МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Самарский государственный аэрокосмический университет имени
академика С.П. Королева
(национальный исследовательский университет)» (СГАУ)

В.А. Федосеев

ЦИФРОВЫЕ ВОДЯНЫЕ ЗНАКИ И СТЕГАНОГРАФИЯ

Учебное пособие с заданиями
для практических и лабораторных работ

Рекомендовано Редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский государственный аэрокосмический университет им. академика С.П. Королева (национальный исследовательский университет)» (СГАУ) в качестве учебного пособия для студентов, обучающихся по образовательным программам высшего образования укрупненной группы направлений 10.00.00 Информационная безопасность

Самара
Издательство СГАУ
2015

УДК 004.9(075)
ББК 32.97я7
Ф338

Рецензенты: зав. каф. мультисервисных сетей и информационной безопасности ПГУТИ, д.т.н., проф. *В.Г. Карташевский*;
проф. каф. технической кибернетики СГАУ, д.т.н., доцент *А.Г. Храмов*

Федосеев, В.А.

Ф338 **Цифровые водяные знаки и стеганография: учебное пособие с заданиями для практических и лабораторных работ** / В.А. Федосеев. – Электрон. текстовые и граф. дан. – Самара: СГАУ, 2015. – 128 с. – 1 эл. опт. диск (CD-ROM).

ISBN 978-5-7883-1062-6

Учебное пособие посвящено изучению базовых методов защиты информации цифровыми водяными знаками, методов цифровой стеганографии (то есть передачи скрытых сообщений внутри цифровых контейнеров), а также методов противодействия системам, реализующим подобные методы.

Материал разбит на 8 глав, большинство из которых содержат теоретический материал, а также практические и лабораторные задания, заключающиеся в программной реализации и исследовании различных систем встраивания информации в цифровые изображения и видео. В одной из глав приводится сжатый обзор среды MATLAB, используемой для выполнения упражнений.

Пособие предназначено для студентов группы направлений «Информационная безопасность».

УДК 004.9(075)
ББК 32.97я7

ISBN 978-5-7883-1062-6

Оглавление

Указатель рассмотренных систем встраивания информации

Список сокращений

БИХ	–	Бесконечная импульсная характеристика
ДВП	–	Дискретное вейвлет-преобразование
ДКП	–	Дискретное косинусное преобразование
ДОП	–	Дискретное ортогональное преобразование
ДПФ	–	Дискретное преобразование Фурье
ИХ	–	Импульсная характеристика
КИХ	–	Конечная импульсная характеристика
ЛИС-система	–	Линейная система, инвариантная к сдвигу
СВИ	–	Система встраивания информации
НЗБ	–	Наименее значимые биты
НЗБП	–	Наименее значимая битовая плоскость
ЦВЗ	–	Цифровой водяной знак

Список обозначений

Основные обозначения

$\mathbb{N}$	–	Множество натуральных чисел
$\mathbb{N}_0 = \mathbb{N} \cup \{0\}$	–	Множество целых неотрицательных чисел
$\mathbb{B}^n = \mathbb{N}_0 \cap [0, 2^n - 1]$	–	Множество целых неотрицательных чисел, для хранения которых достаточно n бит
$\mathbb{B} = \mathbb{B}^1$	–	Множество, элементы которого равны 0 или 1
$\mathbb{Z}$	–	Множество целых чисел
$\mathbb{R}$	–	Множество действительных чисел
$\mathbb{C}$	–	Множество комплексных чисел
$\mathbb{S}^m_{[N_1 \times N_2 \times \ldots \times N_m]}$	–	m-мерная матрица размерами $N_1 \times N_2 \times \ldots \times N_m$ из элементов некоторого множества $\mathbb{S}$
$\mathbb{S}^m_{[]}$	–	m-мерная матрица некоторого размера из элементов некоторого множества $\mathbb{S}$ (употребляется, когда размеры матрицы не важны в рассматриваемом контексте)
$\mathbb{X}^m_{[]}$, где $\mathbb{X} \subseteq \mathbb{R}$	–	Множество цифровых сигналов
$\mathbb{Y}^l_{[]}$, где $\mathbb{Y} \subseteq \mathbb{C}$	–	Множество матриц признаков цифровых сигналов

Обозначения данных в системах встраивания информации

Обозначение	Множество значений	Название	Употребимые эквиваленты в англоязычной литературе
$\mathbf{b}$	$\mathbb{B}^1_{[N_b]}$	Встраиваемая информация	Secret message, watermarking code
$\mathbf{b}^{\mathbf{R}}$	$\mathbb{B}^1_{[N_b]}$	Извлечённая информация	Recovered {название $\mathbf{b}$}
C	$\mathbb{X}^m_{[]}$	Контейнер	Host asset, container
C^W	$\mathbb{X}^m_{[]}$	Носитель информации	Watermarked asset, cover
$\widetilde{C^W}$	$\mathbb{X}^m_{[]}$	Принятый носитель информации	Transformed watermarked asset
f	$\mathbb{Y}^l_{[]}$	Матрица признаков контейнера	–
f^W	$\mathbb{Y}^l_{[]}$	Матрица признаков носителя информации	–
$\widetilde{f^W}$	$\mathbb{Y}^l_{[]}$	Матрица признаков принятого носителя информации	–
$\mathbf{k}$		Секретный или составной ключ СВИ	–
W	$\mathbb{X}^m_{[]}$	Встраиваемый сигнал	Watermarking message (signal), encoded message
W^R	$\mathbb{X}^m_{[]}$	Извлечённый сигнал	Recovered {название W}
ξ	$\mathbb{B}$	Результат обнаружения	Detection result
Ω	$\mathbb{Y}^l_{[]}$	Матрица признаков встраиваемой информации	–
$\tilde{\Omega}$	$\mathbb{Y}^l_{[]}$	Матрица признаков извлечённой информации	–

My sister Laura's bigger than me
And lifts me up quite easily
I can't lift her, I've tried and tried;
She must have something heavy inside
Spike Milligan

Сестре Лауре – восемь лет,
А мне пока что – пять.
Ей ничего не стоит
Шутя меня поднять.

А я пытался столько раз –
И раз, и два, и три…
Но в ней, наверно, что-то есть
Тяжелое внутри.
Спайк Миллиган
(пер. Григория Кружкова)

Предисловие

Настоящее учебное пособие содержит базовую информацию о методах защиты информации цифровыми водяными знаками, методах цифровой стеганографии (то есть передачи скрытых сообщений внутри цифровых контейнеров), а также методах противодействия системам, реализующим подобные методы.

Тематика пособия относится к весьма популярному направлению информатики, именуемому в англоязычной литературе термином “Information Hiding” и методологически находящемуся на стыке цифровой обработки сигналов и изображений и информационной безопасности.

Учебное пособие содержит теоретический материал, а также практические и лабораторные задания, заключающиеся в программной реализации и исследовании различных систем встраивания информации в цифровые изображения и видео. Практические задания предназначены для совместной реализации группой студентов при помощи преподавателя в компьютерном классе. Они направлены на формирование лучшего понимания рассматриваемых в лекционном курсе методов, алгоритмов и систем. Лабораторные задания содержат индивидуальные варианты для разных студентов и предназначены для проверки усвоенных знаний.

При составлении теоретического материала автор старался включать в пособие только ту информацию, которая будет необходима студентам при выполнении практических и лабораторных заданий. Поэтому за рамками пособия остались несколько важных разделов, представляющих главным образом теоретический интерес или попросту не нашедших своё место в практической части курса. Так, в пособии не рассмотрены особенности восприятия человеком визуальной и звуковой информации, методы обеспечения построения систем цифровых водяных знаков, стойких к геометрическим преобразованиям, системы встраивания информации в звуковые сигналы, методы генерации ключей встраивания, наиболее актуальные стеганографические системы.

Пособие состоит из 8 глав. Первая глава в краткой форме излагает основные теоретические основы рассматриваемого направления (Information Hiding). Во второй главе представлен сжатый обзор среды MATLAB, в которой предлагается выполнять лабораторные и практические зада-

ния. Главы с третьей по седьмую посвящены описанию методов встраивания информации в цифровые сигналы, различающиеся по назначению, типу контейнера и используемым алгоритмам. В восьмой главе рассматриваются атаки на некоторые из рассмотренных систем.

Данное учебное пособие главным образом разработано в поддержку курса «Компьютерной стеганографии», читаемого студентам Самарского государственного аэрокосмического университета, обучающимся по направлению «Информационная безопасность автоматизированных систем». Однако оно также будет полезно и студентам смежных специальностей, в учебные планы и программы которых включены рассматриваемые разделы.

В общей сложности в пособии рассмотрены 19 систем встраивания информации, большинство из которых исследуются в рамках практических и лабораторных работ. Простые системы рассматривались в полном объёме, более сложные иногда упрощались, чтобы сконцентрироваться на их сути. Каждая из пяти лабораторных работ содержит 12 вариантов заданий, из которых первые 6 являются более простыми, остальные – более сложными.

При написании настоящего пособия использовались в основном книги [1, 2] и конспекты лекций, читаемых автором в университете. Все материалы, заимствованные из сторонних источников, сопровождаются ссылками на оригинал. Изображения, приведённые в качестве иллюстраций, взяты из стандартного репозитория [3] Университета Ватерлоо.

Пособие подготовлено при поддержке государственного задания вузу №2014/198, код проекта 2298 (главы 3-8), а также гранта Президента Российской Федерации молодым учёным МК-4506.2015.9 (глава 1).

Автор признателен коллеге Юлии Выборновой за техническую помощь в подготовке издания.

1. Теоретические аспекты встраивания информации в цифровые сигналы

1.1. История и предмет дисциплины

Вопросы, рассматриваемые в рамках настоящего учебного пособия, относятся к области знаний, именуемой в англоязычной литературе “Information Hiding” или “Data Hiding”. Она посвящена методам сокрытия цифровой информации внутри других информационных объектов, хранящихся в цифровом виде. Данное направление информатики сформировалось к середине 90-х годов XX века, а первые крупные монографии появились лишь на рубеже тысячелетий. Наиболее серьёзный вклад в её развитие внесли Ingemar Cox [4, 5, 2], Jessica Fridrich [2, 6], Mauro Barni, Franco Bartolini [1], Fabien Petitcolas [7, 8], Stefan Katzenbeisser [8], Eric Cole [9], Birgit Pfitzmann [10].

В рамках данного пособия мы будем переводить “Information Hiding” как «встраивание информации». Общепринятое русскоязычное название в настоящее время отсутствует, а термин «встраивание информации» кажется автору более предпочтительным по сравнению с термином «сокрытие информации», поскольку в ряде методов защиты данных цифровыми водяными знаками встроенная информация *может* и даже *должна быть* визуально различимой, то есть *не скрытой* от глаз. Кроме того, такое название позволяет явным образом отгородиться от предмета и задач криптографии, которая занимается *сокрытием содержания* информации [11].

Итак, под *встраиванием информации* (в узком смысле) будем понимать область знаний, охватывающую широкий круг проблем внедрения информации (называемой в различных ситуациях *секретной информацией*, *секретным сообщением* или *цифровым водяным знаком*) в содержимое другого информационного объекта (называемого *открыто передаваемой информацией* или *контейнером*).

Методы встраивания информации могут быть разделены на 4 основные категории, как показано в табл. 1.1.

Табл. 1.1. Классификация методов встраивания информации

	Сообщение связано с контейнером	Сообщение не связано с контейнером
Факт наличия сообщения сокрыт	Стеганографическое встраивание ЦВЗ (1)	Скрытая передача информации (стеганографическая) (2)
Факт наличия сообщения известен	Нестеганографическое встраивание ЦВЗ (3)	Открытая опосредованная передача информации (4)

Вслед за авторами книги [2] приведём исторические примеры использования каждой группы методов:

1. В 1981 году фотографические оттиски конфиденциальных документов британского кабинета оказались напечатанными в газетах. Согласно слухам, для определения источника утечки Маргарет Тэтчер установила порядок распространения однозначно идентифицируемых копий документов для каждого из её министров. Каждая копия имела уникальные интервалы между словами, которые были использованы для кодирования личности получателя. Таким образом могли быть установлены источники утечки информации.

2. Скрытая передача информации, не связанной с контейнером, всегда являлась важной задачей для военных. Например, согласно договору ОСВ-II между СССР и США, обеим державам допускалось иметь достаточно много бункеров ракет, но лишь ограниченное число ракет. Для проверки соблюдения договора каждый участник соглашения должен был устанавливать датчики, разработанные в другой стране, в своих хранилищах ракет. Каждый такой датчик должен был только сообщать о наполненности бункера, в котором он установлен, и ничего больше. Однако по свидетельствам некоторых источников [12], внутри законных сообщений удавалось спрятать также и дополнительную информацию, касающуюся, в частности, местонахождения бункера.

3. Пример нестеганографического водяного знака (т. е. водяного знака, наличие которого является известным) можно увидеть, к примеру, на электронных картах Google. Каждая плитка карты имеет слабозаметный водяной знак, защищающий права Google как владельца изображения, и на это обстоятельство указывает сообщение в нижней части каждой веб-страницы. Знание того, что водяные знаки встроены в каждое

изображение, помогает сдерживать несанкционированное использование этих материалов.

4. В качестве примера открытой опосредованной передачи информации можно упомянуть вставки кода времени в радиоэфире на заданной частоте, которые практиковались в конце 1940-х годов. Код внедрялся с периодичностью 15 минут. Его было слышно в эфире, но он не являлся водяным знаком, так как сообщение (текущее время) не было связано с содержанием передачи.

В рамках настоящего учебного пособия будут рассматриваться как стеганографические методы, так и методы встраивания цифровых водяных знаков, но все они будут предназначены для контейнеров, являющихся цифровыми сигналами, имеющими физическую природу. К таким сигналам можно отнести изображения, видео, звуковые сигналы. Наибольшее внимание будет уделено изображениям, однако многие из рассмотренных методов применимы и к контейнерам других типов.

1.2. Краткие сведения о системах встраивания информации

Понятие систем встраивания информации

Совокупность методов и средств, образующих единое решение для встраивания информации в цифровой сигнал, будем называть *системой встраивания информации* (СВИ). К СВИ относятся *стеганографические системы* (*стегосистемы*), предназначенные для скрытой передачи информации, и *системы встраивания цифровых водяных знаков* (ЦВЗ), предназначенные для защиты контейнера. Последние мы будем сокращённо обозначать как *ЦВЗ-системы*. Любая система встраивания информации состоит из двух основных блоков:

1) подсистемы встраивания информации;
2) подсистемы извлечения информации.

В первой происходит внедрение встраиваемой информации в цифровой сигнал-*контейнер* в соответствии с *секретным ключом*. Во второй подсистеме происходит либо извлечение встроенной информации, либо проверка наличия в принятом сигнале встроенной информации. Предполагается, что контейнер со встроенной информацией (который будем называть *носителем информации*) передаётся по открытому каналу, в ко-

тором он может подвергнуться искажениям и атакам. Упрощённая схема СВИ представлена на рис. 1.1.

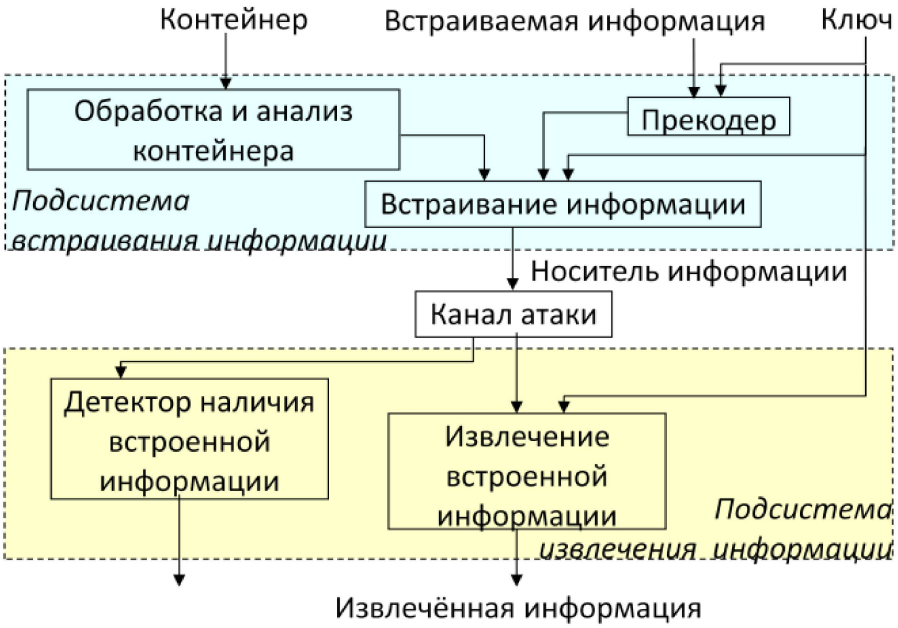


Рис. 1.1 – Упрощённая схема системы встраивания информации

Ключевым требованием, возникающим при проектировании *стеганографических систем*, является недопустимость обнаружения наличия скрытой информации несанкционированным получателем. Поэтому основной целью атак на такие системы является обнаружение факта наличия встроенной информации (извлечение её содержания не является необходимым). Разработка таких атак является задачей *стегоанализа*. Если стегосистема является устойчивой к ним, то говорят, что она обладает *стеганографической стойкостью* [13, 14, 15]. Очевидно, что методы, используемые для скрытой передачи информации, должны позволять встраивать большой объём данных.

Ключевой характеристикой *ЦВЗ-систем* также является стойкость, но она имеет несколько иной смысл. Под *стойкостью ЦВЗ-систем* понимается возможность извлечения встроенной информации из искажённого носителя информации. При этом круг значимых искажений определяется в зависимости от области применения метода встраивания данных. Более того, в ряде задач (защита от изменений, защита от копирования [1]) требуется, чтобы ЦВЗ не был стоек к определённым преобразованиям. Такие водяные знаки могут называться *полухрупкими* или *хрупкими*. Более подробно вопросы стойкости СВИ рассматриваются в параграфе 8.1, а системы хрупких и полухрупких ЦВЗ – в главе 6.

Назначение систем встраивания информации

Существует достаточно широкий круг задач, для решения которых могут использоваться методы встраивания информации. Наиболее значимыми из них являются:

1) защита авторских прав,
2) защита от несанкционированного распространения,
3) защита от изменений,
4) защита от подделки,
5) скрытая передача информации.

Задача *защиты авторских прав* может быть решена с использованием сценария «Демонстрация законного права собственности» [1], при котором автор или владелец объекта авторского права встраивает в него *стойкий ЦВЗ*, однозначно определяющий его как владельца.

Задача *защиты от несанкционированного распространения* может быть решена с использованием сценария «Сдерживание копирования» [1], согласно которому владелец распространяемого информационного объекта, представляющего собой определённую ценность, встраивает в каждую копию различные ЦВЗ (которые в данном случае называются *цифровыми отпечатками пальцев*), однозначно определяющие получателя документа. Если в дальнейшем где-либо будет обнаружена несанкционированная копия, то ее происхождение может быть восстановлено путем извлечения встроенной информации. Таким образом, данная задача тоже решается при помощи стойких ЦВЗ-систем.

Задача *защиты от изменений* может быть решена с использованием *хрупких водяных знаков*, которые разрушаются при какой-либо модификации носителя информации, к примеру, при воспроизведении его на копировальном аппарате. Таким образом, само наличие ЦВЗ является подтверждением подлинности защищаемого сигнала и отсутствия проведённых над ним несанкционированных изменений.

Задача *защиты от подделки* может быть решена посредством встраивания специальных меток, воспроизведение которых является сложной задачей. Эти метки могут быть реализованы в виде стойких ЦВЗ.

Задача *скрытой передачи информации* является ключевой задачей стеганографии. Поэтому для её решения используются стегосистемы.

Свойства систем встраивания информации

При описании систем встраивания информации принято выделять присущие им основные *свойства*. Эти свойства определяют детали подсистем встраивания и извлечения информации, стойкость к различным атакам, а также некоторые численные показатели. Таким образом, они представляют собой важную информацию о системе встраивания информации и в конечном счёте определяют возможности её использования. Ниже перечислены наиболее важные из этих свойств.

1. *Действие, выполняемое подсистемой извлечения информации:* проверка наличия встроенной информации (*детектирование*) или извлечение встроенной информации (*декодирование*).
2. *Знание исходного контейнера подсистемой извлечения информации:* если ни исходный контейнер, ни какие-либо из его параметров не известны на этапе извлечения информации, то такое извлечение называется *слепым*, в противном случае оно называется *неслепым*.
3. *Возможность извлечения встроенной информации:* только санкционированными адресатами или любыми участниками процедуры обмена информацией. В первом случае встроенная информация называется *частной*, во втором – *публичной*.
4. *Тип контейнера*: звук, изображение, видео и пр.
5. *Подбор способа встраивания информации к предопределённому методу извлечения информации:* если это справедливо, то встраивание называют *информированным*, в противном случае – *слепым*.
6. *Способ модификации сигнала при встраивании информации*.
7. *Визуальная различимость* встроенной информации.
8. *Максимально возможный объем встраиваемой информации*, который допускает СВИ.
9. *Возможность повторного встраивания* другой информации в тот же сигнал тем же методом.
10. *Стойкость встроенной информации* к искажениям её носителя. По этому признаку принято разделять системы на секретные, стойкие, полухрупкие и хрупкие. В *секретных СВИ* стойкость встроенной информации должна сохраняться как при преднамеренных атаках, так при непреднамеренных искажениях. *Стойкие*

СВИ защищены только от произвольных непреднамеренных искажений. *Полухрупкие СВИ* устойчивы к одним преобразованиям и неустойчивы к другим, в то время как в *хрупких* системах встроенная информация разрушается даже при незначительных модификациях заполненного контейнера.

Атаки на системы встраивания информации

Можно выделить две классификации атак на СВИ: по *целям*, которые они преследуют, и по *знаниям и возможностям* нарушителей, осуществляющих эти атаки.

В качестве основных целей атак на СВИ выделим следующие:

- обнаружение наличия встроенной информации,
- извлечение встроенной информации без отыскания ключа,
- удаление встроенной информации,
- отыскание секретного ключа,
- подмена встроенной информации,
- подделка носителя информации.

По знаниям и возможностям, которыми обладает нарушитель, можно выделить следующие атаки [11, 13]:

- только с известным носителем информации,
- с известным контейнером,
- с известной встроенной информацией,
- с выбранным контейнером,
- с выбранной встраиваемой информацией.

Последние два вида атак относятся к так называемой модели «активного нарушителя», а остальные рассмотренные атаки – к модели «пассивного нарушителя» [14, 13].

Наиболее сложным типом атаки и в то же время самым распространенным на практике ввиду минимальности требований для её осуществления является атака с известным носителем информации. Нарушитель при этом не обладает никакой априорной информацией о контейнере, ключе и встроенной информации.

Основные обозначения и определения

Одной из важных особенностей функционирования систем встраивания информации является преобразование информационной последовательности из одной формы в другую (с сохранением содержания). Это обусловливает необходимость введения обобщённого термина *внутрен-*

ней информации (внутренней она является по отношению к контейнеру, поскольку передаётся внутри него). Мы редко будем пользоваться этим термином, но важно понимать его суть. Существует три формы внутренней информации: двоичный вектор, цифровой сигнал и матрица признаков. Первая форма соответствует, например, сообщению, передаваемому внутри стеганографического контейнера, или цифровому коду защитного ЦВЗ. Вторая форма соответствует традиционной форме контейнера, в который встраивается информация, то есть это может быть цифровое аудио, изображение, видео и пр. Третья форма индивидуальна для каждой системы и является представлением, в котором непосредственно происходит встраивание информации, то есть модификация данных контейнера.

Под *цифровым сигналом* мы будем понимать величину $X \in \mathbb{X}_{[]}^{m}$, представляющую собой m-мерную матрицу, элементы которой определены на множестве $\mathbb{X} \subseteq \mathbb{R}$. Само множество $\mathbb{X}_{[]}^{m}$ будем называть пространством цифровых сигналов.

Под *матрицей признаков* $y \in \mathbb{Y}_{[]}^{l}$ будем понимать l-мерную матрицу, элементы которой определены на множестве $\mathbb{Y} \subseteq \mathbb{C}$. Само множество $\mathbb{Y}_{[]}^{l}$ мы будем называть пространством признаков.

Изначально внутренняя информация может быть представлена в виде вектора $\mathbf{b}$ двоичных значений длиной N_b (будем обозначать множество таких векторов $\mathbb{B}_{[N_b]}^{1}$) или цифрового сигнала $W \in \mathbb{X}_{[]}^{m}$. Контейнером является цифровой сигнал $C \in \mathbb{X}_{[]}^{m}$. Далее перед встраиванием отыскиваются матрицы признаков контейнера – $f \in \mathbb{Y}_{[]}^{l}$ и встраиваемой информации – $\Omega \in \mathbb{Y}_{[]}^{l}$.

Следует заметить, что в ряде систем встраивание информации осуществляется непосредственно в отсчёты цифрового сигнала. Такие методы применительно к цифровым изображениям называются *встраиванием в пространственной области*. В этом случае просто будем полагать отображение сигнала в матрицу признаков тождественным.

Результатом встраивания Ω в f является матрица признаков носителя информации $f^{W} \in \mathbb{Y}_{[]}^{l}$. Вслед за этим на основе f^{W} отыскивается собственно носитель информации в форме цифрового сигнала $C^{W} \in \mathbb{X}_{[]}^{m}$, который передаётся подсистеме извлечения информации. При передаче он может быть подвергнут атакам или искажениям, поэтому важно отли-

чать отправленный носитель информации от принятого, который обозначается как $\widetilde{C^W}$.

Результатом работы подсистемы извлечения информации является либо результат обнаружения встроенной информации:

$$\xi = \begin{cases} 1, & \text{если } \widetilde{C^W} \text{ содержит } \mathbf{b} \text{ (или } W\text{)}, \\ 0, & \text{если } \widetilde{C^W} \text{ не содержит } \mathbf{b} \text{ (или } W\text{)}, \end{cases}$$

(в случае системы с детектором), либо собственно извлечённая информация в начальной форме, то есть $\mathbf{b^R} \in \mathbb{B}^1_{[\mathrm{N_b}]}$ или $W^R \in \mathbb{X}^m_{[]}$ (где символ R является сокращением от “recovered”, то есть характеризует восстановленную информацию).

Величина ξ на практике рассчитывается по формуле вида

$$\xi = \begin{cases} 1, & \rho(x, x^R) \geq T_\rho, \\ 0, & \rho(x, x^R) < T_\rho, \end{cases} \tag{1.1}$$

где под символами x и x^R подразумевается встроенная и извлечённая информации в *форме детектирования* (то есть в одной из форм внутренней информации, определённой на уровне конкретной системы), $T_\rho \in \mathbb{R}$ – порог, а $\rho(x, x^R)$ – некоторая функция близости величин x и x^R, имеющая в зависимости от формы детектирования одну из трёх форм:

$$\rho : \mathbb{B}^1_{[N_b]} \times \mathbb{B}^1_{[N_b]} \longmapsto \mathbb{R}, \tag{1.2}$$

$$\rho : \mathbb{X}^m_{[]} \times \mathbb{X}^m_{[]} \longmapsto \mathbb{R}, \tag{1.3}$$

$$\rho : \mathbb{Y}^l_{[]} \times \mathbb{Y}^l_{[]} \longmapsto \mathbb{R}. \tag{1.4}$$

Аргументами функции ρ в первом случае являются $(\mathbf{b}, \mathbf{b^R})$, во втором – (W, W^R), в третьем – $\left(\Omega, \tilde{\Omega}\right)$, где $\tilde{\Omega}$ – оценённая матрица признаков внутренней информации. При извлечении информации всегда в первую очередь по принятому носителю информации отыскивается оценка $\tilde{\Omega}$, после чего осуществляется её конвертация в требуемую форму детектирования.

Если формой детектирования является двоичный вектор, то чаще всего используется функция вида

$$\rho(\mathbf{b}, \mathbf{b^R}) = \frac{1}{N_b} \sum_{i=0}^{N_b - 1} (1 - b_i \oplus b_i^R). \tag{1.5}$$

Для формы детектирования $\mathbb{X}^m_{[]}$ может использоваться либо аналогичная побитовая функция, либо какая-либо функция близости двух сигналов. В

случае полутоновых изображений, например, может использоваться мера PSNR [16], рассчитываемая по формуле

$$\rho(W, W^R) = PSNR(W, W^R) = 10lg\frac{255^2}{\varepsilon^2_{\text{кв}}(W, W^R)}, \quad (1.6)$$

где

$$\varepsilon^2_{\text{кв}}(W, W^R) = \frac{1}{N_1 N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\left(W(n_1, n_2) - W^R(n_1, n_2)\right)^2, \quad (1.7)$$

а N_1, N_2 – размеры изображения по вертикали и горизонтали соответственно.

Защищённость информации, передаваемой в СВИ внутри контейнера, обеспечивается *секретным ключом СВИ*, который мы зачастую для удобства будем объединять с открытыми (общеизвестными) параметрами СВИ в виде *составного ключа* $\mathbf{k}$.

В начале данного пособия приведена таблица обозначений всех основных данных, фигурирующих в СВИ.

2. Краткий обзор среды MATLAB и её основных инструментов

Рекомендуемой средой для выполнения практических и лабораторных заданий, приведённых в данном пособии, является MATLAB. MATLAB является сокращением от MATrix LABoratory, то есть представляет собой прежде всего средство удобной работы с матричными данными. MATLAB главным образом состоит из:

- ряда базовых модулей обработки данных (первоначально написанных главным образом на C/C++ и Java),
- интерпретируемого языка программирования, обеспечивающего удобную работу с матрицами и предоставляющего доступ к базовым модулям,
- огромного числа пакетов инструментов для специализированной обработки данных, которые чрезвычайно динамично развиваются и пополняются новыми функциями,
- удобной среды для написания и отладки собственных функций и скриптов.

MATLAB чрезвычайно удобен для выполнения заданий, изложенных в данном пособии, по следующим причинам:

- он содержит методы чтения/записи изображений, видео, звуковых файлов; функции сложной обработки цифровых сигналов и изображений; базовые методы шифрования и генерации псевдослучайных последовательностей; множество удобных функций, облегчающих работу с матрицами;
- программный код на языке MATLAB является очень компактным и относительно лёгким для написания и восприятия;
- среда MATLAB содержит ряд инструментов для анализа полученных результатов и выявления ошибок: многочисленные средства отладки, возможность детализированного просмотра всех матриц, их визуализация в виде графиков или изображений и пр.

Главное окно MATLAB содержит следующие элементы:

- Command Window – окно, функционирующее в режиме командной строки, в котором можно вызывать функции и скрипты;
- Current Folder – содержимое рабочей папки;

– Workspace – перечень текущих данных, сохранённых в памяти в результате работы ранее выполненных команд, с возможностью вызова средств визуализации этих данных;
– Command History – история команд.

Для написания новых или редактирования существующих функций и скриптов специализированной обработки данных (файлов кода на языке MATLAB, имеющих расширение '.m') может быть открыто окно Editor, содержащее инструменты проверки синтаксиса и отладки.

Рассмотрим основные особенности языка MATLAB. Прежде всего следует отметить, что в MATLAB существует довольно большое количество типов данных, однако нет возможности явного объявления переменных с типизацией, вроде следующего:

```
int a;
double b;
```

В MATLAB объявление переменных совмещено с инициализацией:

```
a = 5;
b = 5.1;
```

Используемым по умолчанию типом данных является double. Однако существует возможность явного преобразования типа данных следующим образом:

```
a = int32(a);
a = uint8(a);
b = float(b);
```

Язык MATLAB является чувствительным к регистру. Для первичного освоения основных языковых конструкций рекомендуется пошагово выполнить следующий скрипт, во многом заимствованный из [17]. Дополнительная информация об основах MATLAB может быть найдена, в частности, в [18, 19].

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (1) Основы и работа со скалярными величинами

% Символ "%" используется для создания однострочного комментария
A = 1
A = 1;  % Результат команды, завершающейся символом ";", не выводится в
консоль

a = pi      % Число pi
```

```
disp(a);        % Другой способ вывода
disp(sprintf('2 decimals: %0.2f', a))   % Вывод с форматированием
disp(sprintf('5 decimals: %0.5f', a))
format long % Включение длинного формата вывода чисел с плавающей точкой
a
format short     % Включение короткого формата вывода чисел с плавающей
точкой (используется по умолчанию)
a

3-2     % Разность
5*8     % Произведение
1/2     % Деление
2^6     % Возведение в степень
1 == 2      % Логическое выражение «равно»
1 ~= 2      % Логическое выражение «не равно»
1 || 0      % Логическое выражение «или»
xor(1, 0)   % Логическое выражение «исключающее или»
c = (3 >= 1)    % Создание логической переменной
b = 'bbb';  % Создание строковой переменной
a = pwd     % Текущая рабочая папка
clc     % Очистка командного окна
clear       % Очистка рабочего пространства (всех переменных)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (2) Объявление и индексирование матриц и векторов

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (A) Простое объявление матриц и векторов

A = [1 2; 3 4]; % Создание матрицы 2x2
N = 5       % Создание скалярной переменной (матрицы 1x1)
v = [1, 0, 0]   % Создание вектор-строки
v = [1; 2; 3]   % Создание вектор-столбца
v = v'      % Транспонирование
v = 1:.5:3  % Создание вектора подряд идущих значений от:шаг:до
v = []      % Пустой вектор
length(v)       % Длина вектора (количество элементов)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (B) Создание матриц заданного размера.
% Всегда первое число относится к строкам, второе – к столбцам

m = zeros(2, 3) % Создание матрицы нулей 2x3
m = zeros(3)    % Создание матрицы нулей 3x3
v = ones(1, 3)  % Создание матрицы единиц 1x3
m = eye(3)  % Создание единичной матрицы 3x3
v = rand(3, 1)  % Создание матрицы случайных значений 3x1 (равномерное
распределение [0, 1])
v = randn(3, 1) % Создание матрицы случайных значений 3x1 (нормальное
распределение с параметрами (0, 1))
w = -6 + sqrt(10)*randn(1, 10000);  % Нормально распределённый вектор
hist(w)     % Построение гистограммы
```

```
hist(w, 50);    % Построение гистограммы из 50 столбцов
help randn  % Вывод справки о функции в консоль
doc randn       % Показ html-страницы со справкой о функции

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (C) Индексирование. Индексы всегда начинаются с 1, не с нуля!

v = [1 2 3];
v(3)        % Доступ к элементу вектора

m = [1 2 3 4; 5 7 8 8; 9 10 11 12; 13 14 15 16]
m(1, 3)     % Доступ к элементу матрицы (строка, столбец)
m(2, :)     % Доступ ко всей строке матрицы (второй)
m(:, 1)     % Доступ ко всему столбцу матрицы
m(1, 1:3)       % Доступ к элементам с 1 до 3 строки 1
m(2:3, 2)       % Доступ к элементам с 2 до 3 столбца 2
m(2:end, 3) % Доступ к элементам с 2 до последнего в столбце 3
m(:)        % Разворачивание матрицы в вектор-столбец

m = [1 2 3; 4 5 6]
size(m)     % Размеры матрицы
size(m, 1)  % Число строк
size(m, 2)  % Число столбцов

m1 = zeros(size(m)) % Новая нулевая матрица того же размера, что и m
who     % Список всех текущих переменных
whos        % Детализированный список всех текущих переменных

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (3) Операции над матрицами и векторами

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (A) Поэлементные операции:

a = [1 2 3 4]';
2 * a       % Скалярное умножение
a / 4       % Скалярное деление
b = [5 6 7 8]';
a + b       % Сложение векторов
a .* b      % Поэлементное умножение (за счёт "."!)
a .^ 2      % Поэлементное возведение в квадрат
a ./ b      % Поэлементное деление
log(a)      % Поэлементный логарифм
round([1.5 2; 2.2 3.1]) % Поэлементное округление (также есть floor,
ceil и др. подобные функции

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (B) Векторные операции

a = [1 4 6 3]
sum(a)      % Сумма элементов вектора
mean(a)     % Среднее элементов вектора
```

```
var(a)      % Дисперсия элементов вектора
std(a)      % СКО элементов вектора
max(a)      % Максимальное значение
min(a)      % Минимальное значение
sort(a)     % Сортировка вектора по возрастанию

% Если a является матрицей, то эти операции будут работать над каждым её
столбцом, возвращая в качестве результата вектор-строку
a = [1 2 3; 4 5 6]
max(a)      % Максимум каждого столбца
max(max(a)) % Максимум всей матрицы
max(a(:))       % Максимум всей матрицы
mean(a, 2)  % Среднее строки 2

[1 2 3] * [4 5 6]'  % Произведение векторов с результатом 1x1
[1 2 3]' * [4 5 6]  % Произведение векторов с результатом 3x3

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (C) Матричные операции

a = rand(3,2)
b = rand(2,4)
c = a * b       % Произведение матриц

a = [1 3 2; 6 5 4; 7 8 9];
inv(a)      % Обратная матрица
pinv(a)     % Обратная матрица (более предпочтительный вариант)
eig(a)      % Вектор собственных чисел a
[V, D] = eig(a) % D – диагональная матрица собственных чисел, V – матри-
ца собственных векторов
% Другие матричные операции: det, norm, rank, ...

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (D) Изменение размеров и сборка матриц

a = [1 2; 3 4; 5 6];
b = a(:)        % Создание вектор-столбца из матрицы
a = reshape(b, 2, 3)    % Создание матрицы 2x3 из вектора (по столбцам)
a = [1 2]; b = [3 4];
c = [a, b]  % Горизонтальная конкатенация

a = [1; 2; 3];
c = [a; 4]  % Вертикальная конкатенация
a = [eye(3) rand(3)]    % Горизонтальная конкатенация матриц
b = repmat(5, 3, 2) % Создание матрицы 3x2 из пятёрок
b = repmat([1 2; 3 4], 1, 2) % Продублировать указанную матрицу 2x2 1
раз по строкам и 2 раза по столбцам
b = diag([1 2 3])   % Создание диагональной матрицы с указанными значе-
ниями элементов

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (4) Управляющие конструкции языка и векторизация циклов
```

```
% for VAR = EXPR    % EXPR – вектор, например 1:10, -1:0.5:1 или [1 4 7]
%     STATEMENT
%      ...
%     STATEMENT
% end
%
% while EXPRESSION  % EXPRESSION – логическое выражение, например a < b
%     STATEMENTS
% end
%
% if EXPRESSION
%     STATEMENTS
% elseif EXPRESSION % может отсутствовать
%     STATEMENTS
% else      % может отсутствовать
%     STATEMENTS
% end       %обязательный элемент
%
% Также могут использоваться break, continue, return

% !Циклов в MATLAB лучше избегать, поскольку они медленны.
% Гораздо лучше использовать «векторизованные циклы»,
% реализующиеся через операции над векторами и матрицами

% Примеры
for i=1:2:7
  i                          % Печать значения i
end

for i=[5 13 -1]
  if (i > 10)
    disp('Larger than 10')
  elseif i < 0
    disp('Negative value')
  else
    disp('Something else')
  end
end

% Вычитание вектора из каждой строки матрицы двумя способами
m = 50; n = 10; A = ones(m, n); v = 2 * rand(1, n);
% Реализация через циклы
tic     %Старт таймера
for i=1:m
  A(i,:) = A(i,:) - v;
end
toc     %Вывод времени таймера

% «Векторизованная реализация»
tic
A = ones(m, n) - repmat(v, m, 1);
```

```
toc

% Пример векторизации операций, содержащих условные конструкции
% Копирование положительных элементов из A в B
% Реализация с циклом и условием
B = zeros(m,n);
tic
for i=1:m
  for j=1:n
    if A(i,j)>0
      B(i,j) = A(i,j);
    end
  end
end
toc

% Реализация без циклов и условий
B = zeros(m,n);
tic
ind = find(A > 0);  % Индексы положительных элементов A
B(ind) = A(ind);
toc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(5) Сохранение результатов работы

save myfile % Сохранение всех переменных в файл myfile.mat
save myfile a b % Сохранение только a и b
clear a b       % Удаление a и b из памяти
clear       % Удаление всех переменных
load myfile % Загрузка переменных из файла myfile.mat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%(6) Creating scripts or functions using m-files:
%
% Объявление функции
% function [outarg_1, ..., outarg_m] = myfunction(inarg_1, ..., inarg_n)
% Имя функции должно совпадать с именем файла
% (то есть "myfunction" должна быть сохранена в файле "myfunction.m").
% Функции имеют собственное рабочее пространство для хранения данных.
% Поэтому нет риска конфликта имён переменных.

function y = mySquare(x)
y = x ^ 2;
mySquare(5)
addpath('something')
end

function [y1, y2] = mySquareAndCube(x)
y1 = x ^ 2;
y2 = x ^ 3;
[a, b] = mySquareCube(5)
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%(7) Построение графиков

x = [0 1 2 3 4];
plot(x);        % График значений x от их индекса
pause       % Ожидание нажатия клавиши
plot(x, 2*x);   % График 2*x от x
axis([0 8 0 8]);    % Задание ограничений по координатам

figure;     % Открытие нового окна с графиком/изображением
x = pi*[-24:24]/24;
plot(x, sin(x));
xlabel('radians');  % Подпись по оси x
ylabel('sin value');    % Подпись по оси y
title('title'); % Заголовок графика
print 'myPlot.png'  % Сохранение диаграммы в файл '.png'
figure;
subplot(1, 2, 1);    % Два графика горизонтально в одном окне; обращение
к первому
plot(x, sin(x));    % Первый график
subplot(1, 2, 2);   % Обращение ко второму графику
plot(x, 2*cos(x));  % Второй график

figure;
plot(x, sin(x));
hold on;         % Несколько кривых на одном графике
plot(x, 2*cos(x), '--');    % '--' символизирует другой тип линии

legend('sin', 'cos');   % Отображение легенды
hold off;

figure;
m = rand(64,64);
imagesc(m)  % Отображение матрицы как изображения
colormap gray;  % Выбор градаций серого для отображения
axis image; % Координаты пикселей в качестве значений по осям
axis off;       % Удаление осей

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%(8) Работа с полутоновыми изображениями

I = imread('lena.tif');   % Чтение изображений. На выходе тип данных
uint8

figure
imagesc(I)
colormap gray;
```

```
colorbar        % Отображение шкалы цветов
impixelinfo     % Интерактивное отображение значений пикселей
truesize        % Отображение в масштабе 1:1
truesize(2*size(I))     % Отображение в масштабе 2:1

I2 = imresize(I, 0.5, 'bil');    % Уменьшение размеров в 2 раза с билинейной интерполяцией
I3 = imrotate(I2, 45, 'bil', 'crop');    %Поворот на 45 градусов с обрезкой
I3 = double(I2);     % Преобразование в формат double для осуществления математических операций
imagesc(I3.^2)
imagesc(log(I3))
I3 = uint8(I3); % Возврат обратно в формат uint8 для записи результата в файл
imshow(I3); % Ещё один способ отображения результата
imwrite(I3, 'out.png')  % Сохранение результата

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

В заключение отметим ещё раз наиболее важные особенности программирования на языке MATLAB:

- для реализации какой-либо сложной процедуры обработки данных следует написать скрипт в виде m-файла, не рекомендуется работать напрямую в консоли;
- для повторяемых операций следует заводить отдельные пользовательские функции;
- все матрицы индексируются с 1;
- первый индекс – номер строки, второй – номер столбца;
- формат по умолчанию – double;
- полутоновые изображения, считанные из файла командой imread, имеют тип данных uint8 (8-битное беззнаковое целое);
- поэлементные операции умножения, деления, возведения в степень реализуются добавлением точки к арифметическому символу, например: a .* b, a .^ 2, a ./ b;
- циклы следует использовать только в том случае, если без них совсем не обойтись, необходимо стараться заменять циклы векторными операциями.

3. Базовые алгоритмы встраивания информации в пространственной области изображений

3.1. НЗБ-встраивание

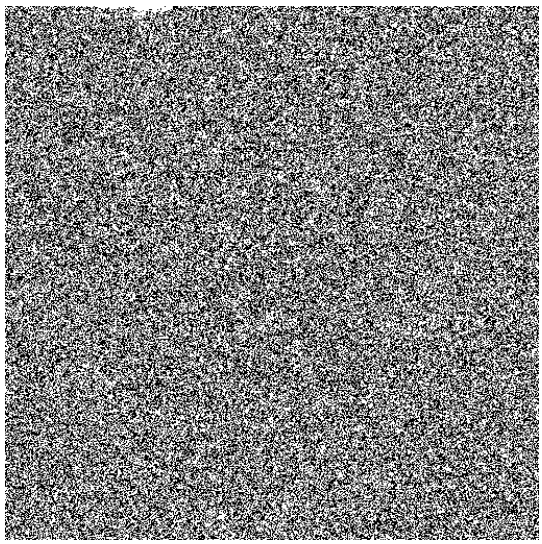
Встраивание информации в наименее значимые биты контейнера (или сокращённо НЗБ-встраивание) – исторически один из первых и, пожалуй, наиболее известный широкой публике подход, который может применяться как для стеганографии, так и для защиты сигналов цифровыми водяными знаками. Он очень прост и позволяет встроить достаточно большое количество информации без сколько-нибудь заметных искажений контейнера, однако методы, использующие данный подход, как правило, обладают низкой стойкостью к искажениям носителя информации и относительно легко могут быть подвергнуты стегоанализу, поэтому имеют весьма ограниченную применимость. Тем не менее, НЗБ-встраивание вполне подходит для задач, в которых отсутствуют жёсткие требования по стойкости к отдельным видам атак.

Основная идея метода заключается в том, что любое полутоновое изображение может быть представлено в виде совокупности битовых плоскостей. Так, контейнер $C(n_1, n_2)$ будет иметь вид:

$$C(n_1, n_2) = C_1(n_1, n_2) + C_2(n_1, n_2) \cdot 2 + \ldots + C_K(n_1, n_2) \cdot 2^{K-1}, \quad (3.1)$$

где $C_k(n_1, n_2) \in [0,1]$ – битовые плоскости, k – номер битовой плоскости, $K = 8$ – их количество.

Наименее и наиболее значащими битовыми плоскостями являются соответственно C_1 и C_8: если изменить значение бита $C_1(n_1, n_2)$, то яркость изменится на единицу; если же изменить значение бита $C_8(n_1, n_2)$, то яркость изменится на 128. Различие между младшими и старшими битовыми плоскостями хорошо заметно на рис. 3.1. Младшие битовые плоскости выглядят как слабокоррелированный шум. Осмысленные детали, как правило, начинают проступать лишь с четвёртой битовой плоскости. Это означает, что наименее значимые битовые плоскости можно модифицировать с целью встраивания скрытого сообщения или цифрового водяного знака.

а)   б)

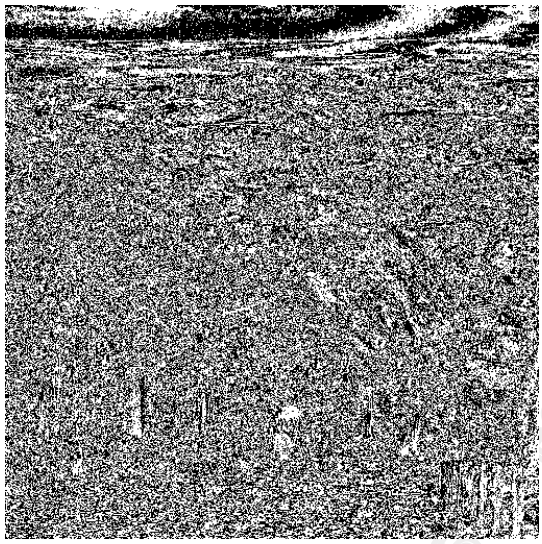
 
в) г)

Рис. 3.1 – Битовые плоскости полутонового изображения: а) исходное изображение; б) 1-я битовая плоскость; в) 4-я битовая плоскость; г) 8-я битовая плоскость

Далее будем рассматривать лишь случай встраивания информации в одну p-ю битовую плоскость. Тогда носитель информации будет иметь вид:

$$C^W(n_1, n_2) = C_1^W(n_1, n_2) + \ldots + C_K^W(n_1, n_2) \cdot 2^{K-1}, \quad (3.2)$$

где $C_k^W(n_1, n_2) = C_k(n_1, n_2)$ для всех $k \neq p$.

Существует достаточно большое количество систем НЗБ-встраивания, которые отличаются способом формирования битовой плоскости $C_p^W(n_1, n_2)$. Ниже мы рассмотрим три таких системы: НЗБ-встраивание ЦВЗ, стеганографическое НЗБ-встраивание и ±-встраивание в полутоновые изображения.

СВИ-1 (НЗБ-встраивание ЦВЗ)

Встраивание цифровых водяных знаков в наименее значимые биты контейнера

Пусть в НЗБ контейнера необходимо встроить изображение (цифровой водяной знак) W того же размера, содержащее бинарные значения. Тогда могут использоваться следующие варианты модификации C_p^W:

1. Непосредственная замена битовой плоскости контейнера битами скрываемой информации:
$$C_p^W(n_1, n_2) = W(n_1, n_2). \quad (3.3)$$

Извлечение информации в этом случае осуществляется, очевидно, путём чтения соответствующей битовой плоскости изображения со встроенной информацией.

2. Побитовое сложение битовой плоскости контейнера с битами скрываемой информации:
$$C_p^W(n_1, n_2) = C_p(n_1, n_2) \oplus W(n_1, n_2). \quad (3.4)$$

Извлечение информации в этом случае происходит путём побитового сложения C_p^W и C_p.

3. Отрицание побитового сложения битовой плоскости контейнера с битами скрываемой информации:
$$C_p^W(n_1, n_2) = \overline{C_p(n_1, n_2) \oplus W(n_1, n_2)}. \quad (3.5)$$

Извлечение информации происходит путём применения той же операции для плоскостей C_p^W и C_p.

■

На рис. 3.2 представлен пример изображения, во вторую битовую плоскость которого по формулам (3.2), (3.3) встроен бинарный орнамент.

Рис. 3.2 – Пример встраивания изображения во вторую битовую плоскость: слева – заполненный контейнер, справа – встроенное изображение

СВИ-2 (стеганографическое НЗБ-встраивание)

Скрытая передача информации в наименее значимых битах контейнера

При стеганографическом встраивании внутри контейнера $C(n_1, n_2)$ передаётся бинарный вектор $\mathbf{b}$ длины $N_b \leq N_1 N_2$. Как правило, встраивание происходит путём замены бит. В простейшем случае информация заносится в НЗБ последовательно:

$$C_p^W(n_1, n_2) = b_{n_1 \cdot N_2 + n_2}, \tag{3.6}$$

где b_i – *i*-й элемент вектора $\mathbf{b}$. Однако такое встраивание легко поддаётся стегоанализу, то есть легко обнаруживается на основе анализа статистических характеристик наименее значимой битовой плоскости (НЗБП), использованной для встраивания информации. Методы стегоанализа НЗБ-встраивания будут рассмотрены в главе 8.

Для противодействия простейшим методам стегоанализа прибегают к следующим мерам:

1) заполняют по возможности небольшую часть НЗБ контейнера, т.е. добиваются того, чтобы величина

$$q = \frac{N_b}{N_1 N_2}, \tag{3.7}$$

называемая заполненностью контейнера, была существенно меньше 1;

2) заполнение контейнера производят в псевдослучайном порядке, который полностью определяется ключом встраивания $\mathbf{k}$.

Ко второму пункту следует добавить, что ключ сам по себе не содержит последовательности координат пикселей, но однозначно определяет её. Например, ключ может представлять собой начальное значение генератора случайных чисел.

Процедура извлечения информации очевидна и представляет собой чтение битов из заданных ключом координат.

■

При встраивании информации в p-ю битовую плоскость яркость отдельно взятого пикселя либо не меняется, либо меняется ровно на p, причём известно, в какую сторону. Пусть для определённости $p = 2$ и стоит задача встроить в пиксель с яркостью 21 значение 1. Число 21 в двоичной записи имеет вид 10101, то есть во второй битовой плоскости стоит 0. Таким образом, встраивая туда 1, мы прибавляем к текущему значению p и в итоге получаем 23. Однако что произойдёт, если мы не прибавим p, а вычтем? Очевидно, что в этом случае разница между искомым и полученным значением составит $2p$, то есть изменения произойдут в более старших разрядах двоичной записи, в то время как p-й бит не претерпит изменений. Действительно, в нашем примере получится число 19, то есть 10011 в двоичной записи. Таким образом, мы имеем два способа изменения значения яркости пикселя, приводящих к идентичным изменениям в нужной битовой плоскости и сопровождаемых равными по абсолютной величине искажениями. Это свойство позволяет несколько модифицировать процедуру стеганографического НЗБ-встраивания путём внесения дополнительной неопределённости, способствующей защите от атак, направленных на обнаружение канала скрытой передачи информации.

СВИ-3 (±1-встраивание)

Скрытая передача информации за счёт изменения отсчётов контейнера на ±1 [6]

Рассуждения выше приводились для общего случая p-й битовой плоскости. Однако на практике чаще всего ограничиваются рассмотрением случая $p = 1$. Более того, само название данной модификации НЗБ-метода, укоренившееся в научной литературе – ±1-встраивание – уже косвенно говорит о номере битовой плоскости (в общем случае следова-

ло бы говорить о ±-встраивании). Мы приведём формулу встраивания для этого частного случая, однако её обобщение не составит труда.

Итак, пусть b_i – i-й элемент вектора $\mathbf{b}$,

$$(n_1, n_2) = \big(n_1(\mathbf{k}, i), n_2(\mathbf{k}, i)\big)$$

– координаты i-го пикселя, в который необходимо встроить бит b_i, а ξ_i – псевдослучайное число, с равной вероятностью принимающее положительные и отрицательные значения (генерация последовательности $\{\xi_i\}$ также происходит на основе ключа). Тогда встраивание информации осуществляется по формуле

$$C^W(n_1, n_2) = \begin{cases} C(n_1, n_2), & C_1(n_1, n_2) = b_i, \\ C(n_1, n_2) + 1, & (C_1(n_1, n_2) \neq b_i) \wedge \big((\xi_i \geq 0) \wedge \\ & \wedge (C(n_1, n_2) < 255) \vee (C(n_1, n_2) = 0)\big), \\ C(n_1, n_2) - 1, & (C_1(n_1, n_2) \neq b_i) \wedge \big((\xi_i < 0) \wedge \\ & \wedge (C(n_1, n_2) > 0) \vee (C(n_1, n_2) = 255)\big), \end{cases} \quad (3.8)$$

то есть случайным образом прибавляется или вычитается единица в том случае, если значение бита не совпадает с требуемым.

Извлечение информации происходит так же, как и в СВИ-2.

■

3.2. Встраивание информации за счёт управляемого переквантования яркости

Рассмотрим ещё один метод, широко используемый для встраивания информации (преимущественно ЦВЗ) в изображения. Изначально предложенный в работе [20], он широко известен под аббревиатурой QIM (Quantization Index Modulation). В русскоязычной литературе он чаще всего именуется методом управляемого переквантования яркости. Общий принцип управляемого переквантования заключается в том, что функция встраивания конкретного значения $\mathcal{E}$ представляется в виде семейства функций-квантователей Q_m, где m – индекс функции, причём

$$\forall m \;\; Q_m(x) \approx x. \quad (3.9)$$

Это условие обеспечивает слабое отклонение значений яркости носителя информации от соответствующих значений контейнера. При встраивании информации индекс используемой функции определяется значением отсчёта ЦВЗ:

$$C^W(n_1, n_2) = \mathcal{E}\big(C(n_1, n_2), W(n_1, n_2)\big) = Q_{W(n_1, n_2)}\big(C(n_1, n_2)\big). \quad (3.10)$$

где $W(n_1, n_2) \in \mathbb{N}_0$.

Достоинством этого метода является его теоретически обоснованная стойкость к аддитивному гауссовскому шуму вплоть до уровня дисперсии, определяемого параметрами метода. Это, в частности, делает его удобным инструментом для проектирования систем хрупких и полухрупких ЦВЗ [21]. Подробную информацию о методе QIM и его модификациях можно найти в книге [2], мы остановимся только на частном случае QIM для встраивания двоичной информации.

СВИ-4 (QIM)

Встраивание ЦВЗ за счёт управляемого переквантования [20]

Пусть C – полутоновой контейнер, а W – бинарный ЦВЗ. Тогда встраивание информации в каждом пикселе (n_1, n_2) осуществляется по формуле:

$$C^W(n_1, n_2) = \left[\frac{C(n_1, n_2)}{2q}\right] \cdot 2q + W(n_1, n_2) \cdot q + \vartheta(n_1, n_2), \quad (3.11)$$

где $q > 0$ – параметр алгоритма, $[x]$ означает целую часть рационального числа x, а $\vartheta(n_1, n_2)$ может рассчитываться одним из следующих способов:

$$\vartheta(n_1, n_2) = 0, \quad (3.12)$$

$$\vartheta(n_1, n_2) = \xi(n_1, n_2), \quad (3.13)$$

где $\xi(n_1, n_2)$ – реализация равномерного белого шума с диапазоном значений от 0 до $q - 1$;

$$\vartheta(n_1, n_2) = C(n_1, n_2)\ (mod\ \ q), \quad (3.14)$$

где $x(\mathrm{mod}\ y)$ – остаток от деления x на y.

Очевидно, что первый способ приводит к наибольшим визуальным искажениям ввиду сокращения множества возможных значений яркости, второй и третий способ имеют целью снизить заметность искажений, возникающих при встраивании.

Формулу извлечения информации предлагается вывести самостоятельно. При этом следует заметить, что корректное извлечение информации в данном методе может осуществляться и без использования исходного контейнера.

■

Упражнения

Реализация и исследование систем стеганографического НЗБ-встраивания

Результатами работы будут являться скрипт `steglsb_run`, а также функции:

`[CW] = lsb_embed(C, b, seed)`

– стеганографическое НЗБ-встраивание в первую битовую плоскость двоичного вектора `b` в контейнер `C`. При `seed<0` порядок записи последовательный, при `seed>=0` это значение определяет случайный порядок записи.

`[bR] = lsb_extract(CW, Nb, seed)`

– извлечение двоичного вектора длины `Nb`, встроенного в контейнер `CW`.

`[CW] = plusminus_embed(C, b, randomOrder)`

– ±1-встраивание.

1. Реализовать функцию `lsb_embed` для случая `seed<0` и написать фрагмент скрипта `steglsb_run`, вызывающего данную функцию.
2. Реализовать функцию `lsb_extract`, выполнить извлечение информации и побитовое сравнение встроенной строки с извлечённой.
3. Реализовать функцию `lsb_embed` для случая `seed>=0` и проверить её работоспособность.
4. Сгенерировать строку, длина которой составляет 20 % от объёма битовой плоскости контейнера. Встроить её последовательно и в случайном порядке.
5. Визуализировать первые битовые плоскости изображений, полученных в предыдущем задании. Сравнить их визуально.
6. Реализовать функцию `plusminus_embed` через вызов функции `lsb_embed`, проверить её работу.

Реализация и исследование СВИ-4 (QIM)

Результатами работы будут являться скрипт `qim_run`, а также функция:

`[CW] = qim_embed(C, W, q, theta_type)`

– встраивание информации методом QIM. `theta_type` может быть равно 1, 2 или 3 и определяет формулу расчёта $\vartheta(n_1, n_2)$: (3.12), (3.13) или (3.14).

1. Реализовать функцию `qim_embed` для случая `theta_type=1` и написать фрагмент скрипта `qim_run`, вызывающего данную функцию.
2. Дополнить функцию `qim_embed` случаем `theta_type=2` и `theta_type=3`.
3. Сгенерировать псевдослучайное поле с приблизительно равным количеством нулей и единиц и встроить его тремя методами. После этого визуализировать и сравнить гистограммы полученных изображений.
4. Повторить предыдущее задание, встроив матрицу нулей в качестве встраиваемой информации.

Лабораторная работа 1: Простейшие методы встраивания информации в полутоновые изображения

Задания

В рамках выполнения лабораторной работы необходимо выполнить задания из списка основных по вариантам, отмеченным в таблице ниже, а также ответить на один контрольный вопрос. Вопросы выбирает преподаватель из списка основных вопросов. По желанию студент может ответить вместо основного на дополнительный вопрос, выбрав его самостоятельно. Также студент по желанию может выполнить дополнительное задание после основных. И то и другое будет отмечено преподавателем.

Основные задания

1. Реализовать встраивание ЦВЗ в одну из наименее значимых битовых плоскостей контейнера одним из рассмотренных способов встраивания (СВИ-1). Номер модифицируемой битовой плоскости и способ модификации определяются вариантом.
2. Реализовать извлечение информации, встроенной в пункте 1.
3. Реализовать встраивание информации при помощи СВИ-4 (QIM). Параметры системы определяются вариантом задания.
4. Реализовать извлечение информации, встроенной в пункте 3.

Дополнительные задания

1. На основе выполненных заданий 1-2 из основного списка реализовать стеганографическое встраивание в НЗБ полутонового контейнера текстовой информации с последующим её извлечением (СВИ-2). Способ преобразования текста в бинарный вектор не принципиален и оставляется на усмотрение студента.

Таблица вариантов заданий

№	Номер НЗБП в СВИ-1 (p)	Способ встраивания в СВИ-1	Значение q в СВИ-4	Способ встраивания в СВИ-4
1	1	(3.3)	5	(3.12)
2	1	(3.3)	8	(3.14)
3	1	(3.4)	10	(3.12)
4	2	(3.3)	5	(3.14)
5	2	(3.3)	8	(3.12)
6	2	(3.4)	10	(3.14)
7	1 и 2	(3.4)	5	(3.13)

№	Номер НЗБП в СВИ-1 (p)	Способ встраивания в СВИ-1	Значение q в СВИ-4	Способ встраивания в СВИ-4
8	1 и 2	(3.5)	8	(3.14)
9	1 и 2	(3.4)	10	(3.13)
10	3	(3.5)	5	(3.14)
11	3	(3.4)	8	(3.13)
12	3	(3.5)	10	(3.14)

Контрольные вопросы

Основные вопросы

1. Какая битовая плоскость изображения является более значимой: четвёртая или шестая? Почему?
2. Напишите и поясните формулу (3.2). С какими коэффициентами в неё входит пятая и седьмая битовые плоскости?
3. Пусть дано фотореалистичное полутоновое изображение, аналогичное представленному на рис. 3.1а. Существенно ли изменится визуально это изображение, если его вторую битовую плоскость заменить седьмой? Если его седьмую битовую плоскость заменить второй?
4. Сравните два метода модификации наименее значимых битов контейнера: побитовое сложение (3.4) и непосредственная замена (3.3) (с точки зрения сложностей при извлечении информации легальным получателем изображения и с точки зрения защищённости от прочтения информации при перехвате нарушителем).
5. Сравните два метода модификации наименее значимых битов контейнера: побитовое сложение (3.4) и обратное ему (3.5) (с точки зрения сложностей при извлечении информации легальным получателем изображения и с точки зрения защищённости от прочтения информации при перехвате нарушителем).
6. Выведите формулу извлечения информации, встроенной при помощи системы QIM, с использованием исходного контейнера.
7. Напишите формулу взаимосвязи параметра q в системе QIM и номера модифицируемой битовой плоскости при НЗБ-встраивании. При каких условиях СВИ-4 эквивалентна СВИ-1?
8. Что приведёт к более существенным искажениям по абсолютной величине ошибки: встраивание при помощи QIM при $q = 12$ или

встраивание методом побитового сложения в четвёртую битовую плоскость?

9. В каких пределах может изменяться параметр q в СВИ-4, чтобы формула встраивания (3.11) оставалась корректной?

Дополнительные вопросы

1. Напишите и поясните формулу извлечения информации, встроенной методом деления с остатком, не использующую исходный контейнер.
2. Придумайте простой способ модификации метода встраивания информации в НЗБ контейнера, в котором при встраивании и извлечении информации используется некий секретный ключ. Что собой представляет этот ключ?

4. Встраивание информации в бинарные изображения

Бинарными называются одноканальные изображения, в которых используется ровно 1 бит для хранения интенсивности каждого пикселя. Иными словами, каждый пиксель может быть только чёрным (значение 0) или белым (значение 1). Несмотря на кажущуюся непрактичность подобных изображений, они представляют собой важный случай, поскольку при печати происходит преобразование полутоновых изображений в бинарные с последующей передачей последних на принтер. Таким образом, возможными способами встраивания информации, стойкой к процедуре печати изображения, являются встраивание информации в бинарные изображения или встраивание информации на этапе преобразования изображения в бинарное.

Очевидно, что рассмотренные ранее методы встраивания информации в полутоновые изображения оказываются неприменимыми для бинарных изображений, поскольку последние содержат лишь одну битовую плоскость. Следовательно, для бинарных изображений требуются специфические методы встраивания информации.

Прежде всего, охарактеризуем вкратце, что собой представляют бинарные изображения, полученные из полутоновых. Принцип их формирования использует особенность человеческого зрения, которое усредняет яркость наблюдаемых фрагментов небольшого размера. Пример на рис. 4.1 показывает, что значения пикселей бинарного изображения формируются таким образом, чтобы их среднее в окрестности каждого пикселя было как можно ближе к яркости полутонового оригинала в этой точке. Процесс формирования таких изображений называется *цифровым растрированием*. Существует довольно большое число методов растрирования: амплитудная и частотная модуляция, диффузия точек, диффузия ошибки (будет рассмотрен ниже), различные методы оптимизации. Подробный обзор этих методов можно найти в книге [22].

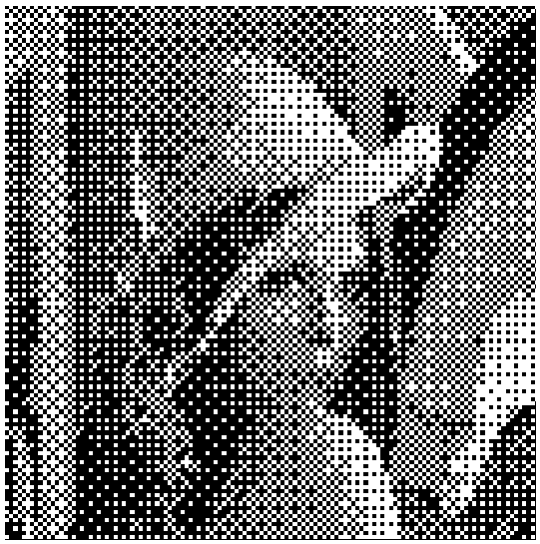
а)   б)

Рис. 4.1 – Изображение Lenna (а) и соответствующее ему бинарное изображение (б), полученное путём растрирования методом частотной модуляции

4.1. Непосредственное встраивание информации в бинарные изображения

СВИ-5 (DHST)

Data Hiding Self-Toggling (DHST) – простое стеганографическое встраивание в бинарный контейнер [23]

Пусть контейнер $C(n_1, n_2)$ размерами $N_1 \times N_2$ представляет собой бинарное изображение, внутри которого необходимо передать бинарный вектор $\mathbf{b}$ длины $N_b < N_1 N_2$. Ключ $\mathbf{k}$ системы представляет собой последовательность координат пикселей изображения длиной $N_k \geq N_b$.

При встраивании информации изначально носитель информации $C^W(n_1, n_2)$ идентичен контейнеру. Затем каждый *i*-й бит вектора $\mathbf{b}$ встраивается по простой формуле

$$C^W\big(n_1(\mathbf{k}, i), n_2(\mathbf{k}, i)\big) = b_i, \tag{4.1}$$

где b_i – элементы вектора $\mathbf{b}$, а $n_1(\mathbf{k}, i), n_2(\mathbf{k}, i)$ – координаты i-го пикселя, определяемые на основе ключа.

Процедура извлечения информации очевидна. Следует отметить, что при большой длине встраиваемого вектора искажения, являющиеся результатом встраивания информации, становятся весьма существенными.

■

СВИ-6 (DHSPT)

Data Hiding by Smart Pair-Toggling (DHSPT) – стеганографическое встраивание в бинарный контейнер с компенсацией искажений [23]

Данная система является модификацией системы СВИ-5 (DHST), в которой искажения, внесённые встраиванием по формуле (4.1), компенсируются путём замены значения одного из соседних пикселей на противоположное. В результате средняя яркость в локальной окрестности изменённого пикселя остаётся неизменной, следовательно, визуальное качество носителя информации повышается.

Пусть рассматривается окрестность изменённого пикселя (n_1, n_2) размерами 3×3 или 5×5. Если в этой окрестности отсутствуют пиксели, имеющие то же значение, что и $C^W(n_1, n_2)$, то компенсации искажений не производится. В противном случае необходимо выбрать один пиксель (m_1, m_2) такой, что $C(m_1, m_2) = C^W(n_1, n_2)$, и инвертировать его.

Если таких пикселей несколько, то в простейшем случае выбирается произвольный. Однако авторы системы предложили и более разумный подход. Для каждого из допустимых пикселей рассчитывается его вес, и инвертированию подвергается пиксель с наибольшим весом.

Пусть окно имеет размер 3×3. Пронумеруем пиксели окрестности в построчном порядке: $x_1, x_2, \ldots x_9$, причем x_5 – центральный пиксель с координатами (n_1, n_2). Тогда вес пикселя $V(m_1, m_2)$ рассчитывается следующим образом:

$$V(m_1, m_2) = \sum_{i=1}^{9} w(i) f(x_5, x_i), \tag{4.2}$$

где

$$f(x, y) = \begin{cases} 1 & x \neq y, \\ 0 & x = y; \end{cases} \tag{4.3}$$

$$w(i) = \begin{cases} 1, & i = 1,3,7,9; \\ 2, & i = 2,4,6,8; \\ 0, & i = 5. \end{cases} \tag{4.4}$$

Веса $w(i)$ в формуле (4.4) соответствуют следующей таблице размерами 3×3:

1	2	1
2	0	2
1	2	1

Наибольший вес пикселя (m_1, m_2) означает, что почти все пиксели его окрестности имеют то же значение, что и $C^W(n_1, n_2)$. Поэтому если инвертированию подвергается пиксель с наибольшим весом, то это влечёт наименьшие визуальные искажения.

В базовом алгоритме DHSPT не описан вид $w(i)$ для случая окрестности 3×3. Однако допустимо, чтобы коэффициенты в этой матрице были обратно пропорциональны расстоянию от центрального отсчёта. В этом случае таблица имеет следующий вид:

$\frac{\sqrt{2}}{4}$	$\frac{1}{\sqrt{5}}$	2	$\frac{1}{\sqrt{5}}$	$\frac{\sqrt{2}}{4}$
$\frac{1}{\sqrt{5}}$	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$	$\frac{1}{\sqrt{5}}$
2	1	0	1	2
$\frac{1}{\sqrt{5}}$	$\frac{\sqrt{2}}{2}$	1	$\frac{\sqrt{2}}{2}$	$\frac{1}{\sqrt{5}}$
$\frac{\sqrt{2}}{4}$	$\frac{1}{\sqrt{5}}$	2	$\frac{1}{\sqrt{5}}$	$\frac{\sqrt{2}}{4}$

■

В алгоритме DHSPT изменяются не только отсчёты, заданные ключом, но и некоторые отсчёты из их окрестности. Поэтому может сложиться ситуация, при которой значение некоторых пикселей изменится дважды. Это, в свою очередь, может привести к неточному извлечению встроенной информации. Поэтому для алгоритма DHSPT ключ должен генерироваться таким образом, чтобы исключить возможность попадания одного пикселя ключа в окрестность другого пикселя ключа.

Перечислим практические способы генерации ключа для систем DHST и DHSPT:

1) простая генерация координат пикселя по вертикали и горизонтали (только для DHST):

$$\left(n_1^i, n_2^i\right): n_1^i = \overline{0..N_1 - 1}, n_2^i = \overline{0..N_2 - 1}, i = \overline{0..N_k - 1}; \quad (4.5)$$

2) генерация координат пикселя на втрое меньшей сетке:

$$(3n_1^k, 3n_2^k): n_1^k = \overline{0..\left[\frac{N_1}{3}\right]-1}, n_2^k = \overline{0..\left[\frac{N_2}{3}\right]-1}, k = \overline{0..N_k-1}; \quad (4.6)$$

3) генерация пар чисел на полной сетке с проверкой попадания в окрестность 3×3:

$$(n_1^k, n_2^k): n_1^k = \overline{0..N_1-1}, n_2^k = \overline{0..N_2-1}, k = \overline{0..N_k-1},$$
$$\min_{k\neq m}(n_1^k - n_1^m) > 1, \min_{k\neq m}(n_2^k - n_2^m) > 1, k, m = \overline{0..N_k-1}; \quad (4.7)$$

4) генерация координат пикселя на впятеро меньшей сетке:

$$(5n_1^k, 5n_2^k): n_1^k = \overline{0..\left[\frac{N_1}{5}\right]-1}, n_2^k = \overline{0..\left[\frac{N_2}{5}\right]-1}, k = \overline{0..N_k-1}; \quad (4.8)$$

5) генерация пар чисел на полной сетке с проверкой попадания в окрестность 5×5:

$$(n_1^k, n_2^k): n_1^k = \overline{0..N_1-1}, n_2^k = \overline{0..N_2-1}, k = \overline{0..N_k-1},$$
$$\min_{k\neq m}(n_1^k - n_1^m) > 2, \min_{k\neq m}(n_2^k - n_2^m) > 2, k, m = \overline{0..N_k-1}. \quad (4.9)$$

Очевидно, безошибочное извлечение информации при использовании для генерации ключа процедуры (4.5) возможно только для алгоритма DHST. Для безошибочного извлечения информации алгоритмом DHSPT с компенсацией пикселя в окне 3x3 целесообразно использовать один из способов (4.6) или (4.7), а в случае окна 5x5 – (4.8) или (4.9).

4.2. Встраивание информации при растрировании изображений

Вторым подходом к встраиванию информации в бинарные изображения (помимо модификации отсчётов подготовленного ранее бинарного контейнера) является внесение дополнительной информации в контейнер на этапе растрирования полутонового изображения. В настоящей главе мы изучим одну систему, реализующую данный подход, однако поскольку она интегрирована с конкретным методом растрирования – диффузией ошибки, то прежде всего необходимо его подробно описать.

Ядро диффузии ошибки

Пусть C – полутоновое изображение размером $N_1 \times N_2$, яркость пикселей которого $C(n_1, n_2)$ принимает целые значения на отрезке $[0{,}255]$. Из него необходимо получить бинарное изображение C^B того же размера.

В алгоритме диффузии ошибки (Error Diffusion) используется матрица h размерами $M_1 \times M_2$, называемая *ядром* или *весовой функцией*. Как правило, размеры ядра невелики: $1 \leq M_1, M_2 \leq 5$. Ядро задаёт на-

правления распространения (диффузии) ошибки растрирования и определяет доли ошибки, передаваемые в каждом из направлений.

Приведём примеры весовых функций, зарекомендовавших себя на практике:

– ядро размерами 2×2:

$$\frac{1}{4}\begin{pmatrix}\odot & 2\\ 1 & 1\end{pmatrix}; \tag{4.10}$$

– ядро из трёх ненулевых элементов:

$$\frac{1}{16}\begin{pmatrix}0 & \odot & 8\\ 2 & 6 & 0\end{pmatrix}; \tag{4.11}$$

– ядро Floyd & Steinberg [24]

$$\frac{1}{16}\begin{pmatrix}0 & \odot & 7\\ 3 & 5 & 1\end{pmatrix}; \tag{4.12}$$

– ядро Fan [25]

$$\frac{1}{16}\begin{pmatrix}0 & 0 & \odot & 7\\ 1 & 3 & 5 & 0\end{pmatrix}; \tag{4.13}$$

– ядро Jarvis et al. [26]

$$\frac{1}{48}\begin{pmatrix}0 & 0 & \odot & 7 & 5\\ 3 & 5 & 7 & 5 & 3\\ 1 & 3 & 5 & 3 & 1\end{pmatrix}; \tag{4.14}$$

– ядро Stucki [27]

$$\frac{1}{42}\begin{pmatrix}0 & 0 & \odot & 8 & 4\\ 2 & 4 & 8 & 4 & 2\\ 1 & 2 & 4 & 2 & 1\end{pmatrix}. \tag{4.15}$$

Символом $\odot$ отмечен пиксель с координатами $(0,0)$. Значение $h(0,0) = 0$.

При практической реализации метода диффузии ошибки может применяться один из двух алгоритмов, которые в конечном счёте приводят к идентичным результатам. Первый из этих алгоритмов реализует подтягивание ошибок растрирования из уже пройденных отсчётов и носит название *pull-модели*. Второй алгоритм, называемый *push-моделью,* осуществляет распространение ошибки из текущего отсчёта в последующие. Рассмотрим подробно оба алгоритма.

Алгоритм диффузии ошибки (pull-модель)

Обозначим за D_h множество точек (n_1, n_2), в которых $h(n_1, n_2) \neq 0$. Тогда pull-модель алгоритма диффузии ошибки может быть записана в виде следующего набора выражений:

$$u(n_1, n_2) = C(n_1, n_2) - \sum_{(m_1, m_2) \in D_h} h\,(m_1, m_2) e(n_1 - m_1, n_2 - m_2), \tag{4.16}$$

$$C^B(n_1, n_2) = \begin{cases} 1, & u(n_1, n_2) \geq T, \\ 0, & u(n_1, n_2) < T, \end{cases} \tag{4.17}$$

$$e(n_1, n_2) = 255 \cdot C^B(n_1, n_2) - u(n_1, n_2). \tag{4.18}$$

В формулах (4.16)–(4.18) $u(n_1, n_2)$ и $e(n_1, n_2)$ – это вспомогательные матрицы размерами $N_1 \times N_2$. Первая характеризует корректируемый в зависимости от ошибки растрирования контейнер, вторая – ошибку в очередной точке. T – пороговое значение, как правило, равное 128. Данный алгоритм схематически проиллюстрирован на рис. 4.2.

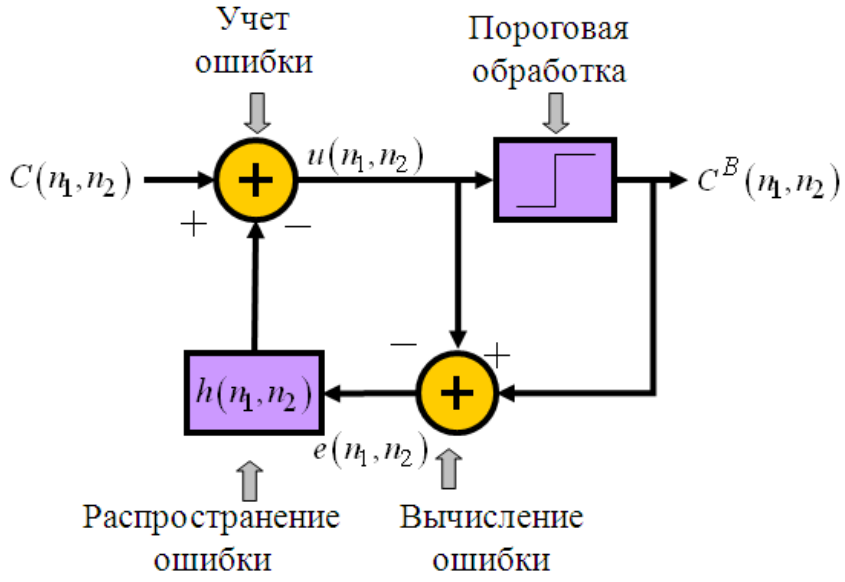


Рис. 4.2 – Схема алгоритма диффузии ошибки (pull-модель)

Алгоритм диффузии ошибки (push-модель)

Пусть D_h, как и ранее, характеризует точки (n_1, n_2), в которых $h(n_1, n_2) \neq 0$. Тогда push-модель алгоритма диффузии ошибки определяется следующими выражениями:

$$u(n_1, n_2) = C(n_1, n_2)\ \forall (n_1, n_2): n_1 = \overline{0..N_1 - 1}, n_2 = \overline{0..N_2 - 1} \tag{4.19}$$

$$C^B(n_1, n_2) = \begin{cases} 1, & u(n_1, n_2) \geq T, \\ 0, & u(n_1, n_2) < T, \end{cases} \tag{4.20}$$

$$e = 255 \cdot C^B(n_1, n_2) - u(n_1, n_2), \tag{4.21}$$

$$\forall (m_1, m_2) \in D_h \ \rightarrow \ u(n_1 + m_1, n_2 + m_2) = u(n_1 + m_1, n_2 + m_2) - e \cdot h(m_1, m_2) \tag{4.22}$$

Как и в pull-модели, $u(n_1, n_2)$ – вспомогательное изображение размерами $N_1 \times N_2$. Поле ошибок уже хранить не обязательно, поскольку на каждом шаге алгоритма ошибка сразу рассеивается по изображению u.

На рис. 4.3 изображён пример работы алгоритма диффузии ошибки. Как видно, качество результирующего изображения является весьма высоким.

Рис. 4.3 – Пример работы алгоритма диффузии ошибки для растрирования полутонового изображения

Встраивание информации при растрировании методом диффузии ошибки

Теперь перейдём собственно к рассмотрению системы встраивания ЦВЗ, использующей алгоритм диффузии ошибки.

СВИ-7 (DHCED)

Data Hiding by Conjugate Error Diffusion (DHCED) – встраивание ЦВЗ за счёт согласованной диффузии ошибки [23]

В данной системе встраиваемая информация представляет собой бинарное изображение $W(n_1, n_2)$, размеры $N_1 \times N_2$ которого равны размерам полутонового контейнера $C(n_1, n_2)$. При встраивании ЦВЗ создаются два бинарных изображения C^B и C^W, являющихся результатом растрирования $C(n_1, n_2)$, таким образом, чтобы в как можно большем числе точек выполнялось равенство

$$W(n_1, n_2) = C^B(n_1, n_2) \oplus C^W(n_1, n_2). \quad (4.23)$$

Для этого изображение C^B создаётся при помощи базового алгоритма диффузии ошибки, рассмотренного выше, а изображение C^W – при помощи модифицированной процедуры диффузии ошибки, в которой на втором этапе вместо формулы (4.17) или (4.20) (в зависимости от используемого алгоритма) используется следующее соотношение:

$$C^W(n_1,n_2)=\begin{cases}1, & u(n_1,n_2)\geq T_1 \ \wedge\ W(n_1,n_2)\oplus C^B(n_1,n_2)=1,\\ 0, & u(n_1,n_2)<T_1 \ \wedge\ W(n_1,n_2)\oplus C^B(n_1,n_2)=1\ ,\\ 1, & u(n_1,n_2)\geq T_2 \ \wedge\ W(n_1,n_2)\oplus C^B(n_1,n_2)=0,\\ 0, & u(n_1,n_2)<T_2 \ \wedge\ W(n_1,n_2)\oplus C^B(n_1,n_2)=0,\end{cases} \quad (4.24)$$

где $T_1 < T < T_2$.

Таким образом, в случае pull-модели для всех пикселей изображения в цикле выполняются последовательно шаги (4.16), (4.24), (4.18) (в последней формуле при этом вместо C^B используется C^W). При использовании push-модели после инициализации изображения $u(n_1,n_2)$ по формуле (4.19) для всех пикселей выполняются последовательно шаги (4.24), (4.21), (4.22) (в формуле (4.21) аналогичным образом используется C^W вместо C^B). Отметим также, что при формировании изображений C^B и C^W возможно использование разных моделей диффузии ошибки.

По умолчанию принято использовать следующие значения параметров системы: $T_1 = 64, T_2 = 192.$

Для извлечения информации используется формула (4.23).

■

Лабораторная работа 2А: Встраивание информации в бинарные изображения

Задания

Лабораторная работа посвящена изучению и программной реализации двух систем стеганографического встраивания в бинарное изображение: СВИ-5 (DHST) и СВИ-6 (DHSPT). В рамках выполнения лабораторной работы необходимо выполнить перечисленные ниже задания, отражающие отдельные этапы реализации данных систем, а также ответить на один контрольный вопрос. Вопросы выбирает преподаватель.

1. Реализовать процедуру генерации встраиваемой информации – двоичной последовательности $\mathbf{b}$ заданной длины N_b (изменяемый параметр).
2. Реализовать процедуру генерации ключа $\mathbf{k}$ – последовательности координат пикселей длины $N_k = N_b$ согласно варианту.
3. Реализовать процедуру встраивания последовательности $\mathbf{b}$ в заданное бинарное изображение при помощи ключа $\mathbf{k}$ одним из алгоритмов DHST или DHSPT (конкретный алгоритм и его параметры определяются вариантом задания).
4. Реализовать процедуру извлечения встроенной последовательности из изображения.

Таблица вариантов заданий

№	Размер окрестности	Выбор компенсирующего пикселя	Способ генерации ключа
1	- (DHST)	-	(4.5)
2	3×3	Случайный	(4.6)
3	3×3	Случайный	(4.7)
4	3×3	Случайный	(4.8)
5	5×5	Случайный	(4.8)
6	5×5	Случайный	(4.9)
7	3×3	Расчёт весов	(4.6)
8	3×3	Расчёт весов	(4.7)
9	3×3	Расчёт весов	(4.8)
10	3×3	Расчёт весов	(4.9)
11	5×5	Расчёт весов	(4.8)
12	5×5	Расчёт весов	(4.9)

Контрольные вопросы

1. Какие способы преобразования из полутонового изображения в бинарное (кроме растрирования) вы знаете? Чем они отличаются друг от друга и от растрирования?
2. Опишите математически алгоритм DHST. Проиллюстрируйте его работу на простом примере (на рисунке).
3. В чём заключаются отличия между различными алгоритмами группы DHST? Проиллюстрируйте различия на простом примере.
4. Зачем производится расчёт весов пикселей в алгоритме DHSPT? Проиллюстрируйте процедуру выбора нужного компенсирующего пикселя по весам.
5. Каким требованиям должны удовлетворять ключи, используемые в алгоритмах группы DHSPT? Расскажите о достоинствах и недостатках каждого из способов генерации ключа, описанных в задании. Можете ли вы предложить какой-либо альтернативный способ?
6. Рассчитайте таблицу весов $w(i)$ при поиске компенсирующего пикселя в окне 7×7 аналогично тому, как эти веса были рассчитаны для случая 5×5.

Лабораторная работа 2В: Встраивание информации при растрировании изображений

Задания

Лабораторная работа посвящена изучению и программной реализации СВИ-7 (DHCED). В рамках выполнения лабораторной работы необходимо выполнить задания из списка основных по вариантам, отмеченным в таблице ниже, а также ответить на один контрольный вопрос. Вопросы выбирает преподаватель. По желанию студент может выполнить дополнительное задание после основных, что будет отмечено преподавателем.

Основные задания

1. Реализовать процедуру растрирования входного полутонового изображения методом диффузии ошибки. Ядро и модель алгоритма определяется вариантом задания.
2. Реализовать процедуру встраивания в растрируемое изображение бинарного изображения алгоритмом DHCED с ядром, использованным в пункте 1, и моделью, определяемой вариантом задания.
3. Реализовать процедуру извлечения информации, встроенной алгоритмом DHCED, и сохранения её в виде бинарного изображения.

Дополнительные задания

1. Математически показать, являются ли pull- и push-модели диффузии ошибки эквивалентными (то есть приводят ли они к идентичному результату растрирования для любого входного изображения) или привести контрпример, доказывающий обратное.
2. Существует модификация системы DHCED, называемая PCED (Pair Conjugate Error Diffusion), которая отличается от DHCED тем, что процессы растрирования двух изображений C^B и C^W выполняются одновременно и оказывают влияние друг на друга. Разумеется, при этом существует разделение вспомогательных величин и изображений на те, которые относятся к растрированию C^B (обозначим их u_1, e_1), и те, которые относятся к растрированию C^W (обозначим их u_2, e_2). Выражение (4.24) при этом изменится, и в нём будут фигурировать обе величины u_1 и u_2. Как и для системы

DHCED, смысл этого выражения будет заключаться в том, чтобы для как можно большего числа точек оказалось выполненным условие (4.23). При выполнении данного задания необходимо сначала записать новое выражение (4.24), согласовать его с преподавателем, после чего приступать к реализации алгоритма.

Таблица вариантов заданий

№	Ядро диффузии ошибки	Алгоритм диффузии ошибки при растрировании	Алгоритм диффузии ошибки при растрировании со встраиванием информации
1	(4.10)	push-модель	push-модель
2	(4.11)	push-модель	push-модель
3	(4.12)	push-модель	push-модель
4	(4.13)	pull-модель	pull-модель
5	(4.14)	pull-модель	pull-модель
6	(4.15)	pull-модель	pull-модель
7	(4.10)	pull-модель	push-модель
8	(4.11)	pull-модель	push-модель
9	(4.12)	pull-модель	push-модель
10	(4.13)	push-модель	pull-модель
11	(4.14)	push-модель	pull-модель
12	(4.15)	push-модель	pull-модель

Контрольные вопросы

1. Какие способы преобразования из полутонового изображения в бинарное (кроме растрирования) вы знаете? Чем они отличаются друг от друга и от растрирования?
2. Опишите общую концепцию метода диффузии ошибки. Приведите примеры весовых функций, используемых на практике. Какие, по-вашему, весовые функции также могут быть использованы и почему?
3. Какие коэффициенты ядра диффузии ошибки должны обязательно быть нулевыми и почему?
4. Как связан общий коэффициент перед матрицами ядер (4.10)-(4.15) с числами внутри матриц? Почему он выбирается именно таким образом?

5. Опишите pull-модель диффузии ошибки. Покажите на рисунке, что происходит в окрестности очередного пикселя (n_1, n_2) при работе этого алгоритма.
6. Опишите push-модель диффузии ошибки. Покажите на рисунке, что происходит в окрестности очередного пикселя (n_1, n_2) при работе этого алгоритма.
7. Опишите алгоритм DHCED. Каков смысл выражения (4.24)?
8. Как результат системы DHCED (изображение C^W, визуальная различимость наличия встроенной информации, точность извлечения) зависит от среднего двух порогов: $\frac{T_1+T_2}{2}$?
9. Как результат системы DHCED (изображение C^W, визуальная различимость наличия встроенной информации, точность извлечения) зависит от разности двух порогов: $T_2 - T_1$?

5. Базовые алгоритмы встраивания информации в области преобразования

5.1. Общая схема

Известно, что крупные объекты на изображении человек воспринимает лучше, чем мелкие детали [1, 16]. Иными словами, система человеческого зрения более чувствительна к низкочастотной информации, нежели к высокочастотной. Поэтому логичным развитием систем встраивания информации в пространственной области является переход от пространственного представления сигнала к частотному с последующей модификацией уже не пикселей изображения, а спектральных компонент.

На практике наиболее популярной альтернативой встраиванию информации в пространственной области является встраивание информации в области дискретных ортогональных преобразований (ДОП), причём главным образом тех, для которых известны быстрые прямые и обратные алгоритмы и спектральные компоненты которых являются слабо коррелированными. К таким преобразованиям в первую очередь относятся дискретное преобразование Фурье (ДПФ), дискретное косинусное преобразование (ДКП) и дискретное вейвлет-преобразование (ДВП).

Итак, пусть дано полутоновое изображение $C(n_1, n_2)$ (контейнер). Встраивание информации в спектральной области обычно состоит из четырёх шагов:

1. Расчёт ДОП изображения C (будем обозначать его C_{DOT}):

$$C_{DOT}(m_1, m_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} C(n_1, n_2) h_{m_1}(n_1) g_{m_2}(n_2), \quad (5.1)$$

где $\{h_{m_1}(n_1)\}_{m_1=0}^{N_1-1}, n_1 = 0..N_1 - 1$, $\{g_{m_2}(n_2)\}_{m_2=0}^{N_2-1}, n_2 = 0..N_2 - 1$ – два семейства ортогональных базисных функций, отличающихся только периодом (то есть при $N_1 = N_2$ они идентичны).

Сокращённо будем обозначать преобразование (5.1) следующим образом (когда тип преобразования известен):

$$C_{DOT}(m_1, m_2) = \mathcal{F}\big(C(n_1, n_2)\big). \quad (5.2)$$

2. Выбор подмножества спектральных компонент для встраивания информации (пространства признаков):

$$f(m) = C_{DOT}\big(\mu(m)\big), m = 0..M-1, \tag{5.3}$$

где отображение $\mu: \mathbb{R} \mapsto \mathbb{R}^2$ либо жёстко фиксировано в рамках конкретного алгоритма, либо определяется на основе секретного ключа $\mathbf{k}$.

3. Собственно встраивание информации в пространстве признаков

$$f^W = \mathcal{E}(f, \Omega, \mathbf{k}), \tag{5.4}$$

в результате которого меняются выбранные спектральные компоненты. Матрицу спектральных компонент после встраивания будем обозначать $C_{DOT}^W\,(m_1, m_2)$.

4. Переход в пространственные координаты при помощи обратного ДОП:

$$C^W(n_1, n_2) = \sum_{m_1=0}^{N_1} \sum_{m_2=0}^{N_2} C_{DOT}^W(m_1, m_2) h_{n_1}^{-1}(m_1) g_{n_2}^{-1}(m_2), \tag{5.5}$$

где $\left\{h_{n_1}^{-1}(m_1)\right\}_{n_1=0}^{N_1-1}, m_1 = 0..N_1-1$, $\left\{g_{n_2}^{-1}(m_2)\right\}_{n_2=0}^{N_2-1}, m_2 = 0..N_2-1$ – два семейства базисных функций обратного преобразования, или в сокращённой записи

$$C^W(n_1, n_2) = \mathcal{F}^{-1}\big(C_{DOT}^W(m_1, m_2)\big). \tag{5.6}$$

Следует отметить, что шаг 2 является распространённым, но не обязательным элементом в приведённой схеме. Нередко всё множество спектральных компонент является пространством признаков. В этом случае f является двумерной матрицей.

В случае использования дискретного преобразования Фурье расчёт спектральных компонент на первом шаге осуществляется по формуле

$$\begin{aligned} C_{DOT}(m_1, m_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} C(n_1, n_2) \times \\ \times exp\left(-\frac{2\pi i m_1 n_1}{N_1}\right) exp\left(-\frac{2\pi i m_2 n_2}{N_2}\right), \end{aligned} \tag{5.7}$$

причём результирующая матрица, как известно, содержит комплексные значения. При использовании дискретного косинусного преобразования формула (5.1) будет иметь вид:

$$C_{DOT}(m_1, m_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} C(n_1, n_2) \times \\ \times cos\left(\frac{\pi m_1}{N_1}\left(n_1 + \frac{1}{2}\right)\right) cos\left(\frac{\pi m_2}{N_2}\left(n_2 + \frac{1}{2}\right)\right). \quad (5.8)$$

Здесь необходимо отметить, что вместо прямого расчёта спектральных компонент по формулам (5.7)–(5.8) всегда используются быстрые алгоритмы, значительно снижающие сложность вычислений.

Под дискретным вейвлет-преобразованием понимается целый класс преобразований, основанных на использовании различных семейств вейвлет-функций [28]. Для большего понимания принципа ДВП рассмотрим его в одномерном случае на примере вейвлетов Хаара и начнём с наглядного примера, рассмотренного в книге [29]. Пусть имеется массив данных:

$$(1,2,3,4,5,6,7,8).$$

Для каждой пары соседних отсчётов осуществляется вычисление полусумм и полуразностей. Результаты записываются последовательно в новый массив данных (сначала все полусуммы, потом все полуразности):

$$\left(\underbrace{\frac{3}{2}, \frac{7}{2}, \frac{11}{2}, \frac{15}{2}}_{L_1}, \underbrace{-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}}_{H_1}\right).$$

В результате массив разделяется на две составляющих: низкочастотную, или аппроксимирующую (первые 4 отсчёта), и высокочастотную, или уточняющую (оставшиеся 4 отсчёта). На следующем шаге аналогичная процедура производится только над низкочастотными отсчётами:

Высокочастотные отсчёты при этом не меняются:

$$\left(\underbrace{\frac{5}{2}, \frac{13}{2}}_{L_2}, \underbrace{-1, -1}_{H_2}, \underbrace{-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}}_{H_1}\right).$$

После ещё одного (заключительного) шага имеем:

$$\left(\underbrace{\frac{9}{2}}_{L_3}, \underbrace{-2}_{H_3}, \underbrace{-1, -1}_{H_2}, \underbrace{-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}}_{H_1}\right).$$

Результат описанной процедуры (если забыть о нормировке) является результатом применения над входным массивом дискретного вейв-

лет-преобразования Хаара. Чтобы преобразование было ортонормированным, необходимо делить не на 2, а на $\sqrt{2}$.

Как видно из примера, в результате подобного преобразования наибольшие по абсолютной величине компоненты оказались сосредоточены в области низких частот. По мере перехода к более высоким частотам значения отсчётов преобразованного массива растут.

На данном примере можно понять общий смысл вейвлет-преобразования: оно заключается в итеративном применении ко входному сигналу двух функций (в данном случае – функций вычисления полусумм и полуразностей) с разными масштабами и сдвигами. Первая из этих функций называется *скейлинг-функцией* (scaling function), вторая – *вейвлетом* (wavelet). Подробнее принцип расчёта коэффициентов ДВП изложен в книге [30] (параграф 8.3).

Важно заметить, что на практике при использовании ДВП для встраивания информации редко доходят до одного аппроксимирующего коэффициента. Вместо этого чаще всего осуществляют 2-3 стадии декомпозиции, после чего осуществляют встраивание информации либо в низкочастотные, либо в высокочастотные отсчёты последнего уровня. На рис. 5.1 проиллюстрировано применение ДВП в двумерном случае. Каждый уровень разложения заключается в последовательном расчёте полусумм и полуразностей (в случае использования вейвлетов Хаара) по горизонтали и по вертикали.

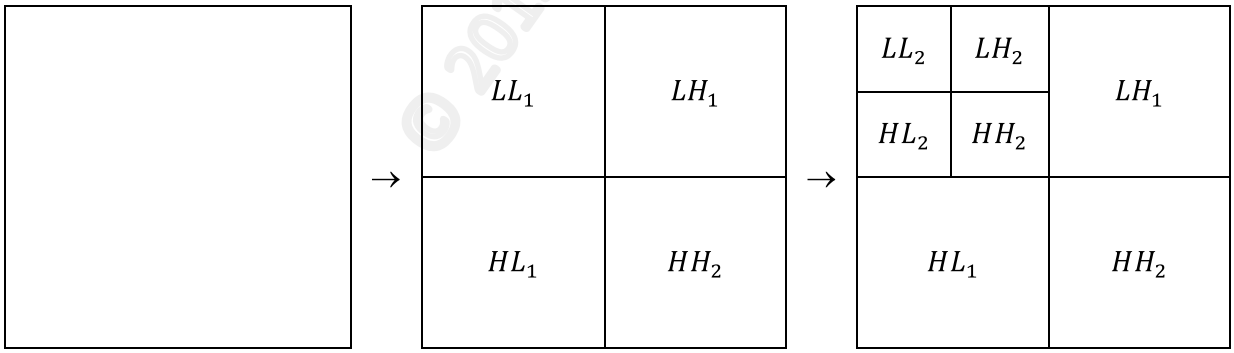


Рис. 5.1 – Принцип пирамидального вейвлет-разложения изображения (два уровня декомпозиции)

Функция $\mathcal{E}$, используемая на третьем шаге для встраивания информации в пространстве признаков, теоретически может иметь любой вид, однако наиболее распространёнными являются функции вида (5.9) и (5.10), реализующие так называемое *аддитивное* и *мультипликативное* встраивание информации:

$$f^W(\mathbf{m}) = f(\mathbf{m}) + \alpha \cdot \beta(\mathbf{m}) \cdot \Omega(\mathbf{m}), \quad (5.9)$$

$$f^W(\mathbf{m}) = f(\mathbf{m})\big(1 + \alpha \cdot \beta(\mathbf{m}) \cdot \Omega(\mathbf{m})\big), \quad (5.10)$$

где $\mathbf{m}$ – частотный аргумент (который может быть одномерным или двумерным), α – постоянный множитель, иногда называемый коэффициентом усиления, а $\beta(\mathbf{m})$ – адаптивная маска усиления (в простейшем случае $\beta(\mathbf{m}) = 1$ для всех $\mathbf{m}$). Ω, как и ранее, обозначает признаковое представление встраиваемой информации.

5.2. Концепция встраивания информации с расширением спектра

Одним из наиболее распространённых методов встраивания информации (главным образом, водяных знаков) в области преобразования является встраивание информации с расширением спектра. Вместе с тем следует подчеркнуть, что этот метод может использоваться и при встраивании информации в пространственной области и не следует его строго ассоциировать со встраиванием в области преобразования.

Первая система, реализующая этот метод, была предложена в 1997 в работе [31], а в настоящее время число подобных систем насчитывает не один десяток. Прежде чем перейти непосредственно к рассмотрению конкретных систем, коротко остановимся собственно на понятии расширения спектра. Этот подход изначально использовался для повышения помехоустойчивости при передаче радиосообщений по каналам связи, характеризующимся высокими шумами или подвергающимся глушению. Суть метода заключается в распределении узкополосного сигнала по каналу с куда большей пропускной способностью. Функция, обеспечивающая распределение сигнала, базируется на секретном ключе, известном только отправителю и получателю сообщения.

Таким образом, концепция встраивания информации в цифровые сигналы с расширением спектра состоит в распределении встраиваемой информации (ЦВЗ или секретного сообщения) внутри контейнера, имеющего гораздо больший размер, на основе функции, зависящей от секретного ключа. Наиболее простым примером расширения спектра является генерация псевдослучайного шаблона, при этом начальное значение генератора однозначно определяется парой «ЦВЗ – ключ».

Для примера рассмотрим простейшую систему встраивания ЦВЗ с расширением спектра в пространственной области, а затем уже перейдём к системам, работающим в спектральной области.

СВИ-8 (E_BLIND/D_LC)

Простейшая ЦВЗ-система с расширением спектра [2]

Данная система позволяет встроить только один бит информации (то есть $\mathbf{b} \in \{0,1\}$) в полутоновой контейнер C размерами $N_1 \times N_2$. Для встраивания информации формируется шаблон ЦВЗ W_r, размерами совпадающий с исходным контейнером и содержащий нормально распределённые числа с нулевым средним и единичной дисперсией:

$$W_r(n_1, n_2) \sim N(0,1). \tag{5.11}$$

Шаблон W_r целиком определяется на основе ключа $\mathbf{k}$ (например, ключ может использоваться в качестве начального значения генератора псевдослучайных чисел).

Встраивание ЦВЗ осуществляется по аддитивной формуле

$$C^W(n_1, n_2) = C(n_1, n_2) + \alpha \cdot (-1)^{\boldsymbol{b}+1} \cdot W_r(n_1, n_2), \tag{5.12}$$

где $\alpha > 0$ – коэффициент усиления ЦВЗ.

Для извлечения встроенного бита информации рассчитывается значение линейной корреляции принятого носителя информации с шаблоном W_r, который может быть заново сформирован на основе ключа $\mathbf{k}$:

$$\rho\left(\widetilde{C^W}, W_r\right) = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \widetilde{C^W}(n_1, n_2) \cdot W_r(n_1, n_2), \tag{5.13}$$

после чего полученная величина сравнивается с порогом для принятия решения о наличии ЦВЗ и его значении:

$$\mathbf{b^R} = \begin{cases} 1, & \rho\left(\widetilde{C^W}, W_r\right) > \tau_{lc}, \\ 0, & \rho\left(\widetilde{C^W}, W_r\right) < -\tau_{lc}, \\ \text{нет ЦВЗ}, & \text{иначе}. \end{cases} \tag{5.14}$$

■

Данная система хорошо иллюстрирует характерную особенность метода расширения спектра: встраивание небольшого объёма информации (в данном случае – одного бита) происходит за счёт изменения значительного числа отсчётов матрицы признаков (в данном случае – всех).

Очевидный недостаток рассмотренной системы – возможность встраивания лишь одного бита информации, однако, во-первых, система

легко конвертируется в систему с детектором, если рассматривать в качестве ключа совокупность векторов $\mathbf{k}$ и $\mathbf{b}$, а во-вторых, на её основе могут быть построены более интересные и практически значимые СВИ. Большое внимание данной системе и её расширениям уделяется в книге [2], мы же к ней вернёмся в главе 6.

5.3. Системы встраивания информации в области преобразования с расширением спектра

Перейдём к рассмотрению СВИ с расширением спектра в области преобразования. Одна из первых и наиболее известных подобных систем была предложена в работе [31] и представляла собой применение концепции встраивания информации с расширением спектра в области дискретного косинусного преобразования.

СВИ-9 (Cox et al.)

Встраивание ЦВЗ в изображения с расширением спектра [31]

Встраивание информации

Встраиваемая информация $\mathbf{b}$ в совокупности с ключом $\mathbf{k}$ кодируют матрицу признаков, имеющую в данном случае вид вектора длиной $N_\Omega = 1000$ чисел: $\Omega \in \mathbb{R}^1_{[N_\Omega]}$, причём элементы Ω представляют собой псевдослучайные числа, распределенные по гауссовскому закону.

Для модификации отбираются 1000 самых больших коэффициентов глобального дискретного косинусного преобразования контейнера (5.8) в *змеевидной развёртке*, как показано на рис. 5.2 (при этом нулевой отсчёт $C_{DOT}(0,0)$ не изменяется). Результатом данного отбора является матрица признаков контейнера $f \in \mathbb{R}^1_{[N_\Omega]}$. Не будем конкретизировать точную формулу расчёта f, поскольку она окажется весьма громоздкой. Встраивание информации в пространстве признаков осуществляется по формуле (5.10) при $\beta = 1$:

$$f^W(m) = f(m)\big(1 + \alpha \cdot \Omega(m)\big). \qquad (5.15)$$

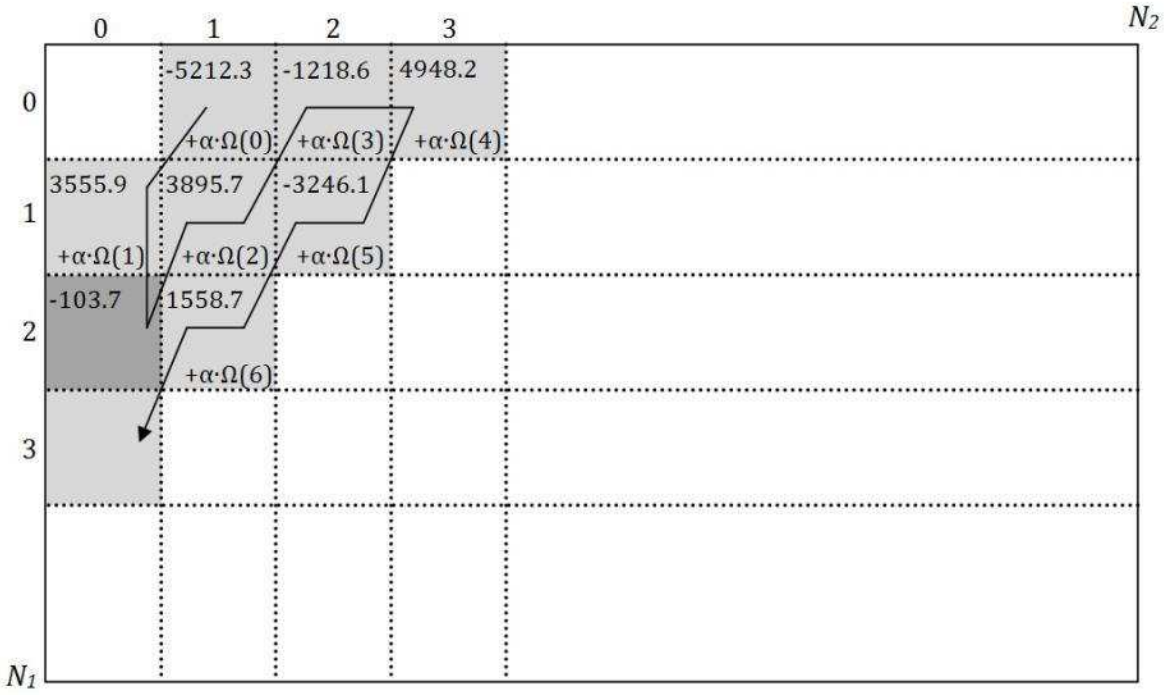


Рис. 5.2 – Схема отбора коэффициентов ДКП для модификации в СВИ-9 (Cox et al.). В каждой ячейке в качестве иллюстрации отображены значения соответствующих компонент ДКП изображения "Lenna"(рис. 4.1а)

Извлечение информации

Для извлечения ЦВЗ используется исходный контейнер, по которому рассчитывается вектор f и которому ставится в соответствие вектор признаков принятого носителя информации $\widetilde{f^W}$. Извлечение признаков встроенного ЦВЗ осуществляется по формуле

$$\tilde{\Omega}(m) = \frac{\widetilde{f^W}(m) - f(m)}{\alpha \cdot f(m)}, \qquad (5.16)$$

то есть путём прямого выражения Ω из формулы (5.15).

Далее осуществляется детектирование (то есть проверка наличия встроенного) ЦВЗ, которое может осуществляться по формуле (1.1) с функцией близости вида*

* В оригинальной работе [27] в знаменателе отсутствует длина вектора Ω, однако отмечается, что использование конкретной функции близости не является принципиальным.

$$\rho\left(\Omega,\tilde{\Omega}\right)=\frac{\sum_{n=0}^{\mathrm{N}_\Omega-1}\Omega(m)\tilde{\Omega}(m)}{\sqrt{\sum_{m=0}^{\mathrm{N}_\Omega-1}\Omega^2(m)}\cdot\sqrt{\sum_{m=0}^{\mathrm{N}_\Omega-1}\tilde{\Omega}^2(m)}}. \quad (5.17)$$

То есть фактически $\rho\left(\Omega,\tilde{\Omega}\right)$ в формуле (5.17) является косинусом угла между векторами Ω и $\tilde{\Omega}$.

Если значение ρ не меньше некоторого порога τ, то детектор ЦВЗ срабатывает, в противном случае принимается решение об отсутствии встроенного водяного знака W в рассматриваемом изображении.

■

Достоинством алгоритма является то, что благодаря выбору наиболее значимых коэффициентов ДКП водяной знак является стойким к сжатию, поэлементным преобразованиям, процедурам обработки скользящим окном, а также многим другим видам обработки изображений. Вместе с тем данная система уязвима для некоторых видов атак. К недостаткам данного алгоритма стоит отнести трудоёмкость операции вычисления двумерного ДКП всего изображения, а также необходимость использования контейнера на этапе извлечения информации.

Для устранения последнего недостатка в работе [32] была предложена модификация СВИ-9 со слепым детектором.

СВИ-10 (Piva et al.)

Слепая модификация системы Cox et al. [32]

Для чего требуется исходное изображение в СВИ-9? Во-первых, чтобы отыскать N_Ω наибольших ДКП-коэффициентов и проранжировать их по порядку, во-вторых, чтобы вычислить оценку $\widetilde{\Omega}$ по формуле (5.16). Для исключения необходимости ранжирования спектральных компонент по убыванию в алгоритм встраивания внесена очевидная корректировка: для встраивания всегда отбираются одни и те же коэффициенты (среднечастотные), отсортированные в порядке змеевидной развёртки. Вместо формулы (5.16) расчёт оценки встроенной последовательности происходит следующим образом:

$$\tilde{\Omega}(m)=\widetilde{f^W}(m), \quad (5.18)$$

то есть сам носитель информации (в форме матрицы признаков) используется в качестве оценки ЦВЗ. Разумеется, такая оценка в случае встраи-

вания по формуле (5.15) далека от истины, поэтому меняется и формула встраивания:

$$f^W(m) = f(m) + \alpha \cdot |f(m)| \cdot \Omega(m). \quad (5.19)$$

Оценка близости рассчитывается как

$$\rho\left(\Omega, \tilde{\Omega}\right) = \sum_{n=0}^{\mathrm{N}_\Omega - 1} \Omega(m)\tilde{\Omega}(m). \quad (5.20)$$

Причину использования (5.19) вместо (5.15) легко понять, подставив (5.18) и (5.19) в (5.20):

$$\rho\left(\Omega, \tilde{\Omega}\right) = \sum_{n=0}^{\mathrm{N}_\Omega - 1} \Omega(m)\widetilde{f^W}(m) \approx \sum_{n=0}^{\mathrm{N}_\Omega - 1} \Omega(m)f(m) + \alpha \sum_{n=0}^{\mathrm{N}_\Omega - 1} \Omega^2(m) \cdot |f(m)|.$$

Второе слагаемое всегда больше нуля, в то время как первое ввиду случайности $\Omega(m)$ и предполагаемой однородности выбранных $f(m)$ по абсолютной величине (поскольку это среднечастотные коэффициенты) может иметь произвольный знак, но небольшое значение по модулю. Таким образом, при детектировании встроенной последовательности $\rho\left(\Omega, \tilde{\Omega}\right)$ будет иметь большое положительное значение.

Для повышения точности детектирования ЦВЗ длина встраиваемой последовательности увеличивается относительно принятого в СВИ-9 значения $N_\Omega = 1000$. В СВИ-10 длина не фиксирована, а может варьироваться в зависимости от размера изображения. В частности, для изображений размером 512×512 для встраивания рекомендуется использовать коэффициенты ДКП со строки 180 до строки 250 (в зигзагообразной развёртке). В этом случае длина последовательности составляет около 15000.

Ещё одно изменение предназначено для снижения визуальных искажений при встраивании ЦВЗ и заключается в том, что $C^W(n_1, n_2)$ формируется не по формуле (5.6), а по формуле

$$\begin{aligned} C^W(n_1, n_2) = \mathcal{F}^{-1}\left(C^W_{DOT}(m_1, m_2)\right) \cdot B(n_1, n_2) + \\ + C(n_1, n_2) \cdot \left(1 - B(n_1, n_2)\right), \end{aligned} \quad (5.21)$$

где

$$B(n_1, n_2) = \frac{C_{MSE,9\times9}(n_1, n_2)}{\max\limits_{i.j} C_{MSE,9\times9}(i, j)}. \quad (5.22)$$

В последней формуле $C_{MSE,9\times9}(n_1, n_2)$ – результат отыскания локального СКО изображения C в скользящем окне размерами 9×9.

Таким образом, в областях низкой дисперсии (то есть достаточно однородных по яркости) $B(n_1, n_2) \to 0$, значит, $C^W(n_1, n_2)$ практически совпадает с $C(n_1, n_2)$. В областях, характеризуемых высокой дисперсией (то есть на границах областей и в текстурированных регионах), напротив, основную часть в сумме (5.21) составляет $\mathcal{F}^{-1}\left(C_{DOT}^W(m_1, m_2)\right)$. На рис. 5.3 приведён пример поля $B(n_1, n_2)$ для конкретного изображения-контейнера.

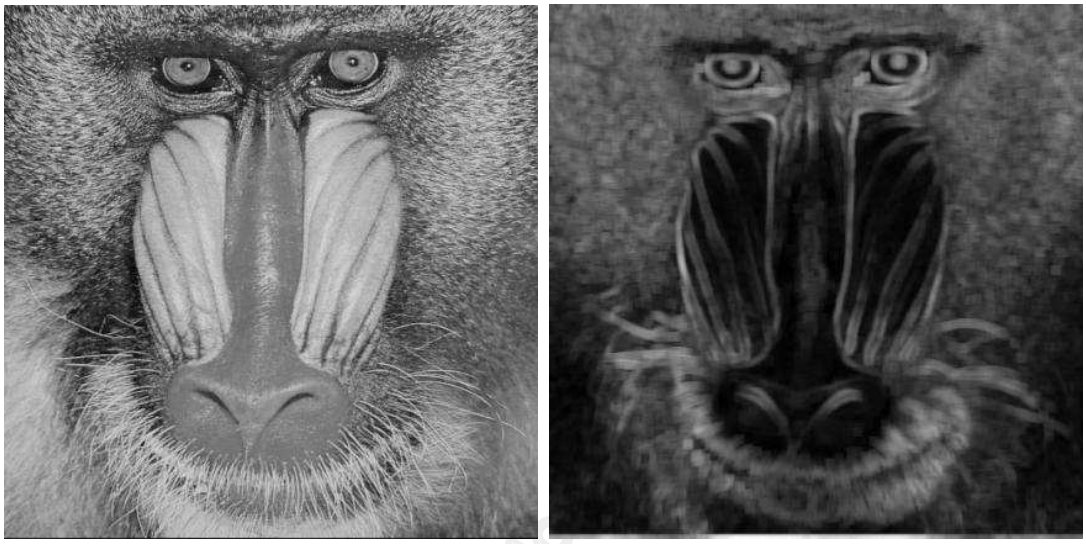


Рис. 5.3 – Полутоновое изображение и его поле локального СКО, соответствующее матрице $B(n_1, n_2)$ в СВИ-10 (Piva et al.)

В заключение отметим, что авторы данной системы уточнили способ расчёта порога T_ρ в формуле детектирования (1.1) исходя из вероятностных соображений:

$$T_\rho = 3.3\sqrt{\frac{2\sigma_{\tilde{f}}^2}{N_\Omega}}, \tag{5.23}$$

где $\sigma_{\tilde{f}}^2$ – оценка дисперсии матрицы $\widetilde{f^W}$.

Вывод данной формулы можно найти в статье [33].

■

СВИ-10, как и его предшественник, обладает высокой стойкостью к ряду преобразований носителя информации, однако может использоваться в сценариях, которые не предполагают возможности доступа к исходному контейнеру на этапе извлечения информации.

Далее рассмотрим кратко ещё два примера СВИ с расширением спектра в области вейвлет-преобразования, которые нам понадобятся в лабораторной работе 3.

СВИ-11 (Corvi & Nicchiotti)

Система мультипликативного встраивания в области ДВП [34]

В данной системе встраиваемая информация (в форме матрицы признаков) представляет собой двумерный массив псевдослучайных чисел, распределенных по гауссовскому закону: $\Omega \in \mathbb{R}^2_{[P\times P]}$, где $P = 32$, то есть всего 1024 числа.

Исходное изображение $C(n_1, n_2)$ подвергается вейвлет-преобразованию на такое число уровней декомпозиции, которое необходимо для получения низкочастотного изображения (в *LL*-поддиапазоне) размером $P \times P$ (то есть предполагается, что размеры изображения N_1 и N_2 кратны P). То есть $f(m_1, m_2)$ также имеет размеры $P \times P$.

Встраивание информации в эти коэффициенты выполняется по формуле

$$f^W(m_1, m_2) = f_{mean} + (f(m_1, m_2) - f_{mean})\big(1 + \alpha\Omega(m_1, m_2)\big), \qquad (5.24)$$

где f_{mean} – среднее значение $f(m_1, m_2)$.

Оценка $\tilde{\Omega}$ находится прямым выражением Ω из (5.24):

$$\tilde{\Omega}(m_1, m_2) = \frac{\widetilde{f^W}(m_1, m_2) - f(m_1, m_2)}{\alpha(f(m_1, m_2) - f_{mean})}. \qquad (5.25)$$

Детектирование происходит по формуле (1.1) с функцией близости (5.17).

■

СВИ-12 (Wang et al.)

Система аддитивного встраивания в области ДВП [35]

В данной системе, как и в СВИ-9 (Cox et al.), встраиваемая информация представляется вектором случайных чисел, распределенных по гауссовскому закону: $\Omega \in \mathbb{R}^1_{[N_\Omega]}$.

При встраивании выполняется две стадии вейвлет-декомпозиции контейнера, и все коэффициенты, за исключением LL-поддиапазона, используются для встраивания данных (см. рис. 5.4).

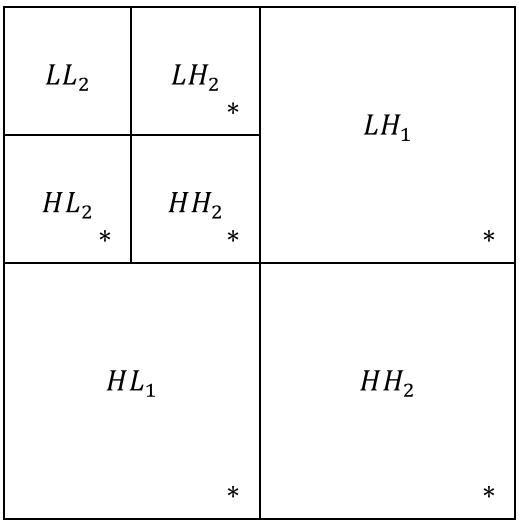


*Рис. 5.4 – Вейвлет-коэффициенты, используемые в СВИ-12 (Wang et al.) для встраивания ЦВЗ (помечены *)*

Собственно встраивание ЦВЗ выполняется по формуле

$$f^W(m_1, m_2) = f(m_1, m_2) + \alpha_s \cdot T_s \cdot \Omega\big(\varphi(m_1, m_2)\big), \tag{5.26}$$

где φ задаёт соответствие координат (m_1, m_2) матрицы вейвлет-коэффициентов и индекса вектора Ω, s – это порядковый номер поддиапазона, к которому относится (m_1, m_2), $\alpha_s \in (0; 1]$ – множитель, управляющий соотношением между сигналом и ЦВЗ в носителе информации, T_s – пороговое значение для текущего поддиапазона, вычисленное при формировании последовательности $\varphi(m)$.

Извлечение в базовом варианте выполняется по обратной формуле

$$\Omega\big(\varphi(m_1, m_2)\big) = \frac{\widetilde{f^W}(m_1, m_2) - f(m_1, m_2)}{\alpha_s \cdot T_s} \tag{5.27}$$

с последующим вычислением функции близости по формуле (5.17). Существует также версия метода со слепым детектором, на которой мы не будем останавливаться.

Следует заметить, что для встраивания отбираются не все вейвлет-коэффициенты в областях, отмеченных звёздочками на рис. 5.4, а только N_Ω. Для их отбора выполняются следующие 3 шага:

1. Для всех поддиапазонов, кроме LL, вычисляется первоначальное значение порога T_s:

$$T_{s,0} = \frac{\beta_s}{2} \max_{(m_1, m_2) \in s} f(m_1, m_2),$$

где β_s – весовой коэффициент поддиапазона, являющийся параметром системы.

2. Просматриваются все поддиапазоны в порядке убывания значений порога T_s и все пиксели в них в порядке построчной развёртки. Все положения (m_1, m_2) коэффициентов $f(m_1, m_2)$, которые превышают порог, добавляются в φ. Это происходит до тех пор, пока ее длина не достигнет N_Ω.
3. Если в результате полного обхода текущее количество элементов φ меньше N_Ω, то пороги для всех поддиапазонов уменьшаются вдвое:

$$T_{s,i+1} = \frac{1}{2} T_{s,i},$$

после чего повторяется предыдущий шаг.

■

Упражнения

Встраивание бинарного логотипа в области ДПФ, ДКП, ДВП

Результатами работы будут являться скрипт `spectrum_run`, а также функции:

`[value] = psnr(C)`
– расчёт PSNR изображения.

`[CW] = simple_dct_embed(C)`
– встраивание логотипа в ДКП изображения.

`[CW] = simple_dft_embed(C)`
– встраивание логотипа в ДПФ изображения.

`[C_dwt] = dwt2_bylevel(C, level)`
– расчёт вейвлет-преобразования изображения вплоть до уровня `level`.

`[C] = idwt2_bylevel(C_dwt, level)`
– расчёт обратного преобразования из вейвлет-спектра изображения `level`.

`[CW] = simple_dwt_embed(C)`
– встраивание логотипа в ДВП изображения.

1. Реализовать функцию `psnr` для расчёта PSNR изображения по формулам (1.6)–(1.7) и проверить её работу (если в используемой версии MATLAB уже имеется реализация расчёта PSNR, то сопоставить результаты работы написанной и стандартной функции).
2. Считать входное изображение, выполнить прямое и обратное ДКП, отобразить результат.
3. Считать бинарный логотип, встроить его в центр спектра, сформировать и отобразить результат подобного встраивания.
4. Посчитать и отобразить поле ошибок, оценить PSNR полученного изображения. Перенести код встраивания информации в отдельную функцию `simple_dct_embed`, проверить её работу.
5. Реализовать аналогичное встраивание логотипа в центр модуля спектра Фурье в виде функции `simple_dft_embed`. В скрипте оценить PSNR полученного изображения, отобразить поле ошибок.

6. Изменить функцию `simple_dft_embed`, реализовав встраивание в фазу спектра. В скрипте оценить PSNR полученного изображения, отобразить поле ошибок. Сравнить с предыдущими результатами.
7. Реализовать один уровень вейвлет-разложения входного изображения, визуализировать результат.
8. Реализовать функцию `dwt2_bylevel` и проверить правильность её работы. Использовать семейство вейвлетов Хаара.
9. Реализовать функцию `idwt2_bylevel` и проверить правильность её работы. Использовать семейство вейвлетов Хаара.
10. Реализовать встраивание логотипа в центр области HH_2 ДВП с использованием семейства вейвлетов Хаара в виде функции `simple_dwt_embed`. В скрипте оценить PSNR полученного изображения, отобразить поле ошибок.

Реализация и исследование системы СВИ-8 (E_BLIND/D_LC)

Результатом работы будет являться скрипт `eblind_run`.

1. Считать имена файлов изображений из заданной папки и их пиксельные размеры $N_1 \times N_2$ (предполагая, что размеры разных изображений идентичны).
2. Сгенерировать случайный нормально распределённый шаблон ЦВЗ (5.11), выполнить его нормировку.
3. Сформировать для каждого изображения в папке два носителя информации, содержащие бит 1 (аддитивно прибавляется шаблон) и бит 0 (вычитается шаблон) (5.12). Использовать коэффициент $\alpha = 1$.
4. Рассчитать три вектора значений линейной корреляции (5.13) шаблона ЦВЗ с исходным изображением, результатом встраивания бита 1, и результатом встраивания бита 0.
5. Рассчитать и отобразить в одном окне гистограммы значений трёх полученных векторов.
6. Найти пороговые значения по критерию минимума модуля разницы между отсчётами гистограммы.
7. Найти доли ошибок извлечения (false positive rate, false negative rate, см. табл. 8.2).

8. Усреднить изображение окном 3×3, увеличить значение α до трёх, вновь найти пороги и оценить ошибки извлечения.

Реализация и исследование системы СВИ-9 (Cox et al.)

Результатами работы будут являться скрипт `cox_run`, а также функции:

`[Omega] = cox_gen_sig(No)`
– генерация встраиваемой последовательности `Omega` длиной `No`.

`[CW] = cox_embed(C, Omega, No)`
– встраивание информации.

`[Omega_r] = cox_extract(C, CW, No)`
– извлечение информации.

`[corr] = cox_cmp(Omega, Omega_r)`
– сравнение встроенной и извлечённой информации.

1. Реализовать функцию `cox_gen_sig)` и написать фрагмент скрипта `cox_run`, вызывающего данную функцию.
2. Начать реализацию функции `cox_embed` встраивания информации: найти ДКП контейнера, в также вектор `No` наибольших спектральных компонент (в порядке убывания) и вектор их позиций.
3. Продолжить реализацию функции `cox_embed`: встроить информацию в пространстве признаков.
4. Завершить реализацию функции `cox_embed`: перейти от пространства признаков к пространству исходного изображения – сначала изменить спектральные компоненты в соответствии с полученным ранее вектором, затем выполнить обратное ДКП.
5. Написать фрагмент скрипта `cox_run`, вызывающего функцию `cox_embed`, визуализировать контейнер и носитель информации, а также их поэлементную разницу.
6. Реализовать функцию `cox_extract` извлечения ЦВЗ по формуле (5.16).
7. Реализовать функцию `cox_cmp` сравнения исходного ЦВЗ с извлечённым по формуле (5.17); написать фрагмент скрипта `cox_run`, вызывающего функции `cox_extract` и `cox_cmp`.

8. Сгенерировать 10000 последовательностей и осуществить сравнение каждой из них с извлечённым ЦВЗ. Найти наибольший показатель близости. Сделать выводы.
9. Внести изменения в написанные функции и скрипт для обеспечения безошибочного извлечения встроенного ЦВЗ.
10. Заменить $3/4$ площади носителя информации отсчётами исходного контейнера (связанную прямоугольную область), извлечь ЦВЗ и оценить его близость со встроенным ЦВЗ. Сделать выводы о стойкости ЦВЗ к потере данных.
11. Заменить $3/4$ площади носителя информации отсчётами исходного контейнера (в псевдослучайных точках), извлечь ЦВЗ и оценить его близость со встроенным ЦВЗ. Сделать выводы о стойкости ЦВЗ к потере данных.

Реализация и исследование системы СВИ-10 (Piva et al.)

Результатами работы будут являться скрипт `piva_run`, а также функции:

`[CW] = piva_embed(C, Omega, No)`
– встраивание информации.

`[Omega_r] = piva_extract(C, CW, No)`
– извлечение информации.
– сравнение встроенной и извлечённой информации.

1. Начать реализацию функции `piva_embed` на базе `cox_embed`: изменить метод выбора спектральных компонент для встраивания информации.
2. Реализовать в скрипте генерацию вспомогательной матрицы $B(n_1, n_2)$ по формуле (5.22).
3. Продолжить реализацию функции `piva_embed`: изменить метод встраивания на базе формул (5.19) и (5.21).
4. Написать фрагмент скрипта `piva_run`, вызывающего функцию `piva_embed`, визуализировать контейнер и носитель информации, а также их поэлементную разницу.
5. Реализовать функцию `piva_extract` извлечения ЦВЗ по формуле (5.18); написать фрагмент скрипта `piva_run`, вызывающего функции `piva_extract` и `piva_cmp`.

6. Сгенерировать 10000 последовательностей и осуществить сравнение каждой из них с извлечённым ЦВЗ. Найти наибольший показатель близости. Сделать выводы.

6. ЦВЗ для аутентификации содержимого

Современные инструменты обработки цифровых сигналов позволяют с лёгкостью обрабатывать фотографии, видео- и аудиофайлы, в том числе изменяя содержимое. В ряде случаев такие изменения могут быть преднамеренно вредоносными или могут непреднамеренно повлиять на интерпретацию содержимого. Например, случайное изменение рентгеновского снимка может привести к неправильному диагнозу, а фальсификация фотографических доказательств в уголовном процессе могут привести к неправильному решению суда. Таким образом, в некоторых задачах существует необходимость проверки подлинности или целостности цифровых объектов – изображений, аудио, видео. В частности, важно иметь в своём арсенале методы, позволяющие ответить на следующие вопросы [2]:

- Был ли объект каким-либо образом изменён?
- Если да, то были ли эти изменения значительными?
- Какие фрагменты подверглись изменению?
- Может ли объект быть восстановлен?

Методы, позволяющие ответить на вопросы из данного списка, могут быть разделены на две группы [36]: пассивные и активные методы аутентификации содержимого. Пассивные методы заключаются в расчёте ряда характеристик объекта и сопоставлении их с типичными или априори известными значениями [37]. Например, если объект представляет собой спутниковый снимок определённой местности, снятый в известное время, то один из способов проверки подлинности заключается в проверке ракурса съёмки и направления теней. Сценарий использования активных методов состоит из двух шагов. На первом нам доступны сырые (неизменённые) данные, что позволяет оценить некоторые их характеристики (например, результат хэширования) или намеренно изменить объект определённым образом. Задача второго шага – проверить подлинность объекта (но уже без доступа к оригиналу). Одним из наиболее эффективных подходов активной защиты является использование цифровых водяных знаков. Далее подробнее остановимся на различных методах встраивания ЦВЗ для решения задач аутентификации изображений и сценариях их использования.

6.1. Точная аутентификация

В задаче точной аутентификации требуется выявить любые изменения, произошедшие с изображением, включая в том числе его изменение вследствие встраивания ЦВЗ. Таким образом, ЦВЗ, встроенный в изображение для защиты от изменений, должен быть полностью удалён после проверки. Таким требованиям удовлетворяют так называемые *удаляемые ЦВЗ*. Сценарий использования удаляемых ЦВЗ для точной аутентификации предполагает следующие шаги (изложим его в популярной в криптографии нотации Алисы и Боба):

1. Алиса вычисляет одностороннюю хэш-функцию изображения и встраивает её результат в это изображение в качестве ЦВЗ.
2. Алиса отправляет результирующий носитель ЦВЗ Бобу.
3. Боб извлекает ЦВЗ из полученного изображения.
4. Боб удаляет ЦВЗ из изображения. Теперь оно должно быть эквивалентно исходному в случае отсутствия изменений при его передаче.
5. Боб вычисляет одностороннюю хэш-функцию полученного изображения и сравнивает её результат с ЦВЗ.
6. Изображение признаётся подлинным тогда и только тогда, когда результаты хэширования полностью совпадают.

Таким образом, эффективность решения задачи точной аутентификации сводится к эффективности построения систем удаляемых ЦВЗ. Однако построение таких систем является непростой задачей, поскольку требования, предъявляемые к ним (применимость к любым изображениям, возможность полного восстановления, минимизация числа ложных срабатываний) являются взаимно-противоречивыми. Рассмотрим примеры систем встраивания удаляемых ЦВЗ.

СВИ-13 (E_MOD/D_LC)

Система встраивания ЦВЗ на основе модульной арифметики [2]

Данная система является модификацией системы СВИ-8 (E_BLIND/D_LC), рассмотренной в главе 5 и предназначенной для встраивания одного бита информации. Для системы удаляемого ЦВЗ нет необходимости встраивать один бит (для передачи информации об исходном контейнере этого недостаточно, для системы с детектором – излишне). Поэтому, учитывая, что контейнер представляет собой полутоновое изо-

бражение, пиксели которого принимают значения от 0 до 255, формула встраивания информации (5.12) примет вид:

$$C^W(n_1, n_2) = \left(C(n_1, n_2) + \alpha \cdot W_r(n_1, n_2)\right) \bmod 256, \quad (6.1)$$

где W_r, как и ранее, псевдослучайный шаблон, совпадающий размерами с исходным изображением, но в данном случае он может опосредованно нести информацию о контейнере.

При детектировании ЦВЗ сначала вычисляется значение линейной корреляции $\rho\left(\widetilde{C^W}, W_r\right)$ по формуле (5.13), после чего принимается решение о наличии встроенного ЦВЗ, если $\rho\left(\widetilde{C^W}, W_r\right) > \tau_{lc}$.

Удаление ЦВЗ осуществляется по формуле

$$\tilde{C}(n_1, n_2) = \left(\widetilde{C^W}(n_1, n_2) - \alpha \cdot W_r(n_1, n_2)\right) \bmod 256. \quad (6.2)$$

■

Следует отметить, что модульное встраивание по формуле (6.1) может приводить к шуму «соль-и-перец». Поэтому визуально носитель информации будет выглядеть плохо. Однако здесь следует помнить о том, что эти искажения полностью устраняются при удалении ЦВЗ.

У рассмотренной системы есть следующие недостатки:

1. Шум типа «соль-и-перец», вызванный изменением формулы встраивания, отрицательно сказывается на качестве детектирования и приводит к росту числа ошибок.
2. Корреляционный детектор будет неработоспособен для обнаружения встраивания вида (6.1), если исходный контейнер C содержит значения, равномерно распределённые на отрезке от 0 до 255. Таким образом, метод будет неприменим для защиты изображения, которое было подвергнуто процедуре эквализации гистограммы.
3. Данная система предполагает детектирование ЦВЗ, что не позволяет полноценно использовать её в предложенном выше сценарии точной аутентификации.

Далее рассмотрим другую систему встраивания удаляемых ЦВЗ, обладающую большей практической значимостью.

СВИ-14 (Lossless-LSB)

Удаляемые ЦВЗ за счёт сжатия НЗБ

В литературе описаны по меньшей мере две системы ([38], [39]), использующие сжатие наименее значимых битовых плоскостей для встраивания дополнительной информации. Здесь мы опишем упрощённую систему, реализующую данный подход. Пусть C – полутоновой контейнер размерами $N_1 \times N_2$, C_k – k-я битовая плоскость контейнера. Как было показано в параграфе 3.1 (см. рис. 3.1), по мере увеличения k битовые плоскости становятся всё менее шумоподобными, и на них начинают проступать очертания крупных объектов. Таким образом, примерно 3-я или 4-я битовая плоскость зачастую являются хорошо сжимаемыми, но в то же время их изменение не слишком существенно сказывается на визуальном качестве.

В рассматриваемой системе осуществляется сжатие без потерь одной из битовых плоскостей $C_k, k = \{3, 4\}$. Далее выбранная битовая плоскость обнуляется, а на её место побитово записывается полученный архив. Далее после метки окончания архива записывается информация об исходном контейнере (например, его хэш). Порядок извлечения и удаления встроенной информации очевиден.

■

6.2. Избирательная аутентификация

Рассмотренные примеры задач защиты медицинских изображений и документальных свидетельств, требующих точной аутентификации, скорее являются исключением из правил. В большинстве же практических приложений допустимы незначительные искажения, вызванные необходимостью внедрить защитный водяной знак. Если ЦВЗ должен разрушаться при малейших изменениях носителя информации, то такие ЦВЗ называются хрупкими. Простейшей система защиты изображений хрупкими ЦВЗ может быть построена на основе СВИ-1 (НЗБ-встраивание ЦВЗ) путём изменения метода извлечения информации с декодера на детектор, проверяющий наличие заданного ЦВЗ в НЗБП. В этом случае если найдётся хотя бы одна точка (n_1, n_2), в которой

$$C_p(n_1, n_2) \neq W(n_1, n_2),$$

где p – номер битовой плоскости, то устанавливается, что изображение изменилось. Для системы хрупких водяных знаков используется $p = 1$.

В то же время весьма распространённой является ситуация, когда незначительные изменения носителя информации считаются допустимыми после встраивания в него защитного ЦВЗ. К таким преобразованиям может относиться слабая фильтрация шума (линейная или медианная), контрастирование, сжатие с потерями (до определённого уровня погрешности), поворот на угол, кратный $\pi/2$, вырезание фрагмента изображения. Водяные знаки, используемые для решения этой задачи, называются полухрупкими.

Для обеспечения стойкости ЦВЗ к незначительным колебаниям яркости может применяться встраивание в битовую плоскость C_p при $p > 1$, а ещё лучше – использование СВИ-4 (QIM) с детектором.

Для каждого из прочих перечисленных искажений применяются специфические модификации базового метода. Например, для достижения стойкости носителя информации C^W размерами $N \times N$ к повороту на угол, кратный $\pi/2$, ЦВЗ, встраиваемый методом QIM, должен удовлетворять следующему ограничению:

$$W(n_1, n_2) = W(N - n_1, n_2) = W(n_1, N - n_2) = W(N - n_1, N - n_2). \quad (6.3)$$

Ниже мы рассмотрим систему, обеспечивающую стойкость к сжатию изображения в формате JPEG с контролируемым уровнем качества. Однако поскольку встраивание информации в этой системе тесно связано с алгоритмом JPEG-сжатия, то прежде всего необходимо остановиться на основных его этапах.

При сжатии изображений в формате JPEG цветное изображение переводится из цветового пространства RGB в YCbCr [16], где компонента Y отвечает за яркость, а Cb и Cr – за цветовую составляющую. Далее нас будет интересовать только яркостная составляющая, поэтому рассмотрим только её. Если изначально изображение является полутоновым, то две других компоненты и вовсе отсутствуют. Пусть $C(n_1, n_2)$ – яркостная компонента. Далее она разбивается на блоки $C_{ij}(n_1, n_2)$ размерами 8×8 (i и j задают положение блока в большой матрице), и на каждом из них осуществляется расчёт ДКП. Результирующие блоки обозначим $f_{ij}(m_1, m_2)$, где $0 \le m_1, m_2 < 8$. Далее все спектральные компоненты поэлементно делятся на отсчёты матрицы η_Q и округляются:

$$p_{ij}(m_1, m_2) = \left[\frac{f_{ij}(m_1, m_2)}{\eta_Q(m_1, m_2)}\right]. \quad (6.4)$$

В матрице η_Q индекс Q определяет параметр качества, представляемый целым числом в диапазоне от 1 до 100. Для любого значения Q матрица η_Q формируется на основе базовой матрицы $\eta = \eta_{50}$ по следующей формуле:

$$\eta_Q(m_1, m_2) = k(Q) \cdot \eta(m_1, m_2), \quad (6.5)$$

где

$$k(Q) = \begin{cases} \left[\frac{5000}{Q}\right], & Q < 50, \\ 200 - 2Q, & Q \geq 50. \end{cases} \quad (6.6)$$

Базовая матрица η задана в табл. 6.1.

Табл. 6.1 – Матрица квантования η в алгоритме JPEG

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Далее осуществляется архивирование полученной квантованной информации, содержащейся в матрицах $p_{ij}(m_1, m_2)$. Более подробно процедура JPEG-сжатия изложена в книге [29]. Теперь перейдём к описанию самой системы встраивания информации.

СВИ-15 (Lin & Chang)

Система полухрупких ЦВЗ, стойких к JPEG-сжатию [40]

Как и в алгоритме сжатия JPEG, контейнер C разбивается на блоки размерами 8×8, которые подвергаются дискретному косинусному преобразованию. В каждый из результирующих блоков $f_{ij}(m_1, m_2)$ встраивается 4 бита информации $b_{ij,k}$, где $k = \overline{0,3}$. Для этого множество из 28 коэффициентов $f_{ij}(m_1, m_2)$, расположенных ниже побочной диагонали, то есть множество

$$D = \{(m_1, m_2): m_1 + m_2 > 7\}, \quad (6.7)$$

разбивается на 4 равных подмножества D_k по 7 коэффициентов в соответствии с ключом встраивания $\mathbf{k}$.

Далее для встраивания каждого бита $b_{ij,k}$ необходимо выполнить следующие шаги:

1. Осуществить деление коэффициентов $f_{ij}(m_1, m_2)$, где $(m_1, m_2) \in D_k$, на элементы матрицы η (табл. 6.1):
$$f'_{ij}(m_1, m_2) = \left[\frac{f_{ij}(m_1, m_2)}{\alpha \cdot \eta(m_1, m_2)}\right], \quad (6.8)$$
где α – параметр, связанный с уровнем качества Q JPEG-сжатия, стойкость к которому требуется обеспечить.
2. Вычислить двоичное значение
$$\beta = \underset{(m_1,m_2)\in D_k}{XOR}\left(f'_{ij}(m_1, m_2)(\mathrm{mod}\ 2)\right). \quad (6.9)$$
3. Если $\beta \neq b_{ij,k}$, то инвертировать младший бит $f'_{ij}(m_1, m_2)$ у того коэффициента (m_1, m_2), которому соответствует наибольшее значение $\eta(m_1, m_2)$.
4. Умножить обратно значения $f'_{ij}(m_1, m_2)$ на элементы матрицы η.
$$f^W_{ij}(m_1, m_2) = \alpha \cdot \eta(m_1, m_2) \cdot f'_{ij}(m_1, m_2). \quad (6.10)$$

Носитель информации C^W формируется в результате применения обратного ДКП к каждому из блоков f^W_{ij}.

Извлечение информации происходит по формуле, эквивалентной (6.9):

$$b^R_{ij,k} = \underset{(m_1,m_2)\in D_k}{XOR}\left(\widetilde{f'_{ij}}(m_1, m_2)(\mathrm{mod}\ 2)\right), \quad (6.11)$$

где $\widetilde{f'_{ij}}$ получено из носителя информации $\widetilde{f^W}$ аналогично f'_{ij}.

■

6.3. Локализация изменений

В задаче аутентификации с локализацией изменений необходимо иметь возможность не только обнаруживать факт изменений, но и строить маску областей, подвергшихся модификации. Рассмотрим одну из простейших систем, предназначенных для решения этой задачи, предложенную в работе [41].

СВИ-16 (Yeung & Mintzer)

Простейшая система встраивания хрупких ЦВЗ с локализацией изменений [41]

Для встраивания и извлечения информации на основе ключа **k** формируется отображение

$$\mu: \mathbb{N}_0 \cap [0{,}255] \longmapsto \{0{,}1\}, \tag{6.12}$$

ставящее в соответствие каждому числу от 0 до 255 двоичное значение. Отображение μ обычно формируют псевдослучайным образом, стараясь избегать больших последовательностей подряд идущих чисел, отображаемых в одно и то же значение. Например, простейшим подходящим отображением является функция отыскания младшего бита:

$$\mu(x) = x(\mathrm{mod}\ 2).$$

Для встраивания формируется бинарный шаблон ЦВЗ W_r размерами $M_1 \times M_2$. В результате встраивания информации должно быть справедливо следующее условие:

$$\mu(C^W(n_1, n_2)) = W_r\big(n_1(\mathrm{mod}\ M_1), n_2(\mathrm{mod}\ M_2)\big). \tag{6.13}$$

Если это условие выполняется для некоторого пикселя (n_1, n_2) исходного контейнера C, то значение $C(n_1, n_2)$ не меняется. В противном случае находится такое $v \in \mathbb{N}_0 \cap [0{,}255]$, что

$$\mu(v) = W_r\big(n_1(\mathrm{mod}\ M_1), n_2(\mathrm{mod}\ M_2)\big) \tag{6.14}$$

и v – ближайшее к $C(n_1, n_2)$ число, удовлетворяющее этому соотношению. То есть

$$v = arg \min_{x:\ \mu(x)=W\left(n_1(\mathrm{mod}\ M_1), n_2(\mathrm{mod}\ M_2)\right)} |x - C(n_1, n_2)|. \tag{6.15}$$

Найденное значение v и будет яркостью текущего пикселя носителя информации:

$$C^W(n_1, n_2) = v. \tag{6.16}$$

Такое изменение в пикселе (n_1, n_2) порождает ошибку относительно исходного контейнера, равную $v - C(n_1, n_2)$. Для снижения визуальных последствий эта ошибка компенсируется в последующих отсчётах по методу диффузии ошибки.

Для проверки подлинности принятого носителя информации в каждой его точке проверяется условие (6.13). Очевидно, изображение будет признано неизменённым, если во всех точках условие соблюдается. Если существует ненулевое множество точек, в которых условие не соблюдается, то строится маска изменений

$$E(n_1, n_2) = \begin{cases} 1, & \mu\left(\widetilde{C^W}(n_1, n_2)\right) \neq W_r\left(n_1 (\mathrm{mod}\ M_1), n_2 (\mathrm{mod}\ M_2)\right), \\ 0 & \text{иначе.} \end{cases} \quad (6.17)$$

Недостатком алгоритма является тот факт, что по статистике половина изменённых точек будет иметь значение $E(n_1, n_2) = 0$. Отчасти он может быть компенсирован исходя из предположения о кластеризации изменённых пикселей в крупные области. В этом случае для уточнения маски E можно осуществить её постобработку, например, применив морфологическое замыкание [16].

■

В работе [21] предложена система, позволяющая осуществлять локализацию изменений при помощи полухрупких водяных знаков. В полной версии системы обеспечивается стойкость к линейному контрастированию, повороту и кадрированию (то есть вырезанию фрагмента изображения без масштабирования), однако мы рассмотрим упрощённый вариант, в котором встраиваемый ЦВЗ является хрупким.

СВИ-17 (Глумов & Митекин)

Хрупкий ЦВЗ с локализацией изменений [21]

Данная система основана на методе QIM, относительно которого произведены две модификации:

1) метод QIM применяется к блокам изображения-контейнера размерами $M \times M$;
2) встраиваемая информация модулируется бинарным шаблоном K, также имеющим размеры $M \times M$ и формируемым на основе секретного ключа системы $\mathbf{k}$.

В данной системе предполагается, что контейнер C имеет квадратный размер $N \times N$, а встраиваемый ЦВЗ W представляется бинарной матрицей, имеющей размеры $[N/M] \times [N/M]$.

Встраивание информации осуществляется по формуле

$$C^W(n_1, n_2) = \left\lfloor \frac{C(n_1, n_2)}{2q} \right\rfloor \cdot 2q + \widehat{W}(n_1, n_2) \cdot q + C(n_1, n_2) \ (\mathrm{mod}\ q), \quad (6.18)$$

где

$$\widehat{W}(n_1, n_2) = \widehat{W}(l_1 \cdot M + m_1, l_2 \cdot M + m_2) = W(l_1, l_2) \oplus K(m_1, m_2), \quad (6.19)$$

где $0 \leq l_1, l_2 < [N/L]$, $0 \leq m_1, m_2 < M$.

При извлечении информации используется следующая формула:

$$W^R(l_1,l_2)=\begin{cases}0, \text{если } \forall m_1,m_2 \\ \widetilde{C^W}(l_1\cdot M+m_1,l_2\cdot M+m_2)-qK(m_1,m_2)<q, \\ 0, \text{если } \forall m_1,m_2 \\ \widetilde{C^W}(l_1\cdot M+m_1,l_2\cdot M+m_2)-q\overline{K(m_1,m_2)}<q, \\ \text{не определено, иначе.}\end{cases} \quad (6.20)$$

В случае, если в результате извлечения ЦВЗ некоторые значения $W^R(l_1,l_2)$ не определены, то делается вывод о том, что блок изображения $\widetilde{C^W}$, соответствующий этому биту, был изменён. Таким образом, здесь в отличие от системы СВИ-16 (Yeung & Mintzer) удаётся однозначно определить маску изменений:

$$E(n_1,n_2)=\begin{cases}1, & W^R\left(\left[\frac{n_1}{M}\right],\left[\frac{n_2}{M}\right]\right) \text{ не определено}, \\ 0, & W^R\left(\left[\frac{n_1}{M}\right],\left[\frac{n_2}{M}\right]\right)\in\{0,1\}.\end{cases} \quad (6.21)$$

Однако следует помнить, что данная маска изменений дискретизирована на блоки размером $M\times M$.

■

Пример проверки данной системы приведён на рис. 6.1. Слева изображён носитель информации, в который были внесены следующие модификации:

- блок 1 изображения подвергся аддитивному зашумлению;
- блок 2 был подвергнут гауссовскому размытию;
- блок 3b был замещен блоком 3a.

В центре показана маска изменений (6.21), а справа – результат извлечения (6.20).

Рис. 6.1 – Пример работы СВИ-17: слева носитель информации с локальными искажениями, в центре маска изменений $E(n_1,n_2)$, справа извлечённый ЦВЗ (белым помечены повреждённые блоки) ([21])

Упражнения

Реализация и исследование СВИ-13 (E_MOD/D_LC)

Результатом работы будет являться скрипт `emod_run`.

1. Реализовать встраивание ЦВЗ в СВИ-13.
2. Реализовать процедуру удаления встроенного ЦВЗ и проверить её работу: сравнить исходное изображение и результат его восстановления.
3. Подобрать параметры встраивания таким образом, чтобы не менее 10 % пикселей в результате встраивания образовывали эффект шума «соль-и-перец».
4. Оценить значение линейной корреляции (5.13) полученного в предыдущем задании изображения с шаблоном ЦВЗ. Реализовать встраивание без модулярной арифметики, оценить значение линейной корреляции в этом случае.

Реализация и исследование СВИ-15 (Lin & Chang)

Результатами работы будут являться скрипт `lin_run`, а также функции:

```
[key] = lin_keygen(seed, Nb)
```

– генерация позиций анализируемых коэффициентов в соответствии с длиной встраиваемой последовательности `Nb`.

```
[CW] = lin_embed(C, b, key)
```

– встраивание информации.

```
[bR] = lin_extract(CW, key)
```

– извлечение информации.

```
[r] = lin_cmp(b, bR)
```

– сравнение встроенной последовательности и извлечённой по формуле (1.5).

1. Определить формат матрицы `key`, которую необходимо сгенерировать в функции `lin_keygen`.
2. Начать реализацию функции `lin_keygen`: определить количество блоков и индексы всех 28 анализируемых коэффициентов внутри блока.

3. Завершить реализацию функции `lin_keygen`: разбить все коэффициенты на 4 группы и сформировать итоговую матрицу.
4. Начать реализацию функции `lin_embed`: выполнить поблочное ДКП.
5. Продолжить реализацию функции `lin_embed`: создать матрицу $\eta(m_1, m_2)$ и реализовать шаг 1 (формула (6.8)).
6. Продолжить реализацию функции `lin_embed`: реализовать шаг 2 (проверку справедливости соотношения (6.9)).
7. Продолжить реализацию функции `lin_embed`: реализовать шаг 3.
8. Завершить реализацию функции `lin_embed`: реализовать шаг 4 и выполнить обратное поблочное ДКП.
9. Реализовать функцию `lin_extract`.
10. Реализовать функцию `lin_cmp`.
11. Проверить работу всего цикла встраивания и извлечения информации на двух последовательностях.
12. Определить связь между параметром алгоритма α и параметром JPEG-сжатия Q. Подобрать значение α, обеспечивающее стойкость вплоть до $Q = 75$.
13. Реализовать эксперимент: сжимать результат встраивания ЦВЗ с параметрами $Q = 50..95$ с шагом 5, извлекать ЦВЗ, каждый раз оценивая точность извлечения. Построить график.

Реализация и исследование СВИ-16 (Yeung & Mintzer)

Результатами работы будут являться скрипт `yeung_run`, а также функции:

`[mapping] = yeung_genmapping(seed, M)`
– генерация вектора, задающего отображение (6.12), по ключу `seed`.

`[CW] = yeung_embed(C, Wr, mapping)`
– встраивание информации.

`[E, uCW] = yeung_extract(CW, mapping)`
– извлечение информации: `E` – маска изменений (6.17), а `uCW` – результат $\mu\left(\widetilde{C^W}(n_1, n_2)\right)$.

1. Реализовать функцию `yeung_genmapping`, генерирующую вектор, задающий отображение (6.12).

2. Начать написание функции `yeung_extract`, а именно – отыскание матрицы `uCW`. В скрипте `yeung_run` считать входное изображение и логотип, попробовать осуществить извлечение ЦВЗ по исходному контейнеру (получить и визуализировать `uCW`).
3. Реализовать упрощённое встраивание информации (функция `yeung_embed`) путём замены значений C^W на меньшие значения, формирующие нужный бит ЦВЗ при отображении. Не применять диффузию ошибки.
4. Завершить реализацию функции `yeung_extract`. В скрипте встроить ЦВЗ, рассчитать PSNR и долю ошибочно извлечённых пикселей.
5. Реализовать полноценный поиск ближайших подходящих значений C^W при встраивании информации. В скрипте встроить ЦВЗ, рассчитать PSNR и долю ошибочно извлечённых пикселей. Сравнить с предыдущим результатом.
6. В скрипте `yeung_run` внести локализованные изменения в носитель информации: размытие фрагмента.
7. В скрипте `yeung_run` внести локализованные изменения в носитель информации: замену одного фрагмента носителя информации другим фрагментом.
8. В скрипте `yeung_run` внести локализованные изменения в носитель информации: замену фрагмента фрагментом другого изображения.
9. [На доске] Написать аналитические выражения, определяющие алгоритм диффузии ошибки (pull-модель) в форме, подходящей для СВИ-16 (Yeung & Mintzer) (используя (4.16)–(4.18)).
10. [На доске] Написать аналитические выражения, определяющие алгоритм диффузии ошибки (push-модель) в форме, подходящей для СВИ-16 (Yeung & Mintzer) (используя (4.19)–(4.22)).
11. Реализовать любой алгоритм диффузии ошибки с ядром Floyd & Steinberg (4.12) для корректировки процедуры встраивания информации. В скрипте встроить ЦВЗ, рассчитать PSNR и долю ошибочно извлечённых пикселей. Сравнить с предыдущими аналогичными результатами.

Реализация и исследование СВИ-17 (Глумов & Митекин)

Результатами работы будут являться скрипт `glu_run`, а также функции:

`[K] = glu_genkey(seed)`
– генерация бинарного шаблона K размерами $M \times M$ по ключу `seed`.

`[CW] = glu_embed(C, W, K, q)`
– встраивание информации.

`[E, WR] = glu_extract(CW, K, q)`
– извлечение информации: `E` – маска изменений (6.17), а `WR` – результат извлечения ЦВЗ.

1. Считать изображение и логотип для встраивания. Определить по их размерам величину M. Реализовать функцию `glu_genkey` и выполнить её.
2. Реализовать функцию `glu_embed` и проверить её работу.
3. Реализовать фунцию `glu_extract` и проверить её работу.
4. В скрипте `glu_run` внести локализованные изменения в носитель информации: размытие фрагмента. Проверить результат извлечения.
5. В скрипте `glu_run` внести локализованные изменения в носитель информации: замену одного фрагмента носителя информации другим фрагментом. Проверить результат извлечения.
6. Определить, какие яркостные искажения не приведут к уничтожению встроенного ЦВЗ.
7. Реализовать встраивание и извлечение найденных в предыдущем задании искажений.
8. Изменить процедуру `glu_genkey`, чтобы обеспечить стойкость к повороту на угол, кратный 90°.
9. Проверить стойкость ЦВЗ к повороту на 90°.

7. Встраивание информации в видеосигналы

7.1. Отличия и особенности СВИ в видео

Предыдущие главы были посвящены изучению различных систем встраивания информации в изображения. В данной главе будут рассмотрены методы, в которых контейнером является видеосигнал.

Области применения методов встраивания информации в видео по большей части те же самые: защита авторских прав, защита от несанкционированного распространения, защита от изменений, скрытая передача информации. Однако появляются и две специфических для видео задачи: контроль копирования и мониторинг телевещания.

Задача контроля копирования заключается в разработке и применении комплексных систем, использующих аппаратные криптографические решения и технологии водяных знаков для воспрепятствования несанкционированному копированию лицензионных дисков с видеоданными, таких как DVD и Blu-ray. Интерес здесь представляют не собственно методы встраивания информации, а сценарии их совместного использования вместе с иными средствами защиты. Эти вопросы рассматриваются в книгах [1, 2], мы же на них останавливаться не будем.

Задача мониторинга вещания актуальна, в частности, для рекламодателей, заказывающих показ их рекламного ролика на телевидении определённое число раз в день и желающих убедиться в точном соблюдении вещателем заключённого контракта. В этом случае рекламодатель будет нуждаться в системе, принимающей видеосигнал в реальном времени и увеличивающий счётчик показов каждый раз, когда обнаружился искомый рекламный ролик. Корреляция видеосигналов в реальном времени является трудоёмкой процедурой. Поэтому вместо этого в рекламный ролик может предварительно внедряться водяной знак, тогда при анализе видеопотока будет постоянно осуществляться попытка извлечения ЦВЗ. Такой подход может оказаться более подходящим для систем реального времени.

В отличие от изображений, которые зачастую могут храниться в формате без потерь, цифровые видео почти всегда сжимаются. Поэтому методы встраивания информации в видео должны быть стойкими к стандартным методам сжатия. Помимо этого, методы встраивания должны

быть стойкими к обрезке или прореживанию видео по временной оси. С другой стороны, как правило, нет необходимости добиваться стойкости встроенной информации к сложным геометрическим искажениям кадров. Ещё одно «облегчение», возникающее при использовании видео в качестве контейнера, заключается в допустимости бóльших по амплитуде искажений для каждого кадра ввиду кратковременности их просмотра. Это, в свою очередь, позволяет повысить точность извлечения встроенной информации.

Существует два основных подхода к встраиванию информации в видео. В первом видео рассматривается как набор независимых кадров (изображений), и в каждый кадр встраиваются одни и те же данные. Такой подход позволяет обеспечить стойкость к потере синхронизации (изменениям по временной оси). Однако в этом случае объем данных, который может быть встроен в контейнер, ограничивается не продолжительностью видео, а разрешением кадра. Таким образом, этот подход не позволяет встроить большой объем информации. Кроме того, некоторые методы этой группы подвержены так называемой атаке с «приближённым вычислением ЦВЗ» (“watermark estimation attack”) [42], которая заключается в оценке сигнала ЦВЗ за счёт усреднения большого числа кадров видео с целью его удаления или восстановления встроенной информации.

Во втором подходе видео рассматривается как набор строго упорядоченных кадров, и встраиваемая информация распределяется между многими кадрами по некоторому правилу. При этом объем встраиваемой информации становится пропорционален продолжительности видео, но встроенная информация становится более уязвимой для атак, связанных с потерей синхронизации.

В данной главе мы рассмотрим две системы, реализующие второй подход и не содержащие в базовом варианте средств защиты от десинхронизирующих атак: это система Hartung & Girod [43], позволяющая встроить очень большой объём данных и, следовательно, подходящая главным образом для задачи стеганографии, а также ЦВЗ-система JAWS, предложенная в работе [44] для мониторинга вещания. А далее мы опишем универсальный подход, позволяющий противостоять атакам потери синхронизации за счёт корректировки встраиваемой информации [45, 46].

7.2. Примеры СВИ в видео

СВИ-18 (Hartung & Girod)

Система встраивания информации в видео с расширением спектра [43]

Встраивание информации

Внутренняя информация представляется в форме $\mathbf{b} \in \mathbb{B}^1_{[N_b]}$ и встраивается в видеосигнал $C \in \mathbb{Z}^3_{[N_1 \times N_2 \times T]}$ ($N_1 \times N_2$ – размеры кадра, а T – число кадров видео) в соответствии с одномерной покадровой построчно-столбцовой развёрткой

$$\varphi(n): n \mapsto (n_1, n_2, t),$$

где $n = \overline{0, N-1}$, $N = N_1 N_2 T$, $n_1 = \overline{0, N_1 - 1}$, $n_2 = \overline{0, N_2 - 1}$, $t = \overline{0, T-1}$. Таким образом, признаки контейнера описываются вектором $f \in \mathbb{Z}^1_{[N]}$:

$$f(n) = C\big(\varphi(n)\big). \tag{7.1}$$

Перед встраиванием осуществляется кодирование информации в пространстве признаков по формуле

$$\Omega(n) = (-1)^{b_i} \text{ для } i \cdot L \le n < (i+1) \cdot L, \tag{7.2}$$

где $L \in \mathbb{N}$ – параметр, характеризующий избыточность встраивания, $i = 0..N_b - 1$, а $n = 0..\min(N_b \cdot L, N) - 1$. Если $N_b \cdot L < N$, то в оставшуюся часть сигнала встраивание не производится.

Встраивание информации осуществляется по формуле

$$f^W(n) = f(n) + \alpha \cdot \lambda(n) \cdot \Omega(n) \cdot (-1)^{k_n}, \tag{7.3}$$

где k_n – n-й бит ключа встраивания $\mathbf{k} \in \mathbb{B}^1_{[N_b]}$, являющегося псевдослучайной двоичной последовательностью длины N_b, $\alpha > 0$ – постоянный множитель при встраиваемом сигнале, а $\lambda(n) > 0$ – множитель при встраиваемом сигнале, адаптивный к локальным особенностям контейнера и меняющийся слабо, настолько, что можно принять, что

$$\forall i \in [0, N_b - 1]\ \forall n \in [i \cdot L, (i+1) \cdot L - 1] \quad \lambda(n) \approx \bar{\lambda}_i, \tag{7.4}$$

где

$$\bar{\lambda}_i = \frac{1}{L} \sum_{n=i \cdot L}^{(i+1) \cdot L - 1} \lambda(n). \tag{7.5}$$

Извлечение информации

При извлечении встроенной информации используется слепой метод, не предполагающий знания исходного контейнера. Результатом яв-

ляется отыскание $\mathbf{b}^{\mathbf{R}} \in \mathbb{B}^1_{[N_b]}$. Оценка матрицы признаков извлечённой информации осуществляется по формуле

$$\tilde{\Omega}(n) = (-1)^{k_n} \cdot h^W(n), \tag{7.6}$$

где h^W – вспомогательная величина, которая подбирается таким образом, чтобы было справедливо приближённое равенство

$$h^W(n) \approx f^W(n) - f(n). \tag{7.7}$$

Поскольку на стадии извлечения информации не известен истинный сигнал-контейнер, то вместо его матрицы признаков $f(n)$ используется оценка $f^W_{mean,S}(n)$ – усреднённый в скользящем окне шириной $S \geq 3$ вектор признаков $f^W(n)$. Таким образом, h^W вычисляется по формуле

$$h^W(n) = f^W(n) - f^W_{mean,S}(n). \tag{7.8}$$

Значение очередного бита b_i^R определяется на основе анализа величины

$$\beta_i = \mathcal{P}_f^{-1}(\tilde{\Omega}) = \sum_{n=i\cdot L}^{(i+1)\cdot L-1} \tilde{\Omega}(n). \tag{7.9}$$

Из (7.9), (7.6) и (7.7) получаем, что

$$\beta_i \approx \sum_{n=i\cdot L}^{(i+1)\cdot L-1} \alpha \cdot \lambda(n) \cdot \Omega(n) \cdot (-1)^{2k_n} = \alpha \cdot L\bar{\lambda}_i \cdot (-1)^{b_i}. \tag{7.10}$$

Поскольку $\alpha L\bar{\lambda}_i$ – величина положительная, то справедливо простое правило извлечения встроенной информации:

$$b_i^R = \begin{cases} 0, & \beta_i > 0, \\ 1, & \beta_i < 0. \end{cases} \tag{7.11}$$

■

СВИ-19 (JAWS)

ЦВЗ-система для мониторинга вещания [44]

JAWS является аббревиатурой от Just Another Watermarking System – такое название дал своей системе автор в работе [44].

Для встраивания цифрового водяного знака используется шаблон P_r размерами $M \times M$, представляющий собой реализацию гауссовского шума, имеющего нормальное распределение. Данный шаблон формируется при помощи ключа $\mathbf{k}$. Далее для каждого кадра t генерируется уникальный шаблон ЦВЗ по формуле вида

$$W_{r,t}(m_1, m_2) = P_r(m_1, m_2) - shift(P_r, \mathbf{b_t}), \tag{7.12}$$

где $m_1, m_2 \in [0, M-1]$, $\mathbf{b_t}$ – фрагмент встраиваемой последовательности $\mathbf{b}$, содержащий биты, встраиваемые в кадр t, а $shift(P_r, \mathbf{b_t})$ обозначает операцию циклического сдвига строк и столбцов P_r в соответствии с битами $\mathbf{b_t}$.

Встраивание происходит по аддитивной формуле

$$C^W(n_1, n_2, t) = C(n_1, n_2, t) + \alpha \cdot \beta(n_1, n_2, t) \cdot \cdot W_{r,t}\big(n_1 (\mathrm{mod}\, M), n_2 (\mathrm{mod}\, M)\big), \quad (7.13)$$

где α – параметр глобального усиления встраиваемого сигнала, $\beta(n_1, n_2, t)$ – маска адаптивного усиления встраиваемого сигнала, рассчитываемая при помощи оператора Лапласа, то есть свёртки (8.17) кадра $C(n_1, n_2, t)$ с маской вида

$$g(n_1, n_2) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (7.14)$$

с последующим взятием модуля полученного поля. Пример подобного расчёта $\beta(n_1, n_2, t)$ для отдельного изображения показан на рис. 7.1.

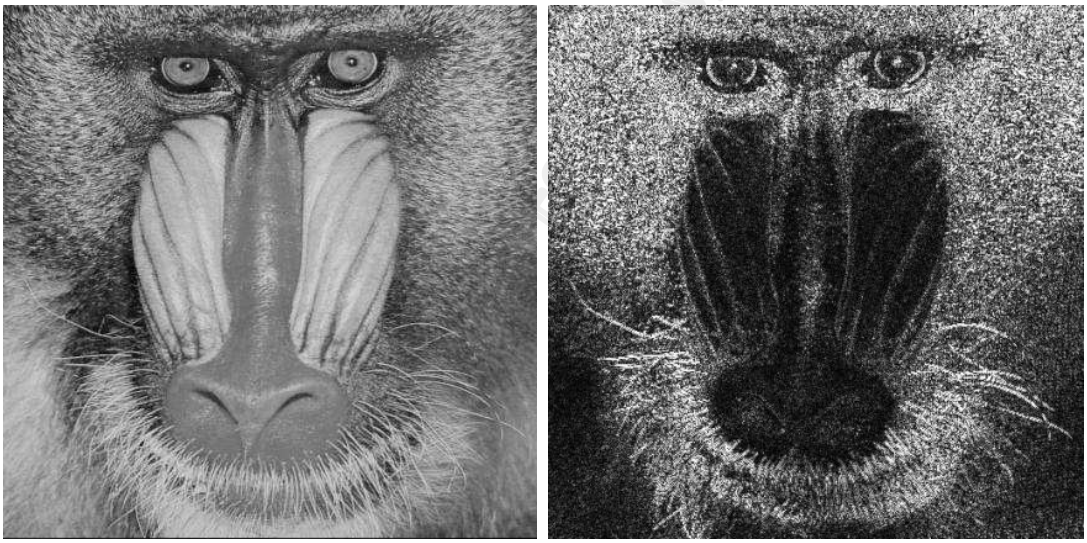


Рис. 7.1 – Полутоновое изображение и его маска адаптивного усиления, рассчитываемая в СВИ-19 (JAWS)

При извлечении информации сначала формируется оценка встроенного шумоподобного сигнала путём усреднения отсчётов в блоках размерами $M \times M$:

$$\widetilde{W}_t(m_1, m_2) = \sum_{i=0}^{[N_1/M]+1} \sum_{j=0}^{[N_2/M]+1} \widetilde{C^W}(i \cdot M + m_1, j \cdot M + m_2, t), \quad (7.15)$$

после чего формируется взаимная корреляционная функция (ВКФ) $\widetilde{W_t}(m_1, m_2)$ и $P_r(m_1, m_2)$, рассчитываемая путём перемножения их спектров [47]

$$B = \mathcal{F}^{-1}\left(\mathcal{F}\left(\widetilde{W_t}\right) \cdot conj\left(\mathcal{F}(P_r)\right)\right), \tag{7.16}$$

где $\mathcal{F}(x)$ означает расчёт двумерного ДПФ (5.7) для x с использованием алгоритма быстрого преобразования Фурье [47]; $\mathcal{F}^{-1}(x)$ – расчёт обратного ДПФ.

Другой похожий способ, предлагаемый автором данной системы в работе [48], заключается в применении так называемой SPOMF-фильтрации (Symmetrical Phase Only Matched Filtering) вместо расчёта ВКФ, в которой перемножаются нормированные спектры:

$$B^* = \mathcal{F}^{-1}\left(\phi\left(\mathcal{F}\left(\widetilde{W_t}\right)\right) \cdot conj\left(\phi\left(\mathcal{F}(P_r)\right)\right)\right), \tag{7.17}$$

$$\phi(x) = \frac{x}{|x|}. \tag{7.18}$$

Далее на полученном корреляционном поле отыскиваются два пика: один положительный, координаты которого задают сдвиг шаблона P_r, а другой – отрицательный, координаты которого задают сдвиг шаблона $shift(P_r, \mathbf{b_t})$ (согласно формуле (7.12)). Таким образом, вектор между двумя этими пиками будет кодировать встроенную информацию $\mathbf{b_t}$.

Благодаря избыточному встраиванию и использованию коррелятора при извлечении информации, встроенный ЦВЗ оказывается стойким не только к пережатию видеофайла, но и к обрезке кадра.

■

7.3. Метод противодействия атакам потери синхронизации

Рассмотренные выше системы не являются стойкими к потере временной синхронизации, то есть сдвигу начала видео или изменению числа кадров в секунду. Однако в работах [45, 46] предложен способ предварительного кодирования встраиваемой информации, позволяющий противодействовать подобной атаке.

Пусть $\mathbf{b}$ – последовательность бит встраиваемой информации, состоящая из L непересекающихся фрагментов длиной N_b/L бит каждый. j-й бит i-го фрагмента $\mathbf{b}$ обозначим как $b_{i,j}, i \in [0, L-1], j \in [0, N_b/L - 1]$. В

каждый кадр исходного видео встраивается одна из L битовых последовательностей $\mathbf{s_i}$, каждая из которых состоит из $N_i + N_b/L$ бит, где

$$N_i = \lceil \log_2 L \rceil + 1, \tag{7.19}$$

$\lceil x \rceil$ обозначает операцию округления в бóльшую сторону.

Битовые последовательности $\mathbf{s_i}$ формируются по следующему правилу:

$$\mathbf{s_i} = \underbrace{i_0 i_1 \dots i_{N_i-1}}_{N_i \text{ бит}} \underbrace{b_{i,0} b_{i,1} \dots b_{i,N_b/L-1}}_{N_b/L \text{ бит}}, \tag{7.20}$$

где $i_0 i_1 \dots i_{N_i-1}$ – бинарное представление индекса i, а $b_{i,0} b_{i,1} \dots b_{i,N_b/L-1}$ – i-й фрагмент встраиваемой последовательности.

Далее при встраивании информации для каждого кадра псевдослучайным образом выбирается одна из последовательностей $\mathbf{s_i}, i \in [0, L-1]$, которая встраивается в кадр видео. Единственным требованием к используемому алгоритму в данном случае является возможность встраивания и слепого извлечения не менее чем $N_i + N_b/L$ бит информации.

Такой подход позволяет защититься от возможной потери синхронизации видео без использования дополнительной информации об исходной нумерации кадров.

Упражнения

Реализация и исследование СВИ-18 (Hartung & Girod)

Результатами работы будут являться скрипт `hartung_run`, а также функции:

`[k] = hartung_keygen(seed, Nb)`
– генерация ключа.

`[CW] = hartung_embed(C, b, k, L)`
– встраивание информации.

`[bR] = hartung_extract(CW, k, L)`
– извлечение информации.

`[r] = hartung_cmp(b, bR)`
– сравнение встроенной последовательности и извлечённой по формуле (1.5).

1. Реализовать функцию генерации ключа `hartung_keygen`.
2. Реализовать скрипт, осуществляющий чтение, запись и отображение файла видео.
3. Начать реализацию `hartung_embed`: подготовить вектор признаков контейнера, вектор признаков встраиваемой информации.
4. Завершить реализацию `hartung_embed`: сгенерировать $\lambda(n)$ как усреднение в окне размером L_λ.
5. Завершить реализацию `hartung_embed`: осуществить аддитивное встраивание информации.
6. Получить формулу расчёта β_i, используемую для извлечения информации.
7. Реализовать функцию извлечения информации `hartung_extract`.
8. Реализовать функцию `hartung_cmp`. Проверить выполнение процедур встраивания и извлечения информации.
9. Проанализировать зависимость точности извлечения от параметра S.
10. Сравнить полученные результаты с результатами, полученными путём неслепого извлечения.
11. Проанализировать зависимость точности извлечения от параметров L_λ и L.

Реализация и исследование СВИ-19 (JAWS)

Результатами работы будут являться скрипт `jaws_run`, а также функции:

`[Pr] = jaws_keygen(seed, M)`
– генерация ключа.

`[Wr] = jaws_shift(Pr, bits)`
– генерация ключа.

`[CW] = jaws_embed(C, Wr)`
– встраивание информации в отдельный кадр видео `C`.

`[WR] = jaws_w_estimation(CW, M)`
– оценка встроенного сигнала по отдельному кадру видео.

`[bR] = jaws_extract(WR, method)`
– извлечение информации из отдельного кадра видео. `method` определяет используемый метод: SPOMF или ВКФ.

`[r] = jaws_cmp(b, bR)`
– сравнение встроенной последовательности и извлечённой по формуле (1.5).

1. Реализовать функцию генерации ключа `jaws_keygen`.
2. Выбрать способ кодирования встраиваемой информации циклическими сдвигами (функция $shift$).
3. Начать реализацию функции `jaws_shift`: реализовать циклические сдвиги по вертикали, кодирующие встраиваемую информацию.
4. Завершить реализацию функции `jaws_shift`: реализовать циклические сдвиги по горизонтали, кодирующие встраиваемую информацию.
5. Проверить работоспособность функции `jaws_shift`.
6. Реализовать обработку заданного изображения оператором Лапласа.
7. Реализовать функцию `jaws_embed`.
8. Реализовать функцию `jaws_w_estimation`.
9. Начать реализацию функцию `jaws_extract`: построить поле B с использованием ВКФ.

10. Начать реализацию функцию `jaws_extract`: построить поле B с использованием SPOMF.
11. Завершить реализацию функцию `jaws_extract`: извлечь встроенную битовую последовательность.
12. Реализовать функцию `jaws_cmp`. Проверить работоспособность метода при встраивании на одном кадре.
13. Проверить влияние размера шаблона на точность извлечения.
14. Реализовать цикл встраивания информации в кадры видео.

8. Атаки на системы встраивания информации

Ранее в параграфе 1.2 были перечислены основные виды атак на системы встраивания информации. Данная глава предназначена для обретения практических навыков по исследованию стойкости СВИ к различного рода атакам и непреднамеренным искажениям.

8.1. Проверка стойкости ЦВЗ-систем

Как отмечалось в главе 1, под стойкостью ЦВЗ-систем понимается возможность корректного извлечения встроенной информации из носителя, который подвергался некоторым искажениям. Иными словами, стойкость ЦВЗ-системы характеризует простоту удаления встроенной информации. Круг возможных искажений, которые могут быть теоретически применены над носителем информации, весьма широк.

Исследование стойкости может осуществляться по одному из двух популярных сценариев. В первом случае проверяется стойкость системы к некоторому множеству преобразований, круг которых главным образом определяется областью применения СВИ, её свойствами и соображениями здравого смысла. То есть это должны быть такие преобразования, которые могут произойти с носителем информации в рамках его использования и которые не нарушают его целостности. Например, если носитель информации представляет собой цветное фотографическое изображение высокого качества, предназначенное для передачи и публикации в цифровом виде, то нет смысла проверять его стойкость к печати-сканированию, поскольку качество результирующего изображения не позволит его полноценно использовать. Также нецелесообразно проверять стойкость системы ЦВЗ для видео к покадровому повороту, поскольку это преобразование нехарактерно для большинства сценариев использования видеофайлов. Рассмотренный сценарий называют *проверкой стойкости к непреднамеренным искажениям носителя информации*. Водяные знаки, сохраняющиеся по результатам данной атаки, принято называть стойкими.

Второй сценарий также предполагает сохранение целостности носителя информации после искажения, но методы преобразования могут быть нестандартными, специально подобранными с целью удаления ЦВЗ. В данном сценарии алгоритм встраивания ЦВЗ предполагается извест-

ным, и он определяет способ искажения. Общая процедура в этом случае называется *проверкой стойкости ЦВЗ к преднамеренным атакам*. Алгоритмы, успешно прошедшие проверку определённой атакой данного типа, называются стойкими к данной атаке. ЦВЗ-системы, стойкие ко всем известным атакам данного типа, иногда называются секретными [1].

Круг непреднамеренных искажений, традиционно рассматриваемых для СВИ в изображения, включает:

- поэлементные изменения функции яркости (контрастирование, цветовая коррекция и др.);
- зашумление (аддитивное и импульсное, различные параметры шума и формы АКФ);
- линейная фильтрация (сглаживание, повышение резкости, нерезкая маска и др.);
- нелинейная фильтрация (медианная, ранговая фильтрация и пр.);
- геометрические искажения (поворот, масштабирование, сдвиг, проективное преобразование и пр.);
- потеря части пространственных данных (обрезка, дублирование фрагмента изображения, замена части отсчётов отсчётами другого изображения);
- сжатие с потерями (в форматах JPEG, JPEG-2000 и пр.);
- печать-сканирование;
- повторное встраивание другого ЦВЗ тем же алгоритмом.

Для систем встраивания информации в видео добавляются всевозможные изменения по оси времени: изменение битрейта, вырезание фрагмента по времени, пропуск отдельных кадров; расширяется список форматов сжатия с потерями. В то же время актуальность теряют геометрические искажения, печать-сканирование. Подробную информацию по всем перечисленным преобразованиям можно найти в книгах [47, 16, 29, 49].

Сценарий проверки стойкости ЦВЗ-систем к непреднамеренным искажениям предлагается к реализации в лабораторной работе 3. Второй сценарий, заключающийся в реализации специфических атак на конкретные ЦВЗ-системы, рассматривается в упражнениях к настоящей главе.

8.2. Методы стегоанализа

Задача и направления стегоанализа

Под *стегоанализом* обычно понимается атака на стеганографические системы, целью которой является обнаружение канала скрытой передачи информации. Также обычно выделяют *целенаправленный стегоанализ* (target steganalysis), при проведении которого считаются известными используемые стеганографические методы и протоколы, и *слепой стегоанализ*, не ориентированный на какие-либо методы.

Поскольку результатом проведения стегоанализа для какого-либо цифрового носителя информации является бинарный ответ: есть встраивание или нет, – то в сущности задача стегоанализа может быть сведена к задаче классификации объекта на два соответствующих класса. В этом случае её решение будет включать два этапа:

1) выбор информативных признаков;
2) классификация векторов признаков с обучением.

На втором этапе может использоваться любой известный классификатор. Выбор конкретного решения может зависеть от характера векторов признаков, их длины, разделимости, количества имеющихся для обучения данных и прочих факторов. Этот материал выходит за рамки нашего курса, однако может быть изучен самостоятельно по книгам [50, 51, 52], электронному ресурсу [53] или в рамках учебных курсов машинного обучения или распознавания образов. Наиболее часто используемые классификаторы (в том числе и в задачах стегоанализа) – линейный и квадратичный дискриминантный анализ, байесовский классификатор, машины опорных векторов, деревья решений и пр.

Обзор простых признаков для НЗБ-стегоанализа

Рассмотрим некоторые популярные методы выбора признаков для решения задачи стегоанализа методов стеганографического встраивания информации в наименее значимые биты полутоновых изображений, а именно собственно НЗБ-встраивания (СВИ-2) и ±1-встраивания (СВИ-3), рассмотренных в главе 3. Все они используют следующее предположение: стеганографическое встраивание разрушает корреляционные связи между соседними отсчётами цифрового сигнала – контейнера. Таким образом, эти признаки должны отражать коррелированность сигнала в пространстве сокрытия.

Первый способ расчёта признаков основывается на расчёте среднего значения и среднего числа переходов в битовой плоскости в скользящем окне. Пусть $C_p^W(n_1, n_2)$ – анализируемая битовая плоскость.

Формула свёртки C_p^W с КИХ-фильтром $g(m_1, m_2)$ размерами $M \times M$ имеет вид:

$$S(n_1, n_2) = C_p^W ** g = \\ = \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} g(m_1, m_2) \cdot C_p^W(n_1 - m_1, n_2 - m_2). \quad (8.1)$$

Локальное среднее тогда рассчитывается как

$$\mu = C_p^W ** g_\mu, \quad (8.2)$$

где для g_μ $M = 2p + 1, p \in \mathbb{N}$, причём отсчёты ИХ постоянны и равны $1/M^2$.

Среднее число переходов бита (в совокупности по горизонтали и вертикали) рассчитывается в несколько этапов:

$$\tau_{hor} = C_p^W ** g_{hor}, \quad (8.3)$$

$$\tau_{ver} = C_p^W ** g_{ver}, \quad (8.4)$$

$$\tau = \left(\frac{1}{2}(|\tau_{hor}| + |\tau_{\text{ver}}|)\right) ** g_\mu, \quad (8.5)$$

где

$$g_{hor} = (-1 \quad 1), \quad g_{\text{ver}} = (g_{hor})^{\text{T}}. \quad (8.6)$$

Разумеется, и μ, и τ являются матрицами значений, по которым, в свою очередь, могут рассчитываться скалярные признаки, такие как:

- среднее;
- дисперсия;
- наибольшее значение;
- наименьшее значение;
- разность наибольшего и наименьшего значений.

Любая комбинация полученных чисел может далее использоваться в качестве вектора признаков, построенного путём анализа среднего значения и среднего числа переходов.

Второй способ предполагает развёртку двумерной битовой плоскости в одномерную последовательность нулей и единиц с последующим расчётом некоторых её статистических характеристик.

При одномерной развёртке близкие на плоскости пиксели должны располагаться как можно ближе в результирующей последовательности.

Наиболее часто используются три развёртки: построчная, серпантинная, а также развёртка Пеано (или Гильберта-Пеано) [54]. Все они проиллюстрированы на рис. 8.1. Развёртка Гильберта-Пеано строится по рекуррентной формуле для областей, размеры которых являются целыми степенями двойки, путём склеивания четырёх шаблонов развёртки области предыдущей степени [55]. Таким образом, данная развёртка постоянно меняет своё направление, охватывая близлежащие пиксели в обоих измерениях. Очевидно, что последние две развёртки имеют преимущество по сравнению с построчной, поскольку не имеют разрывов.

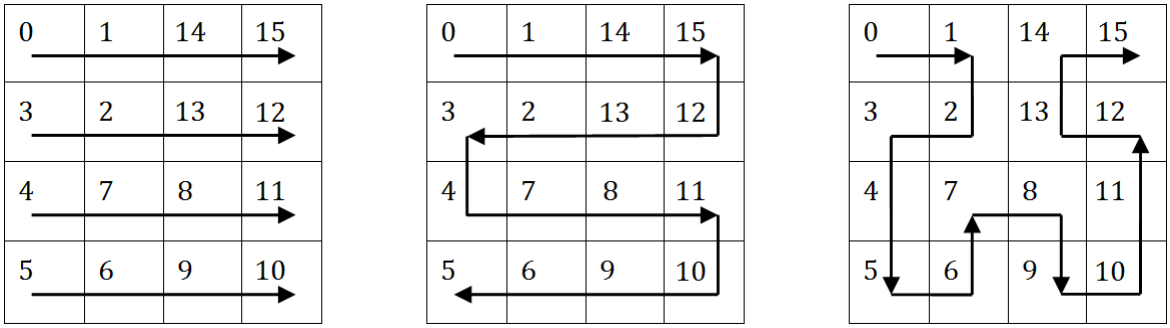


Рис. 8.1 – Развёртки двумерной области 4×4: построчная (слева), серпантинная (в центре), Гильберта-Пеано (справа)

Обозначим полученную последовательность $\{\beta_k\}_{k=0..N-1}$, где $N = N_1 N_2$.

Первый вариант её дальнейшего использования заключается в расчёте относительной частоты переходов между соседними отсчётами последовательности:

$$\pi_{00} = \frac{1}{N-1}\sum_{k=0}^{N-2}\gamma_k^{00}, \tag{8.7}$$

где

$$\gamma_k^{00} = \begin{cases} 1, & (\beta_k = 0) \wedge (\beta_{k+1} = 0), \\ 0, & \text{иначе.} \end{cases} \tag{8.8}$$

По аналогичным формулам находятся также π_{01}, π_{10}, π_{11}, в совокупности образуя вектор из четырёх признаков:

$$(\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11}). \tag{8.9}$$

В случае заполненного контейнера частоты переходов должны быть достаточно близкими, в то время как в пустом контейнере частоты переходов из 0 в 0 и из 1 в 1 значительно превышают частоты переходов двух других видов. На рис. 8.2 показан пример диаграммы частот переходов для разных битовых плоскостей пустого контейнера в сравнении с

заполненной битовой плоскостью, рассчитанных по последовательности, полученной при помощи построчной развёртки. Статистика пустого контейнера считалась по изображению “Lenna” (рис. 4.1а).

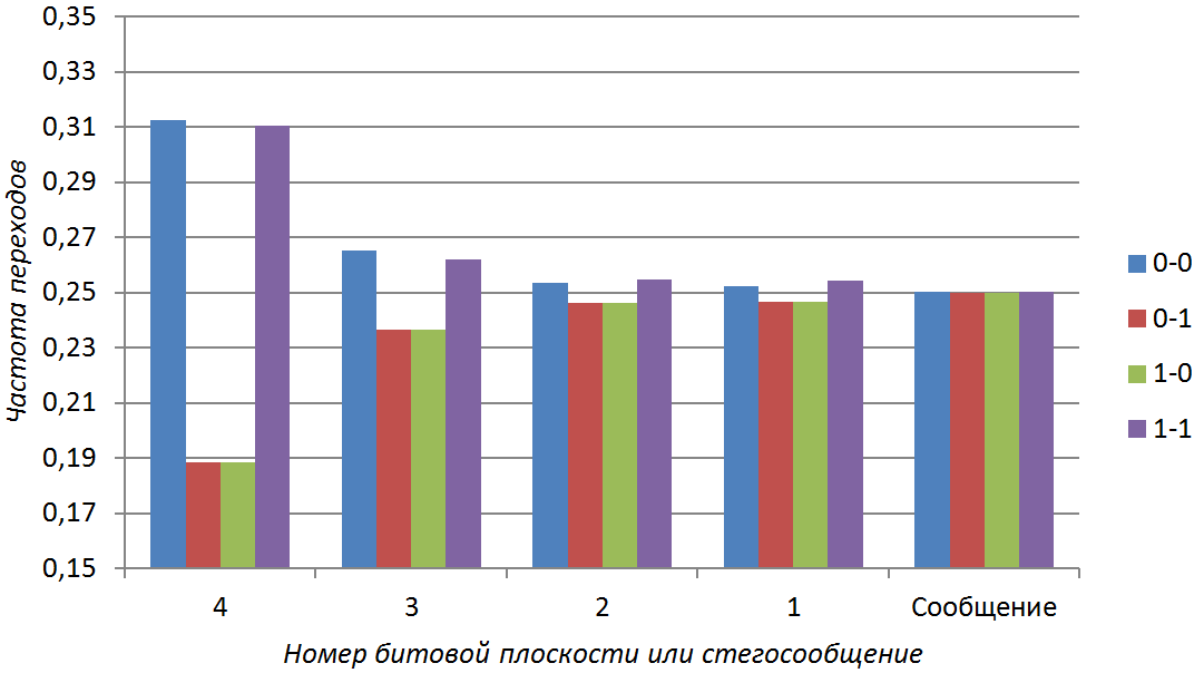


Рис. 8.2 – Диаграмма частоты переходов для пустого и заполненного контейнера

Другой способ формирования признаков по двоичной последовательности – расчёт числа серий разной длины. Серией является фрагмент последовательности, состоящий из одинаковых значений (неважно, единиц или нулей) и ограниченный другими значениями или границей последовательности. Достаточно информативной характеристикой последовательности является статистика, отражающая число серий различной длины. Будем обозначать её $\{s_i\}_i$, где $i > 0$ – длина серии. Иногда эту статистику нормируют на длину последовательности N, чтобы получить значения, не зависящие от объёма контейнера:

$$\{v_i\}_i = \left\{\frac{s_i}{N}\right\}_i. \tag{8.10}$$

В табл. 8.1 приведён пример статистики числа серий последовательностей, полученных при помощи построчной развертки первой битовой плоскости пустого и заполненного контейнера. Статистика пустого контейнера считалась по изображению “Lenna” (рис. 4.1а).

Табл. 8.1 – Пример статистики числа серий в пустом и заполненном контейнере

i	Число серий s_i		i	Число серий s_i	
	Пустой контейнер	Заполненный контейнер		Пустой контейнер	Заполненный контейнер
1	64097	65363	12	48	37
2	32462	32741	13	14	14
3	16143	16596	14	7	11
4	8124	8131	15	7	1
5	4155	4093	16	6	3
6	2075	2054	17	3	1
7	1076	964	18	1	0
8	584	539	19	1	0
9	320	245	20	1	0
10	169	138	21	0	0
11	94	63	22	0	0

Как видно из таблицы, число серий малой длины в заполненном контейнере превышает число серий в пустом контейнере, но начиная с некоторого значения i статистика по пустому контейнеру становится выше. Если рассматривать очень большие серии – длиной в несколько десятков отсчётов, то в заполненном контейнере таковые почти всегда отсутствуют, в то время как в пустом время от времени могут появляться.

Наиболее простой способ формирования вектора признаков по статистике числа серий – выбор в качестве признаков некоторого количества

$$\{s_i\}_{i\in I}. \tag{8.11}$$

При этом множество I также может формироваться различными способами. Некоторым недостатком такого подхода является существенное различие в абсолютных значениях s_i для разных длин i. Эта проблема может решаться путём корректировки векторов признаков либо априори (то есть путём подбора таких $\{k_i\}_{i\in I}$, что величины $\{s_i k_i\}_{i\in I}$ имеют примерно один порядок), либо на основе обучающей выборки путём нормировки векторов признаков.

Рассмотренные признаки просты для изучения, но не слишком хороши для используемых на практике стеганографических методов. Даже простые модификации системы НЗБ-встраивания позволяют обеспечить стойкость некоторым из рассмотренных признаков. Поэтому учёными были разработаны более эффективные признаки для НЗБ-стегоанализа, такие как HCF [56], ALE [57] и др. Более подробно данный материал рассмотрен в работах [58, 57] и книгах [2, 6].

Другие методы НЗБ-стегоанализа

Одним из самых простых и наиболее известных методов целенаправленного стегоанализа НЗБ-встраивания является метод гистограмм пар значений.

Данный метод стегоанализа использует расчёт статистики хи-квадрат для проверки гипотезы о виде распределения яркости контейнера. Пусть для простоты проверяется наличие встраивания информации в первую битовую плоскость. Тогда теоретически значения яркости, отличающиеся только младшим битом (0 и 1, 2 и 3, 4 и 5,…), должны быть равновероятны. Таким образом, метод заключается в расчёте эмпирической гистограммы анализируемого изображения $h_i^e, i = 0..255$, а также соответствующей ей теоретической:

$$h_i^t = \frac{h^e_{2\cdot[i/2]} + h^e_{2\cdot[i/2]+1}}{2}. \tag{8.12}$$

На рис. 8.3 показан пример эмпирической и теоретической гистограмм. Соответствие эмпирической гистограммы теоретической проверяется посредством расчёта статистики хи-квадрат для чётных отсчётов гистограммы:

$$\chi^2 = \sum_{i=0}^{127} \frac{(h^t_{2i} - h^e_{2i})^2}{h^t_{2i}} \tag{8.13}$$

и проверки условия

$$\chi^2 < \chi^2_\alpha(k-1), \tag{8.14}$$

где α – уровень значимости, а $k - 1 = 127$ – число степеней свободы. Если неравенство (8.14) справедливо, то принимается решение о наличии в контейнере встроенной информации.

Данный метод позволяет обнаружить НЗБ-встраивание в условиях полного заполнения битовой плоскости. При низкой заполненности кон-

тейнера (см. формулу (3.7)) метод работает значительно хуже. Кроме того, по понятным причинам он не позволяет обнаружить ±1-встраивание.

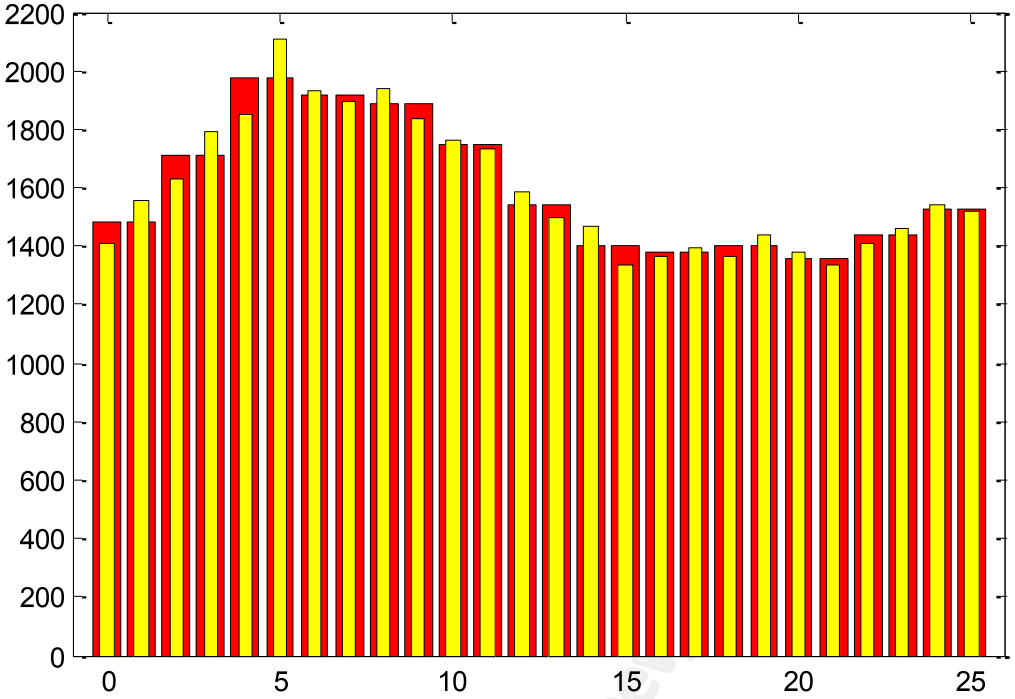


Рис. 8.3 – Пример эмпирической гистограммы изображения (жёлтый цвет) и соответствующей ей теоретической гистограммы (красный цвет)

В книгах [2, 6] рассматривается метод Sample Pair Analysis, являющийся более эффективным для обнаружения НЗБ-встраивания. Помимо собственно факта встраивания информации он позволяет с высокой точностью оценить заполненность контейнера.

Упражнения

Реализация встраивания простейшего видимого ЦВЗ и атаки на него

Результатом работы будет являться скрипт `vis_attack_run`.

1. Считать входное изображение, бинарный логотип и сгенерировать большой ЦВЗ путём замощения плоскости встраивания логотипом (используя цикл for).
2. Сгенерировать аналогичный ЦВЗ без использования циклов (путём конкатенации матриц). Рассчитать время генерации ЦВЗ первым и вторым способом.
3. Написать формулу и реализовать простейшее аддитивное неадаптивное встраивание видимого (визуально заметного) ЦВЗ.
4. Изменить формулу встраивания ЦВЗ для повышения контрастности шаблона в очень тёмных и очень светлых областях.
5. Зная встроенный шаблон ЦВЗ, полностью удалить встроенный в изображение водяной знак.

Реализация атаки усреднением на СВИ-9 (Cox et al.)

Результатом работы будет являться скрипт `cox_attack_run`. В работе используются файлы, полученные при реализации и исследовании СВИ-9 (Cox et al.) (глава 5).

1. Осуществить независимое встраивание $K = 15$ разных ЦВЗ в одинаковые контейнеры, после чего усреднить полученные носители информации.
2. Извлечь ЦВЗ из полученного смешанного контейнера и оценить его близость с каждым из встроенных ЦВЗ.
3. Построить график близости встроенной и извлечённой последовательности в зависимости от $K = 1..30$.
4. Построить график максимальной близости извлечённой последовательности и случайной в зависимости от $K = 1..30$. Сопоставить графики и сделать вывод о стойкости системы к атаке усреднением.

Расчёт признаков носителя информации на основе среднего значения и среднего числа переходов

Результатом работы будет являться скрипт `steg_simple_run`, а также функция

```
[CW] = lsb_embed_for_steg(C, p, q)
```

– заполнение `p`-й битовой плоскости случайным полем с равномерным распределением. `q` задаёт заполненность контейнера (формула (3.7)).

1. Реализовать функцию `lsb_embed_for_steg` для случая `q=1` и проверить её работу.
2. Дополнить функцию `lsb_embed_for_steg` случаем произвольного `q`.
3. Реализовать расчёт локального среднего значения в заданной битовой плоскости заданного изображения.
4. Реализовать расчёт локального числа горизонтальных переходов в заданной битовой плоскости заданного изображения.
5. Реализовать расчёт локального среднего числа переходов в заданной битовой плоскости заданного изображения.
6. Выполнить расчёты изображений согласно заданиям 3 и 5 для пустого контейнера, заполненного на 100 % и заполненного на 50 % (при встраивании в первую битовую плоскость).
7. Осуществить расчёт признаков по полученным изображениям: среднее, дисперсия, наибольшее, наименьшее значения, а также разность последних.
8. Выполнить ручной отбор информативных признаков и придумать эвристические решающие правила, определяющие наличие встроенной в изображение информации.
9. Подготовить тестовую выборку из 300 изображений. Первые 100 оставить неизменными, следующую сотню изображений заполнить на 100 %, последнюю – на 50 %.
10. Применить к выборке решающие правила и получить вектор результатов классификации из 300 элементов.
11. Получить вектор истинных классов, осуществить расчёт точности классификации по выбранным решающим правилам по формуле (8.26).

12. Выполнить расчёт признаков при заполнении второй битовой плоскости.
13. Повторить задание 8 для полученных значений признаков для второй битовой плоскости.
14. Повторить задания 9-10 для встраивания во вторую битовую плоскость.
15. Повторить задание 11 для встраивания во вторую битовую плоскость. Сравнить результаты.

Реализация и проверка метода стегоанализа с расчётом гистограмм пар значений

Результатом работы будет являться скрипт `steg_hist_run` и функция

```
[isStego] = steg_hist(C)
```

– применение метода гистограмм пар значений для обнаружения наличия встраивания в первой битовой плоскости контейнера. Возвращает 1, если контейнер заполнен.

1. Подготовить тестовые данные: пустой контейнер, заполненный на 100 % носитель информации и заполненный на 50 % носитель информации (встраивание в первую битовую плоскость). Использовать ранее написанную функцию `lsb_embed_for_steg`.
2. Отобразить гистограммы трёх изображений, проверить справедливость предположений, используемых в рассматриваемом методе стегоанализа.
3. Начать реализацию функции `steg_hist`: выполнить расчёт эмпирической и теоретической гистограмм.
4. Завершить реализацию функции `steg_hist`: рассчитать значение статистики χ^2, проверить статистическую гипотезу.
5. Применить функцию `steg_hist` для трёх ранее полученных изображений.
6. Реализовать в цикле проверку работоспособности метода для разных значениях заполненности контейнера q.
7. Модифицировать расчёт статистики для обнаружения встраивания информации во вторую битовую плоскость.

8. Повторить задание 6 для обнаружения встраивания во вторую битовую плоскость.

Лабораторная работа 3: Исследование стойкости систем цифровых водяных знаков к искажениям носителя информации

Задания

В лабораторной работе необходимо выполнить исследование устойчивости определённой вариантом системы встраивания ЦВЗ к нескольким искажениям, которые также задаются вариантом. После успешной сдачи практической части задания студентам необходимо ответить на один контрольный вопрос, заданный преподавателем.

Генерация, встраивание, извлечение и сравнение ЦВЗ осуществляется при помощи поставляемой библиотеки исполняемых файлов “Watermarking” [59]. Вместе с библиотекой поставляется пример командного файла, осуществляющего встраивание и извлечение ЦВЗ средствами данной библиотеки, а также функции MATLAB, реализующие эти операции.

Результаты сравнения встроенного и извлечённого ЦВЗ для каждого искажения и для каждого значения параметра необходимо сохранить и вывести на экран в виде графиков.

В лабораторной работе предлагается исследовать стойкость систем к следующим искажениям носителя информации:

1. Линейное изменение динамического диапазона функции яркости

Заключается в линейном поэлементном преобразовании изображения:

$$\widetilde{C^W}(n_1, n_2) = min\{\alpha C^W(n_1, n_2), 255\}, \quad \alpha \in \mathbb{R}, \alpha > 0. \tag{8.15}$$

Параметром является коэффициент α при значении функции яркости.

2. Поворот с последующим восстановлением

В данном искажении необходимо произвести два последовательных поворота изображения: на некоторый угол φ и на обратный ему угол $-\varphi$. При первом повороте важно увеличить размер изображения, чтобы не допустить обрезки углов.

Параметром является угол поворота φ.

3. Масштабирование с последующим восстановлением

Заключается в последовательном выполнении операций изменения размера изображения и его возвращения его в исходный размер.

Параметром является коэффициент масштабирования.

4. Обрезка с заменой данными из исходного контейнера

Данное искажение заключается в вырезании из носителя информации размерами $N_1 \times N_2$ прямоугольной области с теми же пропорциями, начинающейся в точке с координатами $(0,0)$ и составляющей долю ϑ от его площади. Оставшаяся часть заменяется значениями из исходного контейнера

$$\widetilde{C^W}(n_1,n_2) = \begin{cases} C^W(n_1,n_2) & n_1 \le N_1\left[\sqrt{\vartheta}\right], n_2 \le N_2\left[\sqrt{\vartheta}\right], \\ C(n_1,n_2) & n_1 > N_1\left[\sqrt{\vartheta}\right], n_2 > N_2\left[\sqrt{\vartheta}\right]. \end{cases} \quad (8.16)$$

Параметром является доля ϑ.

5. Усреднение в скользящем окне

Под усреднением в скользящем окне будем понимать обработку изображения C^W ЛИС-системой, имеющей конечную импульсную характеристику $g(m_1,m_2)$ размерами $M \times M$, где $M = 2p+1, p \in \mathbb{N}$:

$$\widetilde{C^W}(n_1,n_2) = \sum_{m_1=0}^{M-1}\sum_{m_2=0}^{M-1} g(m_1,m_2)\cdot C^W(n_1-m_1,n_2-m_2), \quad (8.17)$$

причём отсчёты ИХ постоянны и равны $1/M^2$. Например, для $M = 3$

$$g(m_1,m_2) = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Параметром является размер окна M.

6. Гауссовское размытие

Заключается в обработке изображения ЛИС-системой с бесконечной импульсной характеристикой $g(m_1,m_2)$:

$$\widetilde{C^W}(n_1,n_2) = \sum_{m_1=-\infty}^{\infty}\sum_{m_2=-\infty}^{\infty} g(m_1,m_2)\cdot C^W(n_1-m_1,n_2-m_2), \quad (8.18)$$

имеющей вид функции Гаусса:

$$g(m_1,m_2) = \frac{1}{2\pi\sigma^2}\,exp\left\{-\frac{m_1^2+m_2^2}{2\sigma^2}\right\}, \qquad m_1,m_2 \in (-\infty,\infty). \quad (8.19)$$

На практике свёртка с БИХ-фильтром (8.18) заменяется свёрткой с КИХ-фильтром (8.17), который описывается следующим выражением:

$$g(m_1, m_2) = K \cdot exp\left\{-\frac{(m_1 - M/2)^2 + (m_2 - M/2)^2}{2\sigma^2}\right\}, \quad (8.20)$$

где M – размер окна, определяемый по правилу «трёх сигма»:

$$M = 2 \cdot [3\sigma] + 1, \quad (8.21)$$

где $[x]$ – целая часть числа x, а коэффициент K находится из условия нормировки

$$\sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} g(m_1, m_2) = 1. \quad (8.22)$$

Параметром преобразования является значение σ.

7. Повышение резкости

Заключается в следующем преобразовании входного изображения:

$$\widetilde{C^W}(n_1, n_2) = C^W(n_1, n_2) + q\left(C^W(n_1, n_2) - C^W_{smooth}(n_1, n_2)\right), \quad (8.23)$$

где C^W_{smooth} – результат усреднения C^W в окне размерами $M \times M$ (искажение 5 текущего списка), а $q > 0$ – коэффициент усиления разностного изображения.

В качестве изменяемого параметра преобразования будем использовать M, а q примем равным 5.

8. Медианная фильтрация

Медианный фильтр реализуется как процедура локальной обработки скользящим окном различной формы (в настоящей лабораторной работе предлагается использовать квадратное окно размерами $M \times M$), которое включает нечетное число отсчетов изображения: $M = 2p + 1$, $p \in \mathbb{N}$.

Процедура обработки заключается в том, что для каждого положения окна попавшие в него отсчеты упорядочиваются по возрастанию значений. Средний отсчет в этом упорядоченном списке называется *медианой* рассматриваемой группы. Эта медиана заменяет центральный отсчет в окне для обработанного сигнала.

Параметром является размер окна M.

9. Аддитивное зашумление

Заключается в добавлении к изображению поля $\xi(n_1, n_2)$:

$$\widetilde{C^W}(n_1, n_2) = C^W(n_1, n_2) + \xi(n_1, n_2), \quad (8.24)$$

значения которого являются реализацией гауссовской случайной величины с плотностью распределения

$$\rho_\xi(x) = \frac{1}{\sqrt{2\pi D_\xi}} exp\left(-\frac{x^2}{2D_\xi}\right). \quad (8.25)$$

Параметром является дисперсия шума D_ξ.

10. JPEG-сжатие с потерями

Искажение заключается в сохранении носителя информации в формате JPEG и последующем восстановлении его в формате без потерь.

Параметром является показатель качества JPEG-файла Q, изменяемый в пределах от 1 до 100.

Таблица параметров искажений

№ искажения	Параметр p	p_{min}	p_{max}	Δ_p
1	Коэффициент α	0,7	1,3	0,1
2	Угол поворота φ (в градусах)	0	42	7
3	Коэффициент масштабирования	0,55	1,45	0,15
4	Доля площади ϑ	0,2	0,9	0,1
5	Размер окна M	3	15	2
6	Параметр размытия σ	1	4	0.5
7	Размер окна M	3	15	2
8	Размер окна M	3	15	2
9	Дисперсия шума D_ξ	400	1000	100
10	Параметр качества Q	30	90	10

Таблица вариантов заданий

№ варианта	Исследуемая ЦВЗ-система	Список номеров искажений
1	СВИ-11 (Corvi & Nicchiotti)	1, 5, 9
2	СВИ-9 (Cox et al.)	1, 7, 10
3	СВИ-12 (Wang et al.)	1, 8, 9
4	СВИ-11 (Corvi & Nicchiotti)	3, 5, 10
5	СВИ-9 (Cox et al.)	3, 7, 9
6	СВИ-12 (Wang et al.)	3, 8, 10
7	СВИ-11 (Corvi & Nicchiotti)	2, 6, 7, 9
8	СВИ-9 (Cox et al.)	2, 7, 8, 10
9	СВИ-12 (Wang et al.)	2, 6, 7, 9

№ варианта	Исследуемая ЦВЗ-система	Список номеров искажений
10	СВИ-11 (Corvi & Nicchiotti)	4, 7, 8,10
11	СВИ-9 (Cox et al.)	4, 6, 7, 9
12	СВИ-12 (Wang et al.)	4, 7, 8, 10

Контрольные вопросы

1. Классификация систем встраивания информации по стойкости.
2. Как на практике могут применяться стойкие системы ЦВЗ?
3. Как на практике могут применяться хрупкие и полухрупкие системы ЦВЗ?
4. Классификация атак на СВИ по целям.
5. Классификация атак на СВИ по знаниям нарушителя.
6. Опишите принцип медианной фильтрации изображения. Сравните её эффективность для устранения шума типа «соль-и-перец» с усреднением в скользящем окне.
7. Опишите процедуру гауссовского размытия изображения и способ выбора окна фильтра.
8. Опишите смысл и существо процедуры повышения резкости.
9. Перечислите (и при необходимости кратко опишите) основные виды искажений, применяемых к носителю информации для исследования стойкости системы.
10. Перечислите основные свойства СВИ-11 (Corvi & Nicchiotti) и её отличительные особенности: типы процедур встраивания и извлечения информации, область встраивания и пр.
11. Перечислите основные свойства СВИ-9 (Cox et al.) и её отличительные особенности: типы процедур встраивания и извлечения информации, область встраивания и пр.
12. Перечислите основные свойства СВИ-12 (Wang et al.) и её отличительные особенности: типы процедур встраивания и извлечения информации, область встраивания и пр.

Лабораторная работа 4: Реализация и исследование методов НЗБ-стегоанализа изображений

Задания

В лабораторной работе необходимо выполнить исследование качества решения задачи стегоанализа СВИ-2 методами, использующими признаки на основе развёртки НЗБП, в зависимости от заполненности контейнера q (3.7). Параметры встраивания, а также методы классификации и расчёта признаков определяются вариантом задания. После успешной сдачи практической части задания студентам необходимо ответить на один контрольный вопрос, заданный преподавателем.

Входными данными, необходимыми для выполнения лабораторной работы, являются K полутоновых изображений одного размера.

Порядок выполнения лабораторной работы:

1. Реализовать процедуру расчёта всех заданных векторов признаков с использованием заданной развёртки (согласно варианту).
2. Выполнить имитацию работы СВИ-2 для первых $K/2$ изображений: заполнить долю q битовой плоскости p каждого изображения разными реализациями равномерного белого шума. Вторую половину изображений не менять.
3. Произвести обучение заданного классификатора по выборке, содержащей первые 20 % изображений каждого из двух типов (со встраиванием и без) с использованием вектора признаков v. То есть общий объём обучающей выборки составляет $K \cdot 0{,}2$.
4. Применить обученный классификатор на оставшихся 80 % изображений и оценить качество классификации в соответствии с заданным критерием.
5. Повторить пп. 2-4 для прочих долей q и векторов признаков v, сохранить результаты оценки качества классификации в виде таблицы следующего вида:

		Заполненность контейнера q			
		q_1	q_2	q_3	…
Вектор признаков v	v_1				
	v_2				
	v_3				
	…				

Если вариант предполагает использование двух разных развёрток, значит, необходимо получить две такие таблицы.

В задании используются 4 вида развёрток двумерных областей, три из которых были рассмотрены в параграфе 8.2, а четвёртая – зигзагообразная – в параграфе 5.3 при описании СВИ-9 (Cox et al.). В качестве классификаторов используются линейный и квадратичный дискриминантный анализ (ЛДА и КДА), рассмотренные в [53, 60].

Для оценки качества классификации в лабораторной работе предлагается использовать один из двух показателей (по вариантам), рассчитываемых по данным из таблицы ниже. Каждая её ячейка содержит число изображений соответствующей категории.

Табл. 8.2 – Виды ошибок классификации

	Правильная классификация	Неправильная классификация
Информация встроена	TP	FP
Информация не встроена	TN	FN

Показателями качества являются *доля правильных ответов* (англ. “accuracy”), также известная под названием «индекс Рэнда» [61]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{8.26}$$

а также *F-мера* (или F_1 мера), вычисляемая по формуле

$$F = \frac{2}{\frac{1}{Pr} + \frac{1}{R}}, \tag{8.27}$$

где

$$Pr = \frac{TP}{TP + FN}, \tag{8.28}$$

$$R = \frac{TP}{TP + FP}. \tag{8.29}$$

Величины (8.28), (8.29) называются соответственно *точностью* и *полнотой* (англ. – “precision” и “recall”).

Наборы признаков

Заданием предусмотрены 5 наборов векторов признаков, обозначенных литерами A, B, C, D и E. Набор A формируется на основе анализа частоты переходов (см. формулы (8.21)–(8.23)) следующим образом:

$$A = \left\{\left(\frac{\pi_{00} + \pi_{11}}{2}, \frac{\pi_{01} + \pi_{10}}{2}\right), (\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11})\right\}.$$

Остальные наборы формируются на основе анализа числа длин серий (см. формулу (8.11)) с усреднением в соответствии с таблицей:

Набор признаков			Максимальная длина серии			
			4	8	16	32
B	Усреднять по	1	+	+	+	
C		2		+	+	+
D		4		+	+	+
E		8			+	+

Знак «+» означает, что данное сочетание максимальной длины серии и длины усреднения входит в набор. Например, набор E состоит из следующих векторов признаков:

$$E = \left\{\left(\sum_{i=1}^{8} s_i, \sum_{i=9}^{16} s_i\right), \left(\sum_{i=1}^{8} s_i, \sum_{i=9}^{16} s_i, \sum_{i=17}^{24} s_i, \sum_{i=25}^{32} s_i\right)\right\}.$$

Таблица вариантов заданий

№ варианта	p	Развёртка	Классификатор	Наборы признаков	Показатель качества
1	1	Построчная	ЛДА	A	(8.26)
2	1	Серпантинная	ЛДА	B	(8.26)
3	1	Построчная	КДА	E	(8.27)
4	2	Серпантинная	КДА	A	(8.27)
5	2	Построчная	КДА	B	(8.26)
6	2	Серпантинная	ЛДА	E	(8.26)
7	1	Построчная, Серпантинная	ЛДА	C	(8.27)
8	1	Построчная, Зигзагообразная	ЛДА	D	(8.27)
9	1	Построчная, Гильберта-Пеано	КДА	D	(8.26)
10	2	Серпантинная, Зигзагообразная	КДА	D	(8.26)
11	2	Серпантинная, Гильберта-Пеано	КДА	C	(8.27)
12	2	Зигзагообразная, Гильберта-Пеано	ЛДА	C	(8.27)

Для всех вариантов необходимо рассмотреть три разных значения q: 1, 0.7 и 0.3.

Контрольные вопросы

Основные вопросы

1. Задача стегоанализа и различные подходы к её решению.
2. В чём состоит СВИ-2 (стеганографическое НЗБ-встраивание)?
3. В чём состоит СВИ-3 (±1-встраивание)?
4. Опишите возможные способы заполнения контейнера в СВИ-2 (стеганографическое НЗБ-встраивание). Что такое заполненность контейнера?
5. Опишите общий принцип стегоанализа НЗБ-систем и перечислите рассмотренные методы атак на подобные системы.
6. Как осуществляется расчёт числа переходов, почему оно позволяет выявить наличие встроенной информации?
7. Как осуществляется расчёт числа серий, почему оно позволяет выявить наличие встроенной информации? Проиллюстрируйте ответ графиками.
8. Приведите несколько примеров векторов признаков на основе числа серий. Придумайте вектор, не упоминавшийся в тексте пособия.
9. Опишите показатели качества решения задачи стегоанализа, используемые в данной лабораторной работе.
10. Перечислите развёртки двумерных областей, используемые в данной лабораторной работе. Как вы думаете, какие из них лучше других и почему?
11. Кратко опишите используемые в лабораторной работе методы классификации.

Дополнительные вопросы

1. Подходят ли методы стегоанализа СВИ-2, использующие только битовую плоскость, в которой возможно произошло встраивание, для стегоанализа СВИ-3?

Библиографический список

1. Barni, M. Watermarking Systems Engineering [Text] / M. Barni, F. Bartolini. — New-York: Marcel Dekker, Inc., 2004. — 485 p.
2. Cox, I.J. Digital Watermarking and Steganography [Text] / I.J. Cox [et al.]. — 2nd ed. — Morgan Kaufmann Publishers, 2008. — 596 p.
3. Mayer, G. Image Repository [Electronic resource] / G. Mayer // University of Waterloo Fractal coding and analysis group, 2009. — Режим доступа: http://links.uwaterloo.ca/Repository.html (17.10.2015).
4. Miller, M.L. A review of watermarking, principles and practices [Text] / M.L. Miller [et al.] // Digital Signal Processing in Multimedia Systems / Ed. by K.K. Parhi, T. Nishitani. — Marcel Dekker, Inc., 1999. — P. 461–485.
5. Cox, I.J. Digital Watermarking [Text] / I.J. Cox, M.L. Miller, J.A. Bloom. — San Francisco: Morgan Kaufmann Publishers, 2002. — 568 p.
6. Fridrich, J. Steganography in digital media: principles, algorithms, and applications [Text] / J. Fridrich. — Cambridge University Press, 2010. — 450 p.
7. Petitcolas, F.A.P. Information Hiding - A Survey [Text] / F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn // Proceedings of the IEEE. — 1999. — Vol. 87. — No. 7. — P. 1062–1078.
8. Katzenbeisser, S. Information Hiding Techniques for Steganography and Digital Watermarking [Text] / S. Katzenbeisser, F.A.P. Petitcolas. — Boston, London: Artech House, Inc., 2000. — 237 p.
9. Cole, E. Hiding in Plain Sight: Steganography and the Art of Covert Communication [Text] / E. Cole. — Wiley Publishing, Inc., 2003. — 362 p.
10. Pfitzmann, B. Information Hiding Terminology: Results of an informal plenary meeting and additional [Text] / B. Pfitzmann // Springer LNCS, vol. 1174. — 1996. — P. 347–350.
11. Schneir, B. Applied Cryptography [Text] / B. Schneir. — John Wiley & Sons, Inc., 1996. — 662 p.
12. Simmons, G.J. The history of subliminal channels [Text] / G.J. Simmons // LNCS, vol. 1174. —1996. — P. 237–256.
13. Грибунин, В.Г. Цифровая стеганография [Текст] / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. — М.: Солон-Пресс, 2002. — 272 с.
14. Аграновский, А.В. Стеганография, цифровые водные знаки и стеганоанализ [Текст] / А.В. Аграновский [и др.]. — М.: Вузовская книга, 2009. — 220 с.
15. Конахович, Г.Ф. Компьютерная стеганография. Теория и практика [Текст] / Г.Ф. Конахович, А.Ю. Пузыренко. — Киев: МК-Пресс, 2006. — 288 с.
16. Гонсалес, Р. Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс. — М.: Техносфера, 2005. — 1072 с.

17. Roth, S. Introduction to Matlab (Code) / S. Roth // Computer Science at Brown University, 2010. — Режим доступа: URL: https://cs.brown.edu/courses/csci1950-g/docs/matlab/matlabtutorialcode.html (08.12.2015).
18. Sigmon, K. MATLAB primer [Text] / K. Sigmon, T.A. Davis. — CRC Press, 2004.
19. Lazzeretti, R. Introduction to Matlab / R. Lazzeretti // The Visual Information Processing and Protection research group, 2014. — Режим доступа: http://clem.dii.unisi.it/~vipp/files/MultimediaSecurity/Introduction_to_Matlab_Lazzeretti.pdf (08.12.2015).
20. Chen, B. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding [Text] / B. Chen, B. Wornell // IEEE Transactions on Information Theory. — 2001. — Vol. 47. — No. 4. — P. 1423–1443.
21. Глумов, Н.И. Алгоритм встраивания полухрупких цифровых водяных знаков для задач аутентификации изображений и скрытой передачи информации [Текст] / Н.И. Глумов, В.А. Митекин // Компьютерная оптика. — 2011. — Т. 35. — С. 262–267.
22. Lau, D.L. Modern digital halftoning [Text] / D.L. Lau, G.R. Arce. — New-York: Marcel Dekker, Inc., 2001.
23. Fu, M.S. Data hiding watermarking for halftone images [Text]: Ph.D. Thesis / M.S. Fu. — The Hong Kong University of Science and Technology, Hong Kong, 2003.
24. Floyd, R.W. An adaptive algorithm for spatial gray-scale [Text] / R.W. Floyd, L. Steinberg // Proceedings Society Information Display. — 1976. — Vol. 17. — No. 2. — P. 75–78.
25. Fan, Z. Error diffusion with a more symmetric error distribution [Text] / Z. Fan // Proceedings of SPIE, vol. 2179. — 1994. — P. 150–158.
26. Jarvis, J.F. A survey of techniques for the display of continuous-tone pictures on bilevel displays [Text] / J.F. Jarvis, C.N. Judice, W.H. Ninke // Comp. Graf. Im. Pr. — 1976. — Vol. 5. — No. 2. — P. 13–40.
27. Stucki, P. MECCA – a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction [Text]: Technical Report RZ1060 / P. Stucki. — Zurich, 1981.
28. Mallat, S. A Wavelet Tour of Signal Processing [Text] / S. Mallat. — Academic Press, 1999. — 620 p.
29. Сэломон, Д. Сжатие данных, изображений, звука [Текст] / Д. Сэломон. — М.: Техносфера, 2004. — 339 с.
30. Wang, R. Introduction to Orthogonal Transforms: with Applications in Data Processing and Analysis [Text] / R. Wang. — Cambridge University Press, 2011.
31. Cox, I.J. Secure Spread Spectrum Watermarking for Multimedia [Text] / I.J. Cox // IEEE Transactions on Image Processing. — 1997. — Vol. 6. — No. 12. — P. 1673–1687.

32. Barni, M. A DCT-domain system for robust image watermarking [Text] / M. Barni, F. Bartolini, V. Cappellini, A. Piva // Signal Processing. — 1998. — Vol. 66. — No. 3. — P. 357–372.
33. Piva, A. DCT-based watermark recovering without resorting to the uncorrupted original image [Text] / A. Piva [et al.]: Proceedings of 1997 IEEE International Conference on Image Processing. — 1997. — Vol. 1. — P. 520–523.
34. Corvi, M. Wavelet-based image watermarking for copyright protection [Text] / M. Corvi, G. Nicchiotti // Scandinavian conference on image analysis. — 1997. — P. 157–163.
35. Wang, H.J. Wavelet-based digital image watermarking [Text] / H.J. Wang, P.C. Su, C.C.J. Kuo // Optics Express. — 1998. — Vol. 3. — No. 12. — P. 491–496.
36. Перспективные информационные технологии дистанционного зондирования земли [Текст] / под ред. В.А. Сойфера. — Самара: Новая техника, 2015. — 255 с.
37. Popescu, A.C. Statistical Tools for Digital Image Forensics [Text]: Ph. D. Thesis / Dartmouth College, 2004.
38. Celik, M.U. Lossless generalized LSB data embedding [Text] / M.U. Celik, G. Sharma, A.M. Tekalp, E. Saber // IEEE Transactions on Image Processing. — 2005. — Vol. 14. — No. 2. — P. 253–266.
39. Goljan, M. Distortion-Free Data Embedding for Images [Text] / M. Goljan, J. Fridrich, R. Du // Springer LNCS, vol. 2137. — 2001. — P. 27–41.
40. Lin, C.-Y. Issues and solutions for authenticating MPEG video [Text] / C.-Y. Lin, S.-F. Chang // Proceedings of SPIE, vol. 3657. — 1999.
41. Yeung, M. An invisible watermarking technique for image verification [Text] / M. Yeung, F. Mintzer // International Conference on Image Processing. — 1997. — Vol. 1. — P. 680–683.
42. Doërr, G. Security pitfalls of frame-by-frame approaches to video watermarking [Text] / G. Doërr, J.L. Dugelay // IEEE Transactions on Signal Processing. — 2004. — Vol. 52. — No. 10. — P. 2955–2964.
43. Hartung, F. Watermarking of Uncompressed and Compressed Video [Text] / F. Hartung, B. Girod // Signal Processing. — 1998. — Vol. 66. — No. 3. — P. 283–301.
44. Kalker, T. A Video Watermarking System for Broadcast Monitoring [Text] / T. Kalker // Proceedings of SPIE, vol. 3657. — 1999.
45. Митекин, В.А. Метод встраивания информации в видео, стойкий к ошибкам потери синхронизации [Текст] / В.А. Митекин, В.А. Федосеев // Компьютерная оптика. — 2014. — Т. 38. — №3. — С. 564–573.
46. Mitekin, V. A new method for high-capacity information hiding in video robust against temporal desynchronization [Text] / V. Mitekin, V. Fedoseev // Proceedings of SPIE, vol. 9445. — 2015. — P. 94451A-1—94451A-7.
47. Методы компьютерной обработки изображений [Текст] / под ред. В.А. Сойфера. —2-е изд. — М.: Физматлит, 2003. — 784 с.

48. Kalker, T. Analysis of Watermark Detection using SPOMF / T. Kalker, A.J.E.M. Janssen // Proceedings of ICIP. — 1999. — Vol. 1. — P. 316-319.
49. Jähne, B. Digital Image Processing [Text] / B. Jähne. — Springer, 2005. — 631 p.
50. Ту, Д. Принципы распознавания образов [Текст] / Д. Ту, Р. Гонсалес. — М.: Мир, 1978.
51. Вапник, В.Н. Теория распознавания образов. Статистические проблемы обучения [Текст] / В.Н. Вапник, А.Я. Червоненкис. — М.: Наука, 1974.
52. James, G. An introduction to statistical learning [Text] / G. James, D. Witten, T. Hastie, R. Tibshirani. — New York: Springer, 2013.
53. Воронцов, К.В. Машинное обучение (курс лекций) [Электронный ресурс] . — 2015. — Режим доступа: http://www.machinelearning.ru/wiki/index.php?-title=Машинное_обучение_(курс_лекций%2C_К.В.Воронцов) (09.12.2015).
54. Sagan, H. Space-filling curves [Text] / H. Sagan. — New York: Springer, 1994. — 193 p.
55. Сергеев, В.В. Обработка изображений с использованием развертки Гильберта-Пеано [Текст] / В.В. Сергеев // Автометрия. — 1984. — Т. 2. — С. 30–36.
56. Harmsen, J. Higher-order statistical steganalysis of palette images [Text] / J. Harmsen, W. Pearlman // Proceedings of SPIE, vol. 5020. — 2003. — P. 131–142.
57. Cancelli, G. New techniques for steganography and steganalysis in the pixel domain [Text]: Ph.D. Thesis / G. Cancelli. — University of Siena, Siena, 2009.
58. Cancelli, G. A comparative study of ±1 steganalyzers [Text] / G. Cancelli [et al.] // IEEE 10th Workshop on Multimedia Signal Processing. — 2008. — P. 791–796.
59. Meerwald, P. Digital Watermarking Source / P. Meerwald // University of Salzburg, 2010. Режим доступа: http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/-source/ (15.12.2015).
60. Коломиец, Э.И. Байесовская классификация. Методические указания к лабораторной работе [Текст] / Э.И. Коломиец, В.В. Мясников. — Самара: Издательство СГАУ, 2000. — 16 с.
61. Manning, C.D. Introduction to information retrieval [Text] / C.D. Manning, P. Raghavan, H. Schütze. — Cambridge University Press, 2008. — 496 p.

Предметный указатель

Victor Fedoseev

DIGITAL WATERMARKING AND STEGANOGRAPHY:

Textbook with Practical and Laboratory Exercises

This textbook provides basic knowledge on Information Hiding topic: models, methods and terms, and also describes a number of particular techniques for watermarking and steganography, as well as some attacks on such systems. Apart from theoretical material, almost each chapter contains several exercises which can help readers to understand better the considered algorithms and systems. Most of these exercises consist in programming implementation of the systems and its brief investigation.

The book was written as a secondary practical support for the course of Digital Watermarking and Steganography delivered by the author to graduate students of Samara State Aerospace University. Therefore some important aspects of Information Hiding science were not included into the book. Among them are foundations of data concealment within multimedia, actual staganographic systems and opposite steganalysis routines, audio watermarking, geometrically invariant image watermarking, and common approaches to key generation.

The book is organized as follows: Chapter 1 provides introductory information, discusses history of digital watermarking and steganography, and also sets out some definitions. Chapter 2 represents a short MATLAB tutorial which is necessary for exercises given in subsequent chapters. Chapters 3-7 describe a variety of watermarking and steganographic systems differing in their practical applications, types of cover multimedia object, and particular embedding and extraction algorithms. Finally, Chapter 8 considers some simple attacks to watermarking systems and basic methods of steganalysis.

When writing theoretical material for this textbook, the author was primarily relying on the books by Barni and Bartolini [1] and by Cox et al. [2], as well as on his own lecture notes. All materials adopted from foreign sources are properly cited.

Учебное издание

Виктор Андреевич Федосеев

ЦИФРОВЫЕ ВОДЯНЫЕ ЗНАКИ И СТЕГАНОГРАФИЯ

Учебное пособие с заданиями
для практических и лабораторных работ

Редактор Ю.Н. Литвинова
Вёрстка В.А. Федосеев

Электронное издание

федеральное государственное автономное образовательное учреждение высшего образования «Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)» (СГАУ)
443086 Россия, г.Самара, Московское шоссе, 34

Изд-во СГАУ. 443086 Россия, г.Самара, Московское шоссе, 34