

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

МАТЕМАТИЧЕСКИЕ МЕТОДЫ
ОБРАБОТКИ КОСМИЧЕСКИХ
ИЗОБРАЖЕНИЙ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ

*Утверждено Редакционно-издательским советом университета
в качестве учебного пособия*

САМАРА
Издательство СГАУ
2007

УДК 681.3
ББК 22.343
М34



**Инновационная образовательная программа
"Развитие центра компетенции и подготовка
специалистов мирового уровня в области аэро-
космических и геоинформационных технологий"**

**Авторы: *В.В. Сергеев, М.В. Гашиков, Н.И. Глунов,
В.В. Мясников, Е.В. Мясников***

**Рецензенты: д-р техн. наук А. Г. Храмов,
д-р техн. наук В. Г. Карташевский**

**М34 Математические методы обработки космических изображе-
ний. Лабораторный практикум: учеб. пособие / [В.В. Сергеев и
др.]. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2007. – 148 с.: ил.**

ISBN 978-5-7883-0601-8

Пособие представляет собой сборник методических указаний к лабораторным работам, для каждой из которых приводятся описание теоретических основ, задание и список контрольных вопросов. Кроме того, описывается программная система обработки изображений, необходимая для выполнения заданий. Разделы теоретических основ содержат описания наиболее распространенных методов обработки космических изображений: методов поэлементной обработки, выделения контуров, фильтрации шумов и распознавания объектов, а также различные методы компрессии.

Учебное пособие предназначено для студентов факультета информатики, обучающихся по специальности "Прикладная математика и информатика".

УДК 681.3
ББК 22.343

ISBN 978-5-7883-0601-8

© Сергеев В.В., Гашиков М.В., Глунов Н.И.,
Мясников В.В., Мясников Е.В., 2007
© Самарский государственный
аэрокосмический университет, 2007

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ.....	3
1 ПОЭЛЕМЕНТНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ.....	6
1.1 Теоретические основы лабораторной работы	6
1.1.1 Общее описание метода	6
1.1.2 Линейное контрастирование.....	9
1.1.3 Преобразование гистограмм	11
1.1.4 Пороговая обработка.....	13
1.2 Выполнение лабораторной работы	15
1.2.1 Общий план выполнения работы	15
1.2.2 Варианты выполнения лабораторной работы	16
1.2.3 Содержание отчета	17
1.2.4 Контрольные вопросы.....	17
2 ЛОКАЛЬНЫЕ МЕТОДЫ ВЫДЕЛЕНИЯ КОНТУРОВ.....	18
2.1 Теоретические основы лабораторной работы	18
2.1.1 Линейная локальная обработка изображений	18
2.1.2 Постановка задачи выделения контуров.....	21
2.1.3 Градиентный метод	24
2.1.3 Метод оператора Лапласа	27
2.1.4 Метод согласования.....	30
2.1.5 Алгоритм выделения контуров.....	33
2.2 Выполнение лабораторной работы	35
2.2.1 Общий план выполнения работы	35
2.2.2 Порядок проведения обработки	35
2.2.3 Содержание отчета	36
2.2.4 Контрольные вопросы.....	36
3 ФИЛЬТРАЦИЯ ШУМОВ НА ИЗОБРАЖЕНИЯХ.....	37
3.1 Теоретические основы лабораторной работы	37
3.1.1 Модели помех при регистрации изображений	37
3.1.2 Алгоритмы фильтрации изображений	39
3.1.3 Линейный сглаживающий фильтр.....	41
3.1.4 Медианный фильтр.....	45

3.2	Выполнение лабораторной работы.....	48
3.2.1	Общий план выполнения работы.....	48
3.2.2	Вариант 1 - фильтрация аддитивного белого шума	49
3.2.3	Вариант 2 - фильтрация импульсного шума	49
3.2.4	Содержание отчета.....	49
3.2.5	Контрольные вопросы	50
4	ОБНАРУЖЕНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ.....	50
4.1	Теоретические основы лабораторной работы.....	50
4.1.1	Корреляционный метод обнаружения объектов на изображении	50
4.1.2	Метод согласованной фильтрации	54
4.1.3	Согласованный фильтр для фона с биэкспоненциальной корреляционной функцией.....	59
4.2	Выполнение лабораторной работы.....	61
4.2.1	Общий план выполнения работы.....	61
4.2.2	Схема исследования.....	61
4.2.3	Варианты выполнения лабораторной работы.....	62
4.2.4	Содержание отчета.....	63
4.2.5	Контрольные вопросы	63
5	СОКРАЩЕНИЕ ИЗБЫТОЧНОСТИ ИЗОБРАЖЕНИЙ МЕТОДОМ КОДИРОВАНИЯ С ПРЕОБРАЗОВАНИЕМ	63
5.1	Теоретические основы лабораторной работы.....	64
5.1.1	Проблемы сжатия данных	64
5.1.2	Кодирование с преобразованием	67
5.1.3	Выбор преобразования	67
5.1.4	Кодирование спектральных компонент	71
5.2	Выполнение лабораторной работы.....	73
5.2.1	Общий план выполнения работы.....	73
5.2.2	Порядок проведения обработки.....	74
5.2.3	Содержание отчета.....	74
5.2.4	Контрольные вопросы	75
6	МЕТОД ДИФФЕРЕНЦИАЛЬНОГО КОДИРОВАНИЯ ИЗОБРАЖЕНИЙ	75
6.1	Теоретические основы лабораторной работы.....	75
6.1.1	Общее описание метода ДИКМ.....	75
6.1.2	Алгоритмы предсказания отсчетов для метода ДИКМ.	78
6.1.3	Квантование разностного сигнала в методе ДИКМ.....	81
6.2	Выполнение лабораторной работы.....	84
6.2.1	Общий план выполнения работы.....	84
6.2.2	Порядок проведения обработки.....	85

6.2.3 Содержание отчета	85
6.2.4 Контрольные вопросы	86
7 МЕТОД КОМПРЕССИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ИЕРАРХИЧЕСКОЙ СЕТОЧНОЙ ИНТЕРПОЛЯЦИИ.....	86
7.1 Теоретические основы лабораторной работы	87
7.1.1 Общая схема метода компрессии на основе ИСИ.....	87
7.1.2 Алгоритм интерполяции для метода ИСИ.....	93
7.2. Порядок выполнения лабораторной работы.....	94
7.2.1 Общий план выполнения работы	95
7.2.2 Контрольные вопросы	96
8 ИНСТРУМЕНТАЛЬНАЯ ПРОГРАММНАЯ СИСТЕМА ОБРАБОТКИ И АНАЛИЗА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ.....	96
8.1 Общее описание системы.....	96
8.2 Библиотека прикладных модулей.....	97
8.2.1 Выполнение прикладных модулей	98
8.2.2 Генерация детерминированных полей (Gnd.dll)	102
8.2.3 Генерация случайных полей (Gnr.dll)	104
8.2.4 Обработка скользящим окном 3x3 (Local3.dll)	111
8.2.5 Рекурсивная обработка окном произвольного размера (Local.dll).....	114
8.2.6 Поэлементные преобразования (Clc.dll)	118
8.2.7 Генерация таблиц для поэлементных преобразований (Table.dll).....	121
8.2.8 Оценка статистических характеристик изображений (Stat.dll).....	123
8.2.9. Оценка и анализ автоковариационной функции (ACF.dll).....	126
8.2.10 Геометрические преобразования (Geom.dll).....	131
8.2.11 Спектральные преобразования (Spectr.dll)	134
8.2.12 Работа с фрагментами изображений (Frg.dll).....	138
8.2.13 Работа с атрибутами и пикселями изображений (Img.dll).....	142
ГЛОССАРИЙ	146
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	147

1 ПОЭЛЕМЕНТНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ

Цель работы - изучение и экспериментальное исследование простых поэлементных методов цифровой обработки видеoinформации; получение навыков работы с автоматизированной системой обработки изображений.

В лабораторной работе изучаются некоторые простые поэлементные методы цифровой обработки видеoinформации; даются примеры их практического применения при работе с помощью автоматизированной системы обработки изображений.

1.1 Теоретические основы лабораторной работы

1.1.1 *Общее описание метода*

Как известно, при цифровой обработке изображения обычно используется его представление в памяти ЭВМ в виде прямоугольной $M \times N$ -матрицы (двумерного массива) чисел - элементов функции яркости -

$f(m, n), 0 \leq m \leq M-1, 0 \leq n \leq N-1$ Обработка изображения в общем случае заключается в выполнении какого-либо преобразования указанной матрицы, в результате которого формируется набор ее числовых характеристик или новое, обработанное изображение $g(k, l), 0 \leq k \leq K-1, 0 \leq l \leq L-1$. Преобразование может касаться значений элементов или их координат (индексов), выполняться над матрицей в целом, группой элементов или над каждым элементом в отдельности.

В данной лабораторной работе рассматривается простейший вид цифровой обработки изображений, заключающийся в выполнении одного и

того же функционального преобразования для каждого элемента матрицы вне зависимости от его положения и значений других (соседних) элементов. Такая обработка получила название пространственно-инвариантного поэлементного преобразования изображений. Она переводит значение каждого элемента f в новое значение g :

$$g = g(f). \quad (1.1)$$

Размеры входного и выходного изображений здесь, очевидно, совпадают ($K=M, N=L$).

При практической реализации поэлементных преобразований можно непосредственно вычислять каждое значение преобразованного элемента в соответствии с конкретным видом функции (1.1). Однако для достаточно сложных функций такое построение процедуры обработки оказывается неудобным из-за больших затрат машинного времени на вычисления. Скорость обработки возрастает при переходе к табличному заданию функции преобразования. Алгоритм работы с таблицей очень прост: по значению f вычисляется адрес (номер строки) таблицы с выходным значением g . Очевидные преимущества такого подхода - высокое быстродействие, а также гибкость процедуры обработки (таблица преобразования по сути является параметром процедуры и может легко меняться). Недостаток - приближенность результатов из-за ограниченного числа строк таблицы.

Указанный недостаток снимается, если сама функция принимает конечное число квантованных значений. На практике обычно встречается именно такая ситуация, т.к. почти все устройства ввода изображений в ЭВМ формируют целочисленные элементы яркости, лежащие в ограниченном диапазоне. Например, часто изображения в памяти ЭВМ представляются в формате «байт на элемент», при котором каждый элемент функции яркости кодируется восьмиразрядным двоичным числом без знака и, следовательно, может принимать целочисленные значения $0, 1, 2, \dots, 255$. Очевидно, в данном случае таблица преобразования должна содержать всего 256 строк (по числу значений элементов исходного изображения f), а сами эти элементы могут интерпретироваться как адреса таблицы с выходными значениями g .

Однако такая дискретность (в частном случае, целочисленность) элементов имеет и отрицательные последствия. Истинная функция яркости лишь приближенно описывается своими квантованными значениями, т.е. в цифровые данные вносится некоторая погрешность (шум) квантования.

В результате поэлементных преобразований шум квантования может резко возрасти. Это явление иллюстрирует рис. 1.1, где для значений f и g выбраны равномерно расположенные 16 квантованных уровней.

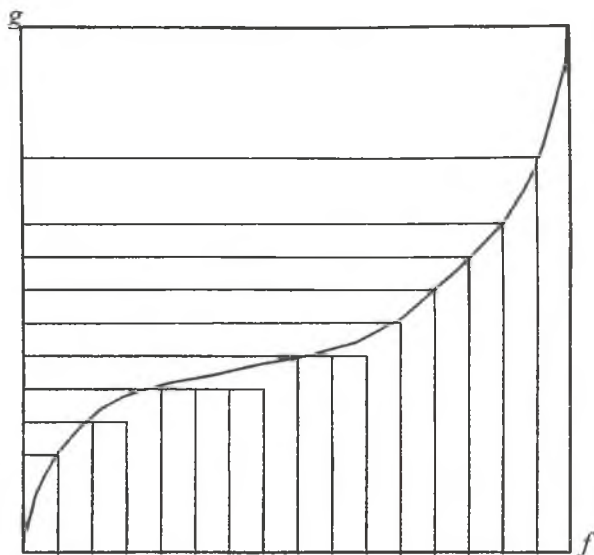


Рис. 1.1 Перераспределение значений яркости по уровням квантования при нелинейном поэлементном преобразовании

Пунктирная кривая на этом рисунке отображает функцию преобразования (1.1). На участках, где производная функции преобразования мала, происходит слияние квантованных уровней изображения f , т.е. разным его значениям соответствует одинаковый результат. Там, где производная велика, происходит увеличение расстояния между используемыми квантованными уровнями в преобразованном изображении g . В результате фактическое число квантованных уровней после преобразования уменьшается и, следовательно, погрешность квантования возрастает. Это обстоятельство нужно учитывать при выборе числа квантованных уровней исходного изображения и при необходимости выбирать его с запасом.

Несмотря на свою простоту, метод поэлементных преобразований позволяет решить довольно много прикладных задач улучшения качества и анализа изображений. Рассмотрим некоторые из них.

1.1.2 Линейное контрастирование

Изображения, вводимые в ЭВМ, часто являются малоконтрастными, т.е. у них вариации функции яркости малы по сравнению с ее средним значением. Реальный динамический диапазон яркостей $[f_{\min}, f_{\max}]$ для таких изображений оказывается намного меньше допустимого диапазона (шкалы яркости). Задача контрастирования заключается в «растягивании» реального динамического диапазона на всю шкалу.

Контрастирование можно осуществить при помощи линейного поэлементного преобразования

$$g = af + b. \quad (1.2)$$

Параметры этого преобразования - a , b - нетрудно определить, исходя из требуемого изменения динамического диапазона. Если в результате обработки нужно получить шкалу $[g_{\min}, g_{\max}]$, то, как следует из (1.2),

$$\begin{cases} g_{\min} = af_{\min} + b, \\ g_{\max} = af_{\max} + b. \end{cases}$$

Отсюда

$$a = \frac{g_{\max} - g_{\min}}{f_{\max} - f_{\min}}; \quad b = \frac{g_{\min}f_{\max} - g_{\max}f_{\min}}{f_{\max} - f_{\min}}.$$

В частном случае, если $g_{\min} = 0$, $g_{\max} = 255$, имеем:

$$a = \frac{255}{f_{\max} - f_{\min}}; \quad b = - \frac{255f_{\min}}{f_{\max} - f_{\min}}.$$

При диалоговой обработке изображений иногда проще не определять параметры преобразования (1.2), а непосредственно строить его в табличной форме, ориентируясь на границы распределения вероятностей элементов функции яркости (короче - распределения яркости). Так, если плотность распределения имеет вид, показанный на рис. 1.2а, то линейное контрастирование можно осуществить с помощью поэлементного преобразования, соответствующего графику на рис. 1.2б.

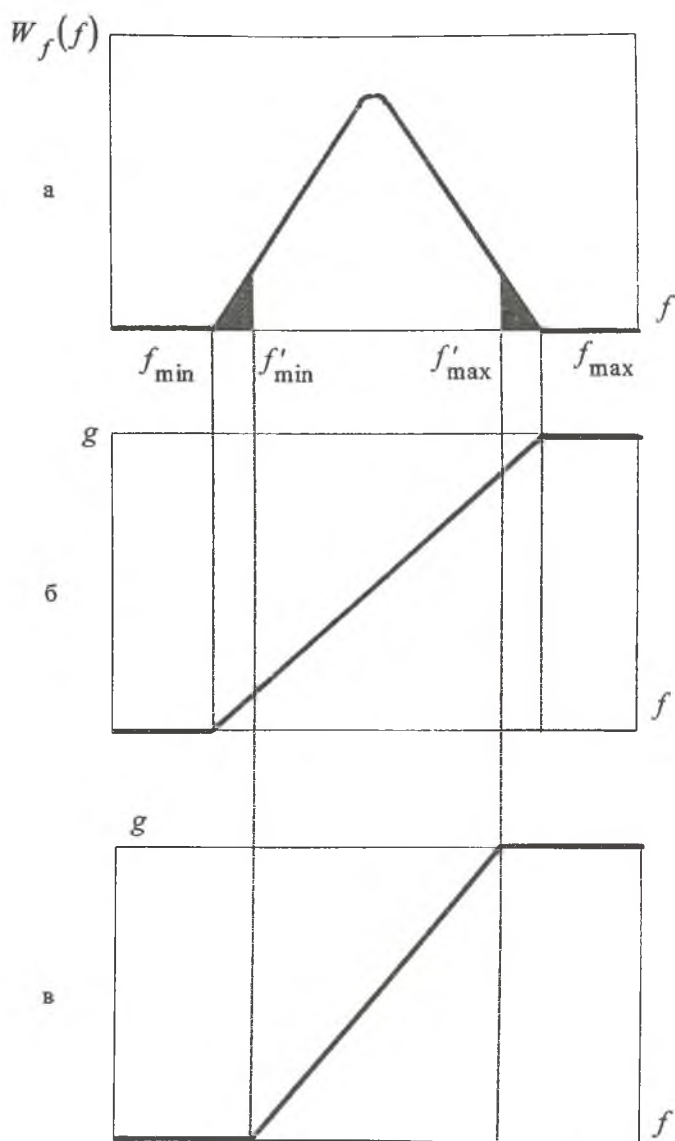


Рис. 1.2 Контрастирование:
 (а) – исходная плотность распределения;
 (б), (в) – возможные функции поэлементного преобразования

Этот способ удобен еще тем, что позволяет не рассматривать истинные экстремальные значения яркости - $f_{\min}, f_{\max}$, а назначать их с некоторыми доверительными вероятностями (см. функцию преобразования на рис. 1.2в, построенную по величинам $f'_{\min}, f'_{\max}$). Эффект контрастирования в этом случае получается сильнее, хотя, естественно, маловероятные элементы, вышедшие после преобразования за пределы шкалы $[g_{\min}, g_{\max}]$, будут потеряны (их значения будут приняты равными $g_{\min}$ или $g_{\max}$).

Оценку плотности распределения яркостей в виде гистограммы получают в результате предварительного анализа изображения.

1.1.3 Преобразование гистограмм

Еще одна процедура повышения контраста изображений заключается в приведении плотности распределения яркостей к некоторому «стандартному» виду. Она реализуется при помощи нелинейного преобразования элементов изображения и, поскольку на практике мы имеем дело с экспериментально полученными гистограммами распределения яркости, называется преобразованием гистограммы.

Чтобы выполнить преобразование гистограммы, нужно сначала найти вид функции поэлементного преобразования (1.1).

Пусть случайная величина f имеет плотность распределения вероятности $W_f(f)$. И пусть $g(f)$ - некоторая монотонно возрастающая функция. Преобразованная величина $g = g(f)$ также случайна и имеет плотность распределения $W_g(g)$. Рассмотрим интегральные функции распределения вероятностей:

$$F_f(f) = \int_{-\infty}^f W_f(\xi) d\xi; \quad F_g(g) = \int_{-\infty}^g W_g(\eta) d\eta.$$

Если случайная величина f принимает значение, меньшее некоторого произвольного f_0 , то по известному свойству интегральной функции распределения вероятность этого события

$$P[f < f_0] = F_f(f_0). \quad (1.3)$$

Аналогично можно записать:

$$P[g < g_0] = F_g(g_0). \quad (1.4)$$

Если $g_0 = g(f_0)$, то в силу возрастания функции преобразования, события, указанные в (1.3) и (1.4), жестко связаны (одно является следствием другого) и их вероятности равны. Приравнивая (1.3) и (1.4) и отбрасывая ставший ненужным индекс, получаем:

$$F_f(f) = F_g[g(f)]. \quad (1.5)$$

Зная требуемый вид плотности распределения вероятностей $W_g(g)$, а значит и $F_g(g)$, из соотношения (1.5) можно выразить функцию поэлементного преобразования $g(f)$.

Покажем, как это делается на примере решения задачи эквализации (выравнивания) гистограммы. В данном случае требуется получить такое изображение, у которого все значения яркости в пределах заданного динамического диапазона $[g_{\min}, g_{\max}]$ равновероятны (рис. 1.3а):

$$W_g(g) = \frac{1}{g_{\max} - g_{\min}}.$$

Интегральная функция распределения на том же интервале получается линейной (рис. 1.3б):

$$F_g(g) = \frac{g - g_{\min}}{g_{\max} - g_{\min}}.$$

Подставив это выражение в (1.5), после элементарных преобразований получаем:

$$g(f) = (g_{\max} - g_{\min}) F(f) + g_{\min}.$$

В частном случае, если $g_{\min} = 0$, $g_{\max} = 255$, имеем:

$$g(f) = 255 F(f).$$

При реализации преобразования гистограмм удобно использовать табличное задание функции поэлементного преобразования $g(f)$. Таблица

$g(f)$ строится следующим образом. Сначала определяется гистограмма распределения яркостей исходного изображения - $W_f(f)$. Затем по массиву значений гистограммы рассчитываются значения интегральной функции распределения $F_f(f)$. И, наконец, массив интегральной функции по соответствующей формуле преобразуется в таблицу значений функции $g(f)$.

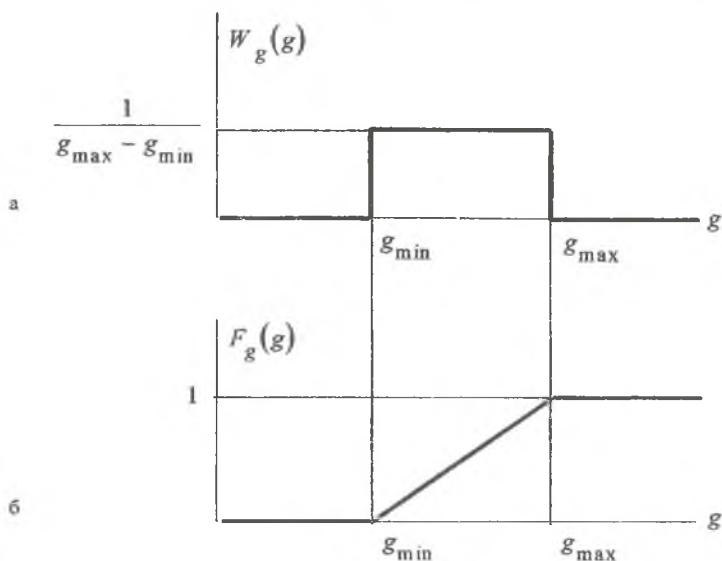


Рис. 1.3 Эквиализация:

- (а) – результирующая плотность распределения;
 (б) – функция поэлементного преобразования

1.1.4 Пороговая обработка

Некоторые задачи обработки видеoinформации связаны с преобразованием полутонового изображения (т.е. такого, которое имеет много градаций яркости) в бинарное (двухградационное). Такое преобразование осуществляется в первую очередь для того, чтобы сократить информационную избыточность изображения, оставив в нем только ту информацию, которая нужна для решения конкретной задачи. В бинарном изображении (или, как говорят, графическом препарате полутонового) должны быть сохранены интересующие нас детали (например, очертания изображенных объектов и исключены несущественные особенности (фон)).

В ряде случаев требуемый графический препарат удастся получить в результате пороговой обработки полутонового изображения. Она заключается в разделении всех элементов изображения на два класса по признаку яркости, т.е., например, в выполнении поэлементного преобразования вида:

$$g = \begin{cases} g_{\max} & \text{при } f \geq f_0, \\ g_{\min} & \text{при } f < f_0, \end{cases}$$

где f_0 - некоторое «пороговое» значение яркости.

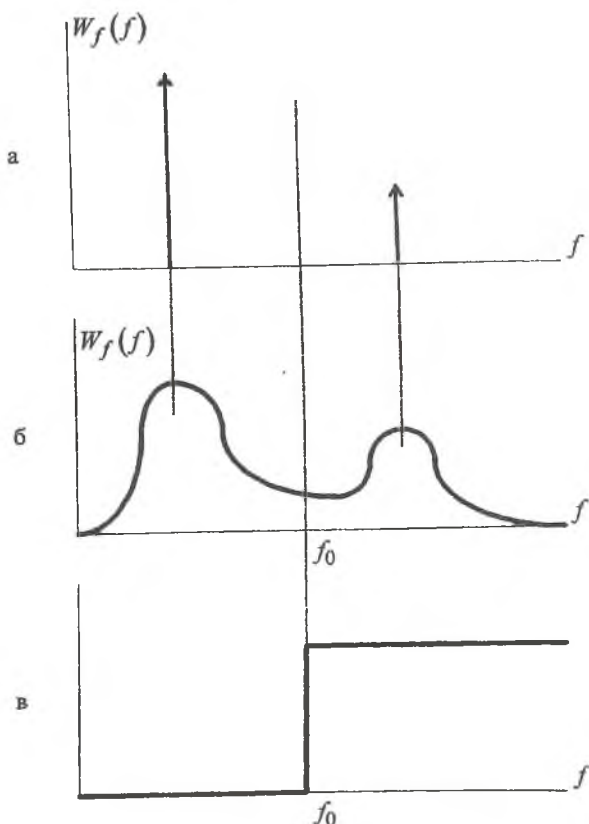


Рис. 1.4 Пороговая обработка:

- (а) – идеальная исходная плотность распределения;
 (б) – реальная исходная плотность распределения;
 (в) – функция поэлементного преобразования

При выполнении пороговой обработки основной вопрос состоит в выборе порога f_0 . Пусть полутоновое изображение содержит интересующие нас объекты одной яркости на фоне другой яркости (типичные примеры: машинописный текст, чертежи, медицинские пробы под микроскопом и т.д.). Тогда в идеале плотность распределения яркостей должна выглядеть как две дельта-функции (рис. 1.4а). В данном случае задача установления порога тривиальна: в качестве f_0 можно взять любое значение между «пиками».

На практике, однако, встречаются определенные трудности, связанные с тем, что, во-первых, изображение искажено шумом, и, во-вторых, как для объектов, так и для фона характерен некоторый разброс яркостей. В результате пики функции плотности распределения «расплываются», хотя обычно ее бимодальность сохраняется (рис. 1.4б). В такой ситуации можно выбрать порог f_0 , соответствующий положению минимума между модами, т.е. использовать функцию поэлементного преобразования, показанную на рис. 1.4в.

В общем случае гистограммы распределения яркостей, измеренные по реальным изображениям, могут оказаться унимодальными или, наоборот, иметь «изрезанный», многомодальный характер. Существуют методики определения порога и в этих ситуациях, однако в рамках данной работы они не рассматриваются.

1.2 Выполнение лабораторной работы

1.2.1 Общий план выполнения работы

1. Изучить метод и описанные задачи поэлементных преобразований изображений.
2. Ознакомиться с описанием программного обеспечения системы обработки изображений и командами входного языка системы.
3. Составить последовательность команд (план проведения обработки) с указанием значений параметров и с комментариями для решения поставленной задачи.
4. Согласовать план проведения обработки с преподавателем, получить разрешение на практическую работу на системе.

5. Выполнить обработку изображений.
6. Составить отчет о выполненной работе.
7. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

1.2.2 Варианты выполнения лабораторной работы

Вариант 1. Линейное контрастирование

1. По указанию преподавателя ввести изображение размером 256x256 элементов.
2. Измерить гистограмму распределения яркостей введенного изображения, определить диапазон ее значений и вывести в виде графика.
3. По графику оценить максимальное и минимальное значение элементов, построить функцию поэлементного преобразования для контрастирования, вывести ее график на экран.
4. Произвести поэлементное преобразование изображения, вывести преобразованное изображение, сравнить с исходным.

Вариант 2. Эквиализация гистограммы

5. По указанию преподавателя ввести изображение размером 256x256 элементов.
6. Измерить гистограмму распределения яркостей введенного изображения, определить диапазон ее значений и вывести в виде графика.
7. Построить функцию поэлементного преобразования для эквиализации гистограммы, вывести ее график на экран.
8. Произвести поэлементное преобразование изображения, вывести преобразованное изображение, сравнить с исходным.
9. Измерить гистограмму распределения яркостей преобразованного изображения, вывести ее в виде графика.

Вариант 3. Пороговая обработка

1. По указанию преподавателя ввести изображение размером 256x256 элементов.

2. Измерить гистограмму распределения яркостей введенного изображения, определить диапазон ее значений и вывести в виде графика.
3. По графику выбрать значение порога, построить функцию поэлементного преобразования для пороговой обработки, вывести ее график на экран.
4. Произвести поэлементное преобразование изображения, вывести преобразованное изображение.

1.2.3 Содержание отчета

Отчет по работе должен содержать:

1. Последовательность команд обработки с комментариями и числовыми значениями параметров.
2. Перечень созданных полей на рабочем файле с указанием их смыслового содержания и имен.
3. Зарисованные графики с экрана монитора с необходимыми пояснениями.
4. Выводы по результатам исследования.

1.2.4 Контрольные вопросы

1. Метод поэлементных преобразований, варианты реализации. Влияние квантования изображений на качество обработки.
2. Линейное контрастирование.
3. Преобразование гистограмм. Общий подход к получению произвольного закона распределения яркостей.
4. Эквализация гистограмм. Функция поэлементного преобразования для эквализации.
5. Пороговая обработка.
6. Препарирование изображений с помощью поэлементных преобразований.
7. Общая характеристика программного обеспечения системы.

2 ЛОКАЛЬНЫЕ МЕТОДЫ ВЫДЕЛЕНИЯ КОНТУРОВ

Цель работы - изучение и экспериментальное исследование методов выделения контуров на изображениях с помощью линейной обработки “окном”; получение навыков работы с автоматизированной системой обработки изображений.

В лабораторной работе изучаются методы выделения контуров на изображении с помощью линейной обработки “окном”; приведены примеры их применения при работе с автоматизированной системой обработки изображений в режиме диалога.

2.1 Теоретические основы лабораторной работы

2.1.1 Линейная локальная обработка изображений

Введенные в ЭВМ изображения часто имеют низкое качество, что затрудняет визуальный анализ и определение параметров изображенных объектов. Ухудшение качества изображений связано со многими причинами: малым динамическим диапазоном видеодатчика, несовершенством оптической части системы регистрации, плохим освещением объекта и др. Визуально такие искажения воспринимаются как дефокусировка, “размытость”, ухудшение резкости, снижение контрастности, появление мелкоструктурных помех (шумов).

Простейший метод повышения качества изображений заключается в пространственно-инвариантном поэлементном преобразовании изображений (см. лабораторную работу №1). При поэлементном преобразовании каждый элемент изображения обрабатывается независимо от других (соседних) элементов. Однако поэлементная обработка не позволяет определять на изображении контуры (границы объектов), которые играют первостепенную роль в восприятии изображенных объектов и, в конечном счете, позволяют классифицировать эти объекты.

В настоящей лабораторной работе рассматривается метод локальной обработки изображений. Локальную обработку проще всего представить себе, рассмотрев перемещающееся по полю изображения “окно” (маску) небольшого размера (его варианты представлены на рис. 2.1), которое

занимает все возможные положения.

При каждом положении “окна” формируется один отсчет (элемент яркости) выходного изображения (обычно соответствующий центру “окна”), который является функцией отсчетов исходного изображения, попавших в “окно”.

Пусть элементы яркости исходного изображения - $f(m, n)$, $0 \leq m \leq M-1$, $0 \leq n \leq N-1$. Тогда отсчеты получаемого изображения

$$e(m, n) = E[f(m+k, n+l) : (k, l) \in W], \quad (2.1)$$

где E - функция преобразования,

а W - область “окна” (точка $(k, l) = (0, 0)$ соответствует его центру).

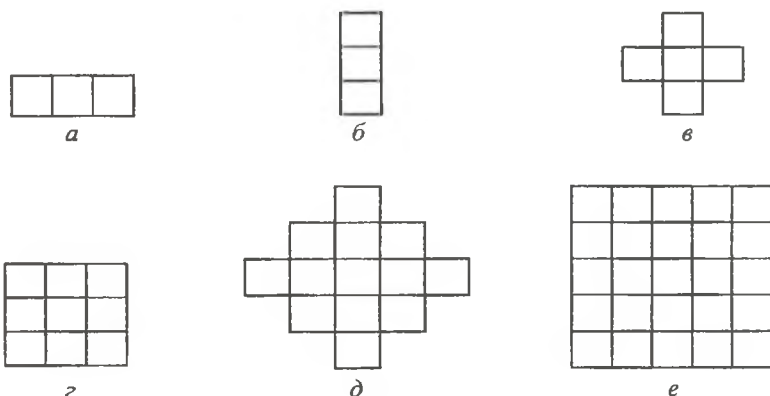


Рис. 2.1 Формы окон при линейной локальной обработке изображений

Например, для окна, показанного на рис. 1г, имеем: $k = -1, 0, 1$ и $l = -1, 0, 1$, т.е.

$$e(m, n) = E[f(m-1, n-1), f(m-1, n), f(m-1, n+1), \\ f(m, n-1), f(m, n), f(m, n+1), \\ f(m+1, n-1), f(m+1, n), f(m+1, n+1)].$$

Из (2.1) следует, что поэлементное преобразование является частным случаем локальной обработки изображений, соответствующей размерам “окна” 1×1 . В общем случае функция E в (2.1) является нелинейной и

может иметь весьма сложный вид. Ясно, что для ее реализации нельзя использовать табличный метод (как при поэлементной обработке), поскольку это потребовало бы неприемлемо больших затрат памяти. Однако для большого числа практических задач локальную обработку можно представить в виде суперпозиции одной или нескольких операций линейной обработки “окном” и поэлементного преобразования. Указанная процедура имеет вид:

$$s_i(m, n) = \sum_{(k, l) \in W} a_i(k, l) f(m - k, n - l) + C_i; i = 1, \dots, I, \quad (2.2)$$

$$e(m, n) = H[s_1(m, n), s_2(m, n), \dots, s_I(m, n)], \quad (2.3)$$

где $s_i(m, n)$ - промежуточные изображения;

$a_i(k, l)$ - коэффициенты линейных масок;

C_i - коэффициенты постоянного смещения;

H - функция совместного поэлементного преобразования изображения $s_i(m, n)$;

I - число параллельных операций линейной обработки “окном”.

В частном случае может быть $I = 1$. В дальнейших рассуждениях для краткости обозначений индекс i будем опускать.

Соотношение (2.2) описывает двумерную свертку функции яркости исходного изображения $f(m, n)$ с импульсной характеристикой $a(k, l)$ некоторой двумерной КИХ-системы. Выбирая определенным образом размеры и коэффициенты маски $a(k, l)$, можно выполнить различные операции по улучшению качества изображения: повышение резкости, сглаживание шумов, подчеркивание контуров, низкочастотную или высокочастотную фильтрацию и т.д. Из соображений простоты обычно применяют “окна” малых размеров: 3×3 или 5×5 .

Коэффициент C в (2.2) не имеет принципиального значения для выполняемых операций. Он необходим для “компенсации” получающихся отрицательных значений отсчетов яркости. (Вспомним, что отсчеты изображения, как правило, кодируются в памяти ЭВМ одним байтом; функция яркости принимает целочисленные значения в диапазоне от 0 до 255).

Иногда требуется, чтобы при обработке изображения линейной маской

(2.2) не менялось среднее значение яркости изображения (его постоянная составляющая). Для выполнения этого условия нужно, чтобы

$$\sum_{(k,l) \in W} a(k,l) = 1, \quad C = 0. \quad (2.4)$$

Если же нужно, чтобы среднее значение яркости после обработки произвольного изображения маской всегда было равно E , следует выбрать

$$\sum_{(k,l) \in W} a(k,l) = 0, \quad C = E. \quad (2.5)$$

При обработке изображения с использованием линейной маски следует учитывать возникающие “краевые эффекты”. Действительно, с практической точки зрения исходное и обработанное изображения удобно считать имеющими одинаковые размеры $M \times N$. Однако из (2.2) следует, что в этом случае получающееся изображение не задано вблизи краев, так как “окно” выходит за границу изображения при некоторых значениях индексов m и n . В этом случае обычно считают, что неопределенные значения элементов яркости вне поля изображения (при $m < 0$, или $n < 0$, или $m \geq M$, или $n \geq N$) равны константе (например, нулю) или значениям ближайших определенных точек.

2.1.2 Постановка задачи выделения контуров

Любое изображение в идеальном случае представляет собой набор однородных областей различной яркости (или, по меньшей мере, допускают такое представление с некоторой погрешностью). В простейшем случае бинарного (двухградационного) изображения имеем области, содержащие отсчеты двух значений яркости, которые соответствуют объектам и фону. Исследованиями психологов установлено, что с точки зрения распознавания и анализа объектов на изображении наиболее информативными являются не значения яркостей объектов, а характеристики их границ - контуров. Другими словами, основная информация заключена не в яркости (или цвете) отдельных областей, а в их очертаниях. Задача выделения контуров состоит в построении изображения именно границ объектов и очертаний однородных областей.

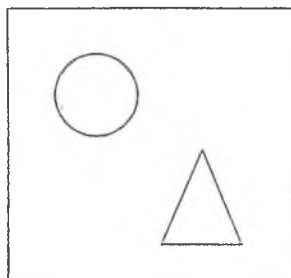
На рис. 2.2а,б показаны соответственно исходное изображение,

состоящее из областей различной яркости, и его графический вариант, состоящий только из границ этих областей.

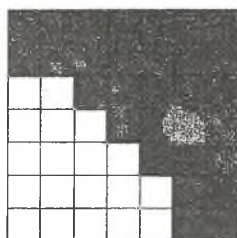
Будем называть контуром изображения совокупность его отсчетов, в окрестности которых наблюдается скачкообразное изменение функции яркости. Так как при цифровой обработке изображение представлено как функция целочисленных аргументов, то контуры представляются линиями шириной как минимум в один элемент. При этом может возникнуть неоднозначность в определении линии контура, как то показано на рис. 2.2г,д, для исходного изображения с перепадом яркости (рис. 2.2в).



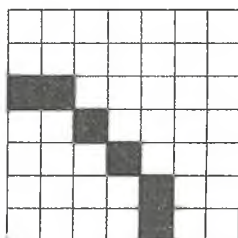
а



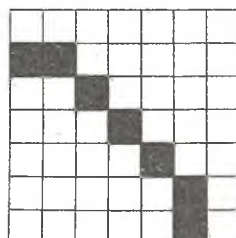
б



в



г



д

Рис. 2.2 Выделение контуров:

(а) – исходное изображение; (б) – контуры;

(в) – часть исходного контура;

(г), (д) – неоднозначность при выделении контура

Если исходное изображение кроме областей постоянной яркости содержит участки с плавно меняющейся яркостью, то введенное определение контура остается справедливым, однако при этом не

гарантируется непрерывность контурных линий: разрывы контуров будут наблюдаться в тех местах, где изменение функции яркости не является достаточно резким (этот эффект иллюстрируется на рис. 2.3).



Рис. 2.3 Разрыв контурной линии на участке с плавно меняющейся яркостью

С другой стороны, если на “кусочно-постоянном” изображении присутствует шум, то, возможно, будут обнаружены “лишние” контуры в точках, которые не являются границами областей.

При разработке алгоритмов выделения контуров нужно учитывать указанные особенности поведения контурных линий. Специальная дополнительная обработка выделенных контуров позволяет устранять разрывы и подавлять ложные контурные линии.

Общую процедуру построения бинарного изображения границ объектов иллюстрирует блок-схема, представленная на рис. 2.4.

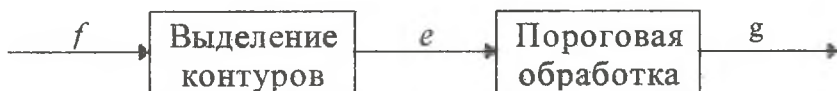


Рис. 2.4 Общая процедура построения бинарного изображения границ объектов

Исходное изображение f подвергается линейной или нелинейной обработке, с тем чтобы выделить перепады яркости. В результате этой операции формируется изображение e , функция яркости которого существенно отличается от нуля только в областях резких изменений изображения f . Затем в результате пороговой обработки из изображения e формируется графический (контурный) препарат g . Правильный выбор порога на втором этапе должен производиться из следующих соображений:

- при слишком высоком пороге могут появиться разрывы контуров, а слабые перепады яркости не будут обнаружены;

- при слишком низком пороге из-за шумов и неоднородности областей могут появиться ложные контуры.

Других особенностей пороговая обработка не имеет. Поэтому обратим основное внимание на первую операцию - **выделение перепадов яркости (контуров)** - и рассмотрим основные методы выполнения этой операции.

2.1.3 Градиентный метод

Одним из наиболее простых способов выделения границ является пространственное дифференцирование функции яркости. Для одномерной непрерывной функции яркости $f(x)$ этот способ иллюстрирует рис. 2.5.

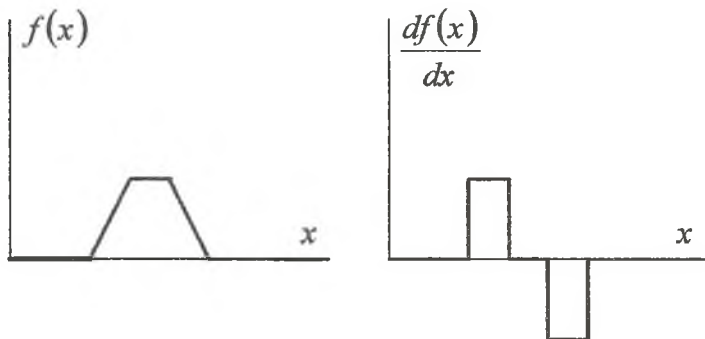


Рис. 2.5 Пространственное дифференцирование функции яркости

Для двумерной функции яркости $f(x,y)$ (считаем пока, что координаты x, y изменяются непрерывно) перепады в направлениях x и y регистрируются частными производными $\frac{\partial f(x,y)}{\partial x}$ и $\frac{\partial f(x,y)}{\partial y}$, которые

пропорциональны скоростям изменения яркости в соответствующих направлениях. Выделение перепадов яркости в двумерном случае иллюстрируется рис. 2.6 (на нем серый цвет соответствует нулевому значению функции, черный - отрицательному, а отсутствие цвета - положительному). Видим, что подчеркивание контуров, перпендикулярных к оси x , обеспечивает $\frac{\partial f(x,y)}{\partial x}$ (рис. 2.6б), а подчеркивание контуров,

перпендикулярных к оси y , $\frac{\partial f(x,y)}{\partial y}$ (рис. 2.6).

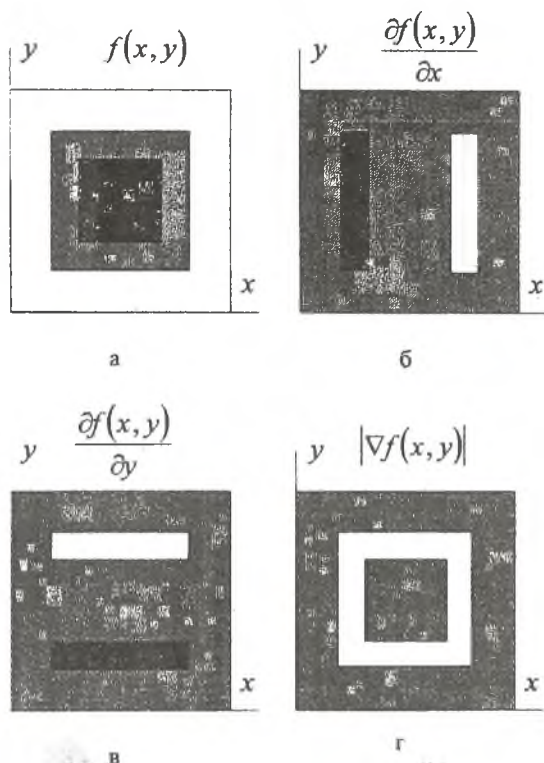


Рис. 2.6 Выделение перепадов яркости в двумерном случае:

- (а) – исходное изображение;
 (б) – подчеркивание контуров, перпендикулярных к оси x ;
 (в) – подчеркивание контуров, перпендикулярных к оси y ;
 (г) – подчеркивание всех контуров (модуль градиента)

В практических задачах требуется выделить контуры, направление которых является произвольным. Для этих целей можно использовать модуль градиента функции яркости

$$|\nabla f(x,y)| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}, \quad (2.6)$$

который пропорционален максимальной (по направлению) скорости изменения функции яркости в данной точке и не зависит от направления контура.

Модуль градиента в отличие от частных производных принимает только неотрицательные значения, поэтому на получающемся изображении (рисунок 2.6г) точки, соответствующие контурам, имеют повышенный уровень яркости. Для цифровых изображений аналогами частных производных и модуля градиента являются функции, содержащие дискретные разности, например:

$$\frac{\partial f(x, y)}{\partial x} \rightarrow \Delta_x f(m, n) = f(m, n) - f(m-1, n);$$

$$\frac{\partial f(x, y)}{\partial y} \rightarrow \Delta_y f(m, n) = f(m, n) - f(m, n-1);$$

$$|\nabla f(x, y)| \rightarrow |\nabla f(m, n)| = \sqrt{[f(m, n) - f(m-1, n)]^2 + [f(m, n) - f(m, n-1)]^2}$$

Таким образом, операция выделения контуров заключается в выполнении нелинейной совместной поэлементной обработке изображений s_1 и s_2 (см. формулу (2.3)) следующего вида:

$$e(m, n) = \sqrt{s_1^2 + s_2^2}, \quad (2.7)$$

каждое из которых получается в результате локальной обработки исходного изображения "окнами" 2×1 и 1×2 :

$$\begin{aligned} s_1 &= f(m, n) - f(m-1, n), \\ s_2 &= f(m, n) - f(m, n-1). \end{aligned} \quad (2.8)$$

Для вычисления модуля градиента вместо частных производных $\frac{\partial f}{\partial x}$ и $\frac{\partial f}{\partial y}$ можно использовать производные по любой паре перпендикулярных направлений. Если в качестве этих направлений взять диагональные и применить соответствующие дискретные разности

следующего вида:

$$\begin{aligned} s_1 &= f(m, n) - f(m-1, n-1), \\ s_2 &= f(m, n-1) - f(m-1, n), \end{aligned} \quad (2.9)$$

то формула (2.7) превратится в так называемый оператор Робертса, который также является приближением модуля непрерывного градиента (2.6).

При реализации процедуры выделения контуров стремятся избежать наиболее трудоемких операций (умножение, извлечение корня), поскольку их приходится выполнять для каждой точки изображения. Для приближенного вычисления модуля градиента часто используют упрощенные выражения, например (используя формулу (2.8)):

$$e(m, n) = |s_1| + |s_2|; \quad (2.10)$$

$$e(m, n) = \max \{ |s_1|; |s_2| \}; \quad (2.11)$$

Однако приближения (2.10)-(2.11) уже не являются одинаково чувствительными к границам при всех их ориентациях: (2.10) более чувствительно к диагонально ориентированным границам, а (2.11) - к вертикальным и горизонтальным.

Иногда используются некоторые другие приближения градиента, например:

$$e(m, n) = \sum_{(k, l) \in W} |f(m, n) - f(m+k, n+l)|, \quad (2.12)$$

$$e(m, n) = \max_{(k, l) \in W} |f(m, n) - f(m+k, n+l)|, \quad (2.13)$$

где W область “окна” (целесообразно использовать “окна”, показанные на рис. 2.1в и 2.1г).

2.1.3 Метод оператора Лапласа

В отличие от градиентного метода выделения границ, базирующегося на использовании первой производной по направлению, метод оператора

Лапласа использует производные второго порядка. Если функция яркости $f(m,n)$ является функцией непрерывных аргументов, то лапласиан (оператор Лапласа) определяется следующим образом:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \quad (2.14)$$

В отличие от градиента лапласиан является скалярной функцией. Его применение для выделения границ основано на том, что лапласиан принимает максимальное (по абсолютной величине) значение на участках “перегибов” функции яркости.

По сравнению с градиентом лапласиан обладает недостатком, заключающимся в раздваивании границы, если она не является достаточно резкой, т.е. если функция яркости содержит участки постоянного наклона. В этом случае необходимо применять специальную дополнительную обработку полученного графического препарата для устранения раздваивания линий.

В дискретном случае оператор Лапласа может быть реализован в виде процедуры локальной линейной обработки по “окну” 3×3 :

$$\begin{aligned} \frac{\partial^2 f(x, y)}{\partial x^2} &\rightarrow f(m+1, n) - 2f(m, n) + f(m-1, n); \\ \frac{\partial^2 f(x, y)}{\partial y^2} &\rightarrow f(m, n+1) - 2f(m, n) + f(m, n-1); \\ \nabla^2 f(x, y) &\rightarrow f(m+1, n) + f(m-1, n) + \\ &\quad + f(m, n+1) + f(m, n-1) - 4f(m, n). \end{aligned} \quad (2.15)$$

Так как лапласиан может принимать как положительные, так и отрицательные значения, в операторе выделения контуров следует взять его абсолютное значение:

$$e(m, n) \rightarrow |f(m+1, n) + f(m-1, n) + f(m, n+1) + f(m, n-1) - 4f(m, n)|. \quad (2.16)$$

Другой вариант дискретного лапласиана получается, если рассмотреть взаимно перпендикулярные диагональные направления для аппроксимации вторых производных:

$$\nabla^2 f(x, y) \rightarrow \frac{\frac{f(m+1, n+1) - f(m, n)}{\sqrt{2}} - \frac{f(m, n) - f(m-1, n-1)}{\sqrt{2}}}{\sqrt{2}} + \frac{\frac{f(m+1, n-1) - f(m, n)}{\sqrt{2}} - \frac{f(m, n) - f(m-1, n+1)}{\sqrt{2}}}{\sqrt{2}}. \quad (2.17)$$

В этом случае оператор выделения контуров принимает вид:

$$e(m, n) = \frac{1}{2} \left| f(m+1, n+1) + f(m+1, n-1) + f(m-1, n+1) + f(m-1, n-1) - 4f(m, n) \right|. \quad (2.18)$$

Множество различных дискретных лапласианов получается суммированием аппроксимаций (2.15) и (2.17) с разными весами. Например, взяв весовые коэффициенты равными $1/3$ и $2/3$, получаем еще один вариант оператора выделения контуров:

$$e(m, n) = \frac{1}{3} \left| f(m+1, n) + f(m-1, n) + f(m, n+1) + f(m, n-1) + f(m+1, n+1) + f(m+1, n-1) + f(m-1, n+1) + f(m-1, n-1) - 8f(m, n) \right|. \quad (2.19)$$

Операторам (2.16), (2.18), (2.19) соответствуют следующие линейные маски:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \quad (2.20)$$

Заметим, что операторы Лапласа (2.16), (2.18), (2.19) являются практически нечувствительными к ориентации выделяемых границ, что является преимуществом по сравнению с аппроксимациями градиента (2.10), (2.11).

2.1.4 Метод согласования

Общим недостатком рассмотренных выше методов выделения границ является высокая чувствительность к шуму. Это объясняется тем, что действие дифференциальных (разностных) операторов состоит в вычислении и комбинировании разностей отсчетов в пределах “окна” малых размеров. Из теории цифровой обработки сигналов известно, что операция дифференцирования приводит к ослаблению низкочастотных составляющих пространственной функции яркости и к усилению высокочастотных составляющих. В то же время высокочастотная составляющая функции яркости в большой степени обусловлена именно действием шума. Поэтому операторы градиента и лапласиана усиливают шум, что ведет к появлению ложных контуров.

Чтобы повысить помехоустойчивость рассмотренных методов, перед выполнением операции дифференцирования производят сглаживание функции яркости в пределах “окна”, например, “подгоняют” или согласуют с функцией яркости выбранную поверхность первого или второго порядков. Такой подход приводит к методу согласования.

Рассмотрим принцип метода согласования на примере “окна” 3×3 . По наблюдаемым значениям функции яркости в пределах этого “окна”

$$\{f(m, n), f(m, n-1), f(m-1, n), f(m-1, n-1)\}$$

построим аппроксимирующую плоскость

$$\bar{f}(x, y) = \alpha x + \beta y + \gamma. \quad (2.21)$$

Определив коэффициенты α, β, γ , можно вычислить частные производные и искомый модуль градиента:

$$\frac{\partial \bar{f}(x, y)}{\partial x} = \alpha, \quad \frac{\partial \bar{f}(x, y)}{\partial y} = \beta, \quad |\nabla \bar{f}(x, y)| = \sqrt{\alpha^2 + \beta^2}.$$

Неизвестные коэффициенты можно определить по методу наименьших квадратов, найдя минимум функционала:

$$\varepsilon^2 = \sum_{k=0}^1 \sum_{l=0}^1 [\bar{f}(m-k, n-l) - f(m-k, n-l)]^2 \rightarrow \min_{\alpha, \beta, \gamma}.$$

В результате этой минимизации получаем:

$$\alpha = \frac{f(m, n) + f(m, n-1)}{2} - \frac{f(m-1, n) + f(m-1, n-1)}{2}, \quad (2.22)$$

$$\beta = \frac{f(m, n) + f(m-1, n)}{2} - \frac{f(m, n-1) + f(m-1, n-1)}{2}. \quad (2.23)$$

Несложно видеть, что коэффициенты α и β формируются при линейной обработке изображения в “окне” 2×2 соответственно масками

$$\frac{1}{2} \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}, \quad \frac{1}{2} \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}, \quad (2.24)$$

то есть могут быть интерпретированы как промежуточные изображения s_1 и s_2 в формуле (2.3).

Частные дискретные разности, рассчитанные по (2.22) и (2.23), являются менее чувствительными к помехам, так как эти разности считаются по усредненным (по горизонтали или вертикали) значениям двух соседних отсчетов изображения. Такое усреднение (сглаживание) перед вычислением дискретной разности приводит к уменьшению шума.

Рассмотренный метод построения аппроксимирующей плоскости допускает очевидное обобщение на “окно” произвольных размеров $K \times L$. Следует, однако, заметить, что размеры “окна” должны быть меньше размеров любой области постоянной яркости на изображении. Иначе вместе со сглаживанием шумов будут “сглаживаться” (и исчезать!) мелкие объекты на изображении.

При аппроксимации плоскостью (2.21) отсчетов в “окне” размером 3×3 по аналогии с предыдущим случаем можно получить следующие маски для формирования изображений s_1 и s_2 :

$$\frac{1}{6} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad \frac{1}{6} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}. \quad (2.25)$$

Вычисление модуля градиента $\sqrt{s_1^2 + s_2^2}$ с использованием масок (2.25) определяет так называемый **оператор Превитта**, который часто

используется на практике. Здесь также можно применить описанные ранее аппроксимации модуля градиента (2.10) или (2.11).

На “окне” размером 3×3 можно построить и аппроксимирующую поверхность второго порядка:

$$\bar{f}(x, y) = \alpha x^2 + \beta y^2 + \gamma xy + \delta x + \zeta y + \xi. \quad (2.26)$$

Тогда величина лапласиана по методу согласования будет равна

$$\nabla^2 \bar{f}(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} = 2\alpha + 2\beta,$$

где коэффициенты α и β формируются масками:

$$\frac{1}{6} \begin{pmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{pmatrix}, \frac{1}{6} \begin{pmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{pmatrix}.$$

Очевидно, дискретный лапласиан может быть вычислен при помощи единственной маски, являющейся удвоенной суммой указанных масок. Эта маска имеет вид:

$$\frac{1}{3} \begin{pmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{pmatrix}. \quad (2.27)$$

На примере лапласиана продемонстрируем, что метод согласования обеспечивает большую помехоустойчивость процедуры выделения контуров по сравнению с методами непосредственного вычисления лапласиана с использованием масок (2.20). В качестве критерия помехоустойчивости будем использовать коэффициент Q усиления мощности белого шума, который показывает во сколько раз увеличивается дисперсия аддитивного белого шума при обработке изображения “окном” с линейной маской. Несложно получить, что

$$Q = \sum_{(k,l) \in W} a^2(k, l),$$

где $a(k, l)$ - коэффициенты линейной маски (2.2).

Для “традиционного” лапласиана с маской $\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} Q=20$.

Для “диагонального” лапласиана с маской $\frac{1}{2} \begin{pmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{pmatrix} Q=5$.

Для “комбинированного” лапласиана с маской $\frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} Q=8$.

Для “согласованного” лапласиана с маской (2.27) $Q=4$, то есть по сравнению с “традиционным” лапласианом, помехоустойчивость повышается в 5 раз.

2.1.5 Алгоритм выделения контуров

При реализации алгоритмов выделения контуров необходимо учитывать некоторые особенности программного обеспечения автоматизированной системы обработки изображений. Эти особенности связаны прежде всего с форматом представления данных (отсчетов функции яркости). Большинство программ пакета ориентировано на работу с изображениями, каждый элемент которых может принимать целочисленные значения из диапазона 0...255. Однако значения лапласиана и проекций градиента принимают и положительные, и отрицательные значения. Для того чтобы не происходило обнуление отрицательных значений дифференциальных операторов, следует предпринять специальные меры.

С этой целью в операторе локальной линейной обработки (2.2) введен коэффициент постоянного смещения C . Если выбрать $C=128$, то отрицательные значения лапласиана и проекций градиента будут отображаться на отрезок $[0,127]$, а положительные - на $[129,255]$. Для получения абсолютных значений дифференциальных операторов можно затем воспользоваться соответствующим поэлементным преобразованием, причем при использовании градиентов метода в форме (2.10) или (2.11) указанную операцию нужно выполнить дважды - для каждой проекции градиента, а затем найти максимум или сумму полученных полей. Впрочем, в используемой программной системе имеется функция, осуществляющая

преобразование входного поля сразу по формуле (2.7) и ее обобщениям.

Следующая операция после выделения контуров - пороговая обработка. Чтобы получить светлые линии контуров на темном фоне, нужно использовать функцию поэлементного преобразования, изображенную на рис. 2.7г. Операции нахождения абсолютной величины и пороговой обработки можно объединить, если используется метод оператора Лапласа. Для этого можно воспользоваться функцией поэлементного преобразования, представленной на рис. 2.7б.

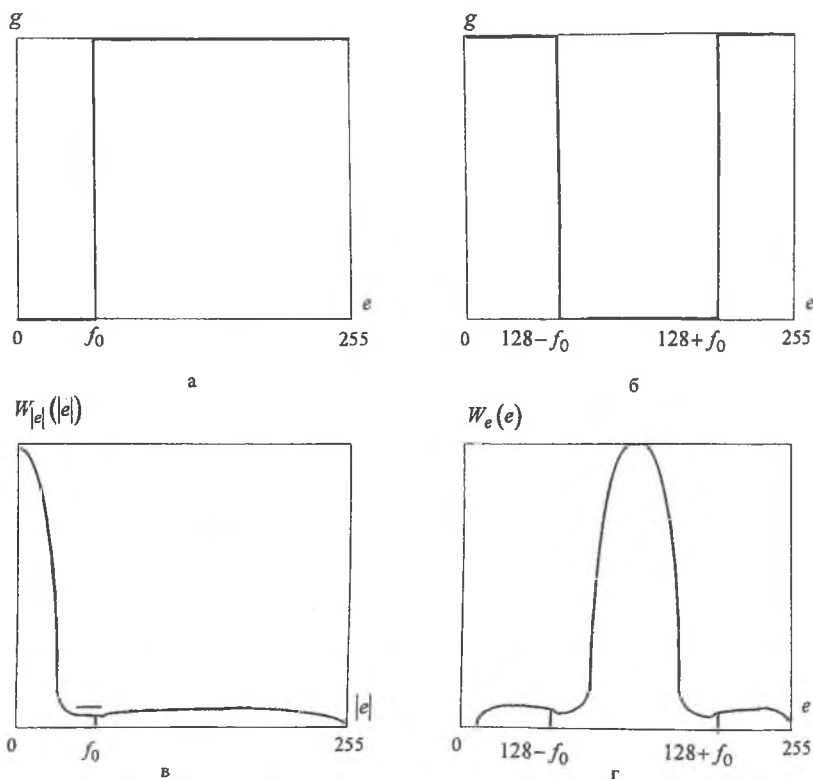


Рис. 2.7 Выставление порога при выделении контуров:
 (в) – плотность распределения модуля градиента яркости;
 (а) – выставление порога для модуля градиента яркости;
 (г) – плотность распределения лапласиана яркости;
 (б) – выставление порога для лапласиана яркости

Проблема заключается в выборе порога f_0 . Его можно определить, проанализировав гистограмму изображения e , полученного в результате

обработки исходного изображения с помощью градиентного метода или лапласиана (рисунок 2.7в,г). Если на гистограмме нет явно выраженных максимумов, соответствующих границам и внутренним областям изображенных объектов, то необходимо выбрать порог экспериментально, визуально контролируя качество получающегося изображения. Заметим, что при использовании масок (2.20), (2.24), (2.25), (2.27) их постоянные множители можно взять единичными, так как они влияют лишь на масштаб гистограмм и, в конечном счете, только на величину порога f_0 .

2.2 Выполнение лабораторной работы

2.2.1 Общий план выполнения работы

1. Изучить методы локальной линейной обработки изображений, используемых для выделения контуров.
2. Ознакомиться с описанием программного обеспечения системы обработки изображений ResLook.
3. Составить последовательность команд с указанием значений параметров и с комментариями для решения задачи выделения контуров следующими методами:
 - 1) метод простого градиента (соотношение (2.7));
 - 2) метод аппроксимации лапласиана (маски (2.20));
 - 3) метод оператора Превитт (маски (2.25));
 - 4) метод согласования для лапласиана (маска (2.27)).
4. Согласовать план проведения обработки с преподавателем, получить разрешение на практическую работу.
5. Выполнить обработку.
6. Составить отчет о выполненной работе.
7. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

2.2.2 Порядок проведения обработки

1. Для заданного изображения выполнить операцию выделения контуров при помощи обработки "окном" с маской, соответствующей

выбранному методу. Вывести на экран промежуточное изображение, полученное в результате обработки “окном”.

2. Рассчитать гистограмму промежуточного изображения с выделенными контурами, вывести ее на экран, произвести выбор порога по гистограмме.
3. Построить функцию поэлементного преобразования для пороговой обработки и вывести ее на экран.
4. Провести пороговую обработку промежуточного изображения и вывести на экран результат обработки.
5. При необходимости повторить выбор порога и пороговую обработку с новым порогом.

2.2.3 Содержание отчета

Отчет по работе должен содержать:

1. Последовательность команд обработки с комментариями и числовыми значениями параметров.
2. Перечень созданных изображений с указанием их смыслового содержания.
3. Зарисованные графики гистограмм и функций для поэлементных преобразований.
4. Выводы по результатам исследования.

2.2.4 Контрольные вопросы

1. Метод локальной линейной обработки. Виды “окон”.
2. Понятие контура. Этапы решения задачи выделения контуров.
3. Градиентный метод выделения контуров и его варианты.
4. Метод оператора Лапласа, его достоинства и недостатки.
5. Выделение контуров методом согласования.
6. Помехоустойчивость алгоритмов выделения контуров.
7. Как выбрать порог для получения изображения контуров?
8. Общая характеристика программного обеспечения

3 ФИЛЬТРАЦИЯ ШУМОВ НА ИЗОБРАЖЕНИЯХ

Цель работы - изучение и экспериментальное исследование методов фильтрации аддитивных и импульсных шумов на изображениях при помощи линейного и медианного фильтров; получение навыков работы на автоматизированной системе обработки изображений.

В лабораторной работе изучаются и экспериментально исследуются методы фильтрации аддитивных и импульсных шумов на изображениях при помощи линейного и медианного фильтров; даются примеры их применения при работе на автоматизированной системе обработки изображений.

3.1 Теоретические основы лабораторной работы

3.1.1 Модели помех при регистрации изображений

Никакая система регистрации не обеспечивает идеального качества изображений исследуемых объектов. Изображения в процессе формирования их изображающими системами (фотографическими, голографическими, телевизионными) обычно подвергаются воздействию различных случайных помех, или шумов. В отличие от так называемых детерминированных искажений, которые часто описываются поэлементными функциональными преобразованиями исходного изображения, для описания случайных воздействий используют модели аддитивного, импульсного и мультипликативного шумов.

Наиболее распространенным видом помех является случайный аддитивный шум, статистически независимый от видеосигнала. Модель аддитивного шума используется тогда, когда сигнал на выходе изображающей системы или на каком-нибудь промежуточном этапе преобразования может рассматриваться как сумма полезного сигнала и некоторого случайного сигнала (шума). Модель аддитивного шума хорошо описывает действие зернистости фотопленки, флуктуационный шум в радиотехнических системах, шум квантования в аналого-цифровых преобразователях и т.п.

Обычно аддитивный шум рассматривают как стационарное случайное

поле и характеризуют дисперсией и корреляционной функцией. Часто аддитивный шум можно считать некоррелированным (белым) или слабокоррелированным. Пусть $x(m,n)$ - отсчёты (элементы функций яркости) исходного (идеального) изображения, а $y(m,n)$ - отсчёты наблюдаемого (зашумлённого) изображения. Тогда действие аддитивного белого шума можно задать выражением:

$$y(m,n) = x(m,n) + v(m,n) , \quad (3.1)$$

где $v(m,n)$ - отсчёты шумового случайного поля с нулевым средним и корреляционной функцией

$$B_v(k,l) = E\{v(m,n)v(m+k,n+l)\} = D_v\sigma(k,l) , \quad (3.2)$$

где $E\{\cdot\}$ - оператор математического ожидания;

D_v дисперсия шума;

$\sigma(k,l)$ - двумерный дельта-импульс.

Интенсивность аддитивного шума характеризуется отношением "сигнал/шум"

$$d^2 = \frac{D_x}{D_v} ,$$

где D_x - дисперсия сигнала (изображения).

Если действие шума сказывается не по всей протяженности поля изображения, а только в случайно расположенных точках, в которых значения функции яркости заменяются случайными величинами, то шум называют импульсным. На изображении такие помехи выглядят изолированными контрастными точками. Будем считать, что искаженные точки равномерно распределены по всему полю изображения, а яркость искаженных точек имеет равномерное распределение в некотором диапазоне. Импульсный шум характерен для систем передачи изображений по радиоканалам с использованием нелинейных методов модуляции, а также для цифровых систем передачи и хранения изображений. В частности, импульсный шум присущ устройствам ввода изображений с телевизионной камеры.

Действие импульсного шума можно описать следующей математической моделью:

$$y(m, n) = \begin{cases} \xi(m, n) & \text{с вероятностью } p; \\ x(m, n) & \text{с вероятностью } (1-p), \end{cases} \quad (3.3)$$

где $\xi(m, n)$ - отсчеты шумового поля, статистически независимые друг от друга и равномерно распределенные в диапазоне $[\xi_{\min}, \xi_{\max}]$.

Таким образом, импульсный шум характеризуется вероятностью p искажения точки ($0 \leq p \leq 1$) и диапазоном значений $[\xi_{\min}, \xi_{\max}]$.

Следующая модель шума - мультипликативная. Эта модель используется тогда, когда отсчеты наблюдаемого изображения получаются умножением отсчетов исходного изображения на случайный сигнал. Такая модель хорошо описывает шум диффузности в когерентно-оптических и голографических изображающих системах. Так как логарифм произведения равен сумме логарифмов сомножителей, то логарифмическое поэлементное преобразование зашумленного изображения приводит мультипликативную модель шума к аддитивной модели (3.1).

Более общей линейной моделью наблюдения изображения в условиях помех является модель, учитывающая наряду с аддитивным шумом динамические пространственные искажения. Если такие искажения можно описать пространственно-однородной (инвариантной к сдвигу) линейной системой с импульсной характеристикой $h(k, l)$, то модель наблюдения принимает вид:

$$y(m, n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h(k, l) x(m-k, n-l) + v(m, n). \quad (3.4)$$

Модель (3.4) описывает искажения, вызванные движением системы регистрации относительно объекта, турбулентностью атмосферы, абберациями оптической системы, неточностью фокусировки и т.п.

3.1.2 Алгоритмы фильтрации изображений

Рассмотрим схему искажения и фильтрации (восстановления, реставрации) изображений, представленную на рис. 3.1.

Целью восстановления искаженного изображения $y(m, n)$ является получение из него при помощи некоторой обработки изображения $\hat{x}(m, n)$,

которое близко к идеальному изображению $x(m, n)$ по заданному критерию. (Для краткости формулировок мы отождествляем понятия изображения и описывающей его функции яркости.) Получающееся в результате обработки изображение $\hat{x}(m, n)$ будем называть оценкой исходного (идеального) изображения $x(m, n)$.

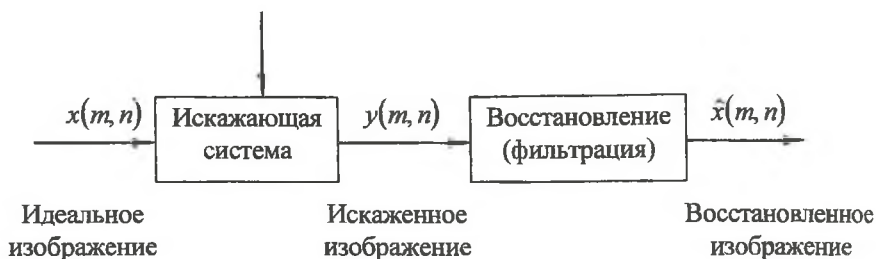


Рис. 3.1 Схема искажения и фильтрации (восстановления) изображений

Определим ошибку оценивания в каждой точке изображения

$$\varepsilon(m, n) = \hat{x}(m, n) - x(m, n), \quad m = \overline{0, M-1}, n = \overline{0, N-1}, \quad (3.5)$$

а также среднюю квадратичную ошибку (СКО) через ее квадрат, т.е. дисперсию ошибки:

$$\bar{\varepsilon}^2 = \frac{1}{MN} \sum \sum [\hat{x}(m, n) - x(m, n)]^2. \quad (3.6)$$

Критерий минимума квадрата СКО ($\bar{\varepsilon}^2 \rightarrow \min$) является наиболее универсальным и распространенным критерием качества восстановления при проектировании алгоритмов фильтрации изображений из-за математической простоты. Однако этот критерий имеет недостаток, заключающийся в том, что он не всегда согласуется с субъективным (психовизуальным) критерием качества, основанным в основном на точности передачи контуров.

Указанный критерий является конструктивным и позволяет теоретически рассчитывать оптимальные (дающие минимумы квадрата СКО) алгоритмы фильтрации при рассмотренных моделях наблюдения. Однако оптимальные алгоритмы оказываются весьма сложными для расчета и реализации. Например, оптимальный винеровский фильтр требует

выполнения двумерного дискретного преобразования Фурье над полем размера $M \times N$, что приводит к значительным затратам машинного времени. В интерактивной автоматизированной системе обработки изображений предпочтение отдается так называемым квазиоптимальным алгоритмам, которые дают минимум квадрата СКО в некотором классе алгоритмов с заданной структурой и незначительно отличаются от оптимальных по этому критерию. Далее будут рассмотрены простейшие линейные или нелинейные алгоритмы фильтрации, основанные на локальной обработке зашумленного изображения «окном» малых размеров.

3.1.3 Линейный сглаживающий фильтр

Линейный фильтр с конечной импульсной характеристикой можно описать уравнением двумерной свертки

$$\hat{x}(m, n) = \sum_{(k, l) \in W} a(k, l) y(m - k, n - l) = a(k, l) ** y(m, n), \quad (3.7)$$

где $a(k, l)$ - коэффициенты линейной маски фильтра (импульсная характеристика восстанавливающей ЛПП-системы); W - область «окна».

Коэффициенты $a(k, l)$ можно выбрать оптимальным образом из условия минимизации дисперсии ошибки фильтрации, которая есть математическое ожидание средней квадратичной ошибки:

$$\varepsilon^2 = E \left\{ \left[\hat{x}(m, n) - x(m, n) \right]^2 \right\} \rightarrow \min. \quad (3.8)$$

Будем считать идеальное и зашумленное изображения стационарными двумерными случайными последовательностями с нулевыми математическими ожиданиями и известными корреляционными функциями

$$\begin{aligned} B_x(k, l) &= E \left[x(m, n) x(m + k, n + l) \right], \\ B_y(k, l) &= E \left[y(m, n) y(m + k, n + l) \right], \\ B_{xy}(k, l) &= E \left[x(m, n) y(m + k, n + l) \right]. \end{aligned} \quad (3.9)$$

Из (3.7) и (3.8) следует, что

$$\begin{aligned} \varepsilon^2 = & \sum_{(k,l) \in W} \sum_{(p,q) \in W} \sum_{(r,s) \in W} a(k,l) a(p,q) B_y(k-p, l-q) - \\ & - 2 \sum_{(k,l) \in W} a(k,l) B_{xy}(-k, -l) + D_x. \end{aligned} \quad (3.10)$$

Для нахождения оптимальных значений коэффициентов $a(k,l)$, минимизирующих квадратичную форму (3.10), приравняем к нулю частные производные $\frac{\partial \varepsilon^2}{\partial a(k,l)}$. При этом получаем систему линейных уравнений относительно неизвестных коэффициентов $a(k,l)$:

$$\sum_{(k,l) \in W} a(k,l) B_y(m-k, n-l) = B_{xy}(-m, -n), \quad (m,n) \in W. \quad (3.11)$$

Коэффициенты $a_*(k,l)$, удовлетворяющие приведенной системе и доставляющие минимум (3.10), составляют оптимальную линейную маску

$$A_* = \{a_*(k,l)\}, \quad (k,l) \in W.$$

Соответствующая оптимальной маске минимальная дисперсия ошибки фильтрации равна

$$\varepsilon_*^2 = D_x - \sum_{(k,l) \in W} a_*(k,l) B_{xy}(-k, -l). \quad (3.12)$$

Для важного частного случая, соответствующего модели аддитивного белого шума (3.1), (3.2), система уравнений (3.11) и дисперсия ошибки (3.12) имеют вид:

$$\sum_{(k,l) \in W} a(k,l) B_x(m-k, n-l) + a(m,n) D_v = B_x(m,n), \quad (m,n) \in W, \quad (3.13)$$

$$\varepsilon_*^2 = a_*(0,0) D_v. \quad (3.14)$$

Будем измерять эффективность фильтрации коэффициентом снижения шума, который равен отношению дисперсии ошибки фильтрации к дисперсии искажений:

$$K_c = \frac{\varepsilon^2}{\varepsilon_n^2} = \frac{E[\hat{x}(m,n) - x(m,n)]^2}{E[y(m,n) - x(m,n)]^2} \quad (3.15)$$

Можно показать, что в случае аддитивного белого шума

$$K_c = a_*(0,0).$$

Из системы (3.13) найдем оптимальную маску размера 3×3

$$A_* = \{a_*(k,l)\}_{k,l=-1}^1$$

для фильтрации аддитивного белого шума при изотропной экспоненциальной корреляционной функции идеального изображения:

$$B_x(k,l) = D_x \rho^{\sqrt{k^2 + l^2}}.$$

При коэффициенте корреляции соседних отсчетов $\rho = 0.95$ и отношениях «сигнал/шум» $d^2 = 10$ и $d^2 = 1$ получаем соответственно

$$A_* = \begin{bmatrix} 0.060 & 0.100 & 0.060 \\ 0.100 & 0.360 & 0.100 \\ 0.060 & 0.100 & 0.060 \end{bmatrix}, A_* = \begin{bmatrix} 0.093 & 0.101 & 0.093 \\ 0.101 & 0.137 & 0.101 \\ 0.093 & 0.101 & 0.093 \end{bmatrix}. \quad (3.16)$$

Коэффициенты снижения шума K_c^* (предельно допустимые) в этих случаях равны 0.360 (для $d^2 = 10$) и 0.137 (для $d^2 = 1$).

На практике измерение или теоретический расчет корреляционных функций (3.9), а также решение системы (3.11) не всегда возможны. Поэтому часто используют другой подход, основанный на построении так называемого сглаживающего фильтра. Поскольку шум (как аддитивный, так и мультипликативный) обычно является пространственно некоррелированным (белым), его спектр содержит более высокие пространственные частоты, чем спектр идеального изображения. Этот факт наводит на мысль, что простая низкочастотная фильтрация может служить эффективным средством подавления шумов. В принципе, любой КИХ-фильтр с неотрицательными коэффициентами обладает сглаживающими свойствами. Ориентируясь на образцы оптимальных масок (3.16), можно

предложить следующие сглаживающие маски:

$$A_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, A_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, A_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (3.17)$$

Коэффициенты масок (3.17) нормированы ($\sum_{(k,l) \in W} a(k,l) = 1$) с тем,

чтобы процедура подавления помех не вызывала смещения средней яркости обработанного изображения относительно исходного. Таким свойством лишь приближенно обладают оптимальные маски (3.16), так как оптимальный КИХ-фильтр рассчитан на нулевые средние значения яркости идеального и зашумленного изображений.

В общем случае отсутствие предварительного центрирования при обработке изображения линейными масками эквивалентно искусственному завышению действительного отношения «сигнал/шум» и приводит в конечном счете к увеличению СКО фильтрации. Если отношение квадрата среднего значения к дисперсии изображения велико, то потеря оптимальности может быть значительной.

Маски (3.17) отличаются степенью сглаживания шумов (у маски A_1 она максимальная, у A_3 - минимальная). Выбор коэффициентов маски должен производиться экспериментально. При увеличении степени сглаживания шумов происходит также подавление высокочастотной составляющей полезного изображения, что вызывает исчезновение мелких деталей и размазывание контуров. Например, при отношении «сигнал/шум» $d^2 = 10$ использование вместо оптимальной маски (3.16) сглаживающей маски A_1 (3.17) приводит к $K_c = 1.28$, а маски A_3 - к $K_c = 0.380$; при отношении «сигнал/шум» $d^2 = 1$ применение этих сглаживающих масок дает соответственно следующие коэффициенты снижения шума: $K_c = 0.147$ и $K_c = 0.164$. Таким образом, простой сглаживающий фильтр может даже увеличить СКО фильтрации при неудачном выборе коэффициентов маски.

Если требуемая степень сглаживания с применением маски размера 3×3 не достигается, то следует использовать сглаживающие маски больших размеров (5×5 , 7×7 , ...). Другой метод заключается в итеративном

применении сглаживающей маски 3×3 :

$$\hat{x}(m, n) = \underbrace{a(k, l) ** a(k, l) ** \dots ** a(k, l)}_R ** y(m, n), \quad (3.18)$$

где R - число проходов.

Заметим, что при неограниченном увеличении числа проходов R приходим к изображению с постоянным значением яркости:

$$\hat{x}(m, n) \rightarrow const = E[(m, n)].$$

Поэтому для конкретной модели наблюдения существует оптимальное число проходов, которое можно определить экспериментально.

3.1.4 Медианный фильтр

Медианный фильтр в отличие от сглаживающего фильтра является нелинейной процедурой подавления шумов. Медианный фильтр представляет собой скользящее по полю изображения «окно» W , охватывающее нечетное число отсчетов. Центральный отсчет заменяется медианой всех элементов изображения, попавших в «окно». Медианой дискретной последовательности $x_1, x_2, \dots, x_L$ для нечетного L называют такой ее элемент, для которого существуют $(L-1)/2$ элементов, меньших или равных ему по величине, и $(L-1)/2$ элементов, больших или равных ему по величине. Другими словами, медианой является средний по порядку член ряда, получающегося при упорядочении исходной последовательности. Например, $med(20, 10, 3, 7, 7) = 7$.

Двумерный медианный фильтр с «окном» W определим следующим образом:

$$\hat{x}(m, n) = \underset{W}{med} y(m, n) = med [y(m+k, n+l) : (k, l) \in W]. \quad (3.19)$$

Как и сглаживающий фильтр, медианный фильтр используется для подавления аддитивного и импульсного шумов на изображении. Характерной особенностью медианного фильтра, отличающей его от сглаживающего, является сохранение перепадов яркости (контуров). Причем, если перепады яркости велики по сравнению с дисперсией аддитивного белого шума, то медианный фильтр дает меньшее значение

СКО (3.6) по сравнению с оптимальным линейным фильтром. Особенно эффективным медианный фильтр является в случае импульсного шума (3.3). На рис. 3.2 показано воздействие сглаживающего и медианного фильтров с трехэлементным «окном» на зашумленный аддитивным шумом перепад яркости для одномерного сигнала.

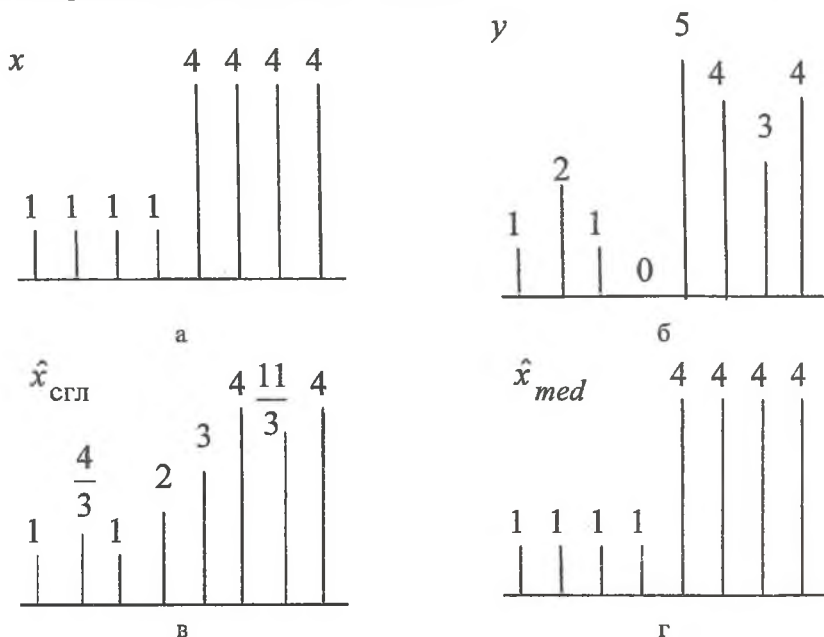


Рис. 3.2 Воздействие сглаживающего и медианного фильтров с трехэлементным «окном»:

а - исходный сигнал;

б - зашумленный сигнал;

в - результат сглаживания;

г - результат медианной фильтрации

Что касается импульсного шума, то, например, медианный фильтр с «окном» 3×3 полностью подавляет одиночные выбросы на равномерном фоне, а также группы из двух, трех и четырех импульсных выбросов. В общем случае для подавления группы импульсных помех размеры «окна» должны быть по меньшей мере вдвое больше размеров группы помех.

Обобщением рассмотренного медианного фильтра является взвешенный медианный фильтр. В простом медианном фильтре все элементы изображения в пределах «окна» одинаково влияют на результат нахождения медианы. Но иногда желательно придать больший вес точкам, расположенным ближе к центру «окна». Основная идея взвешенного

медианного фильтра состоит в изменении числа элементов, попавших в «окно», путем повторения каждого элемента заданное число раз и нахождения медианы получившийся растянутой последовательности. Например, для одномерной последовательности взвешенная медиана по пятиэлементному «окну» с весами (1, 3, 1) определяется как

$$\begin{aligned}\hat{x}(m) &= med_{(1,3,1)}(m) = med \left[y(m-1), 3^o y(m), y(m+1) \right] = \\ &= med \left[y(m-1), y(m), y(m), y(m), y(m+1) \right],\end{aligned}$$

где k^o - коэффициент повторения элемента в последовательности.

В двумерном случае, если весовые коэффициенты объединить в матрицу $W=\{w(k,l)\}$, взвешенный медианный фильтр записывается как

$$\hat{x}(m,n) = med_W y(m,n) = med \left[w(k,l)^o y(m+k, n+l) \right]. \quad (3.20)$$

В (3.20) считается, что сумма весов $\sum \sum w(k,l)$ является целым нечетным числом. Простой медианный фильтр (3.19) является частным случаем взвешенного (3.20) при $w(k,l) \equiv 1$.

Среди взвешенных и простых медианных фильтров с «окном» 3×3 наиболее распространены следующие:

$$\begin{aligned}W_1 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{bmatrix}, W_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \\ W_3 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, W_4 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.\end{aligned} \quad (3.21)$$

Простые медианные фильтры в общем случае подавляют импульсные помехи большей кратности, чем взвешенные медианные фильтры. Если на изображении присутствуют мелкие детали размером 3×3 и менее, они могут быть подавлены простыми медианными фильтрами. Заметим также, что взвешенный медианный фильтр меньше искажает форму объектов (не уничтожает крайние точки).

Как и при линейном сглаживании, возможно итеративное применение медианного фильтра. В отличие от сглаживающего фильтра результат итераций не обязательно сходится к изображению с постоянной яркостью. Иногда целесообразно использовать следующий метод: итеративно применять медианный фильтр до тех пор, пока не получится изображение, инвариантное к данному медианному фильтру. Получившееся изображение называется стабильной точкой медианного фильтра. Например, бинарное изображение в строке 4 приведенной таблицы является стабильной точкой для медианных фильтров с масками W_1 , W_2 , W_3 , а изображение в строке 10 - стабильная точка для всех рассмотренных фильтров.

В заключение отметим, что так как медианный фильтр является существенно нелинейной процедурой оценивания, то теоретический анализ его характеристик чрезвычайно сложен. К настоящему времени не разработаны методы построения оптимальных медианных фильтров. Поэтому медианная фильтрация с целью подавления шумов является в целом эвристическим методом. Ее нельзя применять вслепую. В каждом конкретном случае необходимо убедиться в достоверности получаемых результатов при помощи моделирования фильтра и оценивания качества фильтрации по выбранному критерию. Таким образом, единственным методом построения и исследования характеристик медианных фильтров является статистическое моделирование.

3.2 Выполнение лабораторной работы

3.2.1 Общий план выполнения работы

1. Изучить методы фильтрации аддитивных и импульсных шумов на изображениях при помощи сглаживающего и медианного фильтров.
2. Ознакомиться с описанием команд входного языка системы.
3. Составить последовательность команд с указанием значений параметров для выполнения исследования алгоритмов фильтрации по каждому из приведенных ниже вариантов.
4. Согласовать план проведения обработки с преподавателем, получить разрешение на практическую работу на системе.
5. Выполнить обработку.

6. Составить отчет о выполненной работе. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

3.2.2 Вариант 1 - фильтрация аддитивного белого шума

Сгенерировать изображение «шахматное поле» размером 128x128. Рассчитать дисперсию сгенерированного изображения. Наложить аддитивный белый шум (отношение «сигнал/шум» принять равным 10).

Произвести фильтрацию искаженного изображения медианным и линейным сглаживающим фильтром. Рассчитать квадрат среднего квадратичного отклонения фильтрации и коэффициент снижения шума. Произвести повторную фильтрацию и расчет квадрата СКО.

Повторить исследование, для чего отношение «сигнал/шум» принять равным 1.

3.2.3 Вариант 2 - фильтрация импульсного шума

Сгенерировать изображение «шахматное поле» размером 128x128 и наложить на него импульсный шум интенсивности $p=0.1$. Рассчитать среднее квадратичное искажение.

Произвести фильтрацию искаженного изображения медианным и линейным сглаживающим фильтром. Рассчитать квадрат среднего квадратичного отклонения фильтрации и коэффициент снижения шума. Произвести повторную фильтрацию и расчет квадрата СКО. Повторить исследование, приняв интенсивность импульсного шума $p=0.3$.

3.2.4 Содержание отчета

Отчет по работе должен содержать:

1. Последовательность команд обработки с комментариями и числовыми значениями параметров.
2. Перечень созданных изображений с указанием их смыслового содержания.
3. Результаты исследования алгоритмов фильтрации, оформленные в виде таблицы значений коэффициента снижения шума.

3.2.5 Контрольные вопросы

1. Модели аддитивного, импульсного и мультипликативного шумов.
2. Постановка задачи фильтрации. Критерии качества фильтрации.
3. Метод построения оптимального КИХ-фильтра.
4. Как выбрать линейную маску сглаживающего фильтра?
5. Какие преимущества и недостатки у медианного фильтра?
6. Взвешенный медианный фильтр.
7. Итеративное использование фильтров. Стабильные точки.
8. Как выбрать маску медианного фильтра?
9. При каких моделях шумов применим сглаживающий (медианный) фильтр?

4 ОБНАРУЖЕНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

Цель работы - изучение и экспериментальное исследование методов обнаружения объектов на изображениях при помощи линейной обработки "окном", дальнейшее совершенствование навыков работы на автоматизированной системе обработки изображений в режиме диалога.

В лабораторной работе изучаются методы обнаружения объектов на изображении при помощи линейной обработки "окном"; приведены способы их реализации при работе на автоматизированной системе обработки изображений в режиме диалога.

4.1 Теоретические основы лабораторной работы

4.1.1 Корреляционный метод обнаружения объектов на изображении

Задача обнаружения заключается в установлении наличия на изображении объектов (яркостных областей) с определенными свойствами, а также, если объекты обнаружены, в определении их координат на плоскости изображения.

Основной принцип обнаружения объектов на изображении заключается

в сопоставлении функции яркости изображения с некоторым "эталоном" - фрагментом поля яркости, содержащим искомый объект. При реализации процедуры обнаружения эталон последовательно перемещается по полю изображения, и в каждом положении исследуется его сходство с реальной функцией яркости на фрагменте. Полного совпадения эталона и изображения, как правило, не бывает из-за шумов и искажений, а также из-за того, что обычно отсутствует полная информация относительно формы и структуры объекта (приходится использовать эталон, лишь приближенно описывающий объект).

Обозначим через $t(k,l)$, $(k,l) \in D$, функцию яркости эталонного объекта, заданную на некоторой области D в собственной системе координат эталона (обычно рассматривается прямоугольная область D с началом координат в центре). Пусть $x(m,n)$ - отсчеты функции яркости наблюдаемого изображения. В качестве меры различия между эталоном и изображением в точке (m,n) чаще всего берут квадратичную меру:

$$\varepsilon^2(m,n) = \sum_{(k,l) \in D} [x(m+k, n+l) - t(k,l)]^2. \quad (4.1)$$

Считается, что имеется сходство фрагмента изображения с эталоном в точке (m,n) , если

$$\varepsilon^2(m,n) < L_{\varepsilon},$$

где L_{ε} - некоторый порог, зависящий от интенсивности шумов.

На рис. 4.1 показана схематическая иллюстрация принципа сравнения с эталоном.

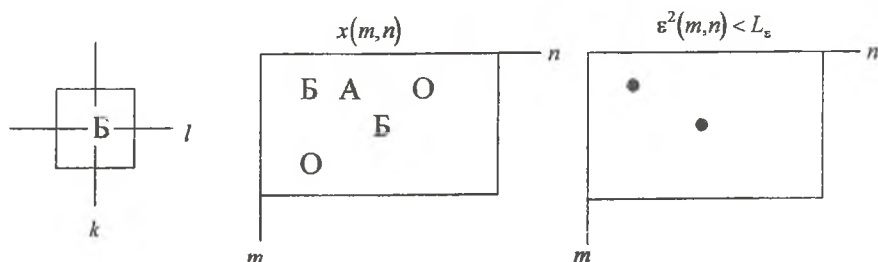


Рис. 4.1 Схематическая иллюстрация принципа сравнения с эталоном

На практике меру (4.1) обычно не вычисляют, а переходят к связанным с ней, но более просто вычисляемым величинам. Преобразуем выражение для квадратичной меры различия:

$$\varepsilon^2(m, n) = \sum_{(k, l) \in D} x^2(m + k, n + l) - 2 \sum_{(k, l) \in D} x(m + k, n + l) t(k, l) + \sum_{(k, l) \in D} t^2(k, l).$$

Здесь первое слагаемое характеризует энергию изображения в пределах "окна" D . Эта энергия обычно меняется довольно медленно в зависимости от (m, n) и практически не характеризует искомый объект. Третье слагаемое характеризует энергию эталона и не зависит от (m, n) . Для обнаружения существенным является лишь второе слагаемое, которое с точностью до постоянного множителя задает **взаимную корреляцию** изображения и эталона:

$$B(m, n) = \sum_{(k, l) \in D} x(m + k, n + l) t(k, l). \quad (4.2)$$

При совпадении изображения и эталона взаимная корреляция (4.2) велика, что и обеспечивает малость квадратичной меры (4.1). Однако непосредственное использование взаимной корреляции в качестве меры сходства обычно дает плохое качество обнаружения. Это объясняется тем, что взаимная корреляция может увеличиваться даже при отсутствии соответствия изображения эталону, если яркость изображения в окрестности точки с координатами (m, n) велика. Эту трудность можно обойти, если использовать **нормированную взаимную корреляцию**:

$$R(m, n) = \frac{\sum_{(k, l) \in D} x(m + k, n + l) t(k, l)}{\sqrt{\left[\sum_{(k, l) \in D} x^2(m + k, n + l) \right] \left[\sum_{(k, l) \in D} t^2(k, l) \right]}}. \quad (4.3)$$

Величина $R(m, n)$ равна максимальному значению (единице), если эталон совпадает с изображением с точностью до постоянного неотрицательного множителя. В данном случае объект в точке (m, n) считается обнаруженным, если $R(m, n) > L'_R$, где L'_R - порог для меры (4.3).

Вид функции (4.3) можно упростить, если предварительно выполнить

нормализацию исходного изображения $x(m,n)$, выбрав энергию эталона равной единице, то есть обеспечить

$$\sum_{(k,l) \in D} t^2(k,l) = 1,$$

и произведя адаптивную поэлементную обработку изображения с "окном" D для выравнивания энергии (дисперсии) изображения

$$\sum_{(k,l) \in D} x^2(m+k, n+l)$$

по всему полю. Таким образом, дополнительно исключается влияние изменений дисперсии в пределах изображения. Обозначив нормализованное изображение через $x'(m,n)$, вновь приходим к линейной мере взаимной корреляции:

$$B'(m,n) = R'(m,n) = \sum_{(k,l) \in D} x'(m+k, n+l) t(k,l). \quad (4.4)$$

Легко видеть, что функция нормированной взаимной корреляции (4.4) формируется в результате прохождения двумерного сигнала - нормализованного изображения $x'(m,n)$ - через двумерную ЛПП-систему с импульсной характеристикой $h(k,l) = t(-k, -l)$:

$$B'(m,n) = x'(m,n) ** h(k,l) = x'(m,n) ** t(-k, -l).$$

Таким образом, получаем общую схему корреляционного алгоритма обнаружения, которая приведена на рис. 4.2. Здесь на последнем этапе получается бинарное изображение с единицами в точках обнаружения объектов:

$$g(m,n) = \begin{cases} 0, & B'(m,n) \leq L'_R, \\ 1, & B'(m,n) > L'_R. \end{cases}$$

ЛПП-система, которая называется в данном случае коррелятором, является КИХ-системой и может быть реализована обработкой "окном"

(прямой сверткой) или спектральным методом (быстрой сверткой).

Рассмотренный корреляционный метод обнаружения отличается относительной простотой, однако ему свойственны довольно большие вероятности ошибок (ложного обнаружения или пропусков объектов), что объясняется игнорированием свойств шумов при синтезе алгоритма обработки изображения.

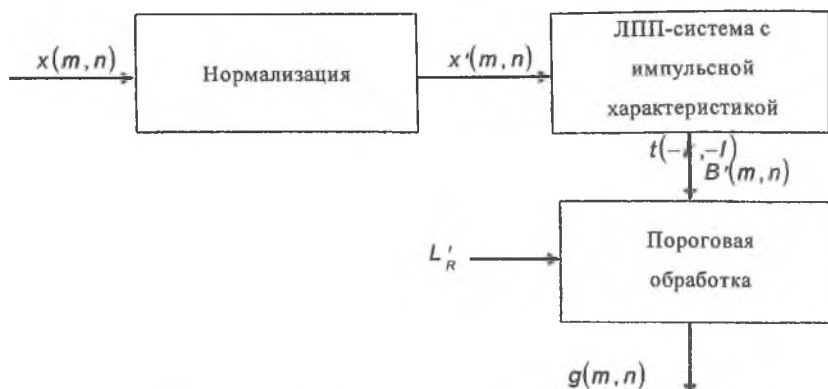


Рис. 4.2 Общая схема корреляционного алгоритма обнаружения

4.1.2 Метод согласованной фильтрации

В корреляционном методе обнаружения, рассмотренном в п.4.1.1, импульсная характеристика коррелятора определялась самим эталоном, т.е. формой и распределением яркости искомого объекта. Можно поставить оптимизационную задачу построения такой ЛПП-системы, которая обеспечивает наилучшее качество обнаружения в классе линейных систем.

Рассмотрим аддитивную модель наблюдения детерминированного объекта $t(m, n)$, наложенного на случайный фон $f(m, n)$. Тогда наблюдаемый сигнал $x(m, n)$ состоит либо из фона:

$$x(m, n) = f(m, n),$$

либо из суммы объекта и фона:

$$x(m, n) = t(m, n) + f(m, n).$$

Будем считать, что фон $f(m, n)$ является стационарным случайным

процессом. Строго говоря, такая модель наблюдения является приближенной, так как обычно наблюдается не наложение, а "врезка" объекта в фон. Однако эксперименты показывают, что такой подход обеспечивает достаточно высокое качество обнаружения.

Обозначим импульсную характеристику искомой ЛПП-системы через $h(k, l)$. Будем судить о наличии объекта по уровню сигнала $y(m, n)$ на выходе этой системы. Если объект отсутствует, то

$$y(m, n) = y_f(m, n) = h(m, n) ** f(m, n) .$$

Если объект есть, то

$$y(m, n) = y_t(m, n) + y_f(m, n) = h(m, n) ** t(m, n) + h(m, n) ** f(m, n)$$

Допустим, что центр объекта находится в точке $(0, 0)$. Тогда, для того чтобы сравнение $y(0, 0)$ с порогом обеспечивало наиболее надежное обнаружение, нужно, чтобы превышение сигнала $y_t(0, 0)$ над фоном $y_f(m, n)$ было бы максимально возможным. Иначе говоря, в точке $(0, 0)$ превышение энергии полезного сигнала (объекта) над энергией фона (шумом) должно быть максимально возможным:

$$\delta^2 = \frac{y_t^2(0, 0)}{\bar{y}_f^2} \rightarrow \max , \quad (4.5)$$

где $\bar{y}_f^2$ - дисперсия фона (считаем, что фон имеет нулевое среднее).

Оптимальная по критерию (4.5) ЛПП-система называется согласованным фильтром. Решим задачу построения согласованного фильтра, воспользовавшись частотным методом.

Обозначим через $H(e^{j\omega_1}, e^{j\omega_2})$ частотную характеристику искомого фильтра. Тогда

$$y_t(0, 0) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(e^{j\omega_1}, e^{j\omega_2}) T(e^{j\omega_1}, e^{j\omega_2}) d\omega_1 d\omega_2 ,$$

где $T(e^{i\omega_1}, e^{i\omega_2})$ - спектр объекта.

Дисперсия фона $y_f(m, n)$ на выходе согласованного фильтра равна:

$$\bar{y}_f^2 = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left| H(e^{i\omega_1}, e^{i\omega_2}) \right|^2 \Phi_f(e^{i\omega_1}, e^{i\omega_2}) d\omega_1 d\omega_2,$$

где $\Phi(e^{i\omega_1}, e^{i\omega_2})$ - энергетический спектр фона $f(m, n)$ на входе.

Согласованный фильтр должен обеспечивать максимум величины

$$\delta^2 = \frac{1}{4\pi^2} \frac{\left[\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} H(e^{i\omega_1}, e^{i\omega_2}) T(e^{i\omega_1}, e^{i\omega_2}) d\omega_1 d\omega_2 \right]^2}{\int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \left| H(e^{i\omega_1}, e^{i\omega_2}) \right|^2 \Phi_f(e^{i\omega_1}, e^{i\omega_2}) d\omega_1 d\omega_2}. \quad (4.6)$$

Воспользовавшись неравенством Коши-Буняковского-Шварца, можно показать, что максимум (4.6) достигается при частотной характеристике

$$H(e^{i\omega_1}, e^{i\omega_2}) = \frac{T(e^{-i\omega_1}, e^{-i\omega_2})}{\Phi_f(e^{i\omega_1}, e^{i\omega_2})}. \quad (4.7)$$

В том случае, если фон описывается белым шумом с дисперсией D_f , согласованный фильтр (4.7) принимает вид:

$$H(e^{i\omega_1}, e^{i\omega_2}) = \frac{1}{D_f} T(e^{-i\omega_1}, e^{-i\omega_2}).$$

Его импульсная характеристика $h(k, l) = \frac{1}{D_f} t(-k, -l)$ равна (с

точностью до постоянного множителя) искомому объекту (т.е. эталону), развернутому на 180 градусов. Следовательно, если фон описывается белым шумом, то имеем тот же результат, что и в корреляционном методе обнаружения, то есть корреляционный метод является в данном случае оптимальным. Однако для изображений фон зачастую не описывается

белым шумом: фон - это тоже изображение со своей структурой, с другими объектами. Поэтому корреляционный метод обнаружения для изображений не всегда эффективен.

В случае "небелого" фона согласованный фильтр не является коррелятором. Физическую трактовку согласованного фильтра можно дать, условно представив этот фильтр в виде последовательного соединения двух звеньев:

$$H(e^{i\omega_1}, e^{i\omega_2}) = \frac{1}{[\Phi_f(e^{i\omega_1}, e^{i\omega_2})]^{1/2}} \times \frac{T(e^{-i\omega_1}, e^{-i\omega_2})}{[\Phi_f(e^{i\omega_1}, e^{i\omega_2})]^{1/2}} = H_1 \times H_2.$$

Первое звено является разбеливающим фильтром для фона. Его частотная характеристика обеспечивает превращение фона в белый шум с единичной дисперсией:

$$\Phi_{f'}(e^{i\omega_1}, e^{i\omega_2}) = \left| H(e^{i\omega_1}, e^{i\omega_2}) \right|^2 \Phi_f(e^{i\omega_1}, e^{i\omega_2}) = 1.$$

Полезный сигнал (объект) при прохождении через разбеливающий фильтр также изменяется, поэтому и эталон необходимо разбеливать. Спектр "разбеленного" эталона становится равным:

$$T'(e^{i\omega_1}, e^{i\omega_2}) = H_1 \cdot T(e^{i\omega_1}, e^{i\omega_2}) = \frac{T(e^{i\omega_1}, e^{i\omega_2})}{[\Phi_f(e^{i\omega_1}, e^{i\omega_2})]^{1/2}}.$$

Второе звено имеет частотную характеристику

$$H_2(e^{i\omega_1}, e^{i\omega_2}) = T'(e^{-i\omega_1}, e^{-i\omega_2}),$$

т.е. реализует корреляционный метод обнаружения "разбеленного" эталона на фоне белого шума.

Известно, что энергетический спектр изображения (фона) $\Phi_f(e^{i\omega_1}, e^{i\omega_2})$ довольно быстро убывает с увеличением частот (ω_1, ω_2) .

Поэтому действие разбеливающего фильтра H_1 заключается в подъеме верхних частот. Как известно, это приводит к оконтуриванию

(подчеркиванию границ) изображения. Тем самым получает объяснение эмпирически найденная многими исследователями рекомендация подвергать изображение оконтуриванию для надежного обнаружения объектов корреляционным методом.

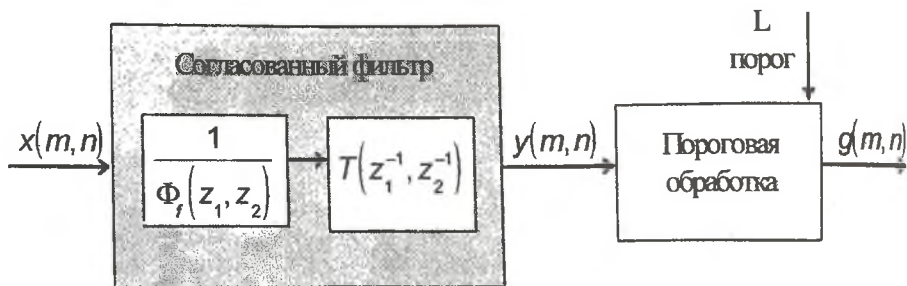


Рис. 4.3 Схема метода согласованной фильтрации для обнаружения объектов

На рис. 4.3 приведена схема метода согласованной фильтрации для обнаружения объектов на изображении. Передаточная функция согласованного фильтра

$$H(z_1, z_2) = \frac{T(z_1^{-1}, z_2^{-1})}{\Phi_f(z_1, z_2)} \quad (4.8)$$

получена из частотной характеристики (4.7) заменой переменных:

$$e^{j\omega_1} \rightarrow z_1, e^{j\omega_2} \rightarrow z_2.$$

В общем случае фильтр с передаточной функцией (4.8) можно реализовать в частотной области при помощи ДПФ. Если энергетический спектр фона $\Phi_f(z_1, z_2)$ является дробно-рациональной функцией, то согласованный фильтр можно построить в виде гибридного (спектрально-рекуррентного) двухпроходного алгоритма обработки изображений. И, наконец, если дробно-рациональный энергетический спектр фона не имеет нулей, то согласованный фильтр является КИХ-системой и легко реализуется в пространственной области в виде процедуры обработки "окном".

4.1.3 Согласованный фильтр для фона с биэкспоненциальной корреляционной функцией

Рассмотрим частный случай построения алгоритма обнаружения, когда фон можно аппроксимировать случайным полем с биэкспоненциальной корреляционной функцией:

$$B_f(m, n) = D_f \rho^{|m|+|n|},$$

где D_f - дисперсия фона,

ρ - коэффициент корреляции между соседними отсчетами фона (одинаковый по строкам и столбцам).

Энергетический спектр такого поля определяется двумерным z - преобразованием от $B_f(m, n)$:

$$\begin{aligned} \Phi_f(z_1, z_2) &= \sum_{m, n = -\infty}^{\infty} B_f(m, n) z_1^{-m} z_2^{-n} = \\ &= \frac{D_f (1 - \rho^2)^2}{(1 - \rho z_1)(1 - \rho z_1^{-1})(1 - \rho z_2)(1 - \rho z_2^{-1})}. \end{aligned}$$

Передаточная функция согласованного фильтра в этом случае не содержит полюсов и равна

$$\begin{aligned} H(z_1, z_2) &= \frac{T(z_1^{-1}, z_2^{-1})}{\Phi_f(z_1, z_2)} = \frac{a_0}{D_f} \left[1 + a_1 (z_1 + z_2 + z_1^{-1} + z_2^{-1}) + \right. \\ &\quad \left. + a_2 (z_1 z_2 + z_1 z_2^{-1} + z_1^{-1} z_2 + z_1^{-1} z_2^{-1}) \right] T(z_1^{-1}, z_2^{-1}), \end{aligned} \quad (4.9)$$

где

$$a_0 = \left[\frac{1 + \rho^2}{1 - \rho^2} \right]; a_1 = -\frac{\rho}{1 + \rho^2}; a_2 = \frac{\rho^2}{(1 + \rho^2)^2}.$$

Постоянный коэффициент $\left(\frac{a_0}{D_f} \right)$ в (4.9) играет роль масштаба яркости

и его можно не учитывать при обработке, так как он оказывает влияние лишь на выбор порога при выделении корреляционных пиков на обработанном согласованном фильтром изображении.

Из (4.9) видим, что согласованный фильтр может быть реализован в два этапа последовательной линейной обработкой "окном":

1) "разбеливание" фона обработкой "окном" 3×3 с маской

$$\begin{pmatrix} a_2 & a_1 & a_2 \\ a_1 & 1 & a_1 \\ a_2 & a_1 & a_2 \end{pmatrix}; \quad (4.10)$$

2) реализация коррелятора обработкой "окном" с маской, соответствующей повернутому на 180 градусов эталону: $\{t(-k, -l)\}$, $(k, l) \in D$.

Заметим, что при $\rho = 0$ имеем фон в виде белого шума, и согласованный фильтр (4.9) становится коррелятором, то есть $a_0 = 1$, $a_1 = a_2 = 0$.

В таблице 4.1 приведены значения коэффициентов согласованного фильтра для различной степени коррелированности фона с биэкспоненциальной корреляционной функцией.

Таблица 4.1 - Значения коэффициентов согласованного фильтра

Коэффициент корреляции ρ	Коэффициенты фильтра		
	a_0	a_1	a_2
0	1	0	0
0,1	1,041	-0,0990	0,0098
0,5	2,778	-0,4000	0,1600
0,8	20,75	-0,4878	0,2380
0,9	90,75	-0,4972	0,2472
0,95	382,8	-0,4993	0,2493
0,98	2451	-0,4999	0,2499
$\rightarrow 1$	$\rightarrow \infty$	-0,5	0,25

4.2 Выполнение лабораторной работы

4.2.1 Общий план выполнения работы

1. Изучить методы обнаружения объектов на изображениях.
2. Ознакомиться с описанием команд входного языка системы.
3. Составить последовательность команд с указанием значений параметров и с комментариями для выполнения исследования методов обнаружения по каждому из приведенных ниже вариантов.
4. Согласовать план проведения обработки с преподавателем, получить разрешение на практическую работу.
5. Выполнить обработку.
6. Составить отчет о выполненной работе.
7. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

4.2.2 Схема исследования

Сначала получаем изображение фона в виде поля с биэкспоненциальной корреляционной функцией. Размеры изображения - 64×64 , среднее - нулевое, дисперсия и коэффициенты корреляции - в зависимости от варианта.

Затем получаем изображение с набором объектов размером 3×3 . Для этого генерируем изображение размером 64×64 , состоящее из нулей, и заносим в него несколько ненулевых отсчетов (например, яркостью 128). Затем пропускаем его через линейный фильтр с одной из масок (х, +, Т, 0, |), которые являются эталоном объектов $\{t(k,l)\}$:

1	0	1	0	1	0	1	1	1	1	1	1	1	0	0
0	1	0	1	1	1	0	1	0	1	0	1	1	1	1
1	0	1	0	1	0	0	1	0	1	1	1	1	0	0

Результирующее изображение получаем суммированием фона и изображения объектов. Для получения достоверных результатов обнаружения изображение не должно содержать пересекающихся

объектов.

Затем производится разбеливание фона при помощи обработки “окном” с маской (4.10) (для метода согласованной фильтрации) и обработка коррелятором. При реализации коррелятора целесообразно выбирать общий масштабный множитель яркости эталонов меньше единицы с тем, чтобы средняя яркость изображения корреляционных пиков не выходила за “видимый” на экране монитора диапазон [0...255]. Например, при выборе коэффициентов коррелятора из соотношения

$$\sum_{(k,l) \in D} t(-k, -l) = 1$$

средняя яркость изображения не изменится.

Если требуется обнаружить объекты нескольких типов, то обработка коррелятором и последующая обработка выполняются отдельно для каждого типа объектов.

Полученное изображение, содержащее корреляционные пики, подвергается пороговой обработке для выделения максимумов с использованием стандартной процедуры (см. лабораторную работу № 1):

- расчет гистограммы;
- ее визуализация;
- выбор порога;
- поэлементное преобразование.

Целью исследования является установление зависимости качества обнаружения (количество ложных объектов, количество пропущенных объектов) от интенсивности шума (дисперсии фона), его коррелированности (ρ) и применяемого метода.

В конце исследования необходимо уничтожить все созданные поля в рабочем файле.

4.2.3 Варианты выполнения лабораторной работы

Вариант 1. Для заданного преподавателем типа объекта исследовать корреляционный метод обнаружения при некоррелированном фоне. Число объектов - 4. Определить число ложных и пропущенных объектов для дисперсии фона 3000 и 5000.

Вариант 2. Для двух типов объектов, заданных преподавателем, провести сравнение двух методов обнаружения (корреляционного и

согласованной фильтрации). Число объектов каждого типа - 2. Дисперсия фона - 3000. Коэффициент корреляции фона - 0,8. Определить число ложных и пропущенных объектов для каждого из методов.

4.2.4 Содержание отчета

Отчет по работе должен содержать:

1. Последовательность команд обработки с комментариями и числовыми значениями параметров.
2. Перечень созданных полей на рабочем файле с указанием их смыслового содержания и шифров.
3. Выводы по результатам исследования.

4.2.5 Контрольные вопросы

1. Корреляционный метод обнаружения объектов на изображении.
2. Реализация корреляционного метода.
3. Метод согласованной фильтрации.
4. В чем преимущества метода согласованной фильтрации?
5. Что такое «разбеливающий фильтр»?
6. Как можно реализовать метод согласованной фильтрации?
7. Согласованный фильтр для фона с биэкспоненциальной корреляционной функцией.

5 СОКРАЩЕНИЕ ИЗБЫТОЧНОСТИ ИЗОБРАЖЕНИЙ МЕТОДОМ КОДИРОВАНИЯ С ПРЕОБРАЗОВАНИЕМ

Цель работы – изучение методов сокращения избыточности изображений при помощи ортогональных преобразований; получение навыков работы на автоматизированной системе обработки изображений в режиме диалога.

В лабораторной работе изучается метод компрессии изображений на кодирования с преобразованием, приведено описание программных средств, необходимых для моделирования этого метода средствами автоматизированной системы обработки изображений.

5.1 Теоретические основы лабораторной работы

5.1.1 Проблемы сжатия данных

Цифровые изображения характеризуются чрезвычайно большим объемом данных. Требования быстрой передачи видеоданных, их хранения вступают в противоречие с техническими возможностями используемой аппаратуры: недостаточной емкостью накопителей, ограниченной пропускной способностью каналов связи, малым быстродействием ЭВМ и т.д. В подобной ситуации большое значение приобретает особый вид обработки изображений – их кодирование с целью сокращения объема (сжатия) изображений.

Технико-экономический эффект от использования сжатия данных весьма многообразен. При передаче данных сжатие позволяет разгрузить канал и, следовательно, повысить достоверность, сократить время, снизить вес передающей аппаратуры. При регистрации изображений сжатие позволяет уменьшить необходимую емкость запоминающих устройств, сократить расходы на хранение и поиск информации. При обработке данных на ЭВМ сжатие иногда дает возможность сократить время обработки и использовать ЭВМ меньшей вычислительной мощности.

Принципиальная возможность сокращения объема данных обусловлена тем, что все реальные изображения обладают высокой степенью избыточности с информационной точки зрения. Это связано, во-первых, с тем, что между близкими точками поля яркости изображения имеется сильная статистическая зависимость. А из теории информации известно, что это приводит к уменьшению количества информации, содержащейся в изображении, т.е. объем изображения используется неэффективно. Другая причина – в неоптимальном вероятном распределении значений яркости отсчетов (оптимальным является равномерное распределение на отрезки). Кроме того, в изображении может присутствовать и семантическая (смысловая) избыточность.

К настоящему времени разработано очень много методов и алгоритмов сокращения избыточности (сжатия) данных, в том числе для изображений. Эти методы можно охарактеризовать совокупностью показателей, применяемых для различных классов изображений и задач обработки.

Показатели эффективности. Для цифровых изображений полезность

(эффективность) процедуры сжатия данных можно охарактеризовать коэффициентом сжатия

$$K_c = \frac{I_0}{I}, (K_c > 1), \quad (5.1)$$

где I_0, I – объемы данных соответственно до и после сжатия.

Этот показатель является относительным, т.к. он характеризует эффективность алгоритма по отношению к корректному способу цифрового представления данных. Для изображений чаще всего используется представление в виде матрицы с форматом данных бит на отсчет, байт на отсчет, слово на отсчет.

Абсолютную эффективность алгоритма сжатия можно охарактеризовать коэффициентом эффективности

$$K_3 = \frac{I_{\min}}{I}, (K_3 < 1),$$

где $I_{\min}$ – минимально достижимый объем данных, соответствующий случаю полного устранения избыточности. Однако этот показатель используется редко из-за сложности определения $I_{\min}$. Эта величина определяется как энтропия источника сообщений и легко вычисляется лишь для некоторых моделей источников.

Показатели точности. Если произвести обработку (кодирование, компрессию) изображения с целью сжатия, а затем его восстановление (декодирование, декомпрессию) по сжатым данным, то в результате может получиться изображение, отличающееся от исходного. Чаще всего в качестве показателей качества используются показатели квадратичной погрешности и максимальной погрешности.

Квадратичная погрешность:

$$\epsilon_{\text{кв}}^2 = \frac{1}{MN} \sum_{m,n=0}^{N-1} [\bar{x}(m,n) - x(m,n)]^2. \quad (5.2)$$

Максимальная погрешность:

$$\epsilon_{\text{max}} = \max_{m,n} |\bar{x}(m,n) - x(m,n)|. \quad (5.3)$$

Здесь $M \times N$ - размеры изображения,
 $x(m, n)$ - отсчеты исходного изображения,
 $\bar{x}(m, n)$ - отсчеты восстановленного (декомпрессированного,
декодированного) изображения.

Для каждого изображения (или класса изображений) и для конкретного алгоритма сжатия данных существует жесткая зависимость показателей эффективности и точности. Чем выше проходит кривая, тем более эффективен алгоритм сжатия (полнее устраняется избыточность).

Максимальная погрешность представляет собой более строгий показатель качества, чем квадратичная. Действительно, при использовании показателя максимальной погрешности требуется, чтобы восстановленное изображение отличалось от исходного в каждой точке не более, чем на величину $\epsilon_{\max}$, в то время как при контроле квадратичной погрешности разница между исходным и восстановленным изображениями в небольшом числе точек может быть очень велика. Поэтому показатель максимальной погрешности обычно используют в задачах, предъявляющих повышенные требования к качеству восстановленных изображений, например в полиграфии.

Различают классы методов компрессии с постоянной и переменной скоростью формирования сжатых данных. Эти классы называют также классами методов с неконтролируемой и контролируемой погрешностью, соответственно.

Методы с постоянной скоростью формирования сжатых данных (методы с неконтролируемой погрешностью) для любого изображения обеспечивают заданный коэффициент сжатия. Эти методы особенно удобны для систем оперативного дистанционного зондирования земной поверхности, т.к. канал связи в таких системах имеет постоянную во времени пропускную способность. Методы с переменной скоростью формирования сжатых данных (методы с контролируемой погрешностью) обычно имеют больший коэффициент сжатия, чем методы с постоянной скоростью, но являются более сложными.

Принципиальная возможность сокращения объема данных при сжатии изображений основана на высокой степени информационной избыточности реальных изображений. Причина этого заключается, прежде всего, в высокой корреляционной зависимости между соседними отсчетами. В

настоящее время разработано много алгоритмов и методов сжатия данных, основанных на уменьшении корреляционной зависимости между отсчетами и последующем кодировании декоррелированных данных.

Системотехнические показатели. Основные из них – сложность алгоритма (в смысле затрат машинного времени) и требуемый объем оперативной памяти ЭВМ. Эти показатели следует относить отдельно к алгоритмам сжатия (кодирования) и к алгоритмам восстановления (декодирования). Сложность может измеряться в числе вычислительных операций на один отсчет или во времени обработки на конкретной ЭВМ. Из большого разнообразия методов сжатия изображений в данной лабораторной работе рассматривается один – метод кодирования с использованием спектральных преобразований.

5.1.2 Кодирование с преобразованием

Основная идея метода кодирования с преобразованием заключается в вычислении спектра изображения в каком-либо ортогональном базисе и квантование спектральных отсчетов. Для большинства реальных изображений значения многих спектральных компонент сравнительно малы. Такие компоненты (отсчета дискретного спектра) можно часто вообще отбросить (положить равными нулю) или отвести на их кодирование малое число двойных разрядов без риска сколько-нибудь значительных искажений изображения. Возникающие погрешности при использовании данного метода обычно оцениваются по среднеквадратичному критерию, т.к. лишь этот критерий инвариантен при спектральных преобразованиях изображений.

При разработке процедуры кодирования с преобразованием приходится решать две задачи:

- 1) выбор вида спектрального преобразования;
- 2) выбор метода обработки (кодирования) спектральных компонент.

Рассмотрим обе эти задачи.

5.1.3 Выбор преобразования

Для рассматриваемой задачи сжатия данных необходимо, чтобы используемое спектральное преобразование отвечало двум требованиям:

- 1) в результате преобразования основной объем информации об изображении должен быть сконцентрирован по возможности в меньшем числе спектральных компонентов;
- 2) преобразования (прямое и обратное) должны достаточно просто выполняться (с точки зрения затрат машинного времени).

Указанные требования являются противоречивыми, и для конкретного класса изображений необходим компромисс в выборе спектрального преобразования. Если известна (или измерена) автоковариационная функция (АКФ) изображения, то можно поставить задачу выбора оптимального базиса спектрального преобразования, исходя из первого требования. Для иллюстрации рассмотрим одномерный сигнал – последовательность $x(n)$, заданную на отрезке $n = 0, N-1$. Запишем спектральное разложение в общей форме:

$$X(m) = \sum_{n=0}^{N-1} x(n) A(m, n) - \text{прямое преобразование,}$$

$$x(n) = \sum_{m=0}^{N-1} X(m) A^{-1}(m, n) - \text{обратное преобразование.}$$

Здесь $X(m)$ – спектральные компоненты, а A и A^{-1} – ядра соответственно прямого и обратного преобразований (считаем, что преобразование является ортогональным, т.е. $A^{-1} = A_m$). Оптимальным преобразованием, идеальным с точки зрения первого требования, является так называемое преобразование Хотеллинга (дискретное преобразование Карунена-Лоэва). Это преобразование строится из условия получения некоррелированных спектральных компонент:

$$\begin{aligned} B_x(m, k) &= E[X(m)X(k)] = \\ &= \sum_{n=0}^{N-1} \sum_{l=0}^{N-1} A(m, n) A(k, l) B_{oc}(n, l) = D_x(m) \delta(m - k). \end{aligned} \quad (5.5)$$

Здесь B_{oc} и B_x – АКФ соответственно исходного сигнала и спектральных компонент, E – оператор математического ожидания, $D_x(m)$ – дисперсия спектральных компонент, $\delta(m)$ – единичный импульс. Из соотношения (5.5) можно получить

$$D_x(m) A(m, p) = \sum_{n=0}^{N-1} B_{oc}(m, p) A(m, n). \quad (5.6)$$

Видим, что функции $A(m, p)$ (m – номер функции) являются собственными функциями (векторами) АКФ (матрицы) $B_x(m, p)$, а $D_x(m)$ – ее собственные значения. Поэтому преобразование Хотеллинга называют еще разложением по собственным векторам АКФ. Можно показать, что кроме некоррелированности спектральных компонент преобразование Хотеллинга обеспечивает максимально быстрое убывание их дисперсии $D_x(m)$. Концентрация энергии (дисперсии) изображения в небольшом числе спектральных компонент и обеспечивает выполнение первого требования к преобразованию (рис. 5.1).

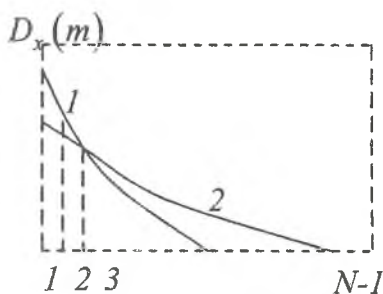


Рис. 5.1 Распределение энергии спектральных компонент преобразования Хотеллинга (1) в сравнении с другими базисами (2)

Все другие базисы дают более медленное убывание, т.е. требуют для той же точности представления данных сохранения большого числа спектральных компонент.

Однако, несмотря на столь важное качество, преобразование Хотеллинга на практике не применяются, так как, во-первых, его базис оказывается жестко привязанным к АКФ сигнала, что требует частой перестройки, а уравнение (5.6) в явном виде не решается. Во-вторых, для преобразования Хотеллинга не существует быстрого алгоритма преобразования (как БПФ), т.е. оно совершенно не удовлетворяет второму требованию, сформулированному выше.

С другой стороны, известно, что преобразование Хотеллинга хорошо аппроксимируется дискретным преобразованием Фурье (ДПФ): ДПФ также декоррелирует сигнал (асимптотически при $N \rightarrow \infty$). Недостатком ДПФ является его комплекснозначность, что затрудняет обработку спектральных компонент. Разработаны и используются вещественные базисы: Фурье в форме Хартли, косинусный, синусный, Уолша, Хаара и много других.

На первое место по близости к оптимальному преобразованию Хотеллинга для большинства реальных сигналов и изображений (например, с экспоненциальной АКФ) можно поставить косинусное преобразование. Немного ему уступает и преобразование Фурье в форме Хартли. Преобразования Уолша и, особенно, Хаара значительно уступают оптимальному преобразованию по показателю точности и применяются тогда, когда предъявляются жесткие требования к скорости вычислений.

Последние два преобразования могут обеспечить высокую эффективность сжатия для некоторых специальных классов изображений (бинарных с прямоугольной формой объектов, таких, например, как шахматное поле).

В заключение приведем аналитические выражения и графики для упомянутых базисных функций (с точностью до постоянного множителя).

Дискретное преобразование Фурье имеет следующий вид:

$$A(m, n) = \exp\left(-i \frac{2\pi}{N} mn\right).$$

Дискретное преобразование Фурье в форме Хартли (ДПФХ) представлено следующим образом:

$$A(m, n) = \cos\left(\frac{2\pi}{N} mn + \frac{\pi}{4}\right).$$

Дискретное косинусное преобразование (ДПК)

$$A(m, n) = \begin{cases} 1, & m = 0 \\ \sqrt{2} \cos \frac{(2m+1)n\pi}{2N}, & 1 \leq m < N. \end{cases}$$

Графики базисных функций Уолша и Хаара показаны на рис. 5.2 и рис. 5.3 (приведены первые 4 базисные функции).

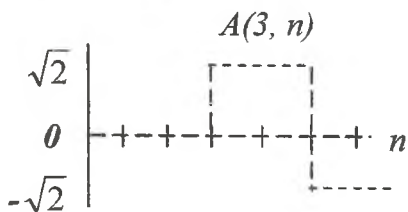
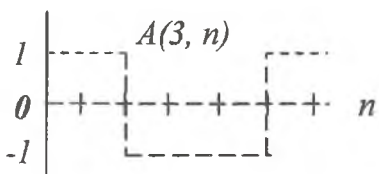
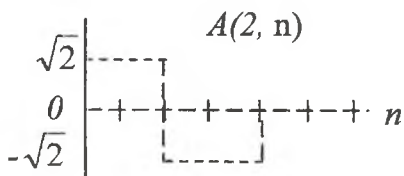
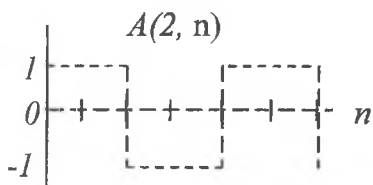
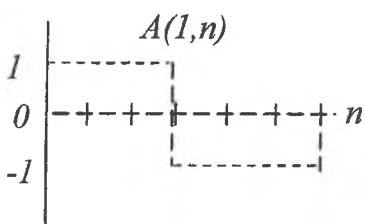
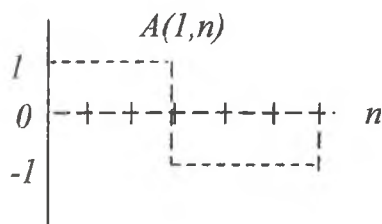
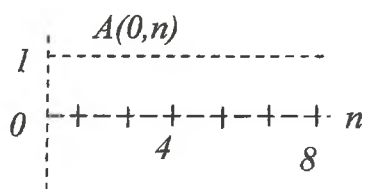
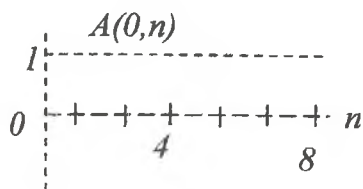


Рис. 5.2. Базисные функции Уолша

Рис 5.3. Базисные функции Хаара

5.1.4 Кодирование спектральных компонент

Обработка результатов спектрального преобразования заключается в выполнении двух новых функций:

- 1) отбор наиболее существенных спектральных компонент;
- 2) их квантование, т.е. представление определенным числом двоичных разрядов.

Обе эти операции приводят к уменьшению объема данных (т.е. обеспечивают эффект сжатия) и в то же время вносят погрешности в обрабатываемый сигнал. Иногда используется только одна из указанных операций: либо отбор, либо квантование.

Осуществлению оптимального квантования спектральных компонент часто препятствует сложность расчета шкал и разрядностей (они должны отдельно строиться для каждой компоненты). Поэтому на практике квантования стараются избежать. В настоящей лабораторной работе мы также ограничимся исследованием эффекта только отбора спектральных компонент.

Для реализации отбора существует много простых эвристических процедур, основанных на практическом опыте. Наиболее популярен так называемый зональный отбор компонент, при котором оставляют только те из них, которые попадают в зону с наибольшими их дисперсиями, а остальные полагаются равными нулю. Чем меньше спектральных компонент сохраняется, тем выше будет эффект сжатия, но тем больше и погрешность (разница между исходным сигналом $x(n)$ и восстановленным $x'(n)$).

В двумерном случае для изображения размера $N \times N$ получаем матрицу спектральных компонент тех же размеров. Процедура зонального отбора эквивалентна умножению этой матрицы на бинарную маску таких же размеров $N \times N$. Матрица маски содержит единицы на месте сохраняемых спектральных компонент и нули на месте отбрасываемых спектральных компонент. Типичные виды бинарных масок представлены на рис. 5.4.

Отношение площади маски ($N \times N$) к площади области, содержащей единицы (на рис. 5.4. она заштрихована), равно коэффициенту сжатия K_c (5.1). Квадратичная погрешность процедуры сжатия/восстановления равна (с точностью до множителя $1/(N \times N)$) энергии спектральных компонент, попадающих в нулевую область.

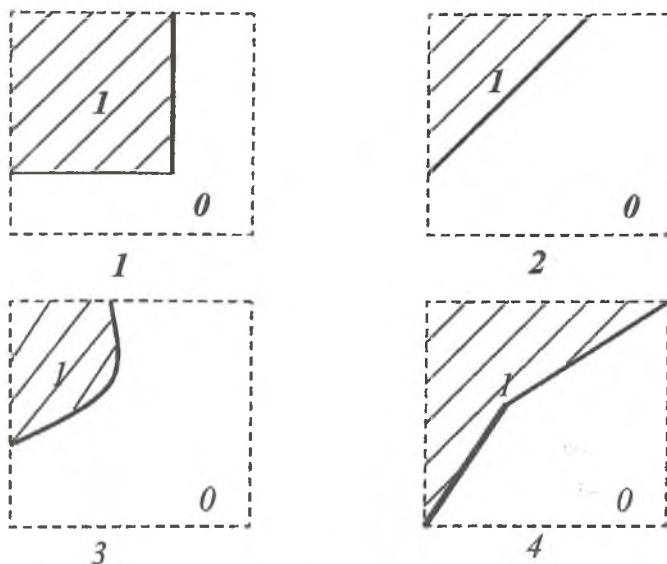


Рис. 5.4 Виды бинарных масок для зонального отбора

5.2 Выполнение лабораторной работы

5.2.1 Общий план выполнения работы

1. Изучить теоретические основы применения спектральных преобразований для сжатия данных.
2. Ознакомиться с описанием команд входного языка.
3. Составить последовательность команд входного языка для экспериментального исследования применения спектральных преобразований для сжатия данных.
4. Согласовать план проведения исследования с преподавателем, получить разрешение на практическую работу.
5. Выполнить работу на автоматизированном комплексе.
6. Составить отчет о выполненной работе.
7. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

5.2.2 Порядок проведения обработки

Написать командный файл (программу в виде bat-файла) с двумя параметрами: имя изображения и коэффициент сжатия.

Программа должна имитировать сжатие и восстановление заданного изображения. То есть программа должна генерировать восстановленное изображение, как если бы сжатие и восстановление было проведено, но не должна генерировать собственно сжатого файла.

С помощью программы должны быть получены следующие результаты:

- 1) восстановленное изображение;
- 2) квадратичная погрешность восстановления для метода компрессии, включающего следующие шаги.

Этап компрессии:

- (1) Вычисление спектра.
- (2) Зональный отбор спектральных компонент с помощью маски.
- (3) Сохранение отобранных спектральных компонент в архиве.

Этап декомпрессии:

- (А) Загрузка спектральных компонент из архива.
- (Б) Обратное спектральное преобразование.

Замечание: требуемые результаты могут быть получены без реализации этапов (3) и (А), поэтому эти этапы реализовывать не нужно.

Исходные данные:

1. Вид преобразования: Фурье, Уолша, Хаара.
2. Вид маски: круг.
3. Коэффициенты сжатия: 2, 4, 10

5.2.3 Содержание отчета

Отчет по работе должен содержать:

1. Последовательность команд обработки с комментариями и числовыми значениями параметров.
2. Перечень созданных изображений с указанием их смыслового содержания.
3. Квадратичные погрешности восстановления изображений для заданных видов преобразования и коэффициентов сжатия.
4. Выводы по результатам исследования.

5.2.4 Контрольные вопросы

1. Постановка задачи сжатия данных.
2. Показатели, характеризующие алгоритмы сжатия данных.
3. Принцип кодирования с преобразованием.
4. Критерий выбора спектрального преобразования.
5. Сравнительный анализ базисов спектральных преобразований.
6. Что такое маска для зонального отбора?
7. Общая характеристика программного обеспечения.

6 МЕТОД ДИФФЕРЕНЦИАЛЬНОГО КОДИРОВАНИЯ ИЗОБРАЖЕНИЙ

Цель работы – изучение методов сокращения избыточности на основе дифференциальной импульсно-кодовой модуляции; получение навыков работы с автоматизированной системой обработки изображений

В лабораторной работе изучается метод компрессии изображений на основе дифференциальной импульсно-кодовой модуляции, приведено описание программных средств сжатия изображений для автоматизированной системы обработки изображений.

6.1 Теоретические основы лабораторной работы

6.1.1 Общее описание метода ДИКМ

Для понимания дальнейшего материала необходимо предварительно ознакомиться с содержанием раздела “Проблемы сжатия изображений” (см. п. 5.1.1 настоящего пособия). Как было отмечено в этом разделе, большинство алгоритмов и методов сжатия изображений основаны на уменьшении корреляционной зависимости между отсчетами и последующем кодировании декоррелированных данных.

Одним из таких методов сжатия является метод на основе дифференциальной импульсно-кодовой модуляции (ДИКМ). Декорреляция в этом методе осуществляется за счет перехода к разностному

представлению сигнала, а собственно уменьшение объема данных достигается за счет кодирования этого разностного сигнала кодовыми словами фиксированной длины, меньшей, чем при представлении исходного сигнала.

При сжатии методом ДИКМ отсчеты изображения $x(m, n)$ обрабатываются в порядке построчной развертки (см. рис. 6.1). Обычно отсчеты реальных изображений сильно коррелированы, а это означает плавное изменение функции яркости. Благодаря этому разница между соседними отсчетами будет почти всегда близка к нулю, что позволяет обойтись меньшим числом разрядов при ее цифровом представлении. На этом и основан данный метод ДИКМ.



Рис. 6.1 Последовательность обхода отсчетов изображения при построчной развертке

Идея заключается в том, что в порядке поступления из каждого отсчета $x(m, n)$ вычитается некоторое предсказанное значение (опорный сигнал) $\hat{x}(m, n)$, т.е. формируется разностный сигнал:

$$f(m, n) = x(m, n) - \hat{x}(m, n). \quad (6.1)$$

Квантованию по уровню в ДИКМ подвергается не исходный, а разностный сигнал (6.1). Квантованный разностный сигнал $\bar{f}(m, n)$ поступает на вход процедуры кодирования (см. рис. 6.2а). В результате получается кодированный (сжатый) сигнал, который записывается в архивный файл или передается в канал связи.

При декомпрессии сначала восстанавливается (декодируется) квантованный разностный сигнал $\bar{f}(m, n)$. Затем к отсчетам $\bar{f}(m, n)$

прибавляется опорный сигнал $\hat{x}(m, n)$. Таким образом формируются значения отсчетов восстановленного сигнала (см. рис. 6.2б):

$$\bar{x}(m, n) = \bar{f}(m, n) + \hat{x}(m, n). \quad (6.2)$$

Важно, чтобы при компрессии и при декомпрессии использовались одинаковые устройства (алгоритмы) предсказания, т.е. формировался один и тот же опорный сигнал $\hat{x}(m, n)$. При восстановлении в нашем распоряжении уже нет исходного сигнала, имеется только восстановленный сигнал $\bar{x}(m, n)$, поэтому предсказание осуществляется именно по нему. Соответственно, то же самое делается и при кодировании. Таким образом, общая схема кодера и декодера метода ДИКМ имеет вид, показанный на рис. 6.2.

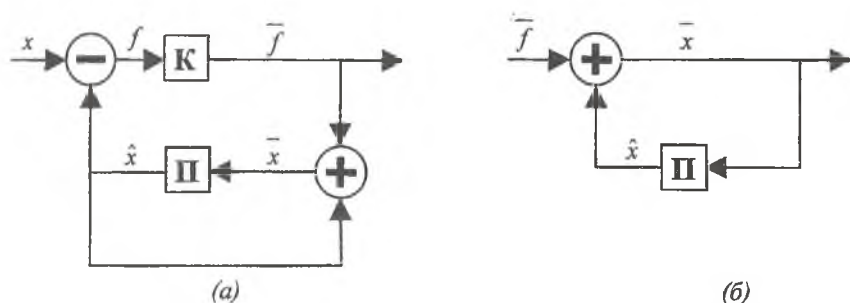


Рис. 6.2. Схема дифференциального кодера (а) и декодера (б):

"К" – квантователь; "П" – предсказатель

В методе ДИКМ в сжатые данные записывается квантованный разностный сигнал. Обычно для представления этого сигнала используется фиксированное число разрядов, меньшее, чем для представления исходного сигнала. Следовательно, коэффициент сжатия метода ДИКМ может быть вычислен следующим образом:

$$K_{\text{сжс}} = \frac{b_0}{b},$$

где b_0 – число разрядов, используемое при исходном представлении сигнала;

b – число разрядов, используемое при представлении квантованного

(кодированного) сигнала (в архивном файле или канале связи).

Таким образом, на этапе компрессии метод ДИКМ включает следующие шаги (для каждого отсчета):

К-1. Предсказание

К-2. Вычисление разностного значения (6.1)

К-3. Квантование разностного значения

К-4. Кодирование квантованного значения

К-5. Вычисление восстановленного значения (6.2)

Этап декомпрессии, в свою очередь, включает следующие шаги:

Д-1. Предсказание (идентично К-1)

Д-2. Декодирование квантованного значения

Д-3. Вычисление восстановленного значения (идентично К-5)

Достоинства метода ДИКМ – простота и постоянная скорость формирования сжатых данных, что делает его чрезвычайно привлекательным для систем оперативного дистанционного зондирования земной поверхности.

Недостаток метода ДИКМ – низкая помехоустойчивость при передаче кодированных данных: если даже всего один отсчет квантованного разностного сигнала содержит сбой, то влияние этого сбоя распространится до конца сигнала.

Качество восстановленного изображения метода ДИКМ оценивают обычно по критериям визуального восприятия или среднеквадратичной ошибки.

6.1.2 Алгоритмы предсказания отсчетов для метода ДИКМ

Эффективность метода ДИКМ в значительной степени определяется точностью алгоритма предсказания: чем точнее предсказание, тем меньше дисперсия разностного сигнала и, следовательно, тем лучше он сжимается. Следовательно, основными требованиями к предсказателю являются как можно большая точность при минимальной вычислительной сложности. Поэтому устройство предсказателя обычно строится как линейная система с постоянными параметрами (ЛПП-система):

$$\hat{x}(m, n) = \sum_{(k, l) \in D} g(k, l) \bar{x}(m - k, n - l),$$

где $g(k, l)$ – импульсная характеристика предсказателя;

D – область ее ненулевых значений, которая должна быть задана таким образом, чтобы обеспечить физическую реализуемость системы при заданном виде развертки. Например, при построчной развертке на область D налагается следующее ограничение:

$$D \subseteq \{(m, n) : m \geq 0, n > 0\} \cup \{(m, n) : m > 0, n \geq 0\}.$$

Проблемы, возникающие при построении линейного предсказателя заключаются, в частности, в том, что сам кодер системы ДИКМ является нелинейным из-за наличия квантователя, вследствие чего анализ его характеристик чрезвычайно сложен. Однако было доказано, что при построении предсказателя наличием квантователя можно пренебречь, и это не приведет к существенному ухудшению результата. Другими словами при построении предсказателя можно считать, что предсказание осуществляется не по восстановленным, а по исходным значениям сигнала.

Импульсная характеристика предсказателя ищется исходя из минимизации квадратичной погрешности предсказания:

$$\sum_{(m, n)} \sum (\hat{x}(m, n) - x(m, n))^2 \rightarrow \min_{g(k, l)}.$$

Другими словами, задача состоит в нахождении параметров физически реализуемой ЛПП-системы, генерирующей оптимальную оценку сигнала на шаг вперед. Эта задача относится к классу задач синтеза линейных восстанавливающих фильтров. При решении этой задачи отсчеты импульсной характеристики восстанавливающего фильтра (т.е. предсказателя) определяются из уравнений Винера-Хопфа, которые в данном случае записываются в виде:

$$\begin{cases} \sum_{(k, l) \in D} g(m, n) R_x(m - k, n - l) = R_x(m, m), & (m, n) \in D, \\ g(m, m) = 0, & (m, n) \notin D; \end{cases} \quad (6.3)$$

где $R_x(m, n)$ – автоковариационная функция (АКФ) сжимаемого

изображения. Дисперсия разностного сигнала, полученного при использовании рассчитанного таким образом предсказателя, задается выражением

$$\sigma_f^2 = \sigma_x^2 \left(1 - \sum_{(m,n) \in D} \sum g(m,n) R_x(m,n) \right),$$

где σ_x^2 - дисперсия сжимаемого изображения.

Практика показывает, что на реальных изображениях зависимость между отсчетами быстро уменьшается с ростом расстояния между ними. Поэтому обычно при предсказании имеет смысл учитывать только ближайшие к предсказываемому отсчеты. Такой предсказатель на основе ближайших отсчетов может быть записан в виде (см. также рис. 6.3):

$$\hat{x}(m,n) = a_1 \bar{x}(m,n-1) + a_2 \bar{x}(m-1,n) + a_3 \bar{x}(m-1,n-1) + a_4 \bar{x}(m-1,n+1),$$

где $A = (a_1, a_2, a_3, a_4)$ - вектор коэффициентов, состоящий из отсчетов импульсной характеристики предсказателя, которые определяются из решения системы уравнений (6.3), причем нетрудно видеть, что

$$a_1 = g(1,0), \quad a_2 = g(0,1), \quad a_3 = g(1,1), \quad a_4 = g(1,-1).$$

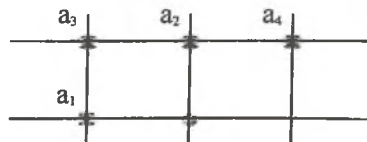


Рис. 6.3 Двумерное линейное предсказание по четырем отсчетам

Однако для того, чтобы реализовать такой способ определения коэффициентов предсказателя, необходимо знать или оценивать АКФ сжимаемого изображения, что приемлемо далеко не для всех прикладных задач сжатия. Поэтому на практике часто используют следующие предсказатели, хорошо работающие на большинстве реальных изображений:

$$\text{П-1. } A = (1, 0, 0, 0);$$

$$\text{П-2. } A = (0.5, 0.5, 0, 0); \quad (6.4)$$

$$\text{П-3. } A = (0.25, 0.25, 0.25, 0.25);$$

$$\text{П-4. } A = (1, 1, -1, 0).$$

Общий недостаток всех линейных предсказателей заключается в том, что они дают всплеск ошибки предсказания (разностного сигнала) при прохождении участков со скачкообразным изменением яркости. На восстановленном изображении этот эффект проявляется в виде размывания границ областей и контурных линий.

6.1.3 Квантование разностного сигнала в методе ДИКМ

Под квантованием произвольного параметра f с плотностью вероятности $W(f)$ будем понимать следующий процесс. Пусть область значений параметра f разбита на L интервалов квантования с границами $d_i, i \in [0, L]$, причем на каждом интервале квантования выбрано представительное значение (квантованный уровень) $f_i, i \in [0, L-1]$ (см. рис. 6.4). Совокупность границ интервалов квантования и квантованных уровней будем называть шкалой квантования.

Квантование заключается в замене квантуемого значения параметра f на квантованный уровень f_i того интервала квантования $[d_i, d_{i+1})$, в который попадает это квантуемое значение:

$$f \rightarrow f_i : f \in [d_i, d_{i+1}).$$

Наиболее часто используется равномерная шкала квантования, в которой размеры всех интервалов квантования одинаковы, а квантованные уровни располагаются в центрах интервалов квантования. Эта шкала наиболее проста и удобна в использовании, однако, как показано ниже, для метода ДИКМ ее применение нецелесообразно.



Рис. 6.4 Шкала квантования

Квантованию в ДИКМ подвергается разностный сигнал $f(m, n)$ (6.1) (для краткости далее аргументы не пишем). При этом, для достижения эффекта сжатия при кодировании квантованного разностного сигнала $\bar{f}$ обычно используется малое число разрядов b . Другими словами, это означает, что шкала содержит малое число уровней квантования $L = 2^b$.

Равномерную шкалу обычно строят таким образом, чтобы она была согласована с дисперсией квантуемого параметра. Например, при построении шкалы по правилу "трех сигм", размах шкалы $(d_L - d_0)$ выбирается равным $6\sigma_f^2$. В этом случае шкала получается "широкой" и вероятность "зашкаливания" очень мала. Однако, так как уровней мало, то расстояние между уровнями получается слишком большим, поэтому квантованный сигнал представляет исходный с недопустимо большой погрешностью (см. рис. 6.5).

Попытка уменьшить эту погрешность путем сужения шкалы приведет к слишком частому «зашкаливанию» разностного сигнала f и ограничению квантованного сигнала $\bar{f}$ по амплитуде, по сравнению с неквантованным. Таким образом, резко возрастает погрешность восстановления сигнала на быстро меняющихся участках – возникают «перегрузки по наклону» (см. рис. 6.6).

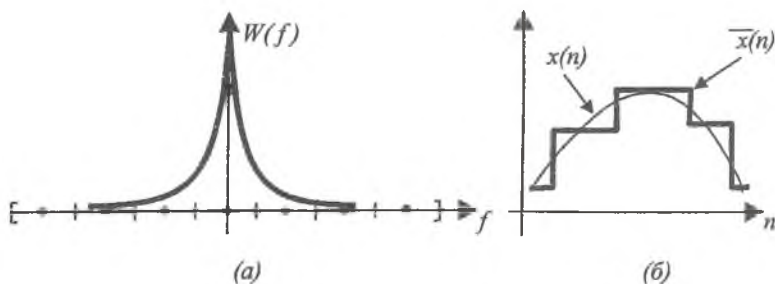


Рис. 6.5 Пример слишком широкой шкалы квантования:
(а) – собственно шкала; (б) – последствия неудачного выбора шкалы

Таким образом, становится ясно, что использование равномерной шкалы для метода ДИКМ неэффективно. Выходом из сложившейся ситуации является применение неравномерных шкал, в которых уровни квантования выбираются тем уже, чем больше вероятность попадания в такой интервал.

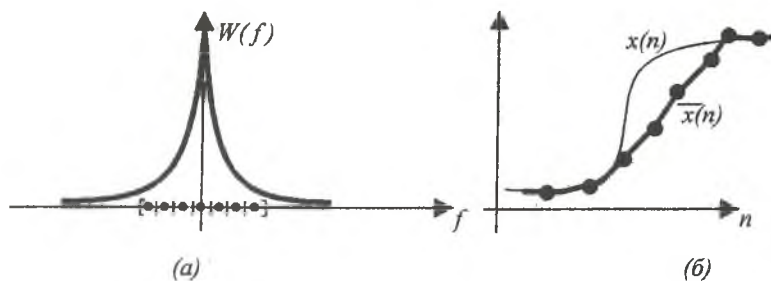


Рис. 6.6 Пример слишком узкой шкалы квантования:

(а) – собственно шкала; (б) – иллюстрация для перегрузок по наклону

Одной из наиболее часто используемых неравномерных шкал является шкала, предложенная Максом в 1960 году. Она строится из условия минимизации квадратичной погрешности квантования

$$\varepsilon_{\text{кв}}^2 = \frac{1}{L} \sum_{i=0}^{L-1} \int_{d_i}^{d_{i+1}} (f - f_i)^2 W(f) df \quad (6.5)$$

при заданном количестве уровней квантования, т.е. является решением задачи

$$\begin{cases} \varepsilon_{\text{кв}}^2 \rightarrow \min, \\ d_i, f_i \\ d_0 = -\infty; d_L = \infty; L = 2^b - \text{fix}. \end{cases} \quad (6.6)$$

Из (6.6) можно получить следующие выражения для квантованных уровней и границ интервалов квантования:

$$f_i = \frac{\int_{d_i}^{d_{i+1}} f W(f) df}{\int_{d_i}^{d_{i+1}} W(f) df}, \quad (6.7)$$

$$d_i = \frac{f_i + f_{i-1}}{2}. \quad (6.8)$$

Так как f_i и d_i зависят друг от друга, то, в общем случае, шкала Макса строится итерационным способом:

Шаг 1. В качестве начального приближения взять равномерную шкалу.

Шаг 2. По формуле (6.7) пересчитывают все f_i .

Шаг 3. По формуле (6.8) пересчитывают все d_i .

Шаг 4. Если при последнем пересчете погрешность (6.5) изменилась менее чем на заданную величину, или квантованные уровни подвинулись менее чем на заданную величину, то стоп. Иначе переход на шаг 2.

В некоторых случаях удастся построить шкалу Макса более простым способом. Например, известно, что для ДИКМ разностный сигнал f имеет распределение вероятностей, близкое к распределению Лапласа:

$$W(f) = \frac{1-\alpha}{1+\alpha} e^{-\alpha |f|}, \quad (6.9)$$

где α – параметр, регулирующий "ширину" распределения.

Нетрудно видеть, что в этом случае все интегралы в формуле (6.7) берутся аналитически, а все f_i и d_i выражаются через величину центрального интервала шкалы Макса.

6.2 Выполнение лабораторной работы

6.2.1 Общий план выполнения работы

1. Изучить теоретические основы применения ДИКМ для сжатия изображений.
2. Ознакомиться с описанием используемых модулей программной системы обработки изображений.
3. Получить от преподавателя исследуемое изображение.
4. Составить последовательность команд для решения поставленной задачи.
5. Согласовать план проведения обработки с преподавателем, получить разрешение на практическую работу на системе.
6. Выполнить работу на автоматизированной системе обработки

изображений.

7. Составить отчет о выполненной работе.
8. Сдать отчет преподавателю, ответить на контрольные вопросы, получить зачет по работе.

6.2.2 Порядок проведения обработки

Задание 1: Выбор самого точного предсказателя для метода ДИКМ

Выбрать самый точный предсказатель для метода ДИКМ из набора (6.4). Исследование проводить на заданном преподавателем изображении при коэффициентах сжатия 2, 4, 8. Обосновать сделанный выбор. При этом считать, что распределение вероятностей разностного сигнала имеет Лапласовский вид (6.9).

Задание 2: Изучение помехоустойчивости метода ДИКМ

Для коэффициента сжатия 8 и для каждого предсказателя из набора (6.4) провести эксперимент следующего вида. Считаем, что в канале связи произошел сбой, имитация которого производится путем инвертирования одного бита в сжатых данных. При этом искажаемый бит должен находиться в первой четверти сжатого файла. Повторить эксперимент еще три раза, располагая сбойный бит во второй, третьей и четвертой четверти файла. Определить наиболее помехоустойчивый предсказатель. Обосновать сделанный выбор.

Вывести на экран контрастированные изображения погрешностей, вызванных сбоем, то есть погрешностей между восстановленным без сбоя изображением и восстановленным после сбоя. Визуально классифицировать полученные изображения погрешностей. Объяснить полученные результаты.

6.2.3 Содержание отчета

1. Последовательность команд обработки с их параметрами, все необходимые командные файлы.
2. Все результаты, достаточные для обоснования выбора самого точного предсказателя для метода ДИКМ, сделанного при

выполнении задания 1

3. Все результаты, достаточные для обоснования выбора самого помехоустойчивого предсказателя для метода ДИКМ, сделанного при выполнении задания 2
4. Перечень всех требуемых в задании 1 и задании 2 изображений с указанием их имен и смыслового содержания.

6.2.4 Контрольные вопросы

1. Показатели эффективности методов сжатия изображений.
2. Показатели точности методов сжатия изображений.
3. Метод сжатия изображений на основе ДИКМ.
4. Предсказание при сжатии изображений на основе ДИКМ.
5. Квантование при сжатии изображений на основе ДИКМ.
6. Шкала Макса.
7. Общая характеристика программного обеспечения.

7 МЕТОД КОМПРЕССИИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ИЕРАРХИЧЕСКОЙ СЕТОЧНОЙ ИНТЕРПОЛЯЦИИ

Цель работы - изучение теоретических основ и экспериментальное исследование метода компрессии на основе иерархической сеточной интерполяции. Моделирование метода иерархической сеточной интерполяции средствами автоматизированной системы обработки изображений.

В лабораторной работе изучается метод компрессии изображений на основе иерархической сеточной интерполяции, приведено описание программных средств, необходимых для моделирования этого метода сжатия средствами автоматизированной системы обработки изображений.

7.1 Теоретические основы лабораторной работы

7.1.1 Общая схема метода компрессии на основе ИСИ

Для понимания дальнейшего материала необходимо предварительно ознакомиться с содержанием раздела “Проблемы сжатия изображений” (см. п. 5.1.1 настоящего пособия). Как было отмечено в этом разделе, большинство алгоритмов и методов сжатия изображений основаны на уменьшении корреляционной зависимости между отсчетами и последующем кодировании декоррелированных данных.

Одним из таких методов сжатия является метод на основе иерархической сеточной интерполяции (ИСИ). Декорреляция в этом методе осуществляется за счет перехода к иерархическому разностному представлению сигнала, а собственно уменьшение объема данных достигается за счет статистического кодирования этого разностного сигнала.

Пусть $\mathbf{X} = \{x(m, n)\}$ – исходное изображение. Рассмотрим представление изображения $\mathbf{X}$ в виде объединения иерархических уровней $\mathbf{X}_l$:

$$\mathbf{X} = \bigcup_{l=0}^{L-1} \mathbf{X}_l, \quad (7.1)$$

$$\mathbf{X}_{L-1} = \{x_{L-1}(m, n)\}, \quad (7.2)$$

$$\mathbf{X}_l = \{x_l(m, n)\} \setminus \{x_{l+1}(m, n)\}, \quad 0 \leq l \leq L-2, \quad (7.3)$$

где L – количество уровней,

$\{x_l(m, n)\}$ – массив отсчетов изображения, взятых с шагом 2^l по каждой координате:

$$x_l(m, n) = x(2^l m, 2^l n).$$

Например, для изображения 9*9 пикселей и числе уровней $L=4$

массивы $\{x_l(m, n)\}$ для $0 \leq l \leq L-1$ располагаются в исходном изображении так, как показано на рис. 7.1. Тогда, соответственно, отсчеты иерархических уровней X_l , $0 \leq l \leq L-1$ будут располагаться на изображении так, как показано на рис. 7.2.

Представление (7.1)-(7.3) дает возможность при компрессии и декомпрессии обрабатывать уровни последовательно, начиная со старшего уровня X_{L-1} , причем отсчеты каждого более старшего (менее детального) уровня будут использоваться для интерполяции отсчетов более младшего (более детального) уровня.

Количество уровней выбирается таким образом, чтобы объем старшего (наиболее прореженного) уровня был пренебрежимо мал по сравнению с объемом данных всего изображения, поэтому старший уровень может быть записан в сжатое изображение и без компрессии.

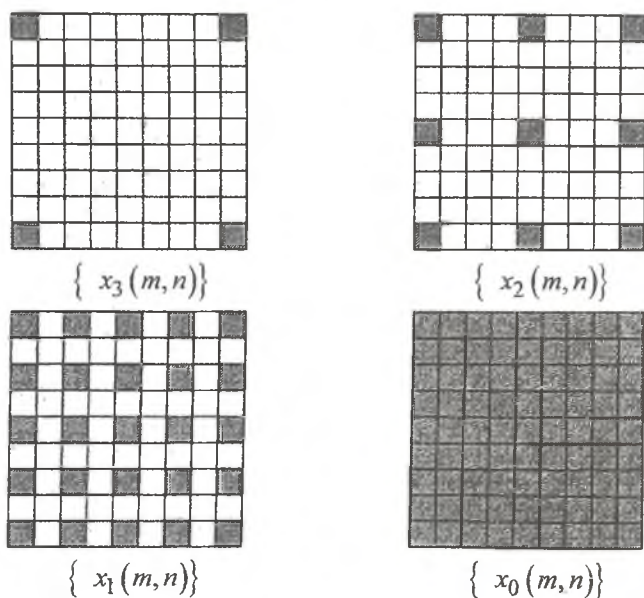


Рис. 7.1 Расположение отсчетов прореженных сеток на изображении

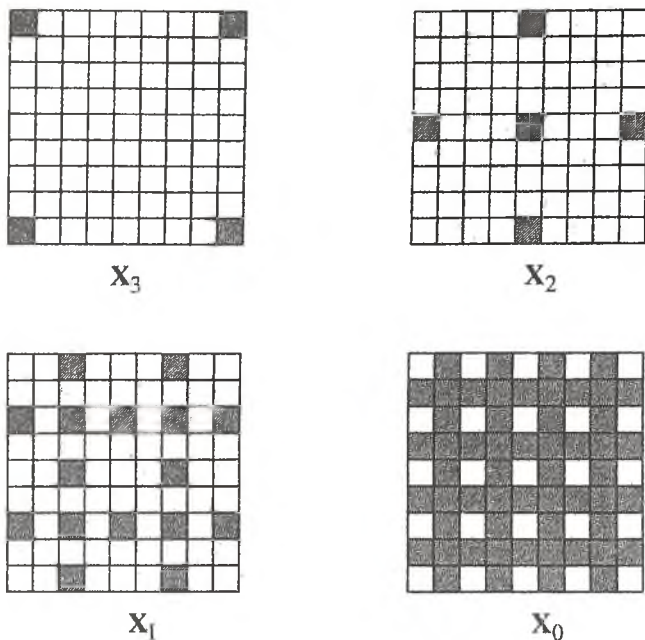


Рис. 7.2 Расположение отсчетов иерархических уровней на изображении (число уровней равно четырем)

Процедура компрессии любого из следующих младших уровней X_l , $l < L-1$ включает следующие этапы (см. также рис. 7.3):

1) Интерполяция

Производится интерполяция отсчетов $\{x_l(m, n)\}$ уровня X_l на основе отсчетов $\{\bar{x}_k(m, n), k > l\}$ уровней $\{X_k, k > l\}$, уже прошедших компрессию и восстановление:

$$\hat{x}_l(m, n) = P \left(\bigcup_{k=l+1}^{L-1} \{ \bar{x}_k(m, n) \} \right), \quad (7.4)$$

где $\hat{x}_l(m, n)$ – интерполирующие значения отсчетов, а $P(\dots)$ – функция, задающая некоторый интерполятор. Конкретный пример интерполятора приведен далее.

2) Вычисление постинтерполяционных остатков

Вычисляется массив разностей истинных и интерполирующих значений отсчетов ("постинтерполяционных остатков"):

$$f_l(m, n) = x_l(m, n) - \hat{x}_l(m, n). \quad (7.5)$$

3) Квантование постинтерполяционных остатков

Выполняется квантование постинтерполяционных остатков (7.5). При этом для обеспечения контроля максимальной погрешности $\varepsilon_{\max}$ (5.3) используется квантователь вида:

$$q_l(m, n) = \text{sign} \left(f_l(m, n) \right) \left[\frac{|f_l(m, n)| + \varepsilon_{\max}}{2\varepsilon_{\max} + 1} \right], \quad (7.6)$$

где $[..]$ обозначает выделение целой части, а sign - функция вычисления знака числа:

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

4) Вычисление восстановленных значений отсчетов

По квантованным значениям постинтерполяционных остатков вычисляются восстановленные значения $\bar{x}_l(m, n)$ отсчетов уровня. Очевидно, что для этого необходимо сначала вычислить восстановленные значения постинтерполяционных остатков:

$$\bar{f}_l(m, n) = q_l(m, n)(2\varepsilon_{\max} + 1), \quad (7.7)$$

а затем осуществить собственно восстановление:

$$\bar{x}_l(m, n) = \bar{f}_l(m, n) + \hat{x}_l(m, n). \quad (7.8)$$

Эти восстановленные отсчеты будут нужны для интерполяции более младших иерархических уровней $\{X_k, k < l\}$.

5) Статистическое кодирование

Осуществляется статистическое кодирование квантованных постинтерполяционных остатков. Поскольку распределение их вероятностей является существенно неравномерным, в результате использования кодов переменной длины можно достичь значительного сокращения объема данных, которое и является целью выполняемой обработки изображения. Для метода ИСИ обычно используется (с небольшими модификациями) кодирование по Хаффмену.

Процедура декомпрессии уровня $X_l, l < L-1$ включает следующие этапы (см. также рис. 7.4):

1) Декодирование

Производится декодирование квантованных значений постинтерполяционных остатков (7.6).

2) Интерполяция

Производится интерполяция вида (7.4) по восстановленным отсчетам.

3) Восстановление

Как и при компрессии, производится деквантование (7.7) и вычисление восстановленных значений отсчетов (7.8).

Достоинства метода ИСИ

1. Высокий коэффициент компрессии.
2. Низкая вычислительная сложность.
3. Контроль максимальной погрешности.
4. Возможность иерархического доступа к сжатым данным (полезно для организации быстрого просмотра, поиска и т.п. на сжатых изображениях).

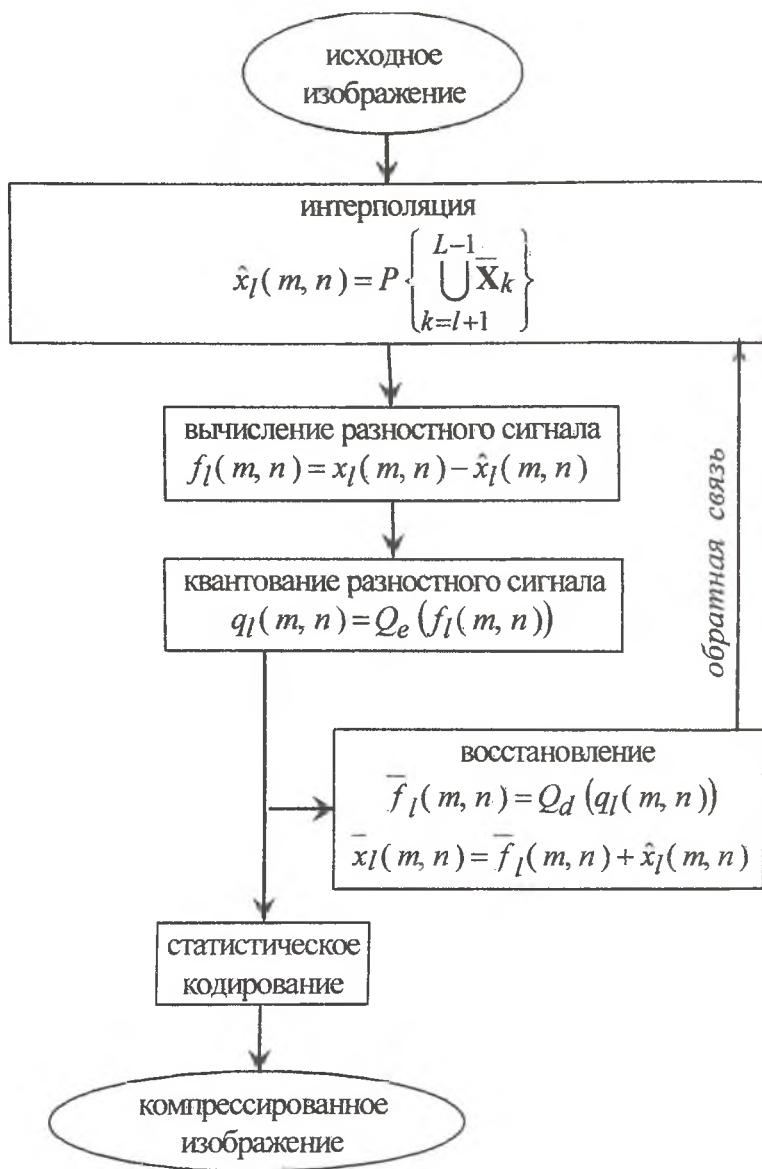


Рис. 7.3 Компрессия иерархического уровня X_l , $l < L-1$

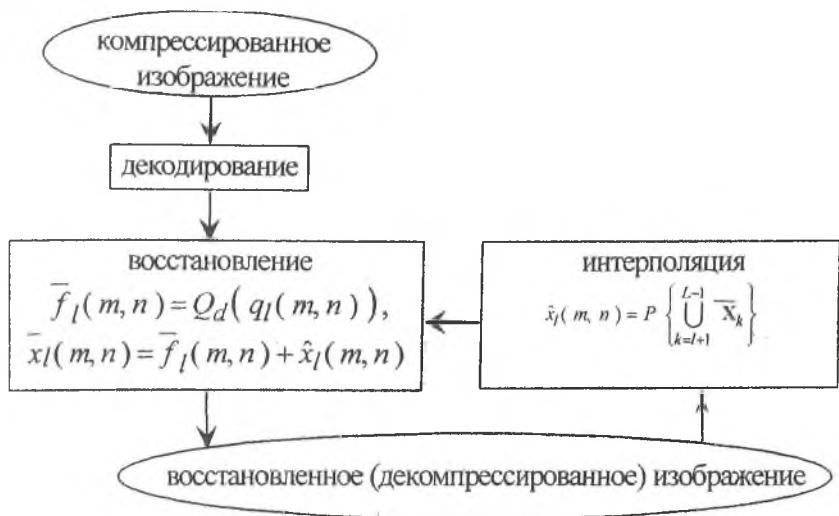


Рис. 7.4 Декомпрессия иерархического уровня $X_l, l < L-1$

7.1.2 Алгоритм интерполяции для метода ИСИ

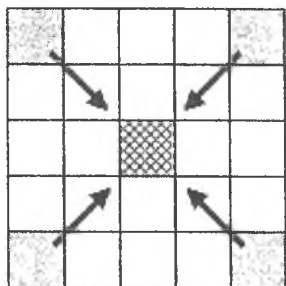
Главным требованием к интерполятору в методе ИСИ является как можно большая точность при как можно меньшей вычислительной сложности. Поэтому обычно для интерполяции используется простое усреднение по ближайшим восстановленным (уже обработанным) отсчетам более старших уровней.

Будем отдельно рассматривать два типа интерполируемых отсчетов. Отсчеты вида $x_l(2m+1, 2n+1)$ будем называть "центральными" (см. рис. 7.5). Нетрудно видеть, что четыре ближайших восстановленных отсчета более старшего уровня находятся по диагонали от центрального. Отсчеты вида $x_l(2m+1, 2n)$ и $x_l(2m, 2n+1)$ будем называть "крайними" (см. рис. 7.5). Два ближайших восстановленных отсчета более старшего уровня находятся по вертикали или по горизонтали от крайнего. Укажем способы интерполяции одного крайнего и центрального отсчетов:

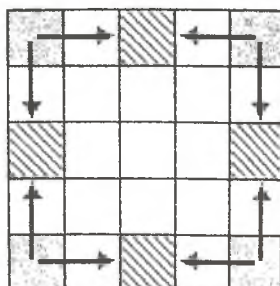
$$\hat{x}_l(2m+1, 2n) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n) \rangle,$$

$$\hat{x}_l(2m+1, 2n+1) = \langle \bar{x}_l(2m, 2n), \bar{x}_l(2m+2, 2n), \bar{x}_l(2m, 2n+2), \bar{x}_l(2m+2, 2n+2) \rangle$$

где $\langle \dots \rangle$ обозначает усреднение с последующим округлением до ближайшего целого.



(a)



(б)

Рис. 7.5 Интерполяция центральных (а) и крайних (б) отсчетов



- опорные (восстановленные) отсчеты более старших уровней



- интерполируемые "центральные" отсчеты вида $x_l(2m+1, 2n+1)$



- интерполируемые "крайние" отсчеты
вида $x_l(2m+1, 2n)$ и $x_l(2m, 2n+1)$

7.2. Порядок выполнения лабораторной работы

Исходные данные:

- 1) Количество иерархических уровней $L = 2$
- 2) Старший уровень сжимается без внесения погрешностей, только статистическим кодером
- 3) В качестве статистического кодера использовать заданную преподавателем программу
- 4) Возможность сжатия с нулевой погрешностью обеспечивать не нужно
- 5) Размер сжимаемых изображений: 256×256

7.2.1 Общий план выполнения работы

Реализовать программу ИСИ-компрессии в виде командного файла с параметрами:

Имя_Изображения Макс_Погрешность Имя_Архива

Имена изображения и архива указываются без расширений. Программа ИСИ-компрессии сжимает изображение *Имя_Изображения.PIC* с максимальной погрешностью *Макс_Погрешность*, сжатые данные помещаются в файл *Имя_Архива*. Никаких других файлов или изображений, кроме файлов *Имя_Архива* и *params.tem*, появляться не должно (нужно удалить все промежуточные изображения командой *del*).

Для изображения *Lena.PIC* при максимальной погрешности, равной четырем, коэффициент сжатия должен быть не меньше двух.

Реализовать программу ИСИ-декомпрессии в виде командного файла с параметрами:

Имя_Архива Макс_Погрешность Имя_Изображения

Имена изображения и архива указываются без расширений. Программа ИСИ-декомпрессии распаковывает изображение *Имя_Изображения.PIC* из архива *Имя_Архива*, полученного при сжатии с погрешностью *Макс_Погрешность*. Никаких других файлов или изображений, кроме *Имя_Изображения.PIC* и *params.tem* появляться не должно (нужно удалить все промежуточные изображения командой *del*).

Построить график зависимости коэффициента сжатия от максимальной погрешности для метода ИСИ на изображении *Lena.PIC*.

Максимальная погрешность изменяется от 1 до 20.

Привести исходные данные для этого графика в виде числовой таблицы.
Рекомендация:

Для промежуточных изображений, диапазон которых превышает 0..255, следует использовать 16-битные изображения.

Преобразование 8 -битного в 16-битное: *rl clc CalcI p16=p8 BPSOut=16*,
преобразование 16-битного в 8 -битное: *rl clc CalcI p8=p16 BPSOut=8*.

7.2.2 Контрольные вопросы

1. Показатели эффективности методов сжатия изображений.
2. Показатели точности методов сжатия изображений.
3. Метод сжатия изображений на основе ИСИ.
4. Иерархическое представление изображения.
5. Интерполяция при сжатии изображений на основе ИСИ.
6. Квантование при сжатии изображений на основе ИСИ.
7. Общая характеристика программного обеспечения.

8 ИНСТРУМЕНТАЛЬНАЯ ПРОГРАММНАЯ СИСТЕМА ОБРАБОТКИ И АНАЛИЗА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

8.1 Общее описание системы

Рассматриваемая в данном разделе программная система предназначена для моделирования и исследования сложных многоэтапных и разветвленных информационных технологий формирования, преобразования и анализа изображений.

Программная система состоит из следующих подсистем:

- подсистема управления данными;
- библиотека прикладных модулей;
- подсистема запуска функций.

Подсистема управления данными обеспечивает взаимодействие прикладных модулей с входными/выходными данными. Данная подсистема поддерживает следующие типы изображений: полутоновые (серые) однокомпонентные изображения, цветные трехкомпонентные изображения, индексированные (палитровые) изображения, цветные четырехкомпонентные изображения в цветовых пространствах CMYK и YCbCr, многокомпонентные изображения (две и более компоненты). Изображения могут быть представлены в форматах TIF, BMP, PIC (внутреннем формате системы, в котором кроме 8-битных поддерживаются целочисленные 16-битные и вещественные 32-битные типы данных).

Большая часть логики обработки сосредоточена в прикладных модулях, реализующих различного рода функции обработки и анализа

изображений, например генерации случайных полей, фильтрации и т.д. Состав библиотеки прикладных модулей может быть расширен путем создания новых прикладных модулей.

Подсистема запуска функций предоставляет пользователю возможность исполнять функции обработки непосредственно из командной строки операционной системы или с помощью команды "Выполнить" меню "Пуск". При этом пользователь помещает исполняемую функцию и параметры запуска в командную строку.

8.2 Библиотека прикладных модулей

Библиотека прикладных модулей программной системы состоит из набора DLL-библиотек, каждая из которых содержит ряд функций, называемых прикладными модулями и предназначенных для решения некоторого класса задач обработки изображений.

Созданные библиотеки могут быть разделены по кругу решаемых задач на группы, приведенные в таблице 8.1.

Таблица 8.1. Библиотеки прикладных модулей

Наименование библиотеки	Назначение
Генерация тестовых изображений	
<i>Gnd.dll</i>	Генерация детерминированных полей
<i>Gnr.dll</i>	Генерация случайных полей
Обработка (в том числе фильтрация) изображений	
<i>Local3.dll</i>	Обработка скользящим окном 3×3
<i>Local.dll</i>	Рекурсивная обработка окном произвольного размера
<i>Clc.dll</i>	Позэлементные преобразования
<i>Table.dll</i>	Генерация таблиц для позэлементных преобразований
Оценка характеристик изображений	
<i>Stat.dll</i>	Оценка статистических характеристик изображений
<i>ACF.dll</i>	Оценка и анализ автоковариационной функции
Преобразования изображений	
<i>Geom.dll</i>	Геометрические преобразования
<i>Spectr.dll</i>	Спектральные преобразования
Служебные программы	
<i>Frg.dll</i>	Работа с фрагментами изображений
<i>Img.dll</i>	Работа с атрибутами и пикселями изображений
<i>Color.dll</i>	Работа с многокомпонентными изображениями
<i>Scalar.dll</i>	Работа со скалярами

8.2.1. Выполнение прикладных модулей

Каждый из прикладных модулей может быть вызван с помощью подсистемы запуска функций с использованием командной строки, задаваемой в виде:

*Exe-модуль DLLName FunctionName Param1 Param2 ...
ParamN [Name1=Value1... NameK=ValueK
[/key1=KeyValue1... /keyM=KeyValueM]]*

Командная строка состоит из

- стандартного имени вызываемого exe-модуля (например, *RL.exe*);
- имени DLL, содержащей прикладной модуль (*DLLName*) (при необходимости - с полным путем доступа, расширение *.dll* можно не указывать);
- имени вызываемого прикладного модуля (*FunctionName*);
- набора обязательных нумеруемых параметров (*Param1 Param2 ... ParamN*), задаваемых:
 - численным значением,
 - строковым значением,
 - массивом строковых или численных значений,
 - идентификатором, значение которого берется из файла параметров (будет описан ниже);
- набора необязательных (именованных) параметров, задаваемых путем присваивания идентификатору необязательного параметра строкового, численного значения, массива строковых или численных значений, или значения идентификатора из файла параметров (*Name1=Value1 Name2=Value2 ... NameK=ValueK*).
- набора необязательных ключей, задаваемых путем присваивания ключу строкового или численного значения (*/key1=KeyValue1 /key2=KeyValue2 ... /keyM=KeyValueM*)

Все параметры разделяются в командной строке пробелами (в произвольном количестве). Порядок следования обязательных параметров

строго задается при создании прикладного модуля. Необязательные параметры следуют в произвольном порядке после обязательных, причем необязательные параметры могут быть заданы не все или отсутствовать. После необязательных параметров следуют ключи, которые также задаются в произвольном порядке и могут отсутствовать.

При обращении к библиотеке без указания имени прикладного модуля и параметров:

Exe-модуль DLLName

выводится информация обо всех прикладных модулях библиотеки. Например, при обращении к библиотеке local.dll

RL.exe local

выводится следующая информация:
DLL "C:\RESLOOK\LOCAL.DLL" содержит модули:
MODULE 0 -----
Name = "Smooth" Caption = "Сглаживание прямоугольным окном"
MODULE 1 -----
Name = "Alc" Caption = "Адаптивное линейное контрастирование"

Возможно обращение к модулю без задания параметров с целью получения информации обо всех параметрах модуля:

Exe-модуль DLLName FunctionName

Например, при обращении к прикладному модулю ALC библиотеки local.dll

RL.exe local Alc

выводится следующее описание:
DLL "C:\RESLOOK\LOCAL.DLL" содержит модули:
MODULE 1 -----

Name = "Alc"		Caption = "Адаптивное линейное контрастирование"		
REQUIRED PARAMETERS				
(Name	Caption	Type	Example)	
ImgIn	Входное изображение	PT_IMAGENAME	"ImgIn"	
Height	Высота окна	PT_INTEGER	"3"	
Width	Ширина окна	PT_INTEGER	"3"	
My	Вых. локальное среднее	PT_DOUBLE	"128"	
Dy	Вых. локальная дисперсия	PT_DOUBLE	"1600"	
ImgOut	Выходное изображение	PT_IMAGENAME	"ImgOut"	
OPTIONAL PARAMETERS				
(Name	Caption	Type	Flag	Default)
C	Регуляризатор	PT_DOUBLE	PF_IN	"0.1"

Обязательные и необязательные параметры

В приведенном выше описании для каждого обязательного параметра (секция **REQUIRED PARAMETERS**) указывается:

- 1) имя параметра (Name) – краткое символьное обозначение;
- 2) описание параметра (Caption), характеризующее назначение параметра;
- 3) тип параметра (Type);
- 4) пример значения (Example).

Для каждого необязательного параметра (**OPTIONAL PARAMETERS**) указывается:

- 1) имя параметра (Name) – краткое символьное обозначение, используемое при вызове прикладного модуля;
- 2) описание параметра (Caption), характеризующее назначение параметра;
- 3) тип параметра (Type);
- 4) атрибут параметра (Flag) - входной (PF_IN), выходной (PF_OUT, входной и выходной (PF_INOUT);
- 5) значение по умолчанию (Default) – используется модулем, если необязательный параметр не задан при вызове модуля.

Синтаксис параметров, задаваемых для прикладного модуля, зависит от типа параметра и определяется следующими правилами:

1) *Целое число* (PT_INTEGER) – 32-разрядное целое число, записанное в десятичной форме.

2) *Вещественное число двойной точности* (PT_DOUBLE) – число с плавающей точкой, записанное, возможно, в экспоненциальной форме.

2) *Строковое значение* (PT_STRING) – любая последовательность символов, ограниченная при необходимости (при наличии пробелов внутри) двойными кавычками.

4) *Строковое значение, задающее имя файла (изображения)* (PT_IMAGENAME) – строковое значение, удовлетворяющее правилам операционной системы задания имен файлов, со следующими уточнениями:

- строка может не содержать расширения файла; если расширение отсутствует, то согласно списку форматов изображений (pic, tif, bmp, raw, hgi) выбирается первое расширение, соответствующее доступному файлу с изображением, или первое из списка расширений, если ни один из файлов не является доступным;

- при задании в имени файла полного пути доступа в качестве разделителей символ '\' используется дважды (например, D:\\Users\\lena.tif);
- строка может содержать номер компоненты для многокомпонентного изображения (например, Name[1])^{*)}.

5) *Массив строковых или численных значений* – набор значений, разделенных запятыми без пробелов.

При задании параметров прикладного модуля вместо непосредственных значений могут быть указаны *идентификаторы* значений, хранящихся в файле параметров. *Идентификатор* представляет собой последовательность символов длиной до 80 символов, начинающуюся с символа '\$' и не содержащую следующие символы:

пробел '=' ':' ';' '-' '+' '*' '/' '\' ','

Необязательные ключи

Необязательные ключи используются для задания параметров, являющихся общими для различных прикладных модулей. В версии 04 программной системы введены следующие ключи:

Имя ключа	Описание	Значение по умолчанию	Возможные значения
/fillMode	Режим дополнения изображения при выходе за его границы	2	0 – доопределение запрещено 1 – доопределение постоянным значением 2 – доопределение по ближайшему пикселю изображения 3 – циклическое доопределение
/fillValue	Значение пикселей изображения, используемое для доопределения при выходе за его границы	0	Произвольное число

Файл параметров

Для обмена информацией (параметрами алгоритмов обработки, численными характеристиками изображений и т.д.) между прикладными модулями используется файл параметров. Файл параметров (по умолчанию файл params.mem) является текстовым, т.е. может создаваться и

^{*)} В версии 04 программной системы не реализовано

редактироваться в любом текстовом редакторе. Файл параметров должен удовлетворять следующим требованиям.

1) каждая строка файла задает значение только одного параметра в виде
`parameter = value; comment,`

где

`parameter` – идентификатор параметра,

`value` – численное, строковое значение или массив численных или строковых значений параметра,

; `comment` – комментарий – строковое значение, начинающееся с символа ';';

2) пробелы в строке, задающей параметр (перед и после идентификатора параметра, перед и после значения параметра), игнорируются;

3) в файле могут присутствовать пустые строки либо строки, не отвечающие вышеуказанному правилу; такие строки при поиске параметра в файле игнорируются.

Следует отметить, что реализация простых сценариев обработки изображений, применяемых при выполнении лабораторных работ, возможна с использованием пакетных файлов (.bat файлов). Например, пакетный файл сценария с внесением гауссовского шума в изображение *Lena* запишется в следующем виде:

```
RL Img GetAttr Lena VerSize=$VerSize HorSize=$HorSize
RL Gnr GnGauss $VerSize $HorSize 128 100 Noise BPS=8
RL Clc Calc1 LenaRes=Lena+Noise-128 BPSOut=8
```

В следующих разделах описываются состав библиотеки прикладных модулей, назначение каждого модуля и его параметры.

8.2.2 Генерация детерминированных полей (Gnd.dll)

Библиотека генерации детерминированных полей включает в себя прикладные модули:

- *GnConst* – генерация изображения-константы;
- *GnChess* – генерация шахматного поля;
- *GnWedge* – генерация «оптического клина».

Генерация изображения-константы

Модуль формирует изображение с заданным постоянным значением отсчетов. Обращение к программе:

RL Gnd GnConst VerSize HorSize Const ImgOut
[BPS=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>Const</i>	Константа	PT_DOUBLE	128
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	ConstImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Бит на отсчет (8, 16, 32)	PT_INTEGER	IN	8

Генерация шахматного поля

Модуль формирует шахматное поле с произвольным количеством, размером и яркостью клеток. Обращение к программе:

RL Gnd GnChess VerSize HorSize Height Width White
Black ImgOut [BPS=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>Height</i>	Высота клетки	PT_INTEGER	20
<i>Width</i>	Ширина клетки	PT_INTEGER	20
<i>White</i>	Яркость светлых клеток	PT_DOUBLE	200
<i>Black</i>	Яркость темных клеток	PT_DOUBLE	100
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	ChessImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Знач. по умолчанию
<i>BPS</i>	Бит на отсчет	PT_INTEGER	IN	8

Генерация «оптического клина»

Модуль формирует байтовое изображение «оптического клина» в диапазоне яркости 0÷255. Обращение к программе:

RL Gnd GnWedge VerSize HorSize ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	WedgeImg

8.2.3. Генерация случайных полей (*Gnr.dll*)

Библиотека генерации случайных полей включает в себя прикладные модули:

- *GnGauss* – генерация нормального "белого" шума;
- *GnEven* – генерация равномерно распределенного "белого" шума;
- *GnIGauss* – генерация нормального импульсного шума;
- *GnIEven* – генерация равномерно распределенного импульсного шума;
- *GnBiExp* – генерация поля с биэкспоненциальной автоковариационной функцией;
- *GnCovar* – генерация поля с изотропной или "эллипсоидальной" АКФ;
- *GnMos_d* – генерация мозаичного поля с детерминированным разбиением.

Генерация нормального "белого" шума

Модуль формирует изображение, отсчеты которого являются независимыми случайными величинами с нормальным (гауссовским) распределением, которое, в свою очередь, задается средним значением и дисперсией. Обращение к программе:

RL Gnr GnGauss VerSize HorSize E D ImgOut
[BPS=<значение> InitValue=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>E</i>	Среднее	PT_DOUBLE	128
<i>D</i>	Дисперсия	PT_DOUBLE	1600
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	GaussImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Бит на отсчет	PT_INTEGER	IN	32
<i>InitValue</i>	Начальное значение датчика случайных чисел	PT_INTEGER	IN	12345

Генерация равномерно распределенного "белого" шума

Модуль формирует изображение, отсчеты которого являются независимыми случайными величинами с равномерным распределением, которое, в свою очередь, задается максимальным и минимальным значением яркости. Обращение к программе:

RL Gnr GnEven VerSize HorSize Min Max ImgOut
[BPS=<значение> InitValue=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT INTEGER	256
<i>Min</i>	Минимальная яркость	PT DOUBLE	128
<i>Max</i>	Максимальная яркость	PT DOUBLE	1600
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	EvenImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Бит на отсчет	PT INTEGER	IN	32
<i>InitValue</i>	Начальное значение датчика случайных чисел	PT_INTEGER	IN	12345

Генерация нормального импульсного шума

Модуль формирует изображение случайно и независимо друг от друга расположенных импульсов, каждый из которых принимает независимое случайное значение в соответствии с нормальным (гауссовским) распределением, которое задается средним значением и дисперсией. Отсчеты, не являющиеся импульсами, имеют нулевое значение. Обращение к программе:

RL Gnr GnIGauss VerSize HorSize E D Prob ImgOut
[BPS=<значение> InitValue=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT INTEGER	256
<i>E</i>	Среднее	PT DOUBLE	128
<i>D</i>	Дисперсия	PT DOUBLE	1600
<i>Prob</i>	Вероятность появления шума	PT DOUBLE	0,2
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	IGaussImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолч.
<i>BPS</i>	Бит на отсчет	PT INTEGER	IN	32
<i>InitValue</i>	Нач. знач. датчика случ. чисел	PT_INTEGER	IN	12345

Генерация равномерно распределенного импульсного шума

Модуль формирует изображение случайно и независимо друг от друга расположенных импульсов, каждый из которых принимает случайное значение в соответствии с равномерным распределением, которое задается максимальным и минимальным значениями яркости. Отсчеты, не являющиеся импульсами, имеют нулевое значение. Обращение к программе:

RL Gnr GnIEven VerSize HorSize Min Max ImgOut
[BPS=<значение> InitValue=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>Min</i>	Минимальная яркость	PT_DOUBLE	0
<i>Max</i>	Максимальная яркость	PT_DOUBLE	255
<i>Prob</i>	Вероятность появления шума	PT_DOUBLE	0,2
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	IEvenImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Бит на отсчет	PT_INTEGER	IN	32
<i>InitValue</i>	Начальное значение датчика случайных чисел	PT_INTEGER	IN	12345

Генерация поля с биэкспоненциальной автоковариационной функцией

Модуль формирует прямоугольный фрагмент псевдослучайного стационарного гауссовского поля с заданным средним (математическим ожиданием) и ковариационной функцией вида

$$\text{Covar}(k, l) = D \cdot RV^{|k|} RH^{|l|}.$$

Формат вызова:

RL Gnr GnBiexp VerSise HorSize E D RV RH ImgOut
[BPS=<значение> InitValue=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>E</i>	Среднее	PT_DOUBLE	128
<i>D</i>	Дисперсия	PT_DOUBLE	1600
<i>RV</i>	Корреляция по вертикали	PT_DOUBLE	0,9
<i>RH</i>	Корреляция по горизонтали	PT_DOUBLE	0,9
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	BiExplmg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
BPS	Бит на отсчет	PT_INTEGER	IN	32
InitValue	Начальное значение датчика случайных чисел	PT_INTEGER	IN	12345

Генерация поля с изотропной или "эллипсоидальной" АКФ

Модуль формирует прямоугольный фрагмент псевдослучайного стационарного гауссовского поля с заданным средним (математическим ожиданием) и ковариационной функцией с эллипсоидальными линиями равных значений. В случае изотропного поля (при $R_{Max} = R_{Min} = R$) АКФ первого типа имеет вид:

$$\text{Covar}(k, l) = D \cdot R^{\sqrt{k^2 + l^2}},$$

а АКФ второго типа имеет вид:

$$\text{Covar}(k, l) = D \cdot R^{k^2 + l^2},$$

Если поле неизотропно, коэффициент корреляции меняется в заданных пределах. Формат вызова:

```
RL Gnr GnCovar VerSise HorSize E D ACFTType RMax
RMin FiRMax ImgOut [FreeOpMem=<значение>
BPS=<значение> InitValue=<значение>]
```

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>E</i>	Среднее	PT_DOUBLE	128
<i>D</i>	Дисперсия	PT_DOUBLE	1600
<i>ACFType</i>	Тип АКФ	PT_INTEGER	1
<i>RMax</i>	Максимальная корреляция	PT_DOUBLE	0,99
<i>RMin</i>	Минимальная корреляция	PT_DOUBLE	0,95
<i>FiRMax</i>	Угол на направление максимальной корреляции (в градусах)	PT_DOUBLE	0
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	CovarImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб)	PT_INTEGER	IN	10
<i>BPS</i>	Бит на отсчет	PT_INTEGER	IN	32
<i>InitValue</i>	Начальное значение датчика случайных чисел	PT_INTEGER	IN	12345

В зависимости от значения параметра *FreeOpMem* и размеров изображения выбираются различные алгоритмы обработки (в оперативной памяти или с созданием временных файлов на жестком диске – для крупноразмерных изображений), существенно влияющие на скорость работы модуля.

Генерация мозаичного поля с детерминированным разбиением

Модуль формирует прямоугольный фрагмент кусочно-постоянного поля с областями, образованными четырьмя семействами прямых параллельных линий, отстоящих друг от друга на заданное расстояние U и ориентированных под углами 0, 45, 90 и 135 градусов к строкам. Изображение в областях постоянной яркости "раскрашивается" независимыми гауссовскими случайными числами с математическим ожиданием E и дисперсией D . Синтезированное изображение является в целом квазистационарным случайным полем, имеющим гауссовское одномерное распределение отсчетов с заданным средним и дисперсией и

автокорреляционную функцию, близкую к изотропной экспоненциальной:

$$\text{Covar}(k, l) = D \cdot R^{\sqrt{k^2 + l^2}}.$$

Коэффициент корреляции между соседними отсчетами поля приблизительно равен:

$$R = \text{Exp}\left\{-2,414/U\right\}.$$

Ниже приведены некоторые конкретные значения:

R	U
0,99	240
0,95	47
0,90	23
0,80	11

Обращение к программе:

RL Gnr GnMos_d VerSise HorSize E D U ImgOut
[BPS=<значение> InitValue=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>VerSize</i>	Размер по вертикали	PT INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT INTEGER	256
<i>E</i>	Среднее	PT DOUBLE	128
<i>D</i>	Дисперсия	PT DOUBLE	1600
<i>U</i>	Расстояние между линиями	PT INTEGER	40
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	MosImg

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Бит на отсчет	PT INTEGER	IN	32
<i>InitValue</i>	Начальное значение датчика случайных чисел	PT_INTEGER	IN	12345

8.2.4 Обработка скользящим окном 3×3 (Local3.dll)

Библиотека обработки скользящим окном 3×3 включает в себя прикладные модули:

- *LinFil3* – линейная обработка окном 3×3;
- *RankFil3* – взвешенная ранговая фильтрация окном 3×3;
- *GradFil3* – градиентное выделение контуров окном 3×3.

Линейная обработка окном 3×3

Модуль осуществляет преобразование вида:

$$Y(i, j) = (X(i-1, j-1) \cdot w11 + X(i-1, j) \cdot w12 + X(i-1, j+1) \cdot w13 + \\ + X(i, j-1) \cdot w21 + X(i, j) \cdot w22 + X(i, j+1) \cdot w23 + \\ + X(i+1, j-1) \cdot w31 + X(i+1, j) \cdot w32 + X(i+1, j+1) \cdot w33) \times \\ \times \frac{MulConst}{DivConst} + AddConst,$$

где $X(i, j)$, $Y(i, j)$ - отсчеты входного и выходного изображений, $w11...w33$, $MulConst$, $DivConst$, $AddConst$ – задаваемые параметры. Обращение к программе:

RL Local 3 LinFil3 *ImgIn w11 w12 w13 w21 w22 w23*
w31 w32 w33 ImgOut [MulConst=<значение>
DivConst=<значение> AddConst=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример знач.
<i>ImgIn</i>	Входное изображение	PT IMAGENAME	<i>ImgIn</i>
<i>w11</i>	Весовой коэффициент (1,1) окна 3×3	PT DOUBLE	1
<i>w12</i>	Весовой коэффициент (1,2) окна 3×3	PT DOUBLE	1
<i>w13</i>	Весовой коэффициент (1,3) окна 3×3	PT DOUBLE	1
<i>w21</i>	Весовой коэффициент (2,1) окна 3×3	PT DOUBLE	1
<i>w22</i>	Весовой коэффициент (2,2) окна 3×3	PT DOUBLE	1
<i>w23</i>	Весовой коэффициент (2,3) окна 3×3	PT DOUBLE	1
<i>w31</i>	Весовой коэффициент (3,1) окна 3×3	PT DOUBLE	1
<i>w32</i>	Весовой коэффициент (3,2) окна 3×3	PT DOUBLE	1
<i>w33</i>	Весовой коэффициент (3,3) окна 3×3	PT DOUBLE	1
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	<i>ImgOut</i>

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>AddConst</i>	Аддитивная константа	PT_DOUBLE	IN	0
<i>MulConst</i>	Мультипликативная константа	PT_DOUBLE	IN	1
<i>DivConst</i>	Делитель	PT_DOUBLE	IN	1

Взвешенная ранговая фильтрация окном 3×3

Значение отсчета $X(i,j)$ выходного изображения определяется следующим образом. В его окрестности размером 3×3 по отсчетам входного изображения строится вариационный ряд, то есть отсчеты упорядочиваются в порядке неубывания значений, причем отсчет

$X(i-1, j-1)$ берется $w11$ раз,
 $X(i-1, j)$ берется $w12$ раз,
 $X(i-1, j+1)$ берется $w13$ раз,
 $X(i, j-1)$ берется $w21$ раз,
 $X(i, j)$ берется $w22$ раз,
 $X(i, j+1)$ берется $w23$ раз,
 $X(i+1, j-1)$ берется $w31$ раз,
 $X(i+1, j)$ берется $w32$ раз,
 $X(i+1, j+1)$ берется $w33$ раз.

В качестве выходного отсчета берется элемент вариационного ряда с номером *NumRank* (нумерация начинается с нуля). Отметим, что при маске вида

0 1 0
 1 1 1
 0 1 0

и *NumRank*=2 получаем медианный фильтр с окном в 5 отсчетов, а при маске вида

1 1 1
 1 1 1
 1 1 1

и *NumRank*=4 получаем медианный фильтр с окном в 9 отсчетов. Обращение к программе:

RL Local 3 RnkFil3 *ImgIn w11 w12 w13 w21 w22 w23*
w31 w32 w33 NumRank ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>w11</i>	Весовой коэффициент (1,1) окна 3×3	PT_INTEGER	1
<i>w12</i>	Весовой коэффициент (1,2) окна 3×3	PT_INTEGER	1
<i>w13</i>	Весовой коэффициент (1,3) окна 3×3	PT_INTEGER	1
<i>w21</i>	Весовой коэффициент (2,1) окна 3×3	PT_INTEGER	1
<i>w22</i>	Весовой коэффициент (2,2) окна 3×3	PT_INTEGER	1
<i>w23</i>	Весовой коэффициент (2,3) окна 3×3	PT_INTEGER	1
<i>w31</i>	Весовой коэффициент (3,1) окна 3×3	PT_INTEGER	1
<i>w32</i>	Весовой коэффициент (3,2) окна 3×3	PT_INTEGER	1
<i>w33</i>	Весовой коэффициент (3,3) окна 3×3	PT_INTEGER	1
<i>NumRank</i>	Ранг выходного значения	PT_INTEGER	4
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Градиентное выделение контуров окном 3×3

Прикладной модуль осуществляет приближенное вычисление модуля градиента изображения по формуле:

$$Y(i, j) = \sqrt{A^2 + B^2},$$

где

$$A = (X(i-1, j-1) \cdot w11 + X(i-1, j) \cdot w12 + X(i-1, j+1) \cdot w13 + \\ + X(i, j-1) \cdot w21 + X(i, j) \cdot w22 + X(i, j+1) \cdot w23 + \\ + X(i+1, j-1) \cdot w31 + X(i+1, j) \cdot w32 + X(i+1, j+1) \cdot w33),$$

$$B = (X(i-1, j-1) \cdot w11 + X(i-1, j) \cdot w21 + X(i-1, j+1) \cdot w31 + \\ + X(i, j-1) \cdot w12 + X(i, j) \cdot w22 + X(i, j+1) \cdot w32 + \\ + X(i+1, j-1) \cdot w13 + X(i+1, j) \cdot w23 + X(i+1, j+1) \cdot w33),$$

где $X(i, j)$, $Y(i, j)$ – отсчеты входного и выходного изображений.

Обращение к программе:

*RL Local 3 Gradi3 ImgIn w11 w12 w13 w21 w22 w23
w31 w32 w33 ImgOut*

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGENAME	<i>ImgIn</i>
<i>w11</i>	Весовой коэффициент (1,1) окна 3×3	PT_DOUBLE	1
<i>w12</i>	Весовой коэффициент (1,2) окна 3×3	PT_DOUBLE	1
<i>w13</i>	Весовой коэффициент (1,3) окна 3×3	PT_DOUBLE	1
<i>w21</i>	Весовой коэффициент (2,1) окна 3×3	PT_DOUBLE	1
<i>w22</i>	Весовой коэффициент (2,2) окна 3×3	PT_DOUBLE	1
<i>w23</i>	Весовой коэффициент (2,3) окна 3×3	PT_DOUBLE	1
<i>w31</i>	Весовой коэффициент (3,1) окна 3×3	PT_DOUBLE	1
<i>w32</i>	Весовой коэффициент (3,2) окна 3×3	PT_DOUBLE	1
<i>w33</i>	Весовой коэффициент (3,3) окна 3×3	PT_DOUBLE	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	<i>ImgOut</i>

8.2.5 Рекурсивная обработка окном произвольного размера (*Local.dll*)

Библиотека рекурсивной обработки окном произвольного размера включает в себя прикладные модули:

- *Smooth* – сглаживание прямоугольным окном;
- *Alc* – адаптивное линейное контрастирование;
- *RankFil* – простой ранговый фильтр;
- *ExtrFil* – экстремальный ранговый фильтр;
- *PeakFil* – пиковый ранговый фильтр;
- *SmkFil* – сглаживание по заданному числу ближайших соседей;
- *SigmFil* – сигма-фильтр.

Сглаживание прямоугольным окном

Прикладной модуль формирует значение каждого отсчета выходного изображения как среднее значение по прямоугольному окну заданного

размера с центром в текущей точке.

Обращение к программе:

RL Local Smooth ImgIn Height Width ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT_INTEGER	3
<i>Width</i>	Ширина окна	PT_INTEGER	3
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Адаптивное линейное контрастирование

Прикладной модуль формирует значение каждого отсчета выходного изображения по формуле:

$$Y(m,n) = \sqrt{\frac{Dy}{Dx(m,n) + C}} (X(m,n) - Mx(m,n)) + My,$$

где $Mx(m,n)$, $Dx(m,n)$ - оценки среднего и дисперсии по прямоугольному симметрично расположенному окну размером $Height \times Width$. Обращение к программе:

RL Local Alc ImgIn Height Width My Dy ImgOut
[C=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT_INTEGER	3
<i>Width</i>	Ширина окна	PT_INTEGER	3
<i>My</i>	Выходное локальное среднее	PT_DOUBLE	128
<i>Dy</i>	Выходная локальная дисперсия	PT_DOUBLE	1600
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>C</i>	Регуляризатор	PT_DOUBLE	IN	0.1

Простой ранговый фильтр

Прикладной модуль формирует значение каждого отсчета выходного изображения как порядковой статистики с номером *NumRank* вариационного ряда, построенного по прямоугольному симметрично расположенному окну размером *Height*×*Width*. Под вариационным рядом понимается последовательность отсчетов, входящих в окно, упорядоченная по неубыванию их значений.

Отметим несколько наиболее часто используемых случаев:

- при $NumRank = \frac{Height \cdot Width - 1}{2}$ модуль реализует медианный фильтр;
- при $NumRank = 0$ - операцию эрозии изображения (вычисления минимального значения по окну);
- при $NumRank = Height \cdot Width - 1$ - операцию дилатации изображения (вычисления максимального значения по окну).

Обращение к программе:

RL Local RnkFil ImgIn Height Width NumRank ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGE	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT INTEGER	3
<i>Width</i>	Ширина окна	PT INTEGER	3
<i>NumRank</i>	Ранг выходного значения	PT INTEGER	4
<i>ImgOut</i>	Выходное изображение	PT IMAGE	<i>ImgOut</i>

Экстремальный ранговый фильтр

Прикладной модуль формирует значение каждого отсчета выходного изображения по формуле:

$$Y = \begin{cases} V(0) & \text{при } X - V(0) < V(Height \cdot Width - 1) - X; \\ V(Height \cdot Width - 1) & \text{иначе,} \end{cases}$$

где X - соответствующий отсчет входного изображения, $V(\cdot)$ - вариационный ряд, построенного по прямоугольному симметрично расположенному окну размером *Height*×*Width*. Обращение к программе:

RL Local ExtrFil ImgIn Height Width ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGENAME	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT INTEGER	3
<i>Width</i>	Ширина окна	PT INTEGER	3
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	<i>ImgOut</i>

Пиковый ранговый фильтр

Прикладной модуль формирует значение каждого отсчета выходного изображения по формуле:

$$Y = \begin{cases} V(Height \cdot Width - 1) & \text{при } X = V(Height \cdot Width - 1); \\ 0 & \text{иначе,} \end{cases}$$

где X - соответствующий отсчет входного изображения, $V(\cdot)$ - вариационный ряд, построенного по прямоугольному симметрично расположенному окну размером $Height \times Width$. Этот фильтр позволяет выделять локальные максимумы на изображении.

Обращение к программе:

RL Local PeakFil ImgIn Height Width ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGENAME	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT INTEGER	3
<i>Width</i>	Ширина окна	PT INTEGER	3
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	<i>ImgOut</i>

Сглаживание по заданному числу ближайших соседей

Прикладной модуль формирует значение каждого отсчета как среднее по K отсчетам в окне, ближайшим по яркости к входному отсчету, стоящему в центре окна. Отсчеты, по которым производится усреднение, являются "ближайшими соседями" входного отсчета в вариационном ряду. Обращение к программе:

RL Local SmkFil ImgIn Height Width K ImgOut

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGE	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT INTEGER	3
<i>Width</i>	Ширина окна	PT INTEGER	3
<i>K</i>	Количество соседних отсчетов	PT INTEGER	5
<i>ImgOut</i>	Выходное изображение	PT IMAGE	<i>ImgOut</i>

Сигма-фильтр

Прикладной модуль формирует значение каждого отсчета как среднее по отсчетам в окне, отличающимся по яркости от входного отсчета, стоящего в центре окна, не более чем на *Sigma*. Обращение к программе:

RL Local SigmFil ImgIn Height Width K ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGE	<i>ImgIn</i>
<i>Height</i>	Высота окна	PT INTEGER	3
<i>Width</i>	Ширина окна	PT INTEGER	3
<i>Sigma</i>	Параметр фильтрации	PT DOUBLE	10
<i>ImgOut</i>	Выходное изображение	PT IMAGE	<i>ImgOut</i>

8.2.6 Поэлементные преобразования (*Clc.dll*)

Библиотека поэлементных преобразований отличается от остальных библиотек нестандартным интерфейсом, так как все прикладные модули, входящие в нее позволяют пользователю задавать произвольное аналитическое выражение для обработки отсчетов изображения, вычисления значения переменной или генерации тестовых одно- и двумерных сигналов.

Библиотека позволяет использовать при записи аналитического выражения:

- математические операции: + - сложение, - - вычитание, * умножение, / - деление, ^ - возведение в степень;
- функции: sin – синус (аргумент считается заданным в радианах), cos – косинус, tan – тангенс, atan – арктангенс (результат выражается в

радианах), sqrt – квадратный корень, $\log$ – натуральный логарифм, exp – экспонента, $\text{rand}(x)$ – генерация случайной величины в диапазоне $0 \div 32767$ ("x" - начальное значение);

– вспомогательные функции:

$$\begin{aligned} \text{bin}(x) &= \begin{cases} 0, & x \leq 0 \\ 1, & x > 0, \end{cases} \\ \text{sign}(x) &= \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0, \end{cases} \\ \text{byte}(x) &= \begin{cases} 0, & x \leq 0 \\ [x], & 0 < x < 255 \\ 255, & x \geq 255, \end{cases} \end{aligned}$$

где $[x]$ – целая часть числа x . Кроме того, в качестве имени функции может выступать имя файла – таблицы, созданного при помощи модулей библиотеки *Table*, описанной ниже. В качестве аргументов функций могут выступать имена файлов, переменные, записанные в файл параметров. Пробелы в аналитическом выражении недопустимы. Пример аналитического выражения:

$$a = \sin(b) * \text{Tabl}(x) - \text{sqrt}(\text{exp}(b^2 + x^2))$$

Библиотека включает в себя следующие прикладные модули:

- *CalcI* – поэлементное преобразование изображений;
- *CalcV* – вычисление значения переменной;
- *CalcF1* – генерация одномерного сигнала, заданного аналитически;
- *CalcF2* – генерация двумерного сигнала (изображения), заданного аналитически.

Поэлементное преобразование изображений

Модуль формирует новое изображение из входных изображений и переменных в соответствии с заданным аналитическим выражением. Обращение к программе:

RL Clc CalcI Expression [BPSOut=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>Expression</i>	Алгебраическое выражение	PT STRING	ImgOut=2*Img1

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPSOut</i>	Количество бит на отсчет для выходного изображения	PT_INTEGER	IN	0

Вычисление значения переменной

Модуль формирует значение в соответствии с заданным аналитическим выражением. Обращение к программе:

RL Clc CalcV Expression

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>Expression</i>	Алгебраическое выражение	PT STRING	a=\$b+\$c

Генерация одномерного сигнала, заданного аналитически

Модуль формирует одномерный сигнал в форме изображения, состоящего из одной строки в соответствии с заданным аналитическим выражением. Значения пикселей вычисляются для каждого значения аргумента x , начиная со значения x_0 с шагом $Step$. Обращение к программе:

RL Clc CalcF1 ImgOut HorSize x0 Step Expression(x) [BPS=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgOut</i>	Выходное изображение	PT IMAGENAME	ImgOut
<i>HorSize</i>	Размер по горизонтали	PT INTEGER	256
<i>x0</i>	Начальное значение переменной	PT_DOUBLE	0
<i>Step</i>	Шаг переменной	PT_DOUBLE	1
<i>Expression(x)</i>	Аналитическое выражение от одной переменной x	PT_STRING	$x+x^2$

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Количество бит на отсчет	PT_INTEGER	IN	32

Генерация двумерного сигнала (изображения), заданного аналитически

Модуль формирует двумерный сигнал (изображение) в соответствии с заданным аналитическим выражением. Значения пикселей вычисляются для каждого значения аргументов x , y , начиная со значений x_0 , y_0 с шагами $StepX$, $StepY$ по каждой координате. Обращение к программе:

*RL Clc CalcF2 ImgOut VerSize HorSize x0 y0 StepX StepY
Expression(x,y) [BPS=<значение>]*

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	ImgOut
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	256
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	256
<i>x0</i>	Начальное значение переменной по горизонтали	PT_DOUBLE	0
<i>y0</i>	Начальное значение переменной по вертикали	PT_DOUBLE	0
<i>StepX</i>	Шаг переменной по горизонтали	PT_DOUBLE	1
<i>StepY</i>	Шаг переменной по вертикали	PT_DOUBLE	1
<i>Expression (x,y)</i>	Аналитическое выражение от двух переменных x и y	PT_STRING	$xy+x^2-y^3$

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>BPS</i>	Количество бит на отсчет	PT_INTEGER	IN	32

8.2.7 Генерация таблиц для поэлементных преобразований (Table.dll)

Библиотека генерации таблиц для поэлементных преобразований включает в себя прикладные модули:

- *TablePL* – генерация таблицы кусочно-линейного преобразования;
- *TableEq* – генерация таблицы для эквализации изображения.

Генерация таблицы кусочно-линейного преобразования

Модуль формирует файл с расширением .TAB, содержащий таблицу кусочно-линейного преобразования для последующего использования библиотекой *Clc*. Таблица строится по заданным узловым точкам. Обращение к программе:

RL Table TablePL ArrayX_Y TabFile

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>X_Y</i>	Массив, содержащий узловые точки кусочно-линейного преобразования в виде $x_1, y_1, x_2, y_2, \dots$	PT_ARRAY	20,0,200,255
<i>FileTab</i>	Имя файла, в который будет помещена таблица	PT_STRING	FileTab

Генерация таблицы для эквализации изображения

Модуль формирует файл с расширением .TAB, содержащий таблицу поэлементного преобразования, обеспечивающего выравнивание функции распределения вероятности отсчетов изображения на интервале [0, 255]. Файл *FileHst* должен иметь расширение .HST и содержать 256 неотрицательных чисел в формате с плавающей точкой (4 байта) - гистограмму изображения, построенную для интервала [0,255] (см. библиотеку *Stat*). Обращение к программе:

RL Table TableEq FileHst TabFile

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>FileHst</i>	Имя файла, содержащего гистограмму изображения	PT_STRING	FileHst
<i>FileTab</i>	Имя файла, в который будет помещена таблица	PT_STRING	FileTab

8.2.8 Оценка статистических характеристик изображений (Stat.dll)

Библиотека оценки статистических характеристик изображений включает в себя прикладные модули:

- *MaxMin* – определение экстремумов изображения и их координат;
- *Stat* – оценка среднего, дисперсии и коэффициента корреляции;
- *Hst* – построение гистограммы;
- *Hist* – построение интервального статистического ряда отсчетов заданного диапазона;
- *StatHst* – оценка статистических моментов по гистограмме;
- *Quantil* – оценка квантиля по гистограмме.

Оценка среднего, дисперсии и коэффициента корреляции

Модуль находит среднее, дисперсию и коэффициент корреляции между отсчетами изображения. Для цветных (многокомпонентных) изображений эти характеристики определяются по каждой цветовой компоненте. Обращение к программе:

```
RL Stat Stat ImgIn [E=имя переменной> D=<имя
переменной>
R=<имя переменной> RH=<имя переменной> RV=<имя
переменной>]
```

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGE	ImgIn

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>E</i>	Среднее	PT ARRAY	OUT	
<i>D</i>	Дисперсия	PT ARRAY	OUT	
<i>R</i>	Корреляция	PT ARRAY	OUT	
<i>RH</i>	Корреляция по горизонтали	PT ARRAY	OUT	
<i>RV</i>	Корреляция по вертикали	PT ARRAY	OUT	

Определение экстремумов изображения и их координат

Модуль находит значение и координаты глобальных экстремумов по изображению. Для цветных (многокомпонентных) изображений определяются экстремумы по каждой цветовой компоненте. Обращение к программе:

```
RL Stat MaxMin ImgIn [Key=<значение >
Max=<имя переменной> Min=<имя переменной>
Vmax=<имя переменной> Hmax=<имя переменной>
Vmin=<имя переменной> Hmin=<имя переменной>]
```

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE	ImgIn

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Key</i>	Режим (0/1) поиска первого/последнего экстремума на изображении	PT_INTEGER	IN	1
<i>Max</i>	Максимум	PT_ARRAY	OUT	
<i>Min</i>	Минимум	PT_ARRAY	OUT	
<i>Vmax</i>	Вертикальная координата максимума	PT_ARRAY	OUT	
<i>Hmax</i>	Горизонтальная координата максимума	PT_ARRAY	OUT	
<i>Vmin</i>	Вертикальная координата минимума	PT_ARRAY	OUT	
<i>Hmin</i>	Горизонтальная координата минимума	PT_ARRAY	OUT	

Построение гистограммы

Модуль обрабатывает только байтовые изображения, для них строится гистограмма в диапазоне значений пикселей 0÷255 и сохраняется в файл с расширением .HST, который потом используется модулями *StatHst*, *Quantil*, *TableEq*. Обращение к программе:

```
RL Stat Hst ImgIn HstFile
```

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>FileHist</i>	Имя файла, в который будет помещена гистограмма изображения	PT_STRING	<i>FileHist</i>

Построение интервального статистического ряда отсчетов заданного диапазона

Модуль формирует изображение, состоящее из одной строки длиной $\frac{Max-Min}{Delta} + 2$. Значения отсчетов выходного изображения равны числу отсчетов входного изображения, попадающих в соответствующий интервал задаваемого диапазона значений входного изображения. Обращение к программе:

RL Stat Hist ImgIn Max Min Delta ImgHist
[TypeHist=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>Max</i>	Максимум	PT_DOUBLE	255
<i>Min</i>	Минимум	PT_DOUBLE	0
<i>Delta</i>	Шаг	PT_DOUBLE	1
<i>ImgHist</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgHist</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>TypeHist</i>	Тип (0 – гистограмма, 1 – интегральная функция распределения)	PT_INTEGER	IN	0

Оценка статистических моментов по гистограмме

Модуль оценивает среднее и дисперсию по файлу гистограммы

изображения. Обращение к программе:

RL Stat StatHst FileHst [E=<имя переменной> D=<имя переменной>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>FileHst</i>	Имя файла, содержащего гистограмму изображения	PT_STRING	FileHst

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>E</i>	Среднее	PT_DOUBLE	OUT	
<i>D</i>	Дисперсия	PT_DOUBLE	OUT	

Оценка квантиля по гистограмме

Модуль вычисляет величину *Value*, для которой неравенство $X < Value$, где X - случайная величина выполняется с вероятностью P . Распределение случайной величины задано файлом гистограммы. Обращение к программе:

RL Stat Quantil FileHst P [Value=<имя переменной>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>FileHst</i>	Имя файла, содержащего гистограмму изображения	PT_STRING	FileHst
<i>P</i>	Вероятность	PT_DOUBLE	0,99

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Value</i>	Среднее	PT_DOUBLE	OUT	

8.2.9 Оценка и анализ автоковариационной функции (ACF.dll)

Библиотека оценки и анализа автоковариационной функции включает в себя прикладные модули:

- *EstCovar* – оценка автоковариационной функции изображения;
- *ApCovarI* – изотропная аппроксимация автоковариационной функции;
- *ApCovarN* – неизотропная ("эллипсоидальная") аппроксимация АКФ;
- *EstPSF* – идентификация импульсной характеристики линейной системы.

Оценка автоковариационной функции изображения

Модуль формирует изображение, представляющее собой прямоугольный фрагмент поля оценки автоковариационной функции входного изображения. Выходное изображение имеет размеры *Height*×*Width* и вещественный тип данных (32 бит на пиксел). Обращение к программе:

RL ACF EstCovar ImgIn VerPos HorPos Height Width
ImgCovar [FreeOpMem=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGENAME	<i>ImgIn</i>
<i>VerPos</i>	Позиция по вертикали	PT INTEGER	0
<i>HorPos</i>	Позиция по горизонтали	PT INTEGER	0
<i>Height</i>	Высота	PT INTEGER	33
<i>Width</i>	Ширина	PT INTEGER	33
<i>ImgCovar</i>	Изображение - оценка АКФ	PT IMAGENAME	<i>ImgCovar</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

Изотропная аппроксимация автоковариационной функции

Модуль производит аппроксимацию массива значений

^{*)} См. замечание к модулю *GnCovar* библиотеки *Gnr.dll*

автоковариационной функции двумерного поля радиально симметричными функциями двух видов. При $TypeCov=1$

$$Covar(k,l) = DS \cdot R^{\sqrt{k^2+l^2}} + DN \cdot \delta(k,l),$$

а при $TypeCov=2$:

$$Covar(k,l) = DS \cdot R^{k^2+l^2} + DN \cdot \delta(k,l),$$

где $\delta(k,l)$ - двумерная дельта-функция Кронекера:

$$\delta(k,l) = \begin{cases} 1, & \text{при } k=l=0 \\ 0, & \text{иначе.} \end{cases}$$

Обращение к программе:

*RL ACF ApCovarI ImgCovar TypeCov [DS=<имя
переменной> DN=<имя переменной>
R=<имя переменной> Eps=<имя переменной>]*

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgCovar</i>	Изображение - оценка АКФ	PT IMAGE	<i>ImgCovar</i>
<i>TypeCov</i>	Тип АКФ	PT INTEGER	1

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>DS</i>	Дисперсия сигнала	PT DOUBLE	OUT	
<i>DN</i>	Дисперсия шума	PT DOUBLE	OUT	
<i>R</i>	Корреляция	PT DOUBLE	OUT	
<i>Eps</i>	Погрешность аппроксимации	PT DOUBLE	OUT	

Неизотропная ("эллипсоидальная") аппроксимация АКФ

Модуль производит аппроксимацию массива значений автоковариационной функции двумерного поля центрально-симметричными функциями с эллипсоидальными линиями равных значений двух видов. При $TypeCov=1$

$$\text{Covar}(k, l) = DS \cdot Rmax \sqrt{A^2 k_0^2 + l_0^2} + DN \cdot \delta(k, l),$$

а при $TypeCov=2$:

$$\text{Covar}(k, l) = DS \cdot Rmax^{A^2 k_0^2 + l_0^2} + DN \cdot \delta(k, l),$$

где $A = \frac{\ln Rmin}{\ln Rmax}$;

$k_0 = k \cdot \cos(FiRMax) + l \cdot \sin(FiRMax)$;

$l_0 = k \cdot \sin(FiRMax) + l \cdot \cos(FiRMax)$; $\delta(k, l)$ - двумерная дельта-функция

Кroneckera:

$$\delta(k, l) = \begin{cases} 1, & \text{при } k = l = 0 \\ 0, & \text{иначе.} \end{cases}$$

Обращение к программе:

*RL ACF ApCovarN ImgCovar TypeCov [DS=<имя
переменной> DN=<имя переменной>
Rmax=<имя переменной> Rmin=<имя переменной>
FiRMax=<имя переменной> Eps=<имя переменной>]*

Обязательные параметры

Имя	Описание	Тип	Пример значения
ImgCovar	Изображение - оценка АКФ	PT_IMAGE	ImgCovar
TypeCov	Тип АКФ	PT_INTEGER	1

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
DS	Дисперсия сигнала	PT_DOUBLE	OUT	
DN	Дисперсия шума	PT_DOUBLE	OUT	
Rmax	Максимальная корреляция	PT_DOUBLE	OUT	
Rmin	Минимальная корреляция	PT_DOUBLE	OUT	
FiRMax	Угол на направление максимальной корреляции в градусах	PT_DOUBLE	OUT	
Eps	Погрешность аппроксимации	PT_DOUBLE	OUT	

Идентификация импульсной характеристики линейной системы

Модуль реализует итерационную процедуру идентификации конечной импульсной характеристики двумерной линейной системы по изображениям на ее входе и выходе. Получаемое изображение импульсной характеристики имеет размеры *Height*×*Width* и вещественный тип данных (32 бит на пиксел). За нулевую точку плоскости импульсной характеристики (центр или пересечение осей симметрии) принимается центр формируемого изображения *ImgPSF* (при нечетных размерах изображения по координатам) или первый отсчет после центра (при четных размерах). Считается, что за пределами изображения отсчеты импульсной характеристики равны нулю. Обращение к программе:

RL ACF EstPSF *ImgIn* *ImgOut* *Height* *Width* *ImgPSF* [*Sym*=<значение>
Reg =<значение> *Eps*=<значение> *Iter*=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Изображение на входе искажающей системы	PT_IMAGENAME	<i>ImgIn</i>
<i>ImgOut</i>	Изображение на выходе искажающей системы	PT_IMAGENAME	<i>ImgOut</i>
<i>Height</i>	Высота оценки ИХ	PT_INTEGER	33
<i>Width</i>	Ширина оценки ИХ	PT_INTEGER	33
<i>ImgPSF</i>	Изображение - оценка ИХ искажающей системы	PT_IMAGENAME	<i>ImgPSF</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Sym</i>	Признак симметрии ИХ (0 - симметрия отсутствует, 1 - центральная симметрия, 2 - осевая симметрия, 3 - круговая симметрия)	PT_INTEGER	IN	0
<i>Reg</i>	Регуляризатор	PT_DOUBLE	IN	1
<i>Eps</i>	Точность (критерий остановки итерационного процесса)	PT_DOUBLE	IN	0,00001
<i>Iter</i>	Максимальное число итераций	PT_INTEGER	IN	5

8.2.10 Геометрические преобразования (Geom.dll)

Библиотека геометрических преобразований включает в себя прикладные модули:

- *Expand* – растяжение (увеличение в целое число раз);
- *Narrow* – сжатие (прореживание);
- *Scale* – вырезание фрагмента с произвольным масштабированием;
- *Transp* – транспонирование;
- *Place* – согласование изображений по паре точек;
- *Affin* – аффинное преобразование с заданными параметрами.

Растяжение изображения (увеличение в целое число раз)

Модуль растягивает изображение в целое число раз с использованием кусочно-постоянной интерполяции (дублированием отсчетов). Обращение к программе:

RL Geom Expand ImgIn VScale HScale ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>VScale</i>	Коэффициент растяжения по вертикали	PT_INTEGER	2
<i>HScale</i>	Коэффициент растяжения по горизонтали	PT_INTEGER	2
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Согласование изображений по паре точек

Модуль реализует геометрическое преобразование изображения, включающее в себя сдвиг, поворот и масштабирование. Коэффициенты преобразования вычисляются с помощью координат двух входных и соответствующих им выходных отсчетов. Обращение к программе:

***RL Geom Place ImgIn XIn1 YIn1 XIn2 YIn2 XOut1 YOut1
XOut2 YOut2 VerSizeOut HorSizeOut ImgOut
[Int=<значение>]***

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>XIn1</i>	Координаты первой точки входного изображения	PT_DOUBLE	0
<i>YIn1</i>		PT_DOUBLE	0
<i>XIn2</i>	Координаты второй точки входного изображения	PT_DOUBLE	1
<i>YIn2</i>		PT_DOUBLE	1
<i>XOut1</i>	Координаты первой точки выходного изображения	PT_DOUBLE	0
<i>YOut1</i>		PT_DOUBLE	0
<i>XOut2</i>	Координаты второй точки выходного изображения	PT_DOUBLE	1
<i>YOut2</i>		PT_DOUBLE	1
<i>VerSizeOut</i>	Высота выходного изображения	PT_INTEGER	1
<i>HorSizeOut</i>	Ширина выходного изображения	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Int</i>	Тип интерполяции (1-ступенчатая, 2-билинейная)	PT_INTEGER	IN	2

Сжатие (прореживание) изображения

Модуль сжимает (прореживает) изображение в целое число раз.
Обращение к программе:

RL Geom Narrow ImgIn VScale HScale ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>VScale</i>	Коэффициент сжатия по вертикали	PT_INTEGER	2
<i>HScale</i>	Коэффициент сжатия по горизонтали	PT_INTEGER	2
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Вырезание фрагмента с произвольным масштабированием

Модуль вырезает фрагмент изображения с использованием билинейной

интерполяции и сохраняет в новое изображение. Обращение к программе:

RL Geom Scale *ImgIn dV dH Height Width ImgOut*
[V0=<значение> H0=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>dV</i>	Шаг дискретизации по вертикали	PT_DOUBLE	1
<i>dH</i>	Шаг дискретизации по горизонтали	PT_DOUBLE	1
<i>Height</i>	Высота выходного фрагмента	PT_INTEGER	1
<i>Width</i>	Ширина выходного фрагмента	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>V0</i>	Начальная координата по вертикали	PT_INTEGER	IN	0
<i>H0</i>	Начальная координата по горизонтали	PT_INTEGER	IN	0

Транспонирование изображения

Обращение к программе:

RL Geom Transp *ImgIn ImgOut [FreeOpMem=<значение>]*

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

^{*)} См. замечание к модулю GnCovar библиотеки Gnr.dll

Аффинное преобразование с заданными параметрами

Модуль реализует геометрическое преобразование изображения общего вида. Координаты входного изображения (i_1, j_1) определяются через координаты выходного изображения (i_2, j_2) по формулам:

$$i_1 = Ai_2 + Bj_2 + C, \quad j_1 = Di_2 + Ej_2 + F$$

Обращение к программе:

**RL Geom Affin ImgIn A B C D E F VerSizeOut HorSizeOut
ImgOut [Int=<значение>]**

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGEName	ImgIn
<i>A B C</i> <i>D E F</i>	Параметры преобразования	PT_DOUBLE PT_DOUBLE	1 0 0 0 1 0
<i>VerSizeOut</i>	Высота выходного изображения	PT_INTEGER	1
<i>HorSizeOut</i>	Ширина выходного изображения	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGEName	ImgOut

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Int</i>	Тип интерполяции (1-ступенчатая, 2- билинейная)	PT_INTEGER	IN	2

8.2.11 Спектральные преобразования (*Spectr.dll*)

Библиотека спектральных преобразований включает в себя прикладные модули:

- *DFT* – двумерное дискретное преобразование Фурье;
- *DHT* – двумерное дискретное преобразование Хартли;
- *DCT* – двумерное дискретное косинусное преобразование;
- *DHaT* – двумерное дискретное преобразование Адамара;
- *DHrT* – двумерное дискретное преобразование Хаара.

Двумерное дискретное преобразование Хартли

Модуль формирует изображение, представляющее собой результат двумерного дискретного преобразования Хартли, примененного к входному изображению. Обращение к программе:

RL Spectr DHT ImgIn Direction ImgOut
[FreeOpMem=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGENAME	<i>ImgIn</i>
<i>Direction</i>	Направление преобразования (1 – прямое, -1 – обратное)	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

Двумерное дискретное преобразование Фурье

Модуль формирует пару изображений, представляющую собой результат комплексного двумерного дискретного преобразования Фурье, примененного к паре входных изображений, являющихся вещественной и мнимой частью входного комплексного сигнала. Обращение к программе:

RL Spectr DFT ImgInRe ImgInIm Direction ImgOutRe
ImgOutIm [FreeOpMem=<значение>]

^{*)} См. замечание к модулю GnCovar библиотеки Gnr.dll

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgInRe</i>	Вещественная часть входного изображения	PT_IMAGENAME	ImgInRe
<i>ImgInIm</i>	Мнимая часть входного изображения	PT_IMAGENAME	ImgInIm
<i>Direction</i>	Направление преобразования (1 – прямое, -1 – обратное)	PT_INTEGER	1
<i>ImgOutRe</i>	Вещественная часть выходного изображения	PT_IMAGENAME	ImgOutRe
<i>ImgOutIm</i>	Мнимая часть выходного изображения	PT_IMAGENAME	ImgOutIm

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

Двумерное дискретное косинусное преобразование

Модуль формирует изображение, представляющее собой результат двумерного дискретного косинусного преобразования, примененного к входному изображению. Обращение к программе:

RL Spectr DCT ImgIn Direction ImgOut
[FreeOpMem=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGENAME	ImgIn
<i>Direction</i>	Направление преобразования (1 – прямое, -1 – обратное)	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	ImgOut

^{*)} См. замечание к модулю GpCovar библиотеки Gnr.dll

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

Двумерное дискретное преобразование Адамара

Модуль формирует изображение, представляющее собой результат двумерного дискретного преобразования Адамара, примененного к входному изображению. Обращение к программе:

RL Spectr DHaT ImgIn Direction ImgOut
[FreeOpMem=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE NAME	<i>ImgIn</i>
<i>Direction</i>	Направление преобразования (1 – прямое, -1 – обратное)	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGE NAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

Двумерное дискретное преобразование Хаара

Модуль формирует изображение, представляющее собой результат двумерного дискретного преобразования Хаара, примененного к входному

^{*)} См. замечание к модулю GnCovar библиотеки Gnr.dll

изображению. Обращение к программе:

RL Spectr DHrT ImgIn Direction ImgOut
[FreeOpMem=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>Direction</i>	Направление преобразования (1 – прямое, -1 – обратное)	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	<i>ImgOut</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>FreeOpMem</i>	Доступная оперативная память (Мб) ^{*)}	PT_INTEGER	IN	10

8.2.12 Работа с фрагментами изображений (*Frg.dll*)

Библиотека работы с фрагментами изображений включает в себя прикладные модули:

- *Extr* – вырезание фрагмента изображения с прореживанием;
- *Ins* – наложение (врезка) изображения;
- *ReadPix* – чтение значений пикселей изображения;
- *WritePix* – запись значений пикселей изображения.

Вырезание фрагмента изображения с прореживанием

Модуль вырезает фрагмент изображения и сохраняет в новое изображение. Обращение к программе:

RL Frg Extr ImgIn V0 H0 Height Width ImgOut
[Scale=<значение>]

^{*)} См. замечание к модулю *GnCovar* библиотеки *Gnr.dll*

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE	ImgIn
<i>V0</i>	Начальная координата по вертикали	PT_INTEGER	0
<i>H0</i>	Начальная координата по горизонтали	PT_INTEGER	0
<i>Height</i>	Высота фрагмента	PT_INTEGER	1
<i>Width</i>	Ширина фрагмента	PT_INTEGER	1
<i>ImgOut</i>	Выходное изображение	PT_IMAGE	ImgOut

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Scale</i>	Коэффициент прореживания	PT_INTEGER	IN	1

Наложение (врезка) изображения

Модуль накладывает одно изображение на другое, выходное изображение формируется в виде:

$$Z(i, j) = \begin{cases} AX(i, j) + BX(i - V0, j - H0) + C, & \text{если } 0 \leq i < V, 0 \leq j < H \\ X(i, j), & \text{иначе,} \end{cases}$$

где $X(i, j), Y(i, j), Z(i, j)$ входное, врезаемое и выходное изображения,

V, H - вертикальный и горизонтальный размеры входного изображения.

Обращение к программе:

RL Frg Ins *ImgIn* *ImgFrg* *V0* *H0* *ImgOut* [*A*=<значение>
B=<значение> *C*=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	ImgIn
<i>ImgFrg</i>	Врезаемое изображение	PT_IMAGE_NAME	ImgFrg
<i>V0</i>	Начальная координата по вертикали	PT_INTEGER	0
<i>H0</i>	Начальная координата по горизонтали	PT_INTEGER	0
<i>ImgOut</i>	Выходное изображение	PT_IMAGE_NAME	ImgOut

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>A</i>	Мультипликативный коэффициент входного фрагмента изображения	PT_DOUBLE	IN	0
<i>B</i>	Мультипликативный коэффициент врезаемого изображения	PT_DOUBLE	IN	1
<i>C</i>	Аддитивная константа	PT_DOUBLE	IN	0

Запись значений пикселей изображения

Модуль записывает массив значений пикселей в изображение в соответствии с заданным направлением *Dir*. Для многокомпонентных изображений интерпретация массива значений производится в соответствии с параметром *PC*. Обращение к программе:

RL Frg WritePix *ImgIn V0 H0 Value* [*Dir*=<значение>
PC=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	ImgIn
<i>V0</i>	Начальная координата по вертикали	PT_INTEGER	0
<i>H0</i>	Начальная координата по горизонтали	PT_INTEGER	0
<i>Value</i>	Значения пикселей	PT_ARRAY	0,1,2

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Dir</i>	Направление записи: по горизонтали/по вертикали	PT_INTEGER	IN	0
<i>PC</i>	Расположение отсчетов в массиве: по пикселям/по слоям	PT_INTEGER	IN	1

Чтение значений пикселей изображения

Модуль читает значение пикселя (или одномерный массив значений пикселей длиной *Num* в соответствии с заданным направлением *Dir*) изображения и записывает в переменную *Value*. Для многокомпонентных изображений вывод значений пикселей реализуется в соответствии с параметром *PC*. Обращение к программе:

***RL Frq ReadPix ImgIn V0 H0 [Num=<значение>
Dir=<значение> PC=<значение>
Value=имя переменной>]***

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGENAME	<i>ImgIn</i>
<i>V0</i>	Начальная координата по вертикали	PT_INTEGER	0
<i>H0</i>	Начальная координата по горизонтали	PT_INTEGER	0

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Num</i>	Количество пикселей	PT_INTEGER	IN	1
<i>Dir</i>	Направление чтения: по горизонтали/по вертикали	PT_INTEGER	IN	0
<i>PC</i>	Расположение отсчетов в массиве: по пикселям/по слоям	PT_INTEGER	IN	1
<i>Value</i>	Значения пикселей	PT_ARRAY	OUT	

8.2.13 Работа с атрибутами и пикселями изображений (Img.dll)

Библиотека работы с атрибутами и пикселями изображений включает в себя прикладные модули:

- *GetAttr* – чтение атрибутов изображения;
- *AddCom* – добавление комментария изображения;
- *Dmp* – вывод численных значений фрагмента изображения;
- *Copy* – конвертация изображений;
- *Synth* – синтез многокомпонентного изображения.

Чтение атрибутов изображения

Обращение к программе:

```
RL Img GetAttr ImgIn [VerSize=<имя переменной>  
HorSize=<имя переменной>  
DataType=<имя переменной> ImgType=<имя переменной>  
BPS=<имя переменной> SPP=<имя переменной>]
```

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT IMAGE	<i>ImgIn</i>

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>VerSize</i>	Размер по вертикали	PT_INTEGER	OUT	
<i>HorSize</i>	Размер по горизонтали	PT_INTEGER	OUT	
<i>DataType</i>	Тип данных	PT_INTEGER	OUT	
<i>ImgType</i>	Тип изображения	PT_INTEGER	OUT	
<i>BPS</i>	Количество бит на отсчет	PT_INTEGER	OUT	
<i>SPP</i>	Количество отсчетов в пикселе	PT_INTEGER	OUT	

Добавление комментария изображения

Программа записывает в изображение комментарий. Если комментарий в изображении уже существует, то добавляемый пишется в него, начиная с позиции

$$Position = \begin{cases} Pos, & \text{если } Pos \geq 0; \\ LengthCom + 1 + Pos, & \text{если } Pos < 0, \end{cases}$$

где *LengthCom* - длина существующего комментария (при *Pos*=-1 комментарий добавляется к существующему). Обращение к программе:

RL Img AddCom ImgIn Comment [Pos=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>Comment</i>	Добавляемый комментарий изображения	PT_STRING	

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Pos</i>	Номер позиции для добавления комментария	PT_INTEGER	IN	-1

Вывод численных значений фрагмента изображения

Модуль выводит численные значения пикселей фрагмента изображения (на предоставляемое внешней программой устройство вывода) в соответствии с задаваемым согласно нотации языка С форматом вывода. Обращение к программе:

RL Img Dmp ImgIn V0 H0 Height Width
[Format=<значение> Attr=<значение>
PC=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGE_NAME	<i>ImgIn</i>
<i>V0</i>	Начальная координата по вертикали	PT_INTEGER	0
<i>H0</i>	Начальная координата по горизонтали	PT_INTEGER	0
<i>Height</i>	Высота фрагмента	PT_INTEGER	1
<i>Width</i>	Ширина фрагмента	PT_INTEGER	1

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>Format</i>	Формат вывода	PT_STRING	IN	"%g"
<i>Attr</i>	Вывод атрибутов да/нет	PT_INTEGER	IN	1
<i>PC</i>	Расположение отсчетов: по пикселям/ по слоям	PT_INTEGER	IN	1

Конвертация изображений

Модуль преобразует изображение из одного формата в другой. Если указан номер компоненты $0 \leq N < SPP$ (где *SPP* – количество компонент во входном изображении), то формируется однокомпонентное изображение, соответствующее номеру компоненты, в противном случае все компоненты переносятся в выходное изображение. Обращение к программе:

RL Img Copy ImgIn ImgOut [N=<значение>]

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn</i>	Входное изображение	PT_IMAGENAME	ImgIn
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	ImgOut

Необязательные параметры

Имя	Описание	Тип	Флаг	Значение по умолчанию
<i>N</i>	Номер компоненты	PT_INTEGER	IN	-1
<i>U_DT</i>	UsDataType	PT_INTEGER	IN	0
<i>U_PC</i>	UsPlanarConfiguration	PT_INTEGER	IN	0
<i>U_IT</i>	UsImageType	PT_INTEGER	IN	0
<i>O_PC</i>	OutPlanarConfiguration	PT_INTEGER	IN	0
<i>O_IT</i>	OutImageType	PT_INTEGER	IN	0

Примечание: параметры *U_DT*, *U_PC*, *U_IT*, *O_PC*, *O_IT* являются служебными, предназначенными для тестирования библиотеки monitor_rl.dll.

Модуль синтезирует цветное (трехкомпонентное) изображение из однокомпонентных. Обращение к программе:

RL Img Synth ImgIn0 ImgIn1 ImgIn2 ImgOut

Обязательные параметры

Имя	Описание	Тип	Пример значения
<i>ImgIn0</i>	Входное изображение (R-компонента)	PT_IMAGENAME	ImgIn0
<i>ImgIn1</i>	Входное изображение (G-компонента)	PT_IMAGENAME	ImgIn1
<i>ImgIn2</i>	Входное изображение (B-компонента)	PT_IMAGENAME	ImgIn2
<i>ImgOut</i>	Выходное изображение	PT_IMAGENAME	ImgOut

ГЛОССАРИЙ

1. Цифровой сигнал – дискретная функция дискретного аргумента, обычно несущая информацию о некотором процессе.
2. Цифровое изображение – графическая форма представления данных, предназначенная для зрительного восприятия.
3. Компрессия (сжатие) – вид обработки данных, имеющий целью сокращение их объема.
4. Сжатие без потерь (lossless compression) – сжатие, при котором изображение после декомпрессии полностью идентично исходному изображению.
5. Сжатие с потерями (Lossy Compression) – сжатие, при котором восстановленное изображение отличается от исходного. Это различие называется искажением. В качестве меры искажения используется среднеквадратическая ошибка.
6. Среднеквадратическая ошибка (СКО) – наиболее распространенная мера искажения, возникающего при сжатии изображений. Определяется как сумма квадратов разностей значений пикселей исходного и восстановленного изображений. Некоторые исследователи считают, что СКО слабо коррелировано с субъективным восприятием искажений изображения.
7. Спектр (Spectr) – представление сигнала в виде конечной или бесконечной суммы некоторых элементарных сигналов, умноженных на некоторые числа. В качестве элементарных сигналов обычно выступают ортогональные функции, такие как Фурье, Уолша, Хаара, Адамара, Хартли, вейвлеты и т.д.
8. Преобразование Карунена-Лоэва – преобразование, для которого базисные функции есть собственные векторы ковариационной матрицы входного сигнала. Является оптимальным по критерию достижения декорреляции входного сигнала.
9. Декорреляция при сжатии изображений – преобразование изображения, имеющее целью уменьшение взаимозависимости между отсчетами. Оптимальным декоррелирующим преобразованием (для гауссовского процесса) является преобразование Карунена-Лоэва. Все остальные преобразования только приближаются к нему.
10. Квантование цифрового сигнала – процесс замены исходного значения отсчета сигнала другим значением, как правило приводящий к уменьшению количества возможных значений цифрового сигнала. За счет этого достигается уменьшение числа требуемых для кодирования отсчета бит, но возникает ошибка кодирования. Расстояния между соседними возможными значениями квантованных отсчетов называют интервалами квантования.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Блейхут, Р. Быстрые алгоритмы цифровой обработки сигналов / Р. Блейхут. - М.: Мир, 1989. - 448 с.
2. Блох, Э.Л. О передаче бинарной последовательности равномерным кодом / Э. Л. Блох // Проблемы Передачи Информации, № 5, 1960. - С.12-22.
3. Вапник, В.Н. Теория распознавания образов / В.Н. Вапник, А.Я. Червоненкис. - М.: Наука, 1974. - 415 с.
4. Голд, Б. Цифровая обработка сигналов / Б. Голд, Ч. Рэйдер. - М.: Сов. радио, 1973. - 368 с.
5. Даджион, Д. Цифровая обработка многомерных сигналов / Д. Даджион, Р. Мерсеро. - М.: Мир, 1988. - 488 с.
6. Давыдов, Э. Г. Исследование операций / Э. Г. Давыдов. - М.: Высш. шк. 1990. - 383 с.
7. Евдокимов, В. П. Методы обработки данных в научных космических экспериментах / В. П. Евдокимов, В. М. Покрас. - М.: Наука, 1977. - 172 с.
8. Колесник, В. Д. Курс теории информации / В. Д. Колесник, Г. Ш. Полтырев. - М.: Наука, 1982. - 416 с.
9. Методы передачи изображений. Сокращение избыточности / У. К. Прэтт, Д. Д. Сакрисон, Х. Г. Мусманн [и др.]. - М.: Радио и связь, 1983. - 264 с.
10. Новик, Д. А. Эффективное кодирование / Д. А. Новик. - М.: Энергия, 1965. - 236 с.
11. Нуссбаумер, Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток / Г. Нуссбаумер. - М.: Радио и связь, 1985. - 248 с.
12. Прэтт, У. К. Цифровая обработка изображений / У. К. Прэтт. - М.: Мир, 1982. - Кн.1. - 310 с.
13. Прэтт, У. К. Цифровая обработка изображений / У. К. Прэтт. - М.: Мир, 1982. - Кн.2. - 480 с.
14. Рао, К. Р. Ортогональные преобразования при обработке цифровых сигналов / Н. Ахмед, К. Р. Рао. - М.: Связь, 1980. - 248 с.
15. Хаффмен, Д. А. Метод построения кодов с минимальной избыточностью / Д. А. Хаффмен // Кибернетический сборник. М.: 1961, вып. 3.
16. Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес. - М.: Мир, 1978. - 411 с.
17. Фукунага, К. Введение в статистическую теорию распознавания образов / К. Фукунага. - М.: Наука, 1979. - 368 с.
18. Ярославский, Л.П. Введение в цифровую обработку изображений / Л.П. Ярославский. - М.: Советское радио, 1979. - 311 с.
19. Ярославский, Л.П. О возможности параллельной и рекурсивной организации цифровых фильтров / Л.П. Ярославский // Радиотехника. - 1984, № 3. - С.87-91.
20. Ярославский, Л.П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику / Л.П. Ярославский. - М.: Радио и связь, 1987. - 296 с.

Учебное издание

*Сергеев Владислав Викторович,
Гашиников Михаил Валерьевич,
Глумов Николай Иванович,
Мясников Владислав Валерьевич,
Мясников Евгений Валерьевич*

**МАТЕМАТИЧЕСКИЕ МЕТОДЫ ОБРАБОТКИ
КОСМИЧЕСКИХ ИЗОБРАЖЕНИЙ.
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

Учебное пособие

Технический редактор В. А. Ф у р с о в
Редакторская обработка Т. Ю. Д е п ц о в а
Корректорская обработка Т. И. К у з н е ц о в а
Доверстка Е. А. В о л ч е н к о в а

Подписано в печать 15.11.07. Формат 60х84 1/16.

Бумага офсетная. Печать офсетная.

Печ. л. 9,25.

Тираж 120 экз. Заказ 189 ИП-97/2007

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

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