества Жулиа образуются по той же самой формуле (1), что и множество Мандельброта. Но при этом **Z[0] ≠ 0**, а **C** – произвольная, наперед заданная, комплексная константа. Пример множества Жулиа представлен на рис. 2.6. При рисовании фрактала с использованием различных начальных точек (чтобы начать процесс итераций), генерируются различные изображения. Это применимо только к множеству Жулиа.

Хотя это нельзя увидеть на картинке, фрактал Мандельброта — это, на самом деле, множество фракталов Жулиа, соединенных вместе. Каждая точка (или координата) множества Мандельброта соответствует фракталу Жулиа. Множества Жулиа можно сгенерировать используя эти точки в качестве начальных значений в уравнении Z=Z[I]+C. Но это не значит, что если выбрать точку на фрактале Мандельброта и увеличить ее, можно получить фрактал Жулиа. Эти две точки идентичны, но только в математическом смысле. Если взять эту точку и просчитать ее по данной формуле, можно получить фрактал Жулиа, соответствующий определенной точке фрактала Мандельброта.

2.2. Пример построения геометрических фракталов

В качестве примера построения фрактальных структур, рассмотрим кривую Коха (см. рис.2.1) и "дракона" Хартера-Хейтуэя (см. рис.2.2). Выделим в этих структурах подобные части и для каждой из них вычислим коэффициенты аффинного преобразования для построения системы итерируемых функций (IFS). IFS представляет собой систему функций из некоторого фиксированного класса функций, отображающих одно многомерное множество (пространство) на другое. Наиболее простая IFS состоит из аффинных преобразований плоскости:

$$X' = A*X + B*Y + C,$$
$$Y' = D*X + E*Y + F.$$

Например, закодировав какое-то изображение двумя аффинными преобразованиями, мы однозначно определяем его с помощью 12 коэффициентов. Если теперь задаться какой-либо начальной точкой (например X=0 Y=0) и запустить итерационный процесс, то мы после первой итерации получим две точки, после второй - четыре, после третьей - восемь и т.д. Через несколько десятков итераций совокупность полученных точек будет описывать закодированное изображение. Но проблема состоит в том, что очень трудно найти подходящие коэффициенты IFS, которая кодировала бы произвольное изображение.

Для построения IFS применяют кроме аффинных и другие классы простых геометрических преобразований, которые задаются небольшим числом параметров. Например, проективные:

$$X' = (A1*X + B1*Y + C1) / (D1*X + E1*Y + F1),$$
$$Y' = (A2*X + B2*Y + C2) / (D2*X + E2*Y + F2)$$

или квадратичные:

$$X' = A1*X*X + B1*X*Y + C1*Y*Y + D1*X + E1*Y + F1,$$
$$Y' = A2*X*X + B2*X*Y + C2*Y*Y + D2*X + E2*Y + F2$$

преобразования на плоскости.

При этом для построения изображения в аффинный коллаж будет включено столько аффинных преобразований, сколько существует частей, подобных целому изображению.

Использование IFS для сжатия обычных изображений (например фотографий) основано на выявлении локального самоподобия в отличие от фракталов, где наблюдается глобальное самоподобие и нахождение IFS не слишком сложно (мы сами только-что в этом убедились). По алгоритму Барнсли происходит выделение в изображении пар областей, меньшая из которых подобна большей, и сохранение нескольких коэффициентов, кодирующих преобразование, переводящее большую область в меньшую. Требуется, чтобы множество "меньших" областей покрывало все изображение. При этом в файл, кодирующий изображения, будут записаны не только коэффициенты, характеризующие найденные преобразования, но и местоположение и линейные размеры "больших" областей, которые вместе с коэффициентами будут описывать локальное самоподобие кодируемого изображения. Восстанавливающий алгоритм в этом случае должен применять каждое преобразование не ко всему множеству точек, получившихся на предыдущем шаге алгоритма, а к некоторому их подмножеству, принадлежащему области, соответствующей применяемому преобразованию

Построим IFS для "дракона" Хартера-Хейтуэя на основе аффинных преобразований. Для этого расположим первое поколение этого фрактала на сетке координат дисплея 640 x 350 (рис.2.7). Обозначим точки получившейся ломаной **A**, **B**, **C**.

По правилам построения у этого фрактала две части, подобные целому – на рис.2.7 это ломаные **ADB** и **BEC**.

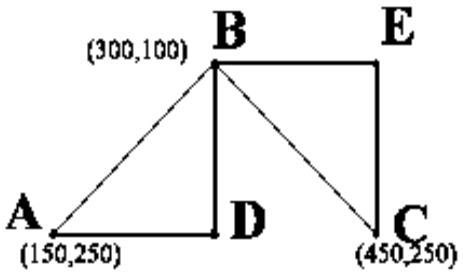


Рис 2.7. Заготовка для построения "дракона" Хартера-Хейтуэя

Зная координаты концов этих отрезков, можно вычислить коэффициенты двух аффинных преобразований, переводящих ломаную **ABC** в **ADB** и **BEC**:

X' = -0.5*X -0.5*Y + 490
Y' = 0.5*X -0.5*Y + 120
X' = 0.5*X -0.5*Y + 340
Y' = 0.5*X +0.5*Y – 110

После чего, задавшись некоторой начальной стартовой точкой (например X=0 Y=0) и итерационно действуя на этой точку, построенной итерационной системой, после десятой итерации на экране получим фрактальную структуру, изображенную на рис.2.8, которая представляет собой "дракон" Хартера-Хейтуэя.

Его кодом (сжатым описанием) является набор коэффициентов двух аффинных преобразований.

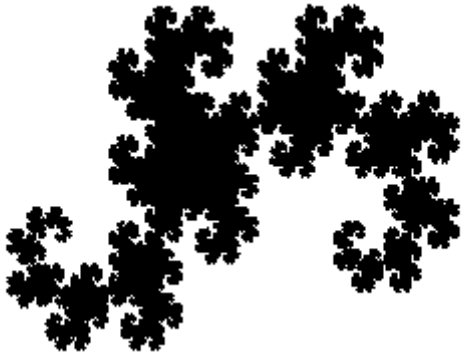


Рис 2.8. "Дракон" Хартера-Хейтуэя, построенный в прямоугольнике 640x350

Аналогично можно построить кривую Кох. Нетрудно видеть, что эта кривая имеет четыре части, подобные целой кривой. Для нахождения системы преобразования опять расположим первое поколение этого фрактала на сетке координат дисплея 640 x 350 (рис.2.9).

Для ее построения требуется набор аффинных преобразований, состоящий из четырех преобразований:

X' = 0.333*X + 13.333
Y' = 0.333*Y + 200
X' = 0.167*X + 0.289*Y + 130
Y' = -0.289*X + 0.167*Y + 256

X' = 0.333*X + 413.333
Y' = 0.333*Y + 200
X' = 0.167*X - 0.289*Y + 403
Y' = 0.289*X + 0.167*Y + 71

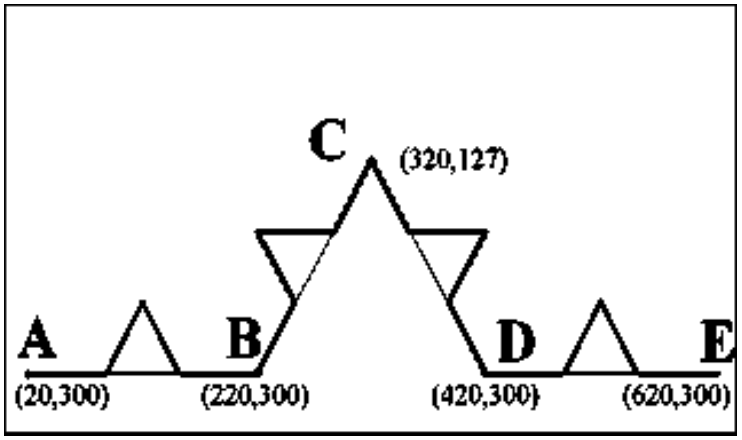


Рис 2.9. Заготовка для построения кривой Кох

Результат применения этого аффинного коллажа после десятой итерации можно увидеть на рис.2.10.

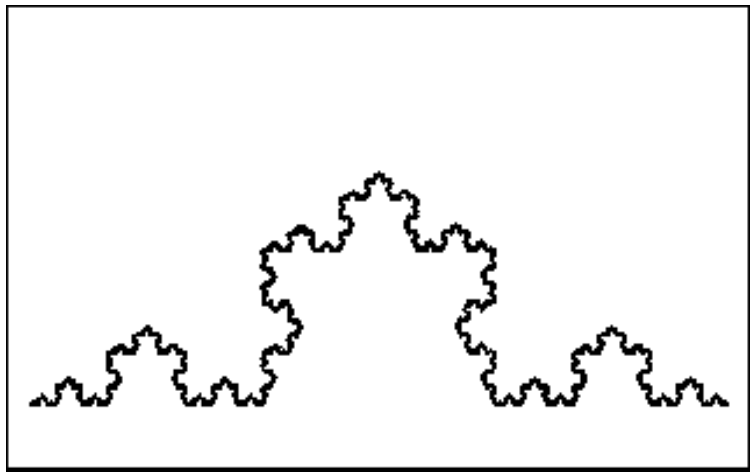


Рис 2.10. Кривая Кох, построенная в прямоугольнике 640x350

2.3. Выполнение лабораторной работы

2.3.1. Общий план выполнения работы

1. Изучить понятие фрактала и их классификацию.
2. Изучить способы построения фракталов (при помощи системы итерируемых функций).
3. Написать подпрограмму создания геометрического фрактала при помощи IFS(см. подробнее Задание 1).
4. Написать подпрограмму создания множества Жулиа (см. подробнее Задание 2.1).
5. Написать подпрограмму создания множества Мандельброта (см. подробнее Задание 2.2).
6. Написать подпрограмму создания множества Жулиа по множеству Мандельброта (см. подробнее Задание 2.3).
7. Составить отчет о выполненной работе.
8. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

2.3.2. Порядок проведения работы

1. Создать геометрический фрактал («дракон» Хартера-Хейтуэя или кривую Кох) пользуясь рекомендациями из «Примера построения».
2. Создание множества Жулиа.

Построить множество Жулиа, используя для построения формулу (2.1), где C = a*i + b – любая заранее выбранная const из интервала [0, 1], а Z[0] = y*i+x (x и y – физические координаты плоскости построения). Подробнее описание процесса построения – в части 2.5.2 «Задание», пункт 2.1.

3. Создание множества Мандельброта.

Построить множество Мандельброта, используя для построения формулу (2.1), C = y*i+x (x и y – физические координаты плоскости построения), а Z[0]=0. Подробнее описание построения – в части 2.5.2 «Задание», пункт 2.2.

Создание множества Жулиа по множеству Мандельброта.

Построить множество Жулиа используя для построения формулу (2.1), где C = y*i + x – const взятая с плоскости построенного Мандельброта (напр. по щелчку мышки).

2.3.3. Содержание отчета

1. Результат выполнения первой части задания:
   а) изображение геометрического фрактала.
2. Результат выполнения первой части задания:
   а) изображение множества Жулиа;
   б) параметр **C**, выбранный при построении множества.
3. Результат выполнения второй части задания:
   а) изображение множества Мандельброта.
4. Результат выполнения третьей части задания:
   а) 2-3 изображения множества Жулиа с разными начальными параметрами **C.**

В результате проделанной работы необходимо получить:

3-4 изображения с множеством Жулиа;

1 изображение с множеством Мандельброта;

1 программный модуль, состоящий из двух подпрограмм – построение множества Жулиа и множества Мандельброта.

2.4. Контрольные вопросы

1. Что такое Фрактал?
2. Что понимать под термином самоподобие?
3. Геометрические фракталы – определение, примеры.
4. Алгебраические фракталы – определение, примеры.
5. Геометрические фракталы – примеры, способ построения при помощи системы итерируемых функций.
6. Алгебраические фракталы – примеры, способ построения.
7. Чем отличаются и чего общего у множеств Жулиа и Мандельброта?

2.5. Данные для выполнения лабораторной работы

2.5.1. Общие данные

При построении геометрических фракталов рекомендуется поэкспериментировать с выбором начальной точки (аттрактора) и количеством итераций.

При построении множеств Жулиа и Мандельброта использовать квадрат не менее ***400x400*** точек, где ось ***y*** соответствует мнимой, а ось ***x*** – действительно части комплексного числа **Z.**

При этом:

► при построении множества Жулиа цифровые координаты по ***x*** и ***y*** на интервале **[0, 400]** соответствуют физическим координатам на интервале **[-2, 2];**

► при построении множества Мандельброта цифровые координаты:

по ***x*** на интервале **[0, 400]** соответствуют физическим координатам на интервале **[-2,25, 0,75];**

по ***y*** на интервале **[0, 400]** соответствуют физическим координатам на интервале **[-1,5, 1,5]**.

2.5.2. Задание

Задание 1: Геометрические фракталы.

Написать программу построения геометрического фрактала («дракон» Хартера-Хейтуэя или кривой Кох), пользуясь рекомендациями из «Примера построения». Для этого использовать 1 начальную точку и соответствующие аффинные преобразования. Количество итераций взять в зависимости от области построения, но не менее 10-11 итераций.

Задание 2: Алгебраические фракталы

Задание 2.1

Написать подпрограмму построения множеств Жулиа с входным параметром **C** ***(это более удобно для дальнейшего выполнения 3-го задания).***

Использовать квадрат не менее ***400x400*** точек, где ось ***y*** соответствует мнимой, а ось ***x*** – действительной части комплексного числа **Z.** При этом

цифровые координаты по ***x*** и ***y*** на интервале **[0, 400]** соответствуют физическим координатам на интервале **[-2, 2].**

Для построения воспользоваться формулой (2.1):

Z[i+1] = Z[i] * Z[i] + C, где **Z**i и **C** - комплексные переменные.

C – наперед заданная константа **ai+b** из интервала [0,1].

Z[0] = Y*I + X – берется из координат каждой точки области построения (то есть уравнение (2.1) рассчитывается для каждой точки исходного квадрата 400x400 т.);

Итерации выполняются для каждой стартовой точки **Z(0)** прямоугольной или квадратной области – подмножестве комплексной плоскости. Итерационный процесс продолжается до тех пор, пока **Z**[i] не выйдет за пределы окружности радиуса 2, центр которой лежит в точке (0,0) (это означает, что аттрактор динамической системы находится в бесконечности), или после достаточно большого числа итераций (**возьмем 255**) **Z**[i] сойдется к какой-нибудь точке окружности.

Итерации продолжаются, пока не выполнится одно из условий:

1) |Z[i+1] – Z[i]| > 4.

2) k = 255.

3) |Z[i+1] – Z[i]| = 0.

Если выход произошел по 2-му условию, то точка [X,Y] закрашивается в черный цвет, иначе в цвет **I** (либо в полутоновой шкале, либо в палитровой, если палитра задана).

Задание 2.2

Написать подпрограмму построения множества Мандельброта**.**

Использовать квадрат не менее ***400x400*** точек, где ось ***y*** соответствует мнимой, а ось ***x*** – действительно части комплексного числа **Z.**

При этом цифровые координаты:

по ***x*** на интервале **[0, 400]** соответствуют физическим координатам на интервале **[-2,25, 0,75];**

по ***y*** на интервале **[0, 400]** соответствуют физическим координатам на интервале **[-1,5, 1,5].**

Для построения воспользоваться формулой (2.1):

Z[i+1] = Z[i] * Z[i] + C, где **Z**i и **C** - комплексные переменные.

C – Y*I + X – берется из координат каждой точки области построения (то есть уравнение (2.1) рассчитывается для каждой точки исходного квадрата 400x400 т.).

Z[0] = 0;

Итерации выполняются для каждой стартовой точки **C** прямоугольной или квадратной области – подмножестве комплексной плоскости. Итерационный процесс продолжается до тех пор, пока **Z**[i] не выйдет за пределы окружности радиуса 2, центр которой лежит в точке (0,0) (это означает, что аттрактор динамической системы находится в бесконечности) или после достаточно большого числа итераций (**возьмем 255**) **Z**[i] сойдется к какой-нибудь точке окружности.

Итерации продолжаются, пока не выполнится одно из условий:

1) |Z[i+1] – Z[i]| > 4.

2) k = 255.

Если выход произошел по 2-му условию, то точка [X,Y] закрашивается в черный цвет, иначе в цвет **I** (либо в полутоновой шкале, либо в палитровой, если палитра задана).

Задание 2.3

Написать подпрограмму построения множеств Жулиа по множеству Мандельброта.

Пользователь щелкает мышкой на построенном во втором задании множестве Мандельброта, и строится множество Жулиа с входным параметром **C** (как в задании 1), взятым из координат мышки (x,y).

Написать отчет по результатам лабораторной работы.

2.5.3. Варианты заданий

Все студенты выполняют одно и то же задание с разными параметрами **C** при начальном построении множества Жулиа.

Примечание. Желательно, но необязательно, задать собственную палитру, состоящую из 256 цветов, при построении фракталов.

2.6. Список рекомендуемой литературы

1. Бондаренко, В.А. Фрактальное сжатие изображений по Барнсли-Слоану/ В.А. Бондаренко // Автоматика и телемеханика. – 1994. - N5. - С.12-20.

2. Ваторин, Д. Применение фракталов в машинной графике/ Д. Ваторин // Computerworld. – Россия.- 1995.- N15.- С.11.

3. Федер, Е. Фракталы / Е. Федер; пер. с англ. – М.: Мир, 1991.-254с. (Jens Feder, Plenum Press, NewYork, 1988).

3. Лабораторная работа № 3

Кратно-масштабный анализ

Цель работы – изучение метода кратно-масштабного анализа изображений; получение навыков программного моделирования вейвлет преобразований.

Содержание:

3.1. Теоретические основы лабораторной работы

3.1.1. Кратно-масштабное представление функций

При анализе сигналов часто полезно представить сигнал и виде совокупности его последовательных приближений. Например, при передаче изображения можно сначала передать грубую его версию, а затем последовательно ее уточнить. Такая стратегия передачи имеет выгоды, например, при осуществлении выбора изображений из некоторой базы данных, когда необходимо быстро просмотреть большое количество картинок. Разрешение и размеры выбранной программы должны затем кратномасштабно увеличиться.

Теория кратно-масштабного анализа базируется на теории функциональных пространств.

$$\ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \qquad (3.1)$$

$$\bigcap_{m\in Z} V_m = \{0\}, \quad \overline{\bigcup_{m\in Z} V_m} = L^2(R).$$

Эти пространства имеют следующее свойство: для любой функции $f(x)\in V_m$, ее сжатая версия будет принадлежать пространству V_{m-1}.

$$f(x)\in V_m \Rightarrow f(2x)\in V_{m-1} \qquad (3.2)$$

И, наконец, последнее свойство кратно-масштабного анализа: существует такая функция $\phi(x)\in V_0$, что ее сдвиги $\phi_{0,n}(x)= \phi(x-n),\ n\in Z$ образуют ортонормированный базис пространства V_0. На рис. 3,1 схематично показаны данные вложенные пространства.

Так как функции $\phi_{0,n}(x)$ образуют ортонормированный базис пространства V_0, то функции

$$\phi_{m,n}(x)=2^{-m/2}\,\phi(2^{-m}x-n) \qquad (3.3)$$

образуют ортонормированный базис пространства V_m. Эти базисные функции называются масштабирующими, так как они создают масштабированные версии функций в $L^2(R)$. Из кратно-масштабного анализа следует, что функция $f(x)$ в $L^2(R)$ может быть представлена множеством последовательных ее приближений $f_m(x)$ в V_m. То есть функция $f(x)$ есть предел аппроксимаций $f_m(x)\in V_m$, при m, стремящемся к $-\infty$:

$$f(x) = \lim_{m\to-\infty} f_m(x). \qquad (3.4)$$

Отсюда появляется возможность анализа функции или сигнала на различных уровнях разрешения, или масштаба. Переменная m называется масштабным коэффициентом, или уровнем анализа. Если значение m велико, то функция в V_m есть грубая аппроксимация $f(x)$, и детали отсутствуют. При малых значениях m имеет место точная аппроксимация. Из определения кратно-масштабного анализа следует, что все функции в V_m могут быть представлены как линейная комбинация масштабирующих функций. В действительности, $f_m(x)$ есть ортогональная проекция $f(x)$ на V_m.

$$f_m(x) = \sum_n \left\langle \phi_{m,n}(x) f(x) \right\rangle \phi_{m,n}(x) = \sum_n c_{m,n}\phi_{m,n}(x). \qquad (3.5)$$

Так как $\phi(x)=\phi_{0,n}(x) \in V_0 \subset V_{-1}$, можно записать

$$\phi_{0,n}(x) = 2^{\frac{1}{2}}\sum_n h_n \phi_{-1,n}(x) = 2\sum_n h_n \phi\,(2x-n), \qquad (3.6)$$

где h_n – некоторая последовательность. Равенство (3.6) является одним из основных в теории вейвлет-анализа и имеет различные названия в литературе. Мы будем называть его далее масштабирующим уравнением.

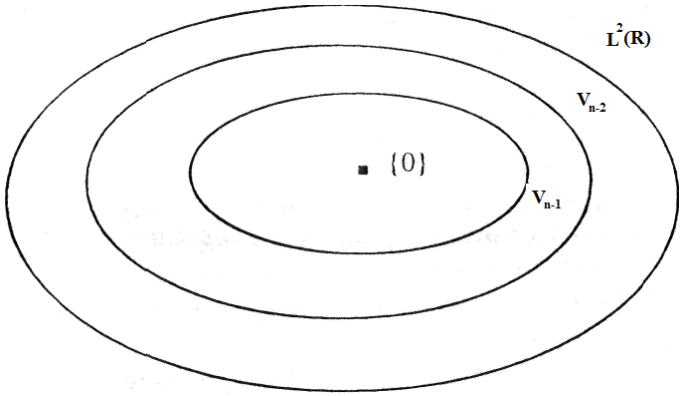


Рис. 3.1. Кратно-масштабное представление $L^2(R)$

Функция $\phi(x)$ и последовательность h_n тесно связаны между собой. Выведем соответствующие отношения. Из (3.6) можно получить

$$\phi_{m-1,k}(x) = 2^{1/2}\sum_p h_p \phi_{m,p-2k}(x) = 2^{(-m+1)/2}\sum_p h_p \phi(2^{-m}x-(p-2k)). \qquad (3.7)$$

Выполним операцию скалярного произведения $\phi_{m,n-2k}(x)$ с обеих сторон равенства (3.7):

$$\left\langle \phi_{m-1,k}(x), \phi_{m,n-2k}(x) \right\rangle = \left\langle 2^{1/2} \sum_p h_p \phi_{m,p-2k}(x), \phi_{m,n-2k}(x) \right\rangle =$$

$$= 2^{1/2} \sum_p h_p \left\langle \phi_{m,p-2k}(x), \phi_{m,n-2k}(x) \right\rangle = 2^{1/2} h_n . \quad (3.8)$$

Отметим, что это равенство выполняется для любого *m*. Далее, если переписать (3.6) в частотной области, можно получить:

$$\Phi(\omega) = \mathrm{H}\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right). \quad (3.9)$$

При рекурсивном повторении формулы (3.9) получается выражение:

$$\Phi(\omega) = \prod_{m=1}^{\infty} \mathrm{H}\left(\frac{\omega}{2^m}\right). \quad (3.10)$$

Итак, последовательность h_n тесно связана с масштабирующей функцией. Кроме того, из концепции кратно-масштабного анализа вытекают следующие свойства. Во-первых, интегрируя (3.6) по всей числовой оси *x*, можно получить:

$$\sum_n h_n = 1, \quad (3.11)$$

так как для построения кратно-масштабного анализа среднее значение функции $\phi(x)$ не должно быть равным нулю. Во-вторых, в силу ортонормальности базисных функций:

$$\delta_{0,k} = \left\langle \phi_{0,0}(x), \phi_{0,k}(x) \right\rangle = 2\sum_n h_n h_{n-2k} . \quad (3.12)$$

Третье свойство последовательности h_n сформулируем в спектральной области. Из записи условия ортонормальности функций $\phi_{m,n}(x)$ в области спектра:

$$\sum_k \left|\Phi(\omega + 2k\pi)\right|^2 = 1 \quad (3.13)$$

можно получить следующее выражение:

$$\left|\mathrm{H}(\omega)\right|^2 + \left|\mathrm{H}(\omega+\pi)\right|^2 = 1. \quad (3.14)$$

Равенство (3.11) эквивалентно тому, что *H(0)=1*. Тогда из (3.14) следует, что $H(\pi) = 0$. Эти свойства последовательности h_n будут использованы

позднее. А пока оставим на время теорию и перейдем к простейшему примеру множества масштабирующих функций $L^2(R)$.

Рассмотрим множество сдвигов и растяжений единичной функции на единичном интервале:

$$\phi(x) = \begin{cases} 1, 0 \le x < 1 \\ 0, \textit{иначе}. \end{cases} \quad (3.15)$$

Таким образом, базисные функции с коэффициентом масштаба -1 имеют следующий вид:

$$\phi_{-1,n}(x) = \begin{cases} \sqrt{2}, n/2 \le x < (n+1)/2 \\ 0, \ \textit{иначе}. \end{cases} \quad (3.16)$$

Базисная функция и соответствующая ей последовательность изображены на рис. 3.2.

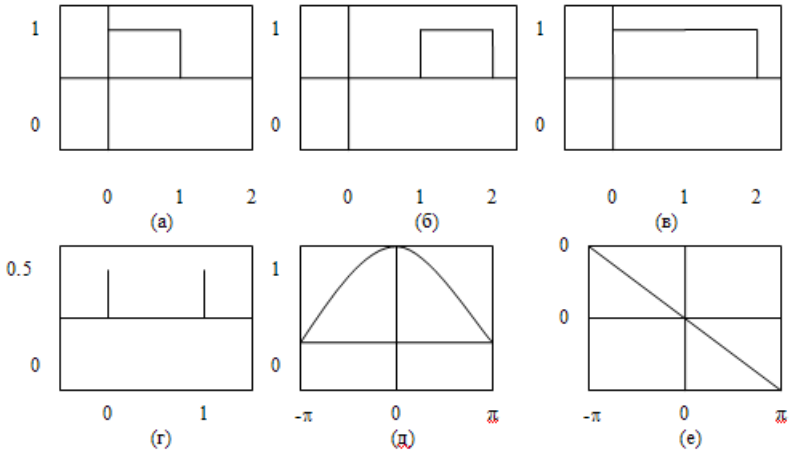


Рис. 3.2. Пример масштабирующей функции: (а) $\phi(x)$; (б) $\phi_{0,n}(x)$; (в) $\phi_{1,n}(x)$; (г) последовательность h_n; (д) $|\mathrm{H}(\omega)|$; (е) $\arg(\mathrm{H}(\omega))$

3.1.2. Представление функций при помощи вейвлетов

При рассмотрении рис. 3.1. видно, что область $L^2(R)$ построена из множества «колец», которые есть разность между двумя соседними про-

странствами. Эти разностные пространства обозначаются через W_m и определяются как ортогональные дополнения областей V_m до V_{m-1}:

$$V_{m-1} = V_m \oplus W_m,\ \bigcap_{m\in Z} W_m = \{0\},\ \overline{\bigcup_{m\in z} W_m} = L^2(R). \tag{3.17}$$

Пусть $\psi(x) = \psi_{0.0}(x)$ есть базисная функция W_0. Так как $\psi_{0,0}(x) \in W_0 \subset V_{-1}$, можно записать

$$\psi_{0,0}(x) = 2^{1/2} \sum_n g_n \phi_{-1,n}(x) \tag{3.18}$$

для некоторой последовательности g_n. По аналогии с ранее рассмотренным множеством функций $\phi_{m,n}(x)$ определим семейство вейвлет-функций:

$$\psi_{0,0}(x) = 2^{-m/2} \psi(2^{-m}x - n) \tag{3.19}$$

Эти функции образуют ортонормированный базис $L^2(R)$.

Существуют строгие зависимости между функциями $\psi(x)$, $\phi(x)$, g_n, h_n. Вначале получим формулу, аналогичную (3.10). Перепишем (3.18) для частотной области:

$$\psi(\omega) = G\left(\frac{\omega}{2}\right)\Phi\left(\frac{\omega}{2}\right), \tag{3.20}$$

заменим $\Phi(\omega)$ бесконечным произведением (3.10) и получим

$$\psi(\omega) = G\left(\frac{\omega}{2}\right)\prod_{m=2}^{\infty} \mathrm{H}\left(\frac{\omega}{2^m}\right). \tag{3.21}$$

Отметим, что $\Psi(\omega)$ пропорционально бесконечному произведению $\mathrm{H}(2^{-m}\omega)$, а не $G(2^{-m}\omega)$, так же как и в (3.18), вейвлет $\psi(x)$ был выражен в виде линейной комбинации масштабирующих функций. Теперь получим выражения, связывающие последовательности g_n и h_n. Так как W_m есть ортогональное дополнение V_m, функции $\psi_{0,0}(x)$ и $\phi_{0,0}(x)$ должны быть ортогональны, и из (3.6) и (3.18) следует, что

$$0 = \langle \phi_{0,n}, \psi_{0,n} \rangle = 2\sum_n \sum_p h_n g_p \langle \phi_{-1,n}, \phi_{-1,p} \rangle = 2\sum_n h_n g_n . \tag{3.22}$$

Легко увидеть, что выбор

$$g_n = (-1)^n h_{-n+2t-1} \tag{3.23}$$

будет корректен для всех $t \in Z$. Эквивалент (3.23) в частотной области представляется в виде

$$G(\omega) = -\mathrm{H}(-\omega + \pi)e^{-i\omega(2t+1)}. \tag{3.24}$$

С учетом этого из (3.20) получим

$$\Psi(\omega) = -e^{i\omega t 2}\mathrm{H}\left(-\frac{\omega}{2} + \pi\right)\Phi\left(\frac{\omega}{2}\right), \tag{3.25}$$

где без потери общности выбрано t=0.

Наконец отметим, что функция $\psi(x)$ и последовательность g_n имеют нулевое среднее. Этот факт легко проверить, подставляя ω=0 в (3.25) и (3.24) и используя свойство $H(\pi)$=0:

$$\Psi(0) = \int_{-\infty}^{\infty} \psi(x)dx = 0 \tag{3.26}$$

$$G(0) = \sum_n g_n = 0 \cdot \tag{3.27}$$

Определение функции вейвлетов позволяет нам записать любую функцию $f(x) \in L^2(R)$ в виде суммы проекций на W_j, $j \in R$:

$$f(x) = \sum_{j=-\infty}^{\infty} e_j(x), \tag{3.28}$$

где

$$e_j(x) = \sum_k \langle \psi_{j,k}(x), f(x) \rangle \psi_{j,k}(x). \tag{3.29}$$

Если осуществлять анализ функций вплоть до некоторого масштаба m, то $f(x)$ будет представлена суммой ее грубой аппроксимации $f_m(x) \in V_m$ и множества деталей $e_j(x) \in W_j$:

$$f(x) = f_m(x) + \sum_{j=-\infty}^{m} e_j(x) = \sum_n \langle \phi_{m,n}(x), f(x) \rangle \phi_{m,n}(x) + \sum_{j=-\infty}^{m} \sum_k \langle \psi_{j,k}(x), f(x) \rangle \psi_{j,k}(x) =$$

$$= \sum_n c_{m,n}\phi_{m,n}(x) + \sum_{j=-\infty}^{m} d_{j,k}\psi_{j,k}(x). \tag{3.30}$$

В качестве примера семейства вейвлет-функций, образующих ортонормальный базис пространства $L^2(R)$, на рис.3.3 показан вейвлет соответствующий вейвлет-функции на рис. 3.2. Это семейство вейвлетов называется вейвлетами Хаара и является одним из наиболее широко-распространенных вейвлетов на сегодняшний день. Кроме того, вейвлеты Хаара считаются одними из первых вейвлетов, предназначенных для кратно-масштабного анализа функций.

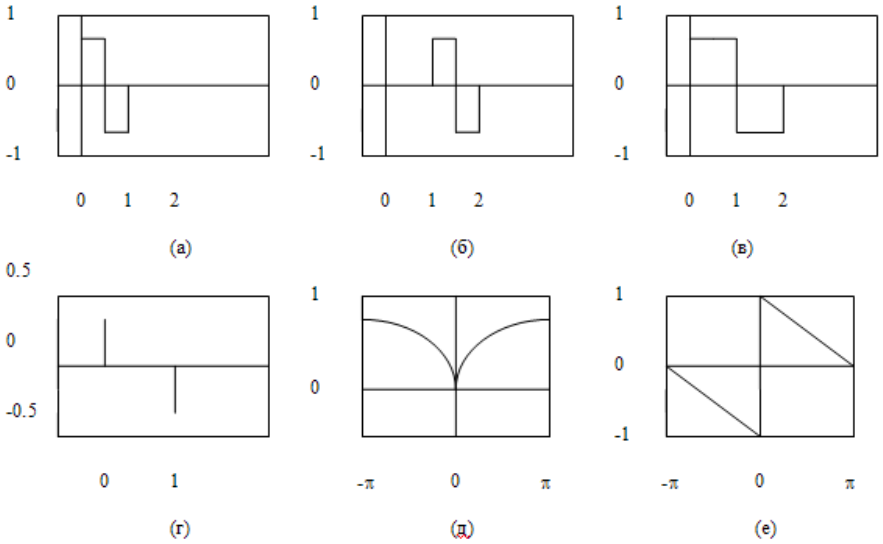


Рис. 3.3. Пример вейвлет-функции: (а) $\psi(x)$; (б) $\psi_{0,1}(x)$; (в) $\psi_{1,0}(x)$; (г) последовательность g_n; (д) $|G(\omega)|$; (е) $\arg(G(\omega))$

Из теории алгебраических структур известно, что в случае ортогональных вейвлетов последовательности h_n и g_n не могут быть симметричными, если длина каждой из них превышает 2. Однако во многих приложениях свойство симметричности является важным. В этом случае отказываются от требования ортогональности и на вейвлет-функции налагают менее строгое требование биортогональности. Выражения для биортогонального кратно-масштабного анализа полностью аналогичны выписанным выше формулам и здесь не приводятся.

3.1.3. Вейвлет-ряды дискретного времени

В большинстве приложений цифровой обработки сигналов и изображений мы имеем дело с дискретными сигналами. Поэтому с практической точки зрения интерес для нас представляют дискретные аналоги алгоритмов CTWT и CTWS, которые предназначены для преобразования дискретного сигнала в непрерывный и непрерывного в дискретный соответственно. К сожалению, формулы для вейвлет-преобразования и рядов вейвлетов дискретного времени (DTWT и DTWS) нельзя получить простой дискретизацией соответствующих формул для непрерывного времени. Кроме того, также невозможно определить кратно-масштабный анализ для дискретных сигналов, так как не существует базисных функций, масштабированные и смещенные версии которых давали бы на базисе пространства $L^2(R)$, пространства квадратично суммируемых последовательностей бесконечной длины.

Попробуем вывести формулы для DTWS из формул кратно-масштабного анализа соответствующего раздела. Пусть имеется некоторая непрерывная функция $f_0(x) \in V_0$. Наш дискретный сигнал c_n представим как последовательность коэффициентов при масштабирующих функциях, по которым раскладывается $f_0(x)$:

$$f_0(x) = \sum_n c_{0,n}\phi_{0,n}(x), \tag{3.31}$$

где $c_{0,n} = c_n$. Другими словами, мы интерпретируем наш сигнал как последовательность коэффициентов разложения, полученную в ходе кратно-масштабного анализа функции $f_0(x)$. Тогда мы можем вычислить аппроксимации этой функции, принадлежащие пространствам $V_1, V_2, \ldots$. Пространства $V_{-1}, V_{-2} \ldots$ не имеют значения при данной интерпретации.

Согласно концепции кратно-масштабного анализа функция $f_0(x)$ декомпозируется на две функции $f_1(x) \in V_1$ и $e_1(x) \in W_1$:

$$f_0(x) = f_1(x) + e_1(x) = \sum_k c_{1,k}\phi_{1,k}(x) + \sum_k d_{1,k}\psi_{1,k}(x). \tag{3.32}$$

Таким образом, получили две новые последовательности – $c_{1,n}$ и $d_{1,n}$. Этот процесс может быть продолжен по $f_1(x)$, и функция $f_0(x)$ (а также и последовательность c_n) будет представлена совокупностью коэффициентов $d_{m,n}, m,n \in Z$.

Итак, концепция DTWS определена. Однако вычисления пока зависят от непрерывных функций $\phi(x)$ и $\psi(x)$. Поэтому покажем, как вычисления DTWS могут быть выполнены с использованием операций только над дискретными сигналами.

С учетом того, что масштабирующая функция образует базис соответствующего пространства, из (3.8) можно получить

$$c_{1,k} = \left\langle \phi_{1,k}(x), f_1(x) \right\rangle = \left\langle \phi_{1,k}, f_0(x) - e_1(x) \right\rangle = \left\langle \phi_{1,k}(x), \sum_n c_{0,n}\phi_{0,n}(x) \right\rangle =$$

$$= \sum_{n\in Z} c_{0,n} \left\langle \phi_{1,k}(x), \phi_{0,n}(x) \right\rangle = 2^{1/2} \sum_{n\in Z} c_{0,n-2k.}. \qquad (3.33)$$

Так что оказывается возможным итеративное вычисление коэффициентов $c_{j,k}$ и $d_{j,k}$ без непосредственного использования функций $\phi(x)$ и $\psi(x)$.

По аналогии с (3.32) можно записать для произвольного *j*:

$$c_{j,k} = 2^{1/2} \sum_n c_{j-1,n} h_{n+2k}, \qquad (3.34)$$

$$d_{j,k} = 2^{1/2} \sum_n c_{j-1,n} g_{n+2k}, \qquad (3.35)$$

получив, таким образом, полностью дискретный процесс декомпозиции.

Последовательности h_n и g_n называют фильтрами. Отметим, что $c_{j,k}$ и $d_{j,k}$ имеют «половинную» длину по сравнению с $c_{j-1.k}$ (хотя, конечно, на данном этапе все последовательности бесконечны). Таким образом, не вводится избыточности.

Обратный процесс заключается в получении c_{j-1} из c_j и d_j:

$$c_{j-1,n} = \left\langle \phi_{j-1,n}, f_{j-1}(x) \right\rangle = \left\langle \phi_{j-1,n}(x), f_j(x) + e_j(x) \right\rangle =$$

$$= \left\langle \phi_{j-1,n}, \sum_k c_{j,k}\phi_{j,k}(x) \right\rangle + \left\langle \phi_{j-1,n}, \sum_k d_{j,k}\psi_{j,k}(x) \right\rangle =$$

$$= \sum_k c_{j,k} \left\langle \phi_{j-1,n}, \phi_{j,k}(x) \right\rangle + \sum_k d_{j,k} \left\langle \phi_{j1-,n}, \psi_{j,k}(x) \right\rangle =$$

$$= 2^{1/2} \sum_k c_{j,k} h_{n+2k} + 2^{1/2} \sum_k d_{j,k} g_{n+2k}. \qquad (3.36)$$

Отметим, что суммирование производится по другим переменным по сравнению с формулами (3.33) и (3.34). Длина последовательности c_{j-1} вдвое больше длины последовательности c_j или d_j.

Подставляя (3.33) и (3.34) в (3.35), получаем следующие ограничения на фильтры h_n и g_n:

$$2\sum_k \left(h_{n+2k} h_{p+2k} + g_{n+2k} g_{p+2k} \right) = \delta_{n,p}, \qquad (3.37)$$

$$2\sum_n h_{n+2k} h_{n+2p} = 2\sum_n g_{n+2k} g_{n+2p} = \delta_{k,p}, \qquad (3.38)$$

$$2\sum_n h_{n+2k} h_{n+2p} = 0. \qquad (3.39)$$

Выражение (3.36) для временной области эквивалентно выражениям (3.14) и (3.24) для частотной. Равенства (3.37) и (3.38) уже появлялись ранее, но в менее общей форме.

3.2. Дискретное вайвлет-преобразование

На практике DTWS должно применяться к сигналам конечной длины. Таким образом, его необходимо модифицировать, чтобы из сигнала какой-то длины получать последовательность коэффициентов той же длины. Получившееся преобразование называется дискретное вайвлет-преобразование (DWT).

Вначале опишем DWT в матричном виде, а затем – на основе банков фильтров, что наиболее часто используется при обработке сигналов.

В обоих случаях мы предполагаем, что базисные функции $\phi(x)$ и $\psi(x)$ компактно определены. Это автоматически гарантирует финитность последовательностей h_n и g_n.

3.2.1. Матричное описание DWT

Обозначим через вектор v^j последовательность конечной длины $c_{j,n}$ для некоторого *j*. Этот вектор преобразуется в вектор v^{j+1}, содержащий последовательности $c_{j+1,n}$ и $d_{j+1,n}$, каждая из которых имеет длину в два раза меньше, чем исходная последовательность. Преобразование может быть записано в виде матричного умножения $v^{j+1} = \mathrm{M}_j v^j$, где матрица M_j - квадратная и состоит из нулей и элементов h_n, умноженных на $\sqrt{2}$. В силу свойств h_n матрица M_j - является ортонормированной, и обратная ей матрица равна транспонированной.

Возьмем фильтр длиной *L=4*, последовательность данных предполагаем имеет длинну *N=8*, а в качестве начального значения возьмем – *j=0*. Последовательность g_n получим из h_n по формуле (3.23), где *t=L/(2-1)=4*. Тогда операцию матрично-векторного умножения возможно будет представить в следующем виде:

$$\begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ d_{1,0} \\ d_{1,1} \\ d_{1,2} \\ d_{1,3} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & -h_2 & h_1 & -h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & 0 & 0 & 0 & 0 & h_3 & -h_2 \end{bmatrix} * \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{0,4} \\ c_{0,5} \\ c_{0,6} \\ c_{0,7} \end{bmatrix}. \quad (3.40)$$

Обратное преобразование эквивалентно умножению v^{j+1} на обратную матрицу M_j^T:

$$\begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ c_{0,4} \\ c_{0,5} \\ c_{0,6} \\ c_{0,7} \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & h_2 & h_3 & 0 & 0 & h_1 \\ h_1 & 0 & 0 & h_3 & -h_2 & 0 & 0 & -h_0 \\ h_2 & h_0 & 0 & 0 & h_1 & h_3 & 0 & 0 \\ h_3 & h_1 & 0 & 0 & -h_0 & -h_2 & 0 & 0 \\ 0 & h_2 & h_0 & 0 & 0 & h_1 & h_3 & 0 \\ 0 & h_3 & h_1 & 0 & 0 & -h_0 & -h_2 & 0 \\ 0 & 0 & h_2 & h_0 & 0 & 0 & h_1 & h_3 \\ 0 & 0 & h_3 & h_1 & 0 & 0 & -h_0 & -h_2 \end{bmatrix} * \begin{bmatrix} c_{1,0} \\ c_{1,1} \\ c_{1,2} \\ c_{1,3} \\ d_{1,0} \\ d_{1,1} \\ d_{1,2} \\ d_{1,3} \end{bmatrix}. \quad (3.41)$$

Таким образом, выражение (3.39) – это один шаг DWT. Полное DWT заключается в итеративном умножении верхней половины вектора v^{j+1} на квадратную матрицу M_{j+1}, размер которой 2^{d-1}. Эта процедура может быть повторена *d* раз, пока длина вектора не станет равна 1 (или столько раз, сколько уровней разложения необходимо получить для дальнейшего анализа данных).

В четвертой и восьмой строках матрицы (39) последовательность h_n циклично сдвинута: коэффициенты, выходящие за пределы матрицы справа,

помещены в ту же строку слева. Это означает, что DWT есть точно один период длины N DTWS сигнала $\tilde{c}_{0,n}$, получаемого путем бесконечного периодического продолжения $c_{0,n}$. Так что DWT, будучи определенным таким образом, использует такую же периодичность сигнала, как и в случае с DFT.

Матричное описание DWT кратко и ясно. Однако при обработке сигналов DWT чаще всего описывается посредством блок-диаграммы, аналогичной диаграмме системы анализа-синтеза.

3.2.2. Описание DWT посредством блоков фильтров

Рассматривая субполосные преобразования, мы интерпретировали равенства, аналогичные (3.33) и (3.34), как фильтрацию с последующим прореживанием в два раза. Так как в данном случае имеется два фильтра h_n и g_n, то банк фильтров является двухполосным и может быть изображен, как показано на рис. 3.4.

Фильтры F и E означают фильтрацию фильтрами h_{-n} и g_{-m} соответственно. В нижней ветви схемы выполняется низкочастотная фильтрация. В результате получается некоторая аппроксимация сигнала, лишенная деталей низкочастотная (НЧ) субполоса. В верхней части схемы выделяется высокочастотная (ВЧ) субполоса. отметим, что при обработке сигналов константа $2^{t/2}$ всегда выносится из банка фильтров и сигнал домножается на 2.

Итак, схема рис. 3.4 делит сигнал уровня j=0 на два сигнала уровня j=1. Далее, вейвлет-преобразование получается путем рекурсивного применения данной схемы к НЧ части. При осуществлении вейвлет-преобразования изображения каждая итерация алгоритма выполняется вначале к строкам, затем – к столбцам изображения (строится так называемая пирамида Малла, позволяющая представить 2-мерное преобразование как последовательность одномерных преобразований). Например, в цифровых программных видеокодеках формата ADV6xx применена модифицированная пирамида Малла, когда на каждой итерации не обязательно выполняется преобразование и по строкам, и по столбцам. Хотя это сделано лишь для более полного учета зрительного восприятия человека.

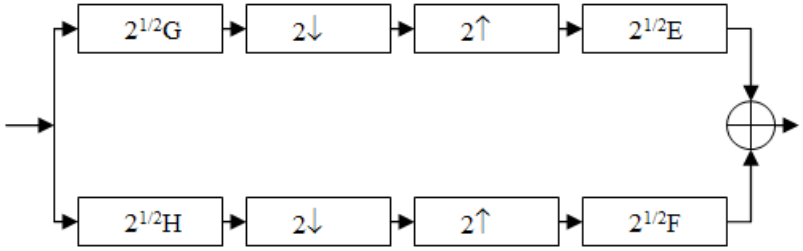


Рис. 3.4. Схема двухполосного банка фильтров

Получившееся преобразование аналогично (3.39). Однако существуют некоторые различия. При фильтрации сигнала конечной длины мы сталкиваемся с проблемой его продолжения на границе. Матричное выполнение DWT эквивалентно периодическому продолжению сигнала на границе. Этот тип продолжения является обязательным для ортогональных фильтров. В случае применения биортогональных фильтров появляются некоторые другие возможности в силу симметричности их характеристик. Подробнее этот вопрос будет рассматриваться позднее.

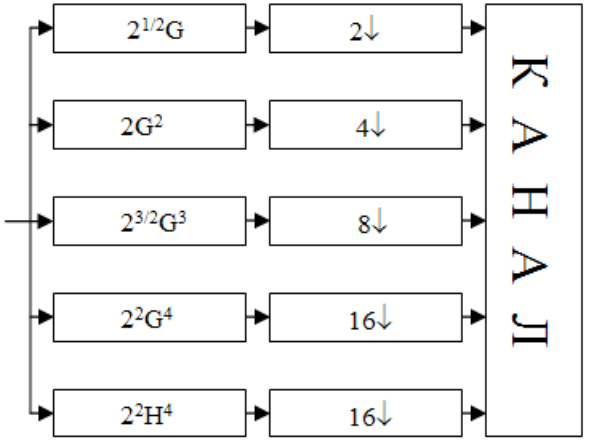


Рис. 3.5. Эквивалентная схема вейвлет-преобразования

Схему, выполняющую DWT, можно представить еще и как показано на рис. 3.5. Здесь рекурсивная фильтрация и прореживание заменены од-

ной операцией фильтрации и одной операцией прореживания на каждую субполосу.

Определение итерационных фильтров h_0^l и g_0^j легче всего дать в частотной области:

$$\begin{aligned} \mathrm{H}^j(\omega) &= \mathrm{H}(\omega)\times \mathrm{H}(2\omega)\times \mathrm{H}(4\omega)\times \ldots \times \mathrm{H}(2^{j-1}\omega) = \prod_{m=1}^{j} \mathrm{H}(2^{m-1}\omega), \\ G^j(\omega) &= G(\omega)\times \mathrm{H}(2\omega)\times \mathrm{H}(4\omega)\times \ldots \times \mathrm{H}(2^{j-1}\omega) = G(\omega)\times \prod_{m=2}^{j} \mathrm{H}(2^{m-1}\omega). \end{aligned} \qquad (3.42)$$

3.3. Пример использования вейвлет-преобразования.

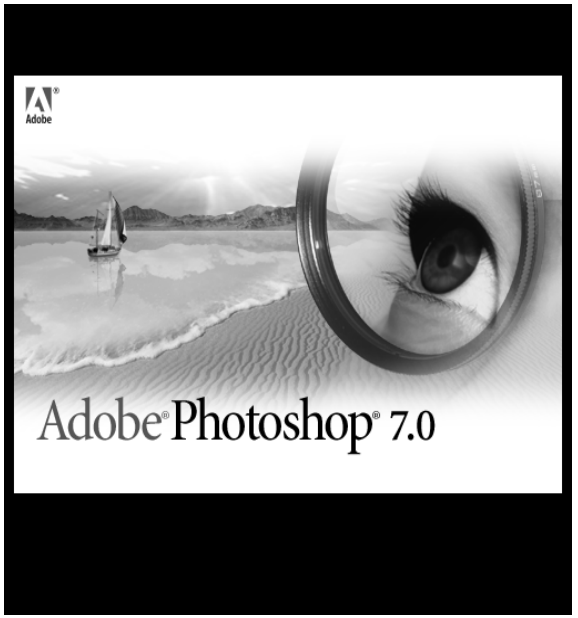


Рис. 3.6. Исходное изображение

Рис 3.7. Построчное преобразование

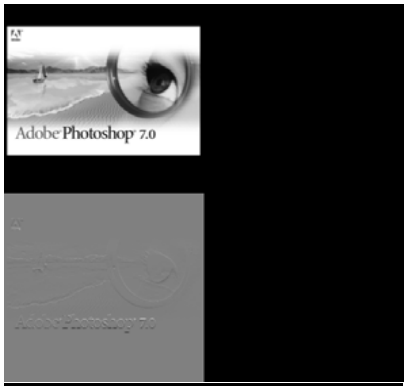


Рис 3.8 Постолбцовое преобразование

Обратные преобразования после фильтрации:

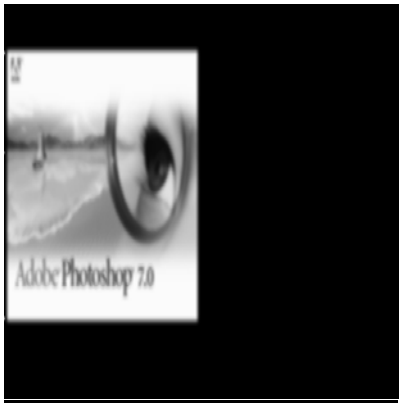


Рис 3.9. Построчное преобразование

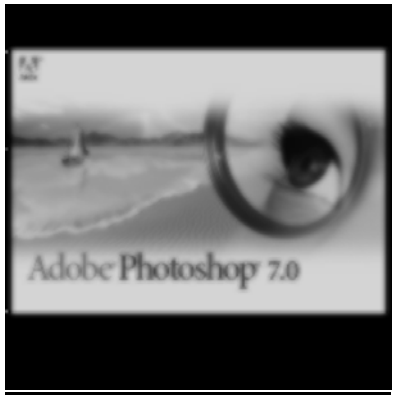


Рис 3.10 Постолбцовое преобразование

3.4. Выполнение лабораторной работы

3.4.1. Общий план выполнения работы

1. Изучить принципы кратно-масштабного представления функций.
2. Изучить понятие дискретного вейвлет-преобразования.
3. Написать подпрограмму разделения функции яркости изображения на низкочастотную и высокочастотную составляющие.
4. Написать подпрограмму получения (сбора) функции яркости изображения по низкочастотной и высокочастотной составляющим.
5. Провести 2 итерации разделения сигнала на низкочастотную и высокочастотную составляющие.
6. Сгладить шумы на получившемся изображении (пороговая обработка).
7. Восстановить изображение (2 итерации сбора сигнала по низкочастотной и высокочастотной составляющим).
8. Составить отчет о выполненной работе.
9. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

3.4.2. Порядок проведения работы

1. Разделение функции яркости изображения на низкочастотную и высокочастотную составляющие.

При осуществлении вейвлет-преобразования изображения каждая итерация алгоритма выполняется вначале к строкам, затем – к столбцам изображения.

Таким образом, применяем формулу (3.40) раздела **(3.4.1)**, используя в качестве входной последовательности **C** сначала строки, а затем столбцы исходного изображения. При этом, в силу используемого фильтра, $h_i = g_i$. Размеры матрицы **H** равны размерам изображения.

При получении результата к правой (а затем нижней) частям изображения необходимо прибавить математическое ожидание сигнала для улучшения визуального восприятия изображения. При выполнении пункта 3 это математическое ожидание необходимо вычесть.

2. Фильтрация шумов.

Необходимо сгладить небольшие шумы (невысоким произвольным порогом) на высокочастотной составляющей (правая и нижняя части изображения). После восстановления такого изображения границы на нем будут более «мягкими».

3. Построение функции яркости изображения по низкочастотной и высокочастотной составляющим.

При осуществлении обратного вейвлет-преобразования изображения каждая итерация алгоритма выполняется вначале к столбцам, затем – к строкам изображения.

Таким образом, применяем формулу (3.41) раздела **(3.4.1)**, используя в качестве входных последовательностей **C** (первая половина строки/столбца) и **D** (вторая половина строки/столбца) сначала столбцы, а затем строки изображения, полученного в пункте 2. При этом, в силу используемого фильтра, $h_i = g_i$. Размеры матрицы **H** равны размерам изображения.

4. При разделении и построении функции яркости изображения попробовать различные типы фильтров.

Например:

4.1. $h_i = \frac{1}{4},\ i = \overline{0,3}.$

4.2. $h_i = \frac{1}{2},\ i = \overline{0,2};\ h_3 = -\frac{1}{2}.$

Объяснить разницу в результатах.

Примечание. Для работы над изображениями использовать матрицы формата* double*, для того чтобы избежать обнуления или переполнения разрядов (и соответствующего «перескока» значений яркости).

3.4.3. Содержание отчета

1. Результат выполнения первой части задания:

а) изображение низкочастотной и высокочастотной составляющих изображения (с любым типом фильтра из п. 4).

2. Результат выполнения второй части задания:

а) изображение низкочастотной и высокочастотной составляющих изображения со сглаженными шумами.

3. Результат выполнения третьей части задания:

а) изображение, воссозданное по низкочастотной и высокочастотной составляющим (с любым типом фильтра из п. 4).

В результате проделанной работы необходимо получить:

3 изображения на разных стадиях преобразования (с любым типом фильтра из п. 4);

1 программный модуль, состоящий из 2-х подпрограмм – прямое и обратное вейвлет-преобразование.

3.5. Контрольные вопросы

1. Что такое кратно-масштабное представление данных?
2. Что понимать под представлением функций при помощи вейвлетов?
3. Какие условия накладываются на последовательности h_n и g_n?
4. Каким образом строится система масштабирующих функций?
5. Примеры масштабирующей функции.
6. Вейвлеты Хаара.
7. Чем отличается дискретное вейвлет-преобразование?
8. Схема вейвлет-преобразования.

3.6. Данные для выполнения лабораторной работы

3.6.1. Общие данные

При работе с изображениями использовать полутоновые или цветные изображения размером, не менее чем ***400x400*** точек, предоставленные преподавателем или выбранные самостоятельно (по договоренности с преподавателем).

3.6.2. Задание

Задание 1

Написать программу, реализующую разделение функции яркости изображения (полутоновое или цветное изображение) на 2 составляющих: низкочастотную и высокочастотную, представляющих собой фон и детали изображения соответственно.

Более подробно см. п. 1 части *Порядок проведения работы*.

Задание 2

Написать функцию фильтрации полученной высокочастотной составляющей изображения с использованием заданного порогового значения.

Подробнее п. 2 части *Порядок проведения работы*.

Задание 3

Написать подпрограмму воссоздания функции яркости изображения по низкочастотной и высокочастотной составляющим.

Более подробно см. п. 3 части *Порядок проведения работы.*

Написать отчет по результатам лабораторной работы.

3.6.3. Варианты заданий:

Все студенты выполняют одно и то же задание с разными исходными данными в виде изображений, которые необходимо получить у преподавателя (см. рис. 3.11).

Вариант1

Вариант 2

Вариант 3

Вариант 4

Вариант 5

Вариант 6

Вариант 7

Вариант 8

Рис. 3.11 – Варианты заданий.

Примечание. Желательно провести 2-3 итерации вейвлет-преобразования. То есть сначала провести 2 итерации разложения, потом фильтрацию воссоздание.

3.7. Список рекомендуемой литературы

1. Чуи, К. Введение в Вейвлеты / К. Чуи. – М.: Мир, 2001.
2. Петухов, А.П. Введение в теорию базисов вейвлетов: учеб. пособие / А.П. Петухов. – СПб: СПбГТУ, 1999.
3. Воробьев, В.И. Теория и практика вейвлет преобразования / В.И. Воробьев, В.Г. Грибунис. – СПб: Изд-во ВУС, 1999.

4. Лабораторная работа № 4
Построение нейронных сетей

Цель работы – ознакомление студентов с теоретическими основами нейронных сетей и их применением; изучение методов обучения нейронных сетей; получение навыков программного моделирования нейронных сетей.

Содержание работы:

4.1. Теоретические основы лабораторной работы

4.1.1. Основные понятия

Функционирование нейронной сети (НС) имитирует функционирование человеческой нервной системы.

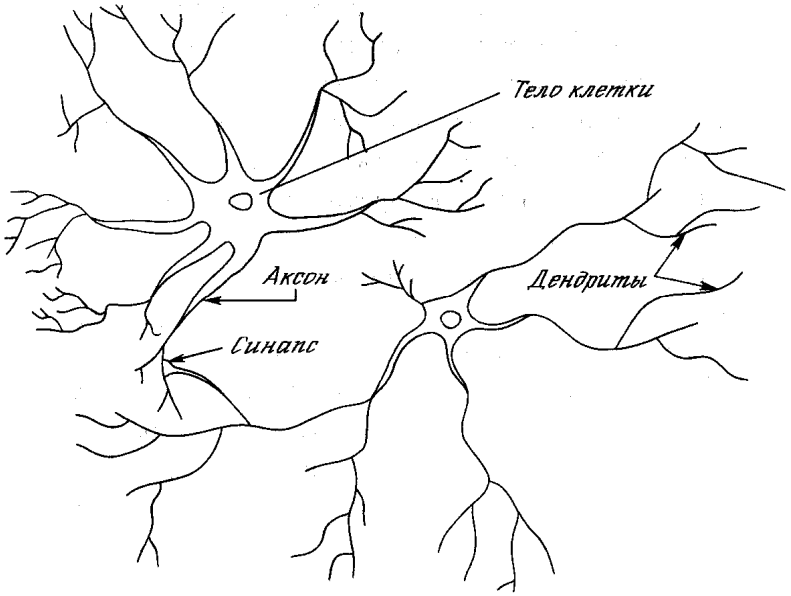


Рис 4.1. Биологический нейрон

Биологический нейрон представляет собой устройство, имеющее несколько входов (дендриты) и один выход(аксон). Каждому входу ставится в соответствие некоторых весовой коэффициент (синапс), характеризующий пропускную способность канала и оценивающий степень влияния сигнала с этого входа на сигнал на выходе. В теле нейрона происходит аккумулирование этих возбуждений, и далее это значение является аргументом активационной функции нейрона.

Для описания алгоритмов и устройств в нейроинформатике выработана специальная "схемотехника", в которой элементарные устройства – сумматоры, синапсы, нейроны и т.п. объединяются в сети, предназначенные для ре-

шения поставленной задачи. Причем внутренняя структура связей между элементами сети не зависит от вида задачи. Стандартный формальный нейрон составлен из входного сумматора, нелинейного преобразователя и точки ветвления на выходе.

Наиболее важный элемент нейросистем – это адаптивный сумматор. Адаптивный сумматор вычисляет скалярное произведение вектора входного сигнала x на вектор параметров α.

На схемах будем обозначать его так, как показано на рис. 4.2. Адаптивным называем его из-за наличия вектора настраиваемых параметров α. Ее вычисление также можно представить с помощью адаптивного сумматора, имеющего n+1 вход и получающего на 0-й вход постоянный единичный сигнал (рис. 4.3).

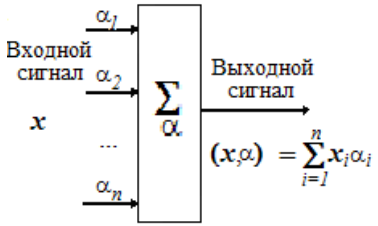


Рис 4.2. Адаптивный сумматор

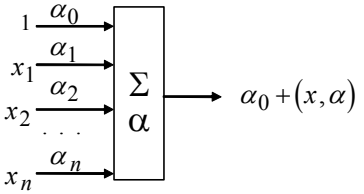
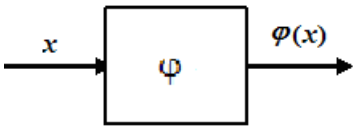


Рис 4.3. Неоднородный адаптивный сумматор

Нелинейный преобразователь сигнала изображен на рис. 4.4. Он получает скалярный входной сигнал x и переводит его в $\varphi(x)$.

Точка ветвления служит для рассылки одного сигнала по нескольким адресам (рис. 4.5). Она получает скалярный входной сигнал x и передает его всем своим выходам.



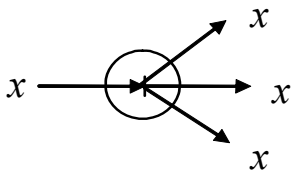


Рис 4.5. Точка ветвления

Рис 4.4. Нелинейный преобразователь сигнала.

В дальнейшем будем рассматривать некий искусственный нейрон, представленный в общем виде (рис. 4.6).

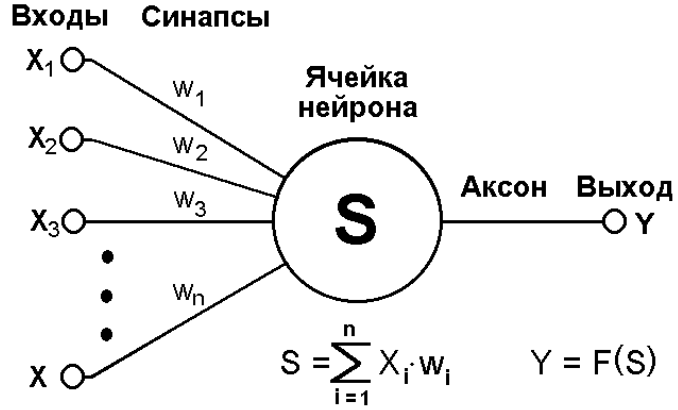


Рис 4.6. Искусственный нейрон

Текущее состояние нейрона определяется как взвешенная сумма всех его входов:

$$s = \sum_{i=1}^{n} x_i \cdot w_i \qquad (4.1)$$

Выход нейрона есть функция его состояния:

$$y = f(s). \qquad (4.2)$$

4.1.2. Активационные функции

Нелинейная функция f называется активационной и может иметь различный вид, как показано на рис. 4.7.

Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоид (т.е. функция S-образного вида):

$$f(x) = \frac{1}{1 + e^{-\alpha x}}. \qquad (4.3)$$

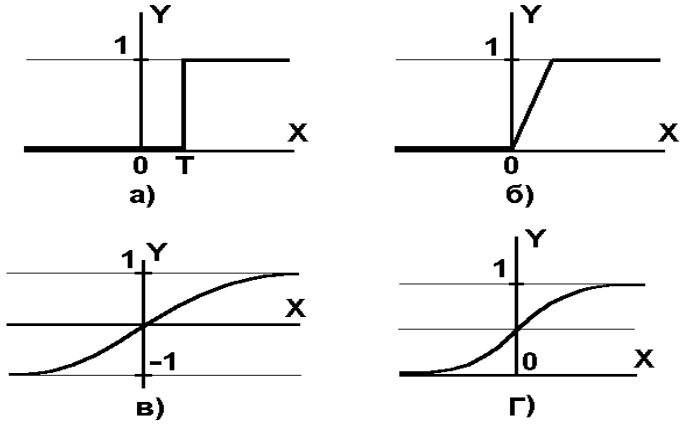


Рис 4.7. Активационные функции: а – функция единичного скачка; б – линейный порог (гистерезис); в – сигмоид (гиперболический тангенс); г – сигмоид (формула (4.3))

Единственным параметром сигмоида является значение коэффициента α. При уменьшении α сигмоид становится более пологим, в пределе при $\alpha=0$ вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоид приближается по внешнему виду к функции единичного скачка с порогом T в точке x=0. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне [0,1]. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной (одно из необязательных требований при построении активационной функции – это простота вычисления производной), применение которого будет рассмотрено в дальнейшем.

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)) \qquad (4.4)$$

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

Когда в качестве активационной функции выступает пороговая функция, то нейрон называется персептроном. В качестве активационной функции также используют нелинейные функции, что дает возможность для решения большего количества задач. Примеры активационных функций представлены в табл. 4.1.

Примеры активационных функций Таблица 4.1.

Название функции	Вид активационной функции
Функция Ферми	$y = f(s) = \frac{1}{1+e^{-s}}$
Гиперболический тангенс	$y = f(s) = th(s)$
Рациональная сигмоида	$y = f(s) = \frac{s}{1+\lvert s \rvert}$

4.1.3. Архитектура нейронных сетей

Хотя один нейрон и способен выполнять простейшие процедуры распознавания – он способен отвечать на вопросы типа «да»-«нет» в зависимости от вида входных импульсов, вся сила нейронных сетей состоит в соединений нейронов в сети.

Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рис. 4.8. Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений и поэтому не считаются слоем (иногда их называют 0-м слоем). Каждый элемент из множества входов X отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. Хотя в искусственных и биологических сетях многие соединения и могут отсутствовать, все соединения показаны в целях общности восприятия картины нейронной сети.

Веса считаются элементами матрицы W. Матрица имеет m строк и n столбцов, где m – число входов, а n – число нейронов. Таким образом, вычисление выходного вектора N, компонентами которого являются выходы OUT нейронов, сводится к матричному умножению

$$\mathbf{Y=F(XW)}, \qquad (4.5)$$

где **X** и **Y** – соответственно входной и выходной сигнальные векторы-строки, **F(V)** – активационная функция, применяемая поэлементно к компонентам вектора **V**.

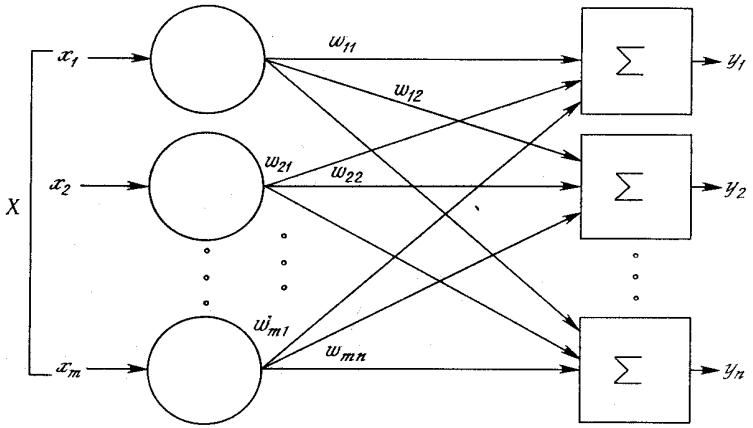


Рис 4.8. Однослойная нейронная сеть

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями.

Многослойные сети могут образовываться каскадами слоев. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рисунке 4.9.

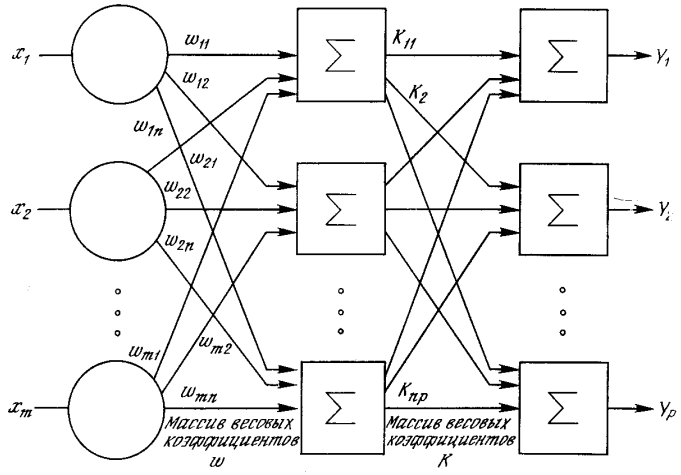


Рис 4.8. Двухслойная нейронная сеть

Среди различных структур нейронных сетей (НС) одной из наиболее известных является многослойная структура, в которой каждый нейрон произ-

вольного слоя связан со всеми аксонами нейронов предыдущего слоя или, в случае первого слоя, со всеми входами НС. Такие НС называются ***полносвязными***. Увеличение вычислительной мощности многослойной сети по сравнению с однослойной сетью возможно лишь в том случае, если активационная функция между слоями будет нелинейной. Вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением (если отсутствует нелинейная активационная функция) результирующего вектора на вторую весовую матрицу.

$$\mathbf{N} = (\mathbf{X}\mathbf{W}_1)\mathbf{W}_2. \qquad (4.6)$$

Так как умножение матриц ассоциативно, то:

$$\mathbf{N} = \mathbf{X}(\mathbf{W}_1\mathbf{W}_2). \qquad (4.7)$$

Уравнения (4.6), (4.7) показывают, что двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью. Таким образом, для расширения возможностей сетей по сравнению с однослойной сетью необходима нелинейная активационная функция.

4.1.4. Применение

Нейронные сети решают широкий круг задач. Это автоматизация процессов распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем, организация ассоциативной памяти и многие другие приложения.

Широкий круг задач, решаемый НС, не позволяет в настоящее время создавать универсальные, мощные сети, вынуждая разрабатывать специализированные НС, функционирующие по различным алгоритмам.

Доказано, что с помощью нейронных сетей можно сколь угодно точно аппроксимировать любую непрерывную функцию и имитировать любой непрерывный автомат. Но далеко не все важные задачи ставятся как задачи аппроксимации.

Системы из одного адаптивного сумматора

Даже *системы из одного адаптивного сумматора* находят очень широкое применение. Вычисление линейных функций необходимо во многих задачах. Вот неполный перечень «специальностей» адаптивного сумматора:

1. Линейная регрессия и восстановление простейших закономерностей.
2. Линейная фильтрация и адаптивная обработка сигналов.
3. Линейное разделение классов и простейшие задачи распознавания образов.

Задача линейной регрессии состоит в поиске наилучшего линейного приближения функции, заданной конечным набором значений: дана выборка значений вектора аргументов x1 , ..., xm, заданы значения функции F в этих точках: F(xi)=fi, требуется найти линейную (неоднородную) функцию φ(x)=(α,x)+α0, ближайшую к F. Чтобы однозначно поставить задачу, необходимо доопределить, что значит «ближайшую». Наиболее популярен метод наименьших квадратов, согласно которому φ ищется из условия

$$\sum_{i=1}^{m}(F(x^i)-\varphi(x^i))^2 \to \min . \qquad (4.8)$$

Координаты классифицируемых векторов представляют собой значения некоторых признаков (свойств) исследуемых объектов.

Эта задача возникает во многих случаях: при диагностике болезней и определении неисправностей машин по косвенным признакам, при распознавании изображений и сигналов и т.п.

Сети Кохонена для кластер-анализа и классификации без учителя

Построение отношений на множестве объектов - одна из самых загадочных и открытых для творчества областей применения искусственного интеллекта. Первым и наиболее распространенным примером этой задачи является классификация без учителя. Задан набор объектов, каждому объекту сопоставлен вектор значений признаков (строка таблицы). Требуется разбить эти объекты на классы эквивалентности.

Для каждого нового объекта мы должны выполнить две операции:

1) найти класс, к которому он принадлежит;

2) использовать новую информацию, полученную об этом объекте, для исправления (коррекции) правил классификации.

Отнесение объекта к классу проводится путем его сравнения с типичными элементами разных классов и выбора ближайшего.

Простейшая мера близости объектов – квадрат евклидового расстояния

между векторами значений их признаков (чем меньше расстояние – расстояние, тем ближе объекты). Соответствующее определение признаков типичного объекта - среднее арифметическое значение признаков по выборке, представляющей класс.

Другая мера близости, естественно возникающая при обработке сигналов, изображений и т.п. – квадрат коэффициента корреляции (чем он больше, тем ближе объекты).

Существует много эвристических алгоритмов классификации без учителя, основанных на использовании мер близости между объектами. Каждый из них имеет свою область применения, а наиболее распространенным недостатком является отсутствие четкой формализации задачи: совершается переход от идеи кластеризации прямо к алгоритму, в результате неизвестно, что ищется (но что-то в любом случае находится, иногда - неплохо).

4.2. Алгоритм Back-Propagation

Будучи соединенными определенным образом, нейроны образуют нейронную сеть. Работа сети разделяется на *обучение* и *применение*. Под обучением понимается процесс адаптации сети к предъявляемым эталонным образцам путем модификации весовых коэффициентов связей между нейронами. Этот процесс является результатом алгоритма функционирования сети, а не предварительно заложенных в нее знаний человека, как это часто бывает в системах искусственного интеллекта.

Рассмотрим полносвязную нейронную сеть. В многослойных сетях оптимальные выходные значения нейронов всех слоев, кроме последнего, как правило, не известны. Поэтому двух или болееслойный перцептрон уже невозможно обучить, руководствуясь только величинами ошибок на выходах НС.

Один из вариантов решения этой проблемы – распространение сигналов ошибки – идет от выходов нейронной сети к ее входам в направлении, обратном прямому распространению сигналов в обычном режиме работы. Этот алгоритм обучения нейронной сети получил название процедуры обратного распространения ошибки – Back Propagation (BP-алгоритм).

Данный алгоритм строится таким образом:

1) Подать на входы сети один из возможных образов сигнала в режиме обычного функционирования нейронной сети, когда сигналы распространяются непосредственно от входов к выходам, и рассчитать значения выходов нейронной сети.

$$s_j^{(n)} = \sum_{i=0}^{M^{n-1}} y_i^{(n-1)} w_{ij}^{(n)}, \qquad (4.9)$$

где M^{n-1} – число нейронов в слое **n-1** с учетом нейрона с постоянным выходным состоянием, равным единице, задающего смещение;

$s_j^{(n)}$ – j-й выход n-го слоя;

$y_i^{(n)} = f(s_i^{(n)})$, где $f(K)$ – активационная функция;

$y_q^{(0)} = x_q$, где x_q – q-я компонента входного образа.

2) Рассчитать $\delta^{(N)}$ для выходного слоя по формуле

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l}, \qquad (4.10)$$

где d_l – идеальное (желаемое) выходное состояние.

Рассчитать по формуле изменения весов $\Delta w^{(N)}$ слоя N:

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}) \qquad (4.11)$$

где μ – коэффициент инерционности;

t – номер текущей итерации;

η – коэффициент скорости обучения, $0<\eta<1$.

3) Рассчитать $\delta^{(N)}$ и $\Delta w^{(N)}$ для всех остальных слоев по формулам (4.11) и (4.10) соответственно, n=N-1,...1:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right] \frac{dy_j}{ds_j}, \qquad (4.12)$$

4) Скорректировать все веса нейронной сети

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \qquad (4.13)$$

5) Если ошибка сети существенна, перейти на шаг 1. В противном случае – выход из алгоритма.

Сети на шаге 1 попеременно предъявляются все тренировочные образы, чтобы сеть, образно говоря, не забывала одни по мере запоминания других.

4.3. Выполнение лабораторной работы

4.3.1. Общий план выполнения работы

1. Познакомиться с применением нейронных сетей.
2. Изучить устройство НС.
3. Изучить способ обучения сети методом ВР.
4. Написать подпрограмму, реализующую НС и ее обучение методом ВР.
5. Составить отчет о выполненной работе.
6. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

4.3.2. Порядок проведения работы

1. Спроектировать НС, т.е. определить структуры данных (векторы, матрицы, классы).
2. Реализовать НС на любом языке программирования.
3. Обучить НС на исходных изображениях. Обучение заключается в том, чтобы из изображения №1 получить наиболее похожее на изображение №2. Параметром схожести является СКО. Кол-во циклов обучения не менее 1000.
4. В режиме использования НС получить преобразованное изображение.

4.3.3. Содержание отчета

В результате проделанной работы необходимо получить:

1. Программный модуль.
2. Изображение, полученное при воздействии НС на данное.

4.4. Контрольные вопросы

1. Что такое НС?
2. Объясните назначение активационной функции.
3. Какие НС называются полносвязными?
4. Имеет ли смысл увеличение числа слоев однослойной сети в случае, если активационная функция между слоями линейная. Почему?
5. В чем заключается метод Back Propogation?

4.5. Данные для выполнения лабораторной работы

4.5.1. Постановка задачи

Даны два изображения(№1 и №2), каждое из которых задано матрицей ***200x200***, элементы матрицы имеют значения от -0.5 до +0.5 – значение соответствует яркости пиксела). Каждому пикселу изображения №2 с координатами i,j соответствует заданное по некоторому правилу множество пикселов изображения №1 – $U=\{(i_1,j_1),(i_2,j_2),(i_3,j_3),(i_4,j_4),(i_5,j_5)\}$. Значения пикселов U подаются на вход НС. Выход НС сравнивается с пикселем (i,j) изображения №2.

Сеть состоит из 5+1=6 входов; единственный скрытый слой содержит N+1 нейрон; выходной слой – 1 нейрон; активационная функция ***f(s)*** – одинакова для всех нейронов. Примерная схема изображена на рис. 4.9.

Написать программу, реализующую такую НС. Реализовать метод ВР. Обучить сеть так, чтобы при подаче на вход изображения №1 получилось изображение наиболее похожее на изображение №2. Данные к заданию представлены в табл. 6.1. Исходное и результирующее изображения можно получить у преподавателя.

Написать отчет по результатам лабораторной работы.

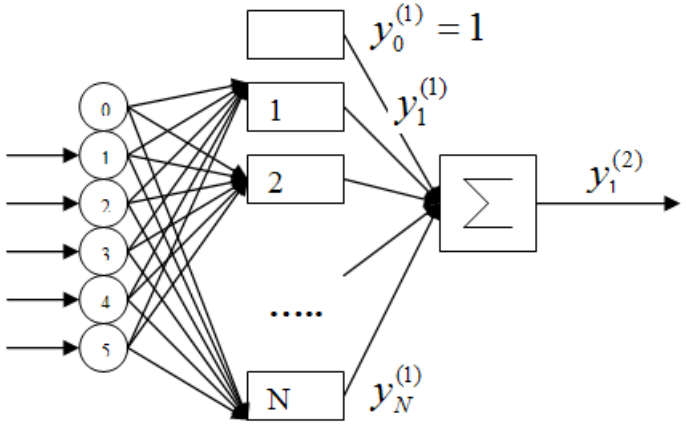


Рис 4.9. Примерная схема сети

4.5.2. Пример расчета формул

1) Подаем на входы сети один из возможных образов в режиме обычного функционирования нейронной сети, когда сигналы распространяются от входов к выходам, рассчитываем значения последних.

$$s_j^{(1)}=\sum_{i=0}^{5} x_i\, w_{ij}^{(1)}, \qquad s_1^{(2)}=\sum_{i=0}^{N} y_i^1 w_{i1}^{(2)},$$

где $y_j^{(n)}=f(s_j^{(n)})$, а выход сети $y=y_1^{(2)}=f(s_1^{(2)})$;

$f(s)=\dfrac{s}{1+abs(s)}$ – активационная функция; $\dfrac{dy_l}{ds_l}=\dfrac{1}{(1+abs(s_i))^2}$.

2) Рассчитываем $\delta^{(2)}$ для выходного слоя по формуле

$$\delta_1^{(2)}=(f(s_1^{(2)})-d)\cdot\frac{1}{(1+abs(s_1^{(2)}))^2},$$

где *d*– идеальное (желаемое) выходное состояние.

Рассчитать по формуле изменения весов $\Delta w^{(2)}$ слоя 2 (*i*=0…10)

$$\Delta w_{i1}^{(2)}(t)=-0.1\cdot\delta_1^{(2)}\cdot y_i^{(1)}.$$

3) Рассчитываем δ^{1} и Δw^1 (*j*=1…10; *i*=0...5):

$$\delta_j^{(1)}=\delta_1^{(2)} w_{j1}^{(2)}\cdot\frac{1}{(1+abs(s_j^{(1)}))^2},$$

$$\Delta w_{ij}^{(1)}(t)=-0.1\cdot\delta_j^{(1)}\cdot x_i.$$

4) Скорректировать все веса нейронной сети

$$w_{ij}^{(n)}(t)=w_{ij}^{(n)}(t-1)+\Delta w_{ij}^{(n)}(t).$$

5) Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Сети на шаге 1 попеременно предъявляются все тренировочные образы, лучше всего в произвольном порядке, чтобы сеть, образно говоря, не забывала одни по мере запоминания других.

4.5.3. Варианты заданий

Активационные функции f(s):

$$f(s) = \frac{1}{1+e^{-s}} - 0.5, \quad (1)$$

$$f(s) = \frac{s}{1+|s|}, \quad (2)$$

$$f(s) = th(s). \quad (3)$$

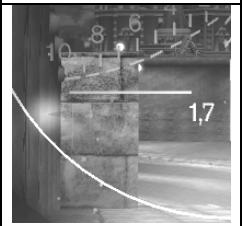
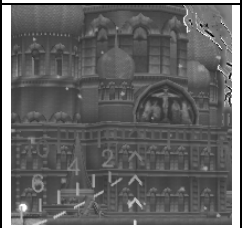
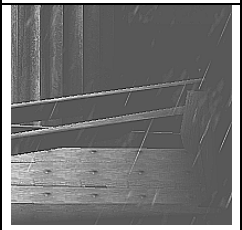
Образ входного сигнала:

$$U_1 = \{(i,j),(i+1,j),(i,j+1),(i-1,j),(i,j-1)\};$$
$$U_2 = \{(i,j),(i+1,j+1),(i-1,j-1),(i+1,j-1),(i-1,j)\};$$
$$U_3 = \{(i,j),(i+1,j+1),(i+2,j+2),(i-1,j\text{-}1),(i-2,j-2)\};$$
$$U_4 = \{(i+1,j),(i,j-1),(i,j+1),(i-1,j-1),(i-1,j+1)\};$$
$$U_5 = \{(i-1,j),(i,j-1),(i,j+1),(i+1,j-2),(i+1,j+2)\}.$$

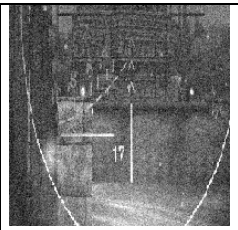
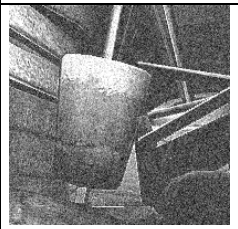
Варианты заданий — Таблица 4.2.

Вариант	U{}	N	η	μ	Image №1	Image №2	f(s)
1	1	30	0.15	0.11			(1)
2	3	29	0.11	0.11			(2)

Продолжение табл. 4.2.

Вари-ант	U{}	N	η	μ	Image №1	Image №2	f(s)
3	2	28	0.12	0.11			(1)
4	4	27	0.13	0.11			(3)
5	4	25	0.14	0.11			(2)
6	1	24	0.11	0.12			(3)
7	5	23	0.12	0.12			(2)

Окончание табл. 4.2.

8	3	22	0.13	0.12			(3)
9	3	21	0.14	0.12			(1)
10	5	20	0.11	0.10			(3)

4.6. Список рекомендуемой литературы

1. Короткий, С. Нейронные сети: алгоритм обратного распространения/ С. Короткий. – http://lii.newmail.ru/NN/KOROTKY/N2/kor_nn2.htm.

2. Иванченко, А.Г. Персептрон – системы распознавания образов/ А.Г. Иванченко. – Киев: Наукова думка, 1972.

3. Анил, К. Джей. Введение в искусственные нейронные сети/ К. Джей. Анил // Открытые системы. №4, 1997.

Глоссарий

1. Шифрование – представление информации в таком виде, что ее нельзя прочесть без использования специального ключа.
2. Частотный анализ – метод раскрытия шифра простой замены при помощи расчета частоты повторяемости букв.
3. Цифровой сигнал – дискретная функция дискретного аргумента, обычно несущая информацию о некотором процессе.
4. Цифровое изображение – графическая форма представления данных, предназначенная для зрительного восприятия.
5. Фрактал – структура, состоящая из частей, которые в каком-то смысле подобны целому.
6. Геометрический фрактал – фракталы, которые получают с помощью некоторой ломаной (или поверхности в трехмерном случае), называемой *генератором*.
7. Алгебраические фракталы – получаются с помощью нелинейных процессов в **n**-мерных пространствах.
8. Кратно-масштабный анализ – анализ функции на разных уровнях разрешения (или масштаба).
9. Базисная функция – функция, которая является базисным вектором, если совокупность всевозможных функций представлять как линейное пространство.
10. Вейвлеты (от англ. wavelet), всплески – это математические функции, позволяющие анализировать различные частотные компоненты данных.
11. Нейроны – нервные клетки, структурно-функциональные единицы нервной системы. Кора головного мозга человека содержит 10-20 миллиардов нейронов.
12. Нейронная сеть — это математическая модель, а также устройства параллельных вычислений, представляющие собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Как математическая модель искусственная нейронная сеть представляет собой частный случай методов распознавания образов или дискриминантного анализа.
13. Линейная регрессия – наилучшее линейное приближение функции, заданной конечным набором значений.
14. Среднеквадратическая ошибка (СКО) – наиболее распространенная мера сравнения изображений. Определяется как сумма квадратов разностей значений пикселов исходного и восстановленного изображений.
15. Квантование цифрового сигнала – процесс замены исходного значения отсчета сигнала другим значением, как правило приводящий к уменьшению количества возможных значений цифрового сигнала.

Учебное издание

Копенков Василий Николаевич, Сергеев Владислав Викторович

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ОБРАБОТКИ ДАННЫХ
ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Учебное пособие

Технический редактор С. Б. П о п о в
Редакторская обработка А. В. Я р о с л а в ц е в а
Корректорская обработка А. А. Н е ч и т а й л о
Доверстка, В. С. Т е л е п о в а

Подписано в печать 16.10.07. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Печ. л. 6.0
Тираж 120 экз. Заказ . ИП-110/2007

Самарский государственный
аэрокосмический университет.
443086 Самара, Московское шоссе, 34.

Изд-во Самарского государственного
аэрокосмического университета.
443086 Самара, Московское шоссе, 34.