

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ

А.Л. Андреев

Автоматизированные телевизионные системы наблюдения

Часть II Арифметико-логические основы и алгоритмы

Учебное пособие
для курсового и дипломного проектирования



Санкт-Петербург

УДК 681.78

Андреев А.Л. Автоматизированные телевизионные системы наблюдения. Часть II. Арифметико-логические основы и алгоритмы. Учебное пособие для курсового и дипломного проектирования. – СПб: СПбГУИТМО, 2005. – 88с.

Приведён анализ общих принципов построения автоматизированных телевизионных систем наблюдения (АТСН) различного типа и назначения.

Во второй части рассмотрены арифметико-логические основы и наиболее часто используемые алгоритмы цифровой обработки изображений. В последнем разделе приводятся конкретные примеры АТСН, иллюстрирующие практическое использование некоторых алгоритмов, а также приёмов проектирования.

Для студентов оптических и оптоэлектронных направлений и специальностей.

Утверждено к изданию Учёным Советом факультета оптико-информационных систем и технологий, протокол № 2 от 08.02.05

© Санкт-Петербургский государственный университет
информационных технологий, механики и оптики,
2005

© А.Л. Андреев, 2005

Содержание

ВВЕДЕНИЕ	4
1. АРИФМЕТИКО-ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ	4
1.1. Системы счисления	4
1.2. Двоичная арифметика	12
1.3. Арифметика повышенной точности	21
1.4. Арифметика чисел с плавающей точкой	23
2. АЛГОРИТМЫ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В АТСН	30
2.1. Методы фильтрации, позволяющие уменьшить уровень помех	31
2.2. Алгоритмы определения интегральных параметров	34
2.3. Дифференциальные алгоритмы обработки изображений	35
2.4. Алгоритмы трансформирования исходных изображений	38
2.5. Алгоритмы выделения признаков контролируемых объектов	42
3. МЕТОДЫ ОБНАРУЖЕНИЯ, РАСПОЗНАВАНИЯ И ИЗМЕРЕНИЯ ПАРАМЕТРОВ ОБЪЕКТОВ НАБЛЮДЕНИЯ	47
3.1. Обнаружение объектов	47
3.2. Идентификация и классификация объектов в АТСН	49
3.3. Об измерении параметров объектов наблюдения	53
3.4. Методы моделирования на этапе проектирования АТСН	53
4. ПРИМЕРЫ ПОСТРОЕНИЯ АВТОМАТИЗИРОВАННЫХ ТЕЛЕВИЗИОННЫХ СИСТЕМ НАБЛЮДЕНИЯ	56
4.1. Оптико-электронные системы астроориентации	56
4.2. Принципы построения обучаемых АТСН	71
4.3. Основные этапы проектирования и реализации АТСН	83
ЗАКЛЮЧЕНИЕ	84
Библиографический список	85

ВВЕДЕНИЕ

Общей особенностью автоматизированных телевизионных систем наблюдения (АТСН) в отличие от других видов телевизионных систем является их способность функционировать без участия человека-оператора, т.е. в полностью автоматическом режиме на основе заранее определённых решающих правил.

Независимо от характера решаемых задач (обнаружение, распознавание или измерение параметров объектов) в АТСН, используются методы цифровой обработки изображений, реализуемые в реальном масштабе времени, т.е. со скоростью протекания реальных процессов. Указанное обстоятельство приходится учитывать при выборе как аппаратных средств, так и конкретных алгоритмов, используемых при обработке сигналов за ограниченное время анализа.

Классификация АТСН, основные понятия, определения, особенности аппаратной структуры и элементная база составляют содержание первой части. Во второй части пособия рассматриваются арифметико-логические основы, наиболее часто используемые алгоритмы цифровой обработки изображений. В последнем разделе приводятся конкретные примеры достаточно интересных видов АТСН, иллюстрирующие практическое использование некоторых алгоритмов, а также приёмов проектирования.

1. АРИФМЕТИКО-ЛОГИЧЕСКИЕ ОСНОВЫ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

На стадии отладки алгоритмов любому специалисту-разработчику АТСН часто бывает нужно контролировать текущие значения операндов в различных регистрах, вводить отдельные константы, рассчитывать необходимый объём памяти СОЗУ и ОЗУ и т. д. Таким образом разработчик аппаратных и программных средств АТСН должен иметь полное представление о двоичной и некоторых других системах счисления, о различных способах кодирования и обработки данных, в том числе чисел повышенной точности и чисел представленных в форме с плавающей точкой.

1.1. Системы счисления

Под системой счисления понимается способ представления чисел с помощью символов, имеющих определённое количественное значение. Множество символов, используемых для представления чисел, называют цифрами. Системы счисления подразделяются на два типа: непозиционные и позиционные.

Непозиционные системы счисления характеризуются сложностью представления чисел и сложностью алгоритмов выполнения арифметических операций, вследствие чего они не получили распространения в вычислительной технике.

Позиционной называется система счисления, в которой число, соответствующее записанной цифре, определяется как значением этой цифры, так и её позицией (разрядом) среди других цифр. Под основанием позиционной системы понимается число различных цифр, используемых в системе счисления. Типичной позиционной системой счисления является хорошо знакомая нам с детства десятичная система. Вспомнив известные правила интерпретации десятичных чисел и правила выполнения арифметических операций над ними, можно без труда понять принцип построения любой другой позиционной системы счисления.

Поскольку десятичная система оперирует десятью символами (0, 1, 2, 3, ..., 9), то говорят, что это система с основанием 10. Каждую позицию цифры в числе принято оценивать весом – показателем степени числа 10. В первой справа позиции целого числа размещены единицы, в соседней с ней второй позиции – десятки, затем сотни и т. д. Иначе говоря, вес крайней справа позиции равен 10^0 , соседней с ней – 10^1 , следующей – 10^2 , затем – 10^3 и т.д. То есть вес позиции в любой системе счисления равен основанию этой системы, возведенному в степень, соответствующую номеру этой позиции.

Пример 1.1

Вес разряда целого числа	10^3	10^2	10^1	10^0
Значение разряда	8	3	6	5

Тогда величину, представленную этими символами определяют следующим образом:

$$8 \times 10^3 + 3 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 = 8365$$

Дробная часть десятичного числа находится справа от десятичной точки, используемой для отделения целой части числа от дробной. (Заметим, что в вычислительной технике принято для отделения целой части от дробной использовать символ точки, а не символ запятой). Каждая позиция справа от десятичной точки также имеет свой вес.

Пример 1.2

Вес разряда десятичной дроби	10^{-1}	10^{-2}	10^{-3}	10^{-4}
Значение разряда	6	4	5	2

Величину, представленную этими символами определяют следующим образом:

$$6 \times 10^{-1} + 4 \times 10^{-2} + 5 \times 10^{-3} + 2 \times 10^{-4} = 0.6452.$$

1.1.1. Двоичная система счисления

Двоичная система проще десятичной. В ней используется только два символа, что хорошо согласуется с техническими характеристиками большинства цифровых схем, имеющих два устойчивых состояния. Как правило, в качестве символов в двоичной системе служат 0 и 1. При отображении состояния отдельных элементов могут использоваться и другие обозначения: «включено – выключено», «знак – пробел» и др.

В двоичной системе счисления, так же как и в десятичной, каждой позиции (разряду) соответствует определённый вес

Пример 1.3

Вес разряда двоичного числа	2^6 (64)	2^5 (32)	2^4 (16)	2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)	2^{-1} (0.5)	2^{-2} (0.25)	2^{-3} (0.125)
Значение разряда	1	0	1	0	0	1	0.	1	0	1

Величину, представленную этими символами определяют следующим образом:

$$1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 82.625$$

т.е. $1010010.101_2 = 82.625_{10}$.

Таким образом, процедура преобразования двоичного числа в десятичное проста: *необходимо сложить десятичные веса всех разрядов, в которых содержатся единицы.*

Если необходимо решить обратную задачу – преобразовать десятичное вещественное число в двоичный эквивалент, то необходимо воспользоваться отдельными различными процедурами для преобразования целой и дробной частей исходного числа. Законченный результат преобразования получают путём записи двоичных эквивалентов этих частей соответственно слева и справа от двоичной точки.

Процедура преобразования целого числа (целой части вещественного числа) в двоичный эквивалент заключается в следующем.

1. *Необходимо разделить исходное число на основание системы счисления, в которой число должно быть представлено – в данном случае на 2. Значение остатка от деления (1 или 0) присваивается младшему значащему разряду (МЗР) искомого числа.*

2. Результат деления на первом шаге (его целую часть) необходимо разделить опять на 2. Остаток от деления используется в качестве значения следующего по значимости разряда.

3. Шаги описанной процедуры повторяются до тех пор, пока частное, полученное в результате очередной операции деления (его целая часть) не станет равным нулю. Тогда остаток от последнего деления используется в качестве старшего значащего разряда (СЗР) искомого числа.

Пример 1.4

Шаг	Деление	Частное	Остаток (значение разряда)
1	134/2	67	0 (МЗР)
2	67/2	33	1
3	33/2	16	1
4	16/2	8	0
5	8/2	4	0
6	4/2	2	0
7	2/2	1	0
8	1/2	0	1 (СЗР)

Результат: $134_{10} = 10000110_2$

Процедура преобразования десятичной дроби в двоичную осуществляется посредством операции умножения.

1. Необходимо умножить дробную часть на основание системы счисления, в которой число должно быть представлено, – в данном случае на 2. Если результат умножения меньше 1, то старшему значащему разряду присваивается значение 0, если больше – 1.

2. Дробная часть результата предыдущей операции опять умножается на 2. Если результат умножения меньше 1, то следующему по значению (ближайшему справа) разряду присваивается значение 0, если больше – 1.

Шаги описанной процедуры повторяются до тех пор, пока либо результат умножения не станет в точности равен 1, либо не будет достигнута требуемая точность.

Пример 1.5

Шаг	Умножение	Произведение	Значение разряда
1	2×0.34375	0.6875	0 (СЗР)
2	2×0.6875	1.375	1
3	2×0.375	0.75	0
4	2×0.75	1.5	1
5	2×0.5	1.0	1
6	2×0	0	0 (МЗР)

Результат: $0.34375_{10} = 0.010110_2$

Шаг	Умножение	Произведение	Значение разряда
1	2×0.3	0.6	0 (СЗР)
2	2×0.6	1.2	1
3	2×0.2	0.4	0
4	2×0.4	0.8	0
5	2×0.8	1.6	1
6	2×0.6	1.2	1
7	2×0.2	0.4	0
8	2×0.4	0.8	0 (МЗР)

Поскольку процедура в примере 1.6 имеет характер бесконечного повторения группы одинаковых результатов, следует ограничиться достаточным количеством полученных разрядов, например восьмью разрядами. Результат: $0.3_{10} = 0.01001100_2$

1.1.2. Восьмеричная и шестнадцатеричная системы счисления

В цифровых вычислительных устройствах непосредственно не используются восьмеричные и шестнадцатеричные сигналы, поскольку они оперируют только с сигналами двух уровней. Однако восьмеричная и шестнадцатеричная системы счисления весьма удобны для компактного представления иногда достаточно длинных двоичных чисел, и они потому широко используются программистами и разработчиками программных средств ЦВУ, в частности микроконтроллеров.

Таблица 1.1

Восьмеричное число	Двоичное число	Десятичное число
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
10	1000	8
11	1001	9
12	1010	10
13	1011	11
...
100	100 0000	64

Восьмеричной является позиционная система счисления с основанием 8. В ней в качестве цифр используются следующие символы: 0, 1, 2, 3, 4, 5, 6, 7. Поскольку символов 8, то каждый восьмеричный символ может быть представлен 3-битовым двоичным числом ($\log_2 8 = 3$) от 000_2 до 111_2 . В таблице 1.1 приведены некоторые восьмеричные числа и их двоичные и десятичные эквиваленты.

Для представления целых двоичных чисел восьмеричными цифрами биты объединяют в группы по три, начиная с младшего значащего бита. Каждую группу битов заменяют соответствующим восьмеричным эквивалентом (см. первые 8 строк таблицы 1.1). Например, двоичное число 1010101111101_2 записывается восьмеричными цифрами как число 25375_8 . Аналогичным образом двоичное число 11000111_2 записывается как 307_8 . В обоих примерах для формирования целого числа групп двоичные числа были дополнены слева незначащими нулями. Очевидно, что такое дополнение не изменяет значения числа.

Дробную часть двоичного числа тоже можно представить восьмеричным эквивалентом аналогичным образом. Но в этом случае биты надо объединять в группы по три, начиная со старшего разряда. При этом незначащие нули добавляются при необходимости справа. Например, двоичная дробь 0.0101101_2 запишется в виде 0.264_8 . А двоичное вещественное число 1011.0101_2 после добавления слева и справа по два незначащих нуля ($001\ 011.010\ 100$) может быть представлено в виде 13.24_8 . Однако очевидно, что $13.24_8 \neq 13.24_{10}$. Во избежание ошибок рекомендуется использовать различные способы чтения восьмеричных и десятичных чисел. Так, восьмеричное число рекомендуется произносить «один – три – точка – два – четыре», в отличие от привычного чтения десятичного числа «тринадцать целых двадцать четыре сотых».

Шестнадцатеричная система – позиционная система счисления с основанием 16. В ней используются следующие символы: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Каждую шестнадцатеричную цифру можно использовать для сокращения записи 4-разрядного двоичного числа.

В таблице 1.2 приведены некоторые шестнадцатеричные числа и их двоичные и десятичные эквиваленты.

Чтобы записать целое двоичное число шестнадцатеричными цифрами биты объединяют в группы по четыре, начиная с младшего значащего бита. Каждую группу битов заменяют соответствующим шестнадцатеричным эквивалентом (см. первые 16 строк таблицы 1.2). Например, чтобы преобразовать двоичное число 1010101111101_2 в шестнадцатеричное необходимо, с целью получения целого числа групп, добавить слева два незначащих нуля ($0010\ 1010\ 1111\ 1101$), которые следует затем заменить на соответствующие шестнадцатеричные эквиваленты. Таким образом, получим число $2AFD_{16}$. Аналогичным образом двоичное число 11000111_2 записывается как $C7_{16}$.

Таблица 1.2

Шестнадцатеричное число	Двоичное число	Десятичное число
0	0	0
1	1	1
2	01	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15
10	10000	16
12	10001	17
...
3F	11 1111	63
40	100 0000	64
...
80	1000 0000	128

Представление двоичных дробей шестнадцатеричными символами осуществляется по правилам, аналогичным для преобразования этих дробей в восьмеричные. Биты дробной части, начиная со старшего, группируются по четыре. По мере необходимости справа могут добавляться незначащие нули. Например:

$$0.01011010_2 = 0.5A_{16};$$

$$1101.0111_2 = D7_{16}.$$

Правила преобразования восьмеричных и шестнадцатеричных чисел в десятичные подобны правилу преобразования двоичных чисел: Каждой позиции числа в любой позиционной системе соответствует определённый вес. Значение веса позиции умножается на цифру, занимающую эту позицию. Результаты операций умножения, выполненных для всех позиций числа, суммируются.

Пример 1.7

Вес разряда восьмеричного числа	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}
	(512)	(64)	(8)	(1)	(0.125)	(0.015625)
Значение разряда	1	1	7	2.	2	5
Результат умножения	512	64	56	2	0.25	0.078125

Величину, представленную символами 1172.25_8 определяют следующим образом:

$$1 \times 8^3 + 1 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 + 2 \times 8^{-1} + 5 \times 8^{-2} = 634.328125,$$

$$\text{т.е. } 1172.25_8 = 634.328125_{10}.$$

Пример 1.8

Вес разряда шестнадцатеричного числа	16^2 (256)	16^1 (16)	16^0 (1)	16^{-1} (0.0625)	16^{-2} (0.00390625)
Значение разряда	2	7	A. (10)	5	4
Результат умножения	512	112	10	0.3125	0.015625

Величину, представленную символами $27A.54_{16}$ определяют следующим образом:

$$2 \times 16^2 + 7 \times 16^1 + 10 \times 16^0 + 5 \times 16^{-1} + 4 \times 16^{-2} = 634.328125,$$

$$\text{т.е. } 27A.54_{16} = 634.328125_{10}.$$

Рассмотрим процедуры преобразования десятичных чисел в восьмеричные и шестнадцатеричные. Эти процедуры в принципе аналогичны процедурам преобразования десятичных чисел в двоичные. Вместо деления (для целых) или умножения (для дробных частей числа) на 2 необходимо выполнять деление или умножение на 8 или на 16. Остатки от деления или целые части произведений используются для формирования результата.

Ниже приводятся два примера преобразования десятичного числа 634.328125 в восьмеричное и шестнадцатеричное. Процедуры преобразования целой и дробной частей десятичного числа отличаются друг от друга. Они выполняются отдельно и объединяются при формировании окончательного результата.

Пример 1.9

Шаг	Преобразование целой части			Преобразование дробной части	
	Деление	Частное (целая часть)	Остаток	Умножение	Произведение (целая часть)
1	$634/8$	79	2(МЗР)	$8 \times 0.328125 =$ $= 2.625$	2 (СЗР)
2	$79/8$	9	7	$8 \times 0.625 = 5.0$	5 (МЗР)
3	$9/8$	1	1		
4	$1/8$	0	1(СЗР)		

Окончательный результат: $634.328125_{10} = 1172.25_8$.

Пример 1.10

Шаг	Преобразование целой части			Преобразование дробной части	
	Деление	Частное (целая часть)	Остаток	Умножение	Произведение (целая часть)
1	634/16	39	$10_{10} = A_{16}$ (МЗР)	$16 \times 0.328125 = 5.25$	5 (СЗР)
2	39/16	2	7	$16 \times 0.25 = 4.0$	4 (МЗР)
3	2/16	0	2(СЗР)		

Окончательный результат: $634.328125_{10} = 27A.54_{16}$.

1.2. Двоичная арифметика

Понимание правил выполнения арифметических операций над двоичными числами помогает специалисту-разработчику АТСН правильно оценивать время реализации того или иного алгоритма цифровой обработки сигналов, когда известны параметры используемого ЦВУ в составе проектируемой системы. Изучение двоичной арифметики тем более необходимо при использовании в составе аппаратных средств специализированных вычислительных устройств на базе так называемых микропрограммируемых микропроцессоров [5], предполагающих разработку вычислительных процедур на уровне микрокоманд.

Поскольку двоичная система счисления, также как и десятичная, относится к позиционным системам, то и правила выполнения арифметических операций над двоичными числами подобны соответствующим правилам выполнения арифметических операций над десятичными числами. Это существенно облегчает нам понимание законов двоичной арифметики.

1.2.1. Двоичное сложение

Сложение двоичных чисел подобно сложению десятичных. В обоих случаях операции начинаются с обработки наименьших значащих цифр, расположенных в крайней справа позиции. Если результат сложения наименьших значащих цифр двух слагаемых не помещается в соответствующем разряде результата, то происходит перенос. Цифра, переносимая в соседний слева разряд, добавляется к содержимому последнего. Сложение цифр любых одноименных разрядов может повлечь за собой перенос в более старший разряд. Перенос возникает, если результат сложения цифр одноименных разрядов больше 9 при использовании десятичной арифметики, и больше 1 в случае двоичной системы (например, $1 + 1 = 2_{10} = 10_2$).

Сходство и различие операций десятичного и двоичного сложения можно продемонстрировать на следующем примере:

Пример 1.10

	Десятичная арифметика	Двоичная арифметика
Перенос	11	11111110
Слагаемое	099	01100011
Слагаемое	095	01011111
Сумма	194	11000010

1.2.2. Двоичное вычитание

Очевидно, что при выполнении операции вычитания может быть получен отрицательный результат. В этой связи возникает вопрос о способе записи отрицательных чисел в двоичной системе счисления.

С учётом специфики функционирования аппаратных средств, в частности арифметико-логических устройств, помимо вышерассмотренного способа кодирования двоичных чисел получил распространение способ представления двоичных чисел посредством *величины и знака*. При этом бит знака всегда занимает самый старший разряд поля представления двоичного числа. Если число положительное, бит знака равен 0, если оно отрицательное, то этот бит равен 1. Рассмотрим, например, десятичное число 28. Его 7-разрядный двоичный эквивалент имеет вид 0011100. Если число положительное (+28), то к указанному двоичному эквиваленту следует добавить слева 0 (бит положительного знака), а именно 00011100. Если число отрицательное (-28), то слева добавляется 1 (бит отрицательного знака): 10011100.

Однако, с точки зрения упрощения вычислительных операций наиболее удобным является способ кодирования отрицательных двоичных чисел посредством *дополнительного кода* (или коротко — *дополнения*). При этом операция вычитания, как будет показано ниже, может быть, заменена операцией сложения, если в качестве первого слагаемого использовать уменьшаемое, а в качестве второго — вычитаемое, но взятое с противоположным знаком (в виде дополнения к исходному значению).

Формирование дополнительного кода состоит из двух простейших процедур, выполняемых любым арифметико-логическим устройством: получение обратного кода (операция инверсии путём замены всех битов двоичного числа на альтернативные значения) и прибавление единицы (инкремент операнда).

Пример 1.11

Число 4_{10} в двоичной форме (прямой код)	00000100
Обратный код числа 4_{10}	11111011
Прибавляемая единица	+00000001
Число $(-4)_{10}$ в дополнительном коде	<u>11111100</u>

При определении дополнений «вручную» можно пользоваться более простым алгоритмом.

Просматривают справа налево разряды числа, начиная с наименьшего по значимости. До тех пор пока встречаются нули, их копируют в одноимённые разряды результата. Первая встретившаяся единица также копируется в соответствующий разряд результата, но каждый последующий бит исходного числа заменяется на обратный.

Следует отметить, что на каждом этапе обработки данных следует не упускать из виду, с какими числами вы имеете дело в данный момент: с двоичными числами без знака или с двоичными числами со знаком. В таблице 1.3 в качестве примера показано, как одни и те же комбинации двоичных символов, составляющие 8-ми разрядные числа могут иногда интерпретироваться по-разному в зависимости от вышеуказанного обстоятельства.

Таблица 1.3

8-ми разрядное двоичное число	Двоичное число со знаком	Двоичное число без знака
00000000	+0	0
00000001	+1	1
00000010	+2	2
00000011	+3	3
...
...
01111100	+124	124
01111101	+125	125
01111110	+126	126
01111111	+127	127
10000000	-128	128
10000001	-127	129
10000010	-126	130
10000011	-125	131
...
...
11111100	-4	252
11111101	-3	253
11111110	-2	254
11111111	-1	255

При сложении или вычитании чисел со знаком результат всегда следует интерпретировать как число со знаком. Если при этом бит

старшего разряда равен единице, то результат – отрицательное число, представленное в дополнительном коде. Чтобы определить абсолютное значение (величину) результата, к последнему нужно применить процедуру вычисления дополнительного кода.

Процедура вычитания при использовании вышерассмотренного способа кодирования двоичных чисел со знаком выполняется посредством сложения следующим образом.

1. *Определяется дополнительный код вычитаемого (т.е. фактически меняется знак вычитаемого на противоположный).*

2. *Производится сложение этого кода с уменьшаемым.*

Если полученный результат (фактически разность) – число положительное (бит старшего разряда равен 0), то он представлен в прямом двоичном коде. Если полученный результат число отрицательное (бит старшего разряда равен 1), то он представлен в дополнительном коде, абсолютное значение которого можно найти путем применения к нему процедуры вычисления дополнения.

Пример 1.12

Вычислим разность чисел $58 - 23$.

1. Определяем дополнительный код вычитаемого (число -23).

00010111	Число 23_{10} (вычитаемое)
+ 11101000	Обратный код числа 23_{10}
<u>00000001</u>	Единица, добавляемая к обратному коду
11101001	Дополнительный код числа 23_{10}

2. Вычисляем разность.

Десятичная арифметика	Двоичная арифметика	
$\begin{array}{r} - 58 \\ - 23 \\ \hline 35 \end{array}$	$\begin{array}{r} 00111010 \\ + 11101001 \\ \hline 00100011 \end{array}$	Число 58_{10} (уменьшаемое) Дополнительный код (число -23_{10}) Разность (число 35_{10} в прямом коде)

Примечание: единица переноса из знакового разряда отбрасывается.

Пример 1.13

Вычислим разность чисел $26 - 34$.

1. Определяем дополнительный код вычитаемого (число -34).

00100010	Число 34 (вычитаемое)
+ 11011101	Обратный код числа 34_{10}
<u>00000001</u>	Единица, добавляемая к обратному коду
11011110	Дополнительный код числа 34_{10}

2. Вычисляем разность.

Десятичная арифметика	Двоичная урифметика	
$\begin{array}{r} 26 \\ - 34 \\ \hline -8 \end{array}$	$\begin{array}{r} 00011010 \\ + 11011110 \\ \hline 11111000 \end{array}$	Число 26_{10} (уменьшаемое)
		Дополнительный код (число -34_{10})
		Разность (число -8 в дополнительном коде (знаковый разряд равен 1))

3. Определяем абсолютное значение разности.

11111000	Разность в дополнительном коде
+ 00000111	Обратный код разности
+ 00000001	Единица, добавляемая к обратному коду
\hline	
00001000	Абсолютное значение разности (8_{10} в прямом коде)

1.2.3. Двоичное умножение

Вспомним сначала десятичное умножение. При умножении одного числа на другое одно из чисел называется *множимым*, другое — *множителем*. Умножение выполняется поразрядно. После умножения множимого на значение младшего разряда множителя получается первое *частичное произведение*. В результате умножения множимого на значение следующего по старшинству разряда множителя формируется второе частичное произведение, которое сдвигается на одну позицию влево (поскольку вес следующего по старшинству разряда в 10 раз больше). Подобным образом формируются и все другие частичные произведения. Для получения результирующего произведения смещенные друг относительно друга частичные произведения складываются с учётом возникающих при этом переносов.

Пример 1.14

$\times 17$	Множимое
$\times 12$	Множитель
\hline	
+ 34	Первое частичное произведение
+ 17	Второе частичное произведение
+ 100	Перенос
\hline	
204	Результирующее произведение

Двоичное умножение аналогично десятичному. Однако, поскольку каждый из разрядов сомножителей может быть равен либо единице, либо нулю, то фактически вместо операции умножения при формировании каждого частичного произведения осуществляется либо копирование множимого (когда соответствующий разряд множителя равен 1), либо регистрация нулевого значения (когда разряд множителя равен 0). Таким образом, процедура двоичного умножения сводится к выполнению определённой последовательности относительно простых операций: сдвигов и сложения.

× 00010001	Множимое (17_{10})
00001100	Множитель (12_{10})
+ 00000000	Первое частичное произведение
+ 00000000	Второе частичное произведение
+ 00010001	Третье частичное произведение
+ 00010001	Четвёртое частичное произведение
+ 00000000	Пятое частичное произведение
+ 00000000	Шестое частичное произведение
+ 00000000	Седьмое частичное произведение
+ 00000000	Восьмое частичное произведение
0000000000000000	Перенос
0000000011001100	Результирующее произведение (204_{10})

Очевидно, что полное число частичных произведений соответствует числу разрядов множителя, хотя некоторые частичные произведения и равны нулю. В данном конкретном примере сложение восьми частичных произведений не вызвало затруднений (связанных с возможностью многоразрядных переносов). В общем же случае получение результирующего произведения может оказаться не такой простой задачей. Кроме того, следует напомнить, что аппаратные средства большинства цифровых вычислительных устройств, в частности АЛУ, могут работать одновременно только с двумя операндами. Поэтому, с целью упрощения практической реализации процедуры двоичного умножения был разработан следующий алгоритм, получивший названия *умножения путём сдвига и сложения*. Основные правила, составляющие этот алгоритм следующие.

1. *Формирование первого частичного произведения.* Если значение младшего разряда множителя равно 0, то и результат равен 0, если значение этого разряда равно 1, то результат является копией множимого.

2. *Правило сдвига.* При использовании очередного разряда множителя для формирования частичного произведения осуществляется сдвиг множимого на один разряд (позицию) влево.

3. *Правило сложения.* Каждый раз, когда значение разряда множителя равно 1, к предыдущему результату необходимо прибавить множимое, расположенное в позиции, определяемой правилом сдвига.

4. *Определение результирующего произведения.* Искомое произведение есть результат выполнения всех операций сдвига и сложения.

× 01011101	Множимое (93_{10})
00101101	Множитель (45_{10})
+ 01011101	Первое частичное произведение
01011101	Множимое, сдвинутое влево на 2 разряда
+ 0111010001	Третье частичное произведение
01011101	Множимое, сдвинутое влево на 3 разряда
+ 10010111001	Четвёртое частичное произведение
01011101	Множимое, сдвинутое влево на 5 разрядов
1000001011001	Результирующее произведение (4185_{10})

Заметим, что в общем случае результирующее произведение может иметь число значащих разрядов, равное сумме разрядов сомножителей. Это необходимо учитывать при определении разрядности ячеек памяти, используемых для хранения операндов.

1.1.1. Двоичное деление

Чтобы понять алгоритм деления двоичных чисел, используемый в цифровых вычислительных устройствах, проанализируем сначала известную нам процедуру деления десятичных чисел.

Пример 1.16 Деление десятичных чисел начинается с анализа делимого (204) и делителя (12). Анализ показывает, что число 12 «укладывается» в числе 20, представляющих два старших разряда делимого, один раз, поскольку остаток (8) меньше делителя (12). Объединяя остаток со значением следующего разряда делимого, получаем число 84 и снова выясняем, сколько раз делитель укладывается в этом числе. В данном случае 7 раз. Таким образом, результатом деления является число 17.

Из рассмотренного примера видно, что операция деления, в принципе, значительно сложнее умножения. Реализовать процедуру «выяснения сколько раз делитель укладывается в том или ином числе» в вычислительном устройстве было бы непросто. Однако при использовании двоичной системы кодирования чисел алгоритм деления существенно упрощается. В самом деле, при двоичном представлении делитель может укладываться в любом двоичном числе (содержащем столько же разрядов) только один раз (если данное число больше делителя или равно ему), или ни одного раза (если это число меньше). Поэтому процедура «выяснения» может быть заменена процедурой вычитания (или прибавления

дополнительного кода) делителя с последующим анализом результата. Таким образом, весь алгоритм деления двоичных чисел сводится к достаточно простым операциям сдвига и сложения.

Ниже вместо подробного описания алгоритма двоичного деления приводится пример, хорошо иллюстрирующий его применение на практике.

Пример 1.17

Выполним деление числа $11001100_2 (204_{10})$ на число $1100_2 (12_{10})$.

Прежде всего, необходимо представить делитель в дополнительном коде.

01100 – число $+12_{10}$ в прямом коде (первый ноль – знаковый разряд);

10011 – обратный код числа 12 ;

$+00001$ – единица, прибавляемая к обратному коду;

10100 – дополнительный код числа 12 (т.е. число -12).

После этого можно непосредственно приступить к процедуре деления, которая выполняется в несколько шагов.

1. Аналогично процедуре десятичного деления на первом шаге необходимо выяснить сколько раз делитель укладывается в числе, образованном соответствующим количеством старших разрядов делимого. Вычислительное устройство, разумеется, не может строить догадок на этот счёт. Но, как уже было указано выше, на практике при двоичном кодировании операндов подобный анализ делается путём вычитания делителя из делимого.

011001100 – делимое (уменьшаемое);

$+10100$ – делитель в дополнительном коде (вычитаемое);

000001100 – результат, полученный на первом шаге деления.

Заметим, что наличие нуля в знаковом разряде результата (бит переноса из знакового разряда отбрасывается) свидетельствует о том, что результат вычитания положительный. Следовательно, делитель полностью укладывается в анализируемом числе, и значит бит старшего разряда частного равен 1.

Таким образом, будущий окончательный результат деления можно пока представить в виде $1xxx\dots$

2. С целью определения следующего разряда частного, следует повторить попытку вычитания делителя, но предварительно осуществив сдвиг первого результата на одну позицию влево. Нетрудно убедиться, что это вполне согласуется с соответствующим шагом процедуры десятичного деления (см. пример 1.16).

Итак:

а) сдвиг влево первого результата:

000001100 — первый результат вычитания до сдвига;

000011000 — первый результат после сдвига на один разряд;

б) операция вычитания:

000011000 — первый результат после сдвига на один разряд;

+10100 — вычитание делителя;

101011000 — второй результат вычитания.

Наличие единицы в знаковом разряде свидетельствует о том, что в данном случае результат вычитания отрицательный. Значит, следующий значащий разряд частного равен 0. Итак, промежуточное значение частного: 10xx...

Прежде чем перейти к очередному шагу алгоритма необходимо вернуться на один шаг назад, т.е. к первому результату, полученному после первого сдвига (так как последнее вычитание дало отрицательный результат).

101011000 — второй результат;

+01100 — число $+12_{10}$ в прямом коде (прибавление делителя);

000011000 — первый результат после сдвига на один разряд.

2. Теперь нужно осуществить второй сдвиг первого результата и повторить попытку вычитания ещё раз.

а) второй сдвиг влево первого результата:

000011000 — первый результат после сдвига на один разряд;

000110000 — первый результат после сдвига на два разряда;

б) операция вычитания:

+000110000 — первый результат после сдвига на два разряда;

+10100 — вычитание делителя;

101110000 — третий результат вычитания.

Наличие единицы в знаковом разряде свидетельствует о том, что и в данном случае результат вычитания отрицательный. Значит, следующий значащий разряд частного также равен 0. Итак, промежуточное значение частного: 100x...

Поскольку третий результат вычитания также отрицательный, необходимо вернуться на один шаг назад к первому результату вычитания, полученному после сдвига на два разряда:

+101110000 — третий результат вычитания;

01100 — число $+12_{10}$ в прямом коде (прибавление делителя);

000110000 — первый результат после сдвига на два разряда.

3. Следующий шаг деления:

а) третий сдвиг влево первого результата:

000110000 – первый результат после сдвига на два разряда;

001100000 – первый результат после сдвига на три разряда;

б) операция вычитания:

001100000 – первый результат после сдвига на три разряда;

+10100 – вычитание делителя;

110100000 – четвёртый результат вычитания.

Как и прежде наличие единицы в знаковом разряде свидетельствует о том, что и в данном случае результат вычитания отрицательный. Значит, следующий значащий разряд частного также равен 0. Итак, промежуточное значение частного: 1000...

Возврат к первому результату вычитания после сдвига на три разряда:

110100000 – четвёртый результат вычитания.

+01100 – число $+12_{10}$ в прямом коде;

001100000 – первый результат после сдвига на три разряда.

4. Ещё один шаг деления:

а) четвёртый сдвиг влево первого результата:

001100000 – первый результат после сдвига на три разряда;

011000000 – первый результат после сдвига на четыре разряда;

б) операция вычитания:

011000000 – первый результат после сдвига на четыре разряда;

+101000000 – вычитание делителя;

000000000 – пятый результат вычитания.

Наличие нуля в знаковом разряде означает, что следующий значащий разряд частного равен 1. К тому же поскольку результат последнего вычитания в точности равен нулю (случай деления без остатка), то процесс деления можно завершить. Таким образом, окончательный результат деления равен $10001_2 = 17_{10}$.

Заметим, что если бы последний результат вычитания не был бы равен нулю, то процесс деления следовало бы продолжить либо до достижения нулевого результата, либо до достижения требуемой точности.

1.3. Арифметика повышенной точности

При реализации алгоритмов цифровой обработки сигналов приходится решать вопрос об определении достаточной разрядности чисел, представляемых двоичным кодом. Например, для первичного кодирования видеосигналов, формируемых телевизионным датчиком АТСН, зачастую достаточно 8-и разрядов двоичного кода. Это соответствует возможной погрешности квантования сигнала $\approx \pm 0,25\%$.

Однако, при выполнении различных алгоритмов обработки таких чисел требуется оперировать с числами, имеющими значительно большую длину.

В самом деле, 8-ми разрядные числа охватывают диапазон значений от 0 до 255, если кодирование осуществляется без учёта знака. При кодировании чисел со знаком диапазон 8-ми разрядных чисел включает значения от -128 до $+127$.

Используя в качестве операнда два 8-ми разрядных слова (т.е. 16-ти разрядные данные), получаем диапазон от $-32\,768$ до $+32\,767$. При этом погрешность кодирования оценивается величиной $\approx \pm 0,0015\%$.

Ещё более высокая точность может быть достигнута при кодировании данных тремя 8-ми разрядными словами: 1 бит для знака и 23 бита для абсолютной величины числа. В этом случае диапазон возможных значений чисел простирается от $-8\,388\,608$ до $+8\,388\,607$, включая 0. При этом погрешность кодирования оценивается величиной меньшей, чем одна миллионная. При необходимости выполнения операций над вещественными числами, часть разрядов двоичного кода может быть использована для записи дробной части. Число разрядов, отводимых для целой и дробной частей операнда, задаётся положением двоичной точки и определяется разработчиком.

Используя числа повышенной точности, следует помнить, что для их хранения требуется значительно больший объём оперативной памяти. Кроме того, к быстрдействию ЦВУ предъявляются более высокие требования. Предположим, при обработке данных, представленных 24-х разрядными числами, используется 8-ми разрядный микроконтроллер. В этом случае для выполнения операции сложения сначала нужно произвести обращение к младшему значащему байту каждого числа. После сложения двух байтов результат записывают в память, а возможные при этом переносы подлежат временному хранению. Затем из памяти извлекают средние по значимости байты, складывают и прибавляют к полученной сумме биты переноса. Результат записывают в ячейку памяти, специально зарезервированную для хранения среднего байта суммы. Наконец из памяти извлекают старшие значащие байты, складывают, а к сумме добавляют биты переноса, полученные при предыдущей операции сложения, после чего результат записывают в ячейку памяти, зарезервированную для хранения старшего байта суммы. Таким образом, при прочих равных условиях сложение чисел тройной точности занимает в три раза больше времени и требует в три раза больший объём памяти. Кроме того, если, например, в процессе сложения чисел тройной точности произойдёт прерывание основной программы, то на время обработки прерывания необходимо в стековой памяти хранить содержимое регистра переноса.

1.4. Арифметика чисел с плавающей точкой

При использовании арифметики повышенной точности не могут быть решены проблемы кодирования очень больших и очень малых чисел.

Эти проблемы могут быть разрешены с помощью, так называемой *арифметики чисел с плавающей точкой (запятой)*.

Арифметика чисел с плавающей точкой нам уже знакома, применительно к десятичной системе счисления. Особенно удобно в форме с плавающей точкой представлять очень малые, например, заряд электрона $e = 1.6 \times 10^{-19}$ Кл, и очень большие числа, например число Авогадро $N_A = 6.02 \times 10^{23}$. При этом каждое число представляется в виде двух составляющих: *мантиссы* и *порядка*. В нормализованном виде мантисса числа должна укладываться в диапазон значений от 0.1 до 1.0. Порядок десятичного числа представляет собой показатель степени 10. Например, число 0.0025 в нормализованном виде представляется 0.25×10^{-2} , а число -1250 – как -0.125×10^4 и т.д.

В двоичной системе счисления для записи мантиссы обычно используются три байта, т.е. 24 разряда. Самый старший разряд выполняет роль знака числа, а остальные 23 разряда служат для записи абсолютного значения мантиссы. Порядок числа занимает четвертый байт, в котором старший разряд выполняет роль знака порядка. Согласно такому представлению число с плавающей точкой имеет следующий формат:

$$\pm \text{XXXXXXXXXXXXXXXXXXXXXXXXXXXX} \times 2^{\pm \text{XXXXXXXX}},$$

где X – условное обозначение двоичной цифры (бита). Следовательно двоичная арифметика с плавающей точкой позволяет оперировать с числами от $-2^{23} \times 2^{-128}$ до $+(2^{23} - 1) \times 2^{127}$.

Действительные числа с плавающей точкой могут храниться в памяти ЦВУ также как и целые числа. Однако не следует забывать, что мантисса – число, принадлежащее диапазону $0.1_2 - 1.0_2$ ($0.5_{10} - 1.0_{10}$).

Представление числа с плавающей точкой в 8-ми разрядном микроконтроллере можно схематически изобразить следующим образом:

Таблица 1.4

№ байта	Содержание байта
1	8 младших битов мантиссы
2	8 средних битов мантиссы
3	\pm (бит знака) и 7 старших битов мантиссы
4	\pm (бит знака) и 7 битов порядка

Таким образом, любое число, включая 0, занимает в памяти микроконтроллера 4 байта.

Перед тем, как выполнить алгебраическое сложение чисел, представленных в форме с плавающей запятой, необходимо осуществить выравнивание их порядков. При этом слагаемое с меньшим порядком приводится к порядку второго слагаемого (с большим порядком). Для этого мантисса слагаемого с меньшим порядком умножается на число $2^{P1 - P2}$. (Здесь $P1$ и $P2$ – порядки слагаемых, причём $P1 < P2$). Это эквивалентно операции сдвига двоичного кода мантиссы вправо на $P2 - P1$ разрядов. Алгебраическое сложение чисел с выравненными порядками сводится к алгебраическому сложению мантисс.

Примечание.

Это хорошо согласуется с известным правилом алгебраического сложения десятичных чисел, представленных в форме с плавающей точкой. Например, чтобы к числу 0.6×10^{-2} прибавить число 0.15×10^3 нужно сначала первое число преобразовать к виду 0.000006×10^3 , а затем получить окончательный результат: 0.150006×10^3 .

При выполнении операции умножении (или деления) чисел, представленных в форме с плавающей точкой, порядок произведения (или частного) определяется посредством суммирования (или соответственно вычитания) порядков этих чисел. Мантисса произведения (или частного) определяется посредством умножения (или соответственно деления) мантисс этих чисел. Полученное произведение необходимо привести к нормализованному виду.

Чтобы лучше понять логику вышеуказанного правила для двоичных чисел, рекомендуем выполнить перемножение, а затем деление, например, следующих десятичных чисел: 0.25×10^{-2} и 0.5×10^3 .

Ниже приводятся два примера, подробно иллюстрирующих алгоритм алгебраического сложения двоичных чисел, представленных в форме с плавающей точкой.

Пример 1.18

**Выполним побайтное сложение двух положительных чисел.
(Формат чисел соответствует таблице 1.4)**

Первое слагаемое		Второе слагаемое
00000000	Младшие байты мантисс	00000000
10011000	Средние байты мантисс	10010010
00110000	Старшие байты мантисс	00110010
11111111	Байты порядков	00000001

1. Находим разность порядков, путём вычитания из большего порядка меньшего.

$$\begin{array}{r}
 00000001 \text{ Порядок второго слагаемого } (+1_{10}) \\
 + 11111111 \text{ Порядок первого слагаемого } (-1_{10}) \text{ в дополнительном коде} \\
 \hline
 00000010 \text{ Разность порядков } (2_{10})
 \end{array}$$

2. Приводим слагаемые к одному порядку, т.е. порядок первого слагаемого (-1) приводим к порядку второго слагаемого (+1). Это процедура реализуется путём сдвига мантиисы первого слагаемого на 2 разряда вправо (деление на 2^2).

Теперь оба слагаемых имеет вид:

00000000	Первое слагаемое после сдвига мантиисы на
00100110	два разряда вправо. Порядок первого
00001100	слагаемого теперь совпадает с порядком
00000001	второго слагаемого.

00000000	Второе слагаемое осталось без
10010010	изменений.
00110010	
00000001	

3. Складываем младшие байты мантиис.

00000000	Младший байт мантиисы первого слагаемого;
+ 00000000	Младший байт мантиисы второго слагаемого;
<u>00000000</u>	Младший байт суммы мантиис.

4. Складываем средние байты мантиис.

00100110	Средний байт мантиисы первого слагаемого;
+ 10010010	Средний байт мантиисы второго слагаемого;
<u>10111000</u>	Средний байт суммы мантиис (бит переноса отсутствует).

5. Складываем старшие байты мантиис.

00001100	Старший байт мантиисы первого слагаемого;
+ 00110010	Старший байт мантиисы второго слагаемого;
<u>00111110</u>	Старший байт суммы мантиис.

6. Запишем окончательный результат.

00000000	Младший байт суммы мантиис;
10111000	Средний байт суммы мантиис;
00111110	Старший байт суммы мантиис;
00000001	Порядок суммы чисел.

Теперь осуществим проверку полученного результата.

1. Проанализируем первое слагаемое и представим его в десятичной системе счисления.

Прежде всего заметим, что первое слагаемое – число положительное, так как знаковый разряд мантиисы равен 0. Следовательно, мантииса первого слагаемого представлена прямым двоичным кодом.

Напомним, что для преобразования двоичного числа в десятичное достаточно сложить десятичные веса всех двоичных разрядов, которые равны единицы.

Итак, мантисса первого слагаемого:

$$\begin{aligned} +(0.110000\ 10011000\ 00000000)_2 &= +(2^{-1}+2^{-2}+2^{-7}+2^{-10}+2^{-11})_{10} \\ &= +0.75927734375_{10}. \end{aligned}$$

Учитывая порядок первого слагаемого (-1), запишем его значение:

$$+0.75927734375 \times 2^{-1} = +0.379638671875.$$

1. Проанализируем второе слагаемое и представим его в десятичной системе счисления.

Второе слагаемое также – число положительное, так как знаковый разряд мантиссы равен 0. Следовательно, мантисса второго слагаемого также представлена прямым двоичным кодом.

Мантисса второго слагаемого:

$$\begin{aligned} +(0.110010\ 10010010\ 00000000)_2 &= +(2^{-1}+2^{-2}+2^{-5}+2^{-7}+2^{-10}+2^{-13})_{10} = \\ &= +0.7901611328125_{10}. \end{aligned}$$

Учитывая порядок (+1), запишем значение второго слагаемого:

$$+0.7901611328125 \times 2^1 = +1.580322265625.$$

2. Теперь проанализируем полученную сумму.

Сумма также число положительное, так как значение знакового разряда мантиссы равно 0.

Мантисса суммы:

$$\begin{aligned} &+(0.111110\ 10111000\ 00000000)_2 = \\ &= +(2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-7}+2^{-9}+2^{-10}+2^{-11})_{10} = +0.97998046875_{10}. \end{aligned}$$

Учитывая порядок полученной суммы (+1), запишем:

$$+0.97998046875 \times 2^1 = +1.9599609375.$$

3. Проверка.

$$(+0.379638671875) + (+1.580322265625) = +1.9599609375.$$

Пример 1.19

Выполним побайтное сложение двух отрицательных чисел, представленных в формате с плавающей точкой.

Первое слагаемое		Второе слагаемое
10000000	Младшие байты мантисс	00000000
00010100	Средние байты мантисс	10000001
11000100	Старшие байты мантисс	11010110
00000001	Байты порядков	00000000

1. Находим разность порядков, путём вычитания из большего порядка меньшего.

$$\begin{array}{r}
 + 00000001 \quad \text{Порядок первого слагаемого (+1}_{10}\text{)} \\
 + 00000000 \quad \text{Порядок второго слагаемого (0)} \\
 \hline
 00000001 \quad \text{Разность порядков (1}_{10}\text{)}
 \end{array}$$

2. Приводим слагаемые к одному порядку, т.е. порядок второго слагаемого (0) приводим к порядку первого слагаемого (+1). Это процедура реализуется путём сдвига мантиисы второго слагаемого на 1 разряд вправо (деление на 2^1).

Теперь оба слагаемых имеет вид:

$$\begin{array}{r}
 10000000 \quad \text{Первое слагаемое без изменений;} \\
 00010100 \\
 11000100 \\
 00000001 \\
 \\
 10000000 \quad \text{Второе слагаемое после сдвига мантиисы} \\
 01000000 \quad \text{на один разряд вправо. Порядок второго} \\
 11101011 \quad \text{слагаемого теперь совпадает с порядком} \\
 00000001 \quad \text{первого слагаемого.}
 \end{array}$$

Примечание.

Важно заметить, что при сдвиге отрицательного числа, представленного в дополнительном коде, необходимо в позиции знакового разряда сохранять 1.

3. Складываем младшие байты мантиис.

$$\begin{array}{r}
 + 10000000 \quad \text{Младший байт мантиисы первого слагаемого;} \\
 + 10000000 \quad \text{Младший байт мантиисы второго слагаемого;} \\
 \hline
 00000000 \quad \text{Младший байт суммы мантиис (имеется бит} \\
 +1(\text{перенос}) \quad \text{переноса в следующий разряд);}
 \end{array}$$

4. Складываем средние байты мантиис.

$$\begin{array}{r}
 00010100 \quad \text{Средний байт мантиисы первого слагаемого;} \\
 + 01000000 \quad \text{Средний байт мантиисы второго слагаемого;} \\
 \hline
 01010100 \quad \text{Сумма средних байтов мантиис;} \\
 + \quad \quad \quad 1 \quad \text{Прибавление единицы бита переноса;} \\
 \hline
 01010101 \quad \text{Средний байт суммы мантиис;}
 \end{array}$$

5. Складываем старшие байты мантиис.

$$\begin{array}{r}
 + 11000100 \quad \text{Старший байт мантиисы первого слагаемого} \\
 + 11101011 \quad \text{Старший байт мантиисы второго слагаемого} \\
 \hline
 10101111 \quad \text{Старший байт суммы мантиис}
 \end{array}$$

6. Запишем окончательный результат.

00000000	Младший байт суммы мантисс;
01010101	Средний байт суммы мантисс;
10101111	Старший байт суммы мантисс;
00000001	Порядок суммы чисел.

Осуществим проверку полученного результата.

1. Проанализируем первое слагаемое и представим его в десятичной системе счисления.

Заметим, что первое слагаемое – число отрицательное, так как знаковый разряд мантиссы равен 1. Следовательно, мантисса первого слагаемого представлена в дополнительном коде.

В данном случае перед операцией преобразования двоичного кода мантиссы в десятичный необходимо определить абсолютное значение мантиссы. Для этого к дополнительному коду мантиссы необходимо применить операцию вычисления дополнительного кода («взять дополнение к дополнению»). Воспользовавшись одним из алгоритмов определения дополнительного кода (см. раздел 1.2.2), находим абсолютное значение мантиссы первого слагаемого:

11000100 00010100 10000000	мантисса первого слагаемого;
00111011 11101011 10000000	абсолютное значение мантиссы.

Представим мантиссу первого слагаемого в десятичном виде:

$$\begin{aligned} & -(0.111011\ 11101011\ 10000000)_2 = \\ & = -(2^{-1} + 2^{-2} + 2^{-3} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-9} + 2^{-11} + 2^{-13} + 2^{-14} + 2^{-15})_{10} = \\ & = -0.936248779296875_{10}. \end{aligned}$$

С учетом порядка (+1), запишем значение первого слагаемого:

$$-0.936248779296875 \times 2^1 = -1.87249755859375.$$

2. Проанализируем второе слагаемое и представим его в десятичной системе счисления.

Второе слагаемое – также число отрицательное, так как значение знакового разряда мантиссы равно 1. Следовательно, мантисса второго слагаемого представлена в дополнительном коде.

11010110 10000001 00000000	мантисса второго слагаемого ;
00101001 01111111 00000000	абсолютное значение мантиссы.

Представим мантиссу второго слагаемого в десятичном виде:

$$\begin{aligned} & -(0.101001\ 01111111\ 00000000)_2 = \\ & = -(2^{-1} + 2^{-3} + 2^{-6} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12} + 2^{-13} + 2^{-14})_{10} = -0.64837646484375_{10} \end{aligned}$$

Поскольку порядок второго слагаемого в исходном виде равен 0, то окончательно значение второго слагаемого:

$$-0.64837646484375 \times 2^0 = -0.64837646484375.$$

3. Теперь проанализируем полученную сумму.

Сумма также число отрицательное, так как знаковый разряд мантиссы равен 1.

10101111 01010101 00000000	Мантисса суммы;
01010000 10101011 00000000	Абсолютное значение суммы.

Представим мантиссу суммы в десятичном виде:

$$-(1.010000 10101011 00000000)_2 = -(2^{-0} + 2^{-2} + 2^{-7} + 2^{-9} + 2^{-11} + 2^{-13} + 2^{-14})_{10} = -1.26043701171875_{10}.$$

Учитывая значение порядка суммы (+1), окончательно получим:

$$-1.26043701171875 \times 2^1 = -2.5208740234375.$$

1. Проверка:

$$(-1.87249755859375) + (-0.64837646484375) = -2.5208740234375.$$

Вопросы и задания для самопроверки:

1. Что такое позиционная система счисления. Приведите пример позиционной и непозиционной систем счисления?

2. Преобразуйте следующие десятичные числа в двоичные эквиваленты: 251.125; 170.002. Проверьте правильность полученного результата, выполнив обратное преобразование.

3. Представьте следующие числа в десятичной, восьмеричной и шестнадцатеричной системах счисления: 11111111; 101.11111; 1110.1110. Проверьте правильность полученных результатов, выполнив преобразование полученных восьмеричных и шестнадцатеричных эквивалентов в десятичный эквивалент.

4. Определите дополнительный код следующих двоичных чисел: 10110110; 01010101; 1111000011110000; 1100110011000011. Исходя из того, что указанные числа представляют собой числа со знаком, определите, какие из них являются положительными, а какие отрицательными.

5. Выполните умножение приведённых ниже пар двоичных чисел: 01011101×00101101; 00011011×11111100. Преобразуйте сомножители и произведения в десятичную систему счисления. Проверьте правильность выполненных действий, сравнив результаты двоичного и десятичного умножений.

6. Используя описанную выше процедуру двоичного деления путём вычитания и сдвига, выполните деление следующих положительных чисел: 11110/101; 10000100/1100. Проверьте полученные результаты, выполнив деление тех же чисел, представленных в десятичной системе счисления.

7. По аналогии с вышерассмотренными примерами 1.18 и 1.19 запишите два произвольных операнда, представленных в форме с плавающей точкой. Выполните сложение этих чисел и проверьте полученный результат, предварительно преобразовав слагаемые и сумму в десятичные эквиваленты.

2. АЛГОРИТМЫ ЦИФРОВОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ В АТСН

Процедура цифровой обработки изображений в АТСН обычно включает два этапа:

- этап предварительной обработки массива $[E_{ij}]$;
- собственно целевой этап решения задачи обнаружения, распознавания или измерения параметров объектов наблюдения.

Такое разделение носит условный характер, так как одни и те же математические операции могут быть использованы на обоих этапах. Однако, обычно предварительная обработка изображений преследует, как минимум, одну из следующих целей:

- сокращение объёма информации, предназначенной для дальнейшей обработки;
- фильтрацию помех, затрудняющих последующую обработку;
- выделение признаков контролируемых объектов для упрощения дальнейшей процедуры распознавания.

Такие цели могут быть достигнуты следующими способами:

- путём перехода от кодирования сигналов всех элементов изображения к кодированию сигналов центральных точек «окна» из $n \times n$ элементов с последующей интерполяцией сигналов остальных элементов «окна» в соответствии с заданной функцией изменения освещенности;
- уменьшением числа уровней квантования видеосигнала (в предельном случае до 2-х) с учётом информативности отдельных участков или изображения в целом;
- использованием дифференциальных алгоритмов выделения контуров в изображении объекта;
- использованием интегральных алгоритмов обобщенного описания изображения;
- применением методов трансформирования двумерных массивов исходных изображений в двумерные массивы коэффициентов на основе ортогональных преобразований и др.

Рассмотрим некоторые из указанных способов предварительной обработки изображений.

2.1. Методы фильтрации, позволяющие уменьшить уровень помех на предварительном этапе обработки изображений

Простой пороговый метод нелинейной фильтрации импульсных помех.

Сигнал от каждого из элементов массива $[E_{ij}]$ анализируемого изображения сравнивается со средним значением сигнала для небольшой группы $m \times n$ в окрестностях данного элемента

$$\bar{E}_{ij} = \frac{\sum_{k=-(m-1)/2}^{(m-1)/2} \sum_{l=-(n-1)/2}^{(n-1)/2} E_{(i+k),(j+l)}}{m \cdot n} . \quad (2.1)$$

Здесь m и n – нечётные числа. Если значение сигнала E_{ij} превышает \bar{E}_{ij} на заданную пороговую величину ξ , то значение сигнала заменяется на среднее значение сигнала группы. Если – нет, то сохраняется прежнее значение. Таким образом, формируется новый массив

$$E_{ij}^* = \begin{cases} \bar{E}_{ij}, & \text{при: } E_{ij} \geq \bar{E}_{ij} + \xi \\ E_{ij}, & \text{при: } E_{ij} < \bar{E}_{ij} + \xi \end{cases} , \quad (2.2)$$

в котором устранены одиночные «выбросы», т.е. помехи, некоррелированные с соседними элементами.

Анизотропная фильтрация относится к категории линейных процедур цифровой обработки массива $[E_{ij}]$. Это более совершенный, теоретически более обоснованный, но вместе с тем и более сложный алгоритм пространственной фильтрации. Он заключается в выполнении операции свёртки исходного массива изображения формата $M \times N$ со скользящим сглаживающим массивом $[W]$ меньшего формата $m \times n$ (ядро свёртки). Иначе говоря, реализуется процедура программного сканирования исходного изображения скользящей апертурой меньшего формата и вычисление свёртки на каждом шаге сканирования. В результате формируется новый массив $[E_{ij}^*]$, имеющий (также как и исходный) формат $M \times N$, но представляющий собой «сглаженное» изображение, подвергнутое низкочастотной пространственной фильтрации

$$E_{ij}^* = \sum_{k=-(m-1)/2}^{(m-1)/2} \sum_{l=-(n-1)/2}^{(n-1)/2} E_{(i+k),(j+l)} \cdot W_{k,l} . \quad (2.3)$$

Каким образом следует выбирать размерность сглаживающего массива $m \times n$? Размерность массива, очевидно, должна быть согласована с интервалом пространственной корреляции исходного изображения.

Естественно, что для изображений с плавным изменением освещённости от элемента к элементу, т.е. изображений, не содержащих мелкие детали в своей структуре, целесообразно увеличивать размер сглаживающей матрицы. При этом повышается эффективность фильтрации с точки зрения подавления импульсных помех. Однако, следует иметь ввиду, что время обработки исходного изображения при программной реализации алгоритма растёт пропорционально m^2 . А поскольку в АТСН, работающих в реальном масштабе времени, параметр быстродействия системы часто является одним из важнейших, то размерность массива $m \times n$ нужно выбирать и с учётом допустимых временных затрат на вычисление двумерной свёртки.

Ниже даны примеры различных видов сглаживающих массивов, рекомендуемых с учётом свойств оптических систем.

$$[W_1] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}; [W_2] = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; [W_3] = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Заметим, что нормирующие множители $1/9$, $1/10$, $1/16$ (значения которых обратно пропорциональны соответствующим суммам весовых коэффициентов сглаживающих матриц) позволяют сохранить масштаб чисел, представляемых двоичным кодом после процедуры фильтрации.

Более рациональным, с точки зрения использования оперативной памяти ЦВУ, является **рекуррентный алгоритм анизотропной фильтрации**. Математическая форма записи рекуррентного алгоритма практически не отличается от формы записи рассмотренного выше алгоритма анизотропной фильтрации

$$E_{ij}^* = \sum_{k=-(m-1)/2}^{(m-1)/2} \sum_{l=-(n-1)/2}^{(n-1)/2} E_{(i+k),(j+l)}^* \cdot W_{k,l} \quad (2.4)$$

Однако, здесь при вычислении дискретной свертки могут использоваться как элементы массива $[E_{ij}]$, так и элементы, взятые из уже сглаженного массива E_{ij}^* . Благодаря этому, одни и те же ячейки оперативной памяти могут использоваться вначале для хранения исходных значений E_{ij} , а затем новых значений E_{ij}^* , полученных после обработки данных. Если сканирование исходного массива $[E_{ij}]$ осуществляется, например, слева-направо и сверху-вниз, то в пределах программного окна форматом $m \times n$ используются элементы из нового массива $[E_{ij}^*]$ для всех $k = -(m-1)/2; \dots; -1, 0$ и $l = -(n-1)/2; \dots; -1$, а для всех остальных – из исходного массива $[E_{ij}]$.

Можно добавить, что благодаря вторичному использованию при

вычислении свертки элементов, уже прошедших процедуру фильтрации, рекуррентный алгоритм анизотропной фильтрации может давать лучшие результаты сглаживания, чем алгоритм, рассмотренный выше (ф. 2.3).

Медианная фильтрация осуществляется посредством замены значения каждого элемента массива $[E_{ij}]$, находящегося в центре окна скользящей апертуры, медианой исходных значений, находящихся внутри апертуры. В результате такой обработки на выходе медианного фильтра получается как бы сглаженное изображение, в котором отсутствуют малоразмерные, например, точечные детали, занимающие малое (по сравнению с размерами апертуры) число элементов.

Приведём несколько определений.

Медианой (med) последовательности X_1, X_2, \dots, X_n при нечётном n является средний член ряда, получающегося при упорядочении последовательности по возрастанию (или по убыванию). При чётном n медиана определяется как среднее арифметическое двух средних членов упорядоченного ряда.

Одномерный медианный фильтр с апертурой n для одномерной последовательности $\{X_i; i \in Z\}$ при нечётном n определяется как

$$Y_i = \text{med} (X_{i-v}, \dots, X_i, \dots, X_{i+v}), \quad i \in Z, \quad (2.5)$$

где: $v = (n - 1)/2$; Z – множество всех натуральных чисел.

Двумерный медианный фильтр с апертурой L для двумерного массива $\{E_{i,j}; (i, j) \in Z^2\}$ определяется как

$$E_{i,j}^* = \text{med} \{E_{i+r, j+s}; (r, s) \in L\}; \quad i, j \in Z^2. \quad (2.6)$$

На практике могут использоваться различные формы сглаживающих апертур L : линейные сегменты, кресты, квадраты и др. (рис. 2.1).

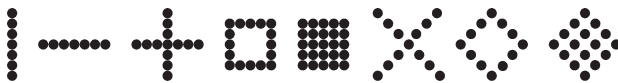


Рис. 2. 1. Примеры апертур медианных фильтров.

Оптимальный выбор формы сглаживающей апертуры зависит от специфики решаемой задачи и формы объектов, находящихся в поле зрения АТСН.

Примечание. Вблизи граничных точек конечных изображений нужно находить медиану только для тех точек, которые вписываются в пределы апертуры. Таким образом, при обработке точек изображения вблизи границ кадра медианы могут определяться для меньшего, чем в L (четного или нечётного) числа точек.

Следует отметить, что алгоритм медианной фильтрация обладает явно выраженной избирательностью по отношению к элементам массива, представляющим собой немонотонную составляющую последовательности чисел в пределах апертуры. В то же время на монотонную составляющую последовательности медианный фильтр не действует, оставляя её без изменений. Благодаря этой особенности, медианные фильтры при оптимально выбранной апертуре могут, например, сохранять без искажений резкие границы объектов, эффективно подавляя некоррелированные или слабо коррелированные помехи и малоразмерные детали. В то же время при аналогичных условиях алгоритм линейной анизотропной фильтрации, осуществляя сглаживание помех, неизбежно «смазывает» резкие границы и контуры объектов. В качестве иллюстрации на рис. 2.2 показан результат обработки медианным и анизотропным фильтрами фрагмента выделенной строки, содержащей малоразмерные детали и резкие границы более крупных объектов.

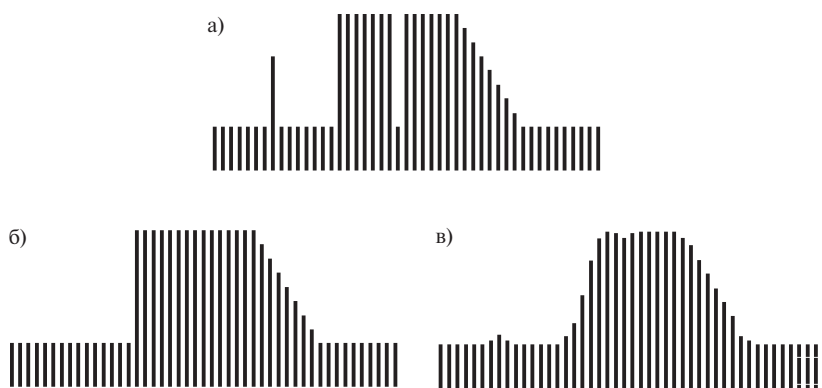


Рис. 2.2. Фрагмент строки исходного массива (а); результаты обработки фрагмента медианным (б) и анизотропным фильтрами (в).

2.2. Алгоритмы определения интегральных параметров дискретных изображений

В некоторых случаях на этапе предварительной обработки изображений целесообразно определение некоторых обобщённых (интегральных) признаков или свойств изображений, которые в сочетании с другими признаками (или самостоятельно) могут быть использованы на последующем этапе распознавания объектов, попавших в поле зрения АТСН.

Гистограмма отсчётов освещённости – зависимость числа одинаковых значений освещённости в анализируемом дискретном изображении, соответствующих какому-либо уровню квантования сигнала, от этого уровня

$$S(h) = \sum_{i=1}^M \sum_{j=1}^N K_{i,j}; \quad K_{i,j} = \begin{cases} 1, E_{i,j} = h \\ 0, E_{i,j} \neq h \end{cases} \quad (2.7)$$

где: $h = 0, 1, 2, \dots, (H - 1)$; H – число уровней квантования видеосигнала.

Средняя освещённость в изображении

$$\bar{E} = \sum_{h=0}^{H-1} h \cdot S(h); \quad h = 0, 1, 2, \dots, H-1; \quad H=2^n, \quad (2.8)$$

где n – число разрядов двоичного кода (число разрядов АЦП).

Среднеквадратическое отклонение освещённости от среднего значения

$$\sigma E = \sum_{h=0}^{H-1} (h - \bar{E})^2 \cdot S(h) \quad (2.9)$$

2.3. Дифференциальные алгоритмы обработки изображений

Дифференциальные алгоритмы используются для определения нормы градиента в точках изображения с целью последующего выделения границ и контуров объектов наблюдения.

Примечание. Строго говоря, термин «дифференциальные алгоритмы обработки» не является вполне корректным применительно к дискретным последовательностям. Более точно рассматриваемые ниже алгоритмы следовало бы называть разностными. Однако, в литературе термин «дифференциальные алгоритмы обработки» традиционно используется, подчёркивая некоторую аналогию с алгоритмами обработки непрерывных аналоговых сигналов, когда поиск градиентов осуществляется путём вычисления производных.

Применительно к дискретным изображениям, с которыми приходится иметь дело при цифровой обработке, могут быть предложены следующие способы приближённого вычисления нормы градиента, отличающиеся друг от друга количеством исходных отсчётов освещённости, используемых на каждом шаге, и объёмом вычислительных затрат.

2.3.1. Простейший алгоритм вычисления нормы градиента

$$G_{i,j} = \sqrt{(E_{(i+1),j} - E_{i,j})^2 + (E_{i,(j+1)} - E_{i,j})^2}, \quad (2.10)$$

где: $G_{i,j}$ – норма градиента для элемента матрицы $[E_{i,j}]$.

Здесь при анализе каждой точки исходного изображения используются три значения сигнала, соответствующие смежным точкам. Отметим, что при незначительном увеличении погрешности вычислений, нормы градиента могут определяться по упрощённой формуле алгоритма

$$G_{ij} = |E_{(i+1),j} - E_{i,j}| + |E_{i,(j+1)} - E_{i,j}|, \quad (2.10a)$$

2.3.2. Меньшую погрешность даёт, так называемый, **оператор Робертса**, благодаря тому, что на каждом шаге вычислений используются четыре исходных значения сигнала

$$G_{ij} = \sqrt{(E_{(i+1),(j+1)} - E_{i,j})^2 + (E_{(i+1),j} - E_{i,(j+1)})^2} \quad (2.11)$$

или

$$G_{ij} = |E_{(i+1),(j+1)} - E_{i,j}| + |E_{(i+1),j} - E_{i,(j+1)}|. \quad (2.11a)$$

2.3.3. Вычислительный **алгоритм Собела** предполагает использование восьми отсчётов освещённости в окрестностях анализируемой точки, однако значение освещённости в самой анализируемой точке в вычислениях не участвует

$$G_{ij} = \sqrt{G_{i,j(x)}^2 + G_{i,j(y)}^2} \quad (2.12)$$

или

$$G_{ij} = |G_{i,j(x)} + G_{i,j(y)}|, \quad (2.12a)$$

где:

$$G_{i,j(x)} = [E_{(i-1),(j-1)} + 2E_{(i-1),j} + E_{(i-1),(j+1)}] - [E_{(i+1),(j-1)} + 2E_{(i+1),j} + E_{(i+1),(j+1)}];$$

$$G_{i,j(y)} = [E_{(i-1),(j-1)} + 2E_{i,(j-1)} + E_{(i+1),(j-1)}] - [E_{(i-1),(j+1)} + 2E_{i,(j+1)} + E_{(i+1),(j+1)}].$$

Такой алгоритм наряду с более точным определением нормы градиента позволяет, в принципе, определять и направление вектора градиента в плоскости анализа изображения

$$\alpha = \arctg (G_{i,j(y)} / G_{i,j(x)}), \quad (2.13)$$

где α – угол между направлением вектора градиента и направлением строк матрицы $[E_{ij}]$.

2.3.4. В тех случаях, когда требуется максимальная точность в определении нормы градиента, может быть рекомендован **многошаговый метод вычислений**

$$\vec{G}_{ij} = \sum_{k=1}^4 \vec{G}(k), \quad (2.14)$$

где: $G(k)$ – скалярное произведение векторов $\vec{a}_{(k)}$ и $\vec{b}_{(k)}$;

$$\vec{a}_1 = [1; \sqrt{2}; 1; 0; 0; -1; -\sqrt{2}; -1]$$

$$\vec{a}_2 = [1; 0; -1; \sqrt{2}; -\sqrt{2}; 1; 0; -1]$$

$$\vec{a}_3 = [0; -1; -\sqrt{2}; 1; -1; \sqrt{2}; 1; 0]$$

$$\vec{a}_4 = [-\sqrt{2}; -1; 0; -1; 1; 0; 1; \sqrt{2}]$$

$$b = [E_{(i-1),(j-1)}; E_{i,(j-1)}; E_{(i+1),(j-1)}; E_{(i-1),j}; E_{(i+1),j}; E_{(i-1),(j+1)}; E_{i,(j+1)}; E_{(i+1),(j+1)}]$$

При проектировании АТСН выбор конкретного алгоритма следует делать с учётом допустимых временных затрат, определяемых техническим заданием.

2.3.5. Выделение границ контролируемых объектов.

В общем случае процедуре выделения границ (контуров) изображений объектов должен предшествовать этап определения нормы градиента для всех точек исходного массива G_{ij} . После этого границы объекта могут быть найдены следующим образом.

В качестве первого этапа осуществляется выбор координат точек изображения, для которых значения нормы градиента превышают установленный порог

$$ij \in \omega_{\text{гр}}, \text{ если } G_{ij} \geq D, \quad (2.15)$$

где: $ij \in \omega_{\text{гр}}$ – множество координат точек, принадлежащих области изображения вблизи границ объекта; D – пороговое значение нормы градиента.

Примечание. Следует, однако, заметить, что решающего правила (2.15), обычно, недостаточно для успешного выделения контуров объекта. Дело в том, что при низком уровне порога D кроме контурных точек могут оказаться выделенными и другие, «лишние» точки, расположенные вблизи контура, для которых также выполняется условие $G_{ij} \geq D$. Если же величина порога D задана высокой, то, наоборот, не все точки контура окажутся выделенными. Изменяя величину D , можно, в принципе, менять соотношение между вероятностью выделения «лишних» точек (ошибки первого рода) и вероятностью пропуска контурных точек объекта (ошибки второго рода). Очевидно, что чем выше уровень помех, действующих в АТСН, тем больше суммарная вероятность ошибок выделения контурных точек. На практике пороговое значение нормы градиента предпочтительнее задавать таким, чтобы вероятность ошибок второго рода была минимальной. При этом «лишние» точки, выделенные на первом этапе, могут быть «отсеяны» на втором этапе, который предполагает дополнительный анализ выделенных точек.

В результате дополнительного анализа в окрестностях каждой из точек с координатами $ij \in \omega_{\text{гр}}$ должны быть оставлены только две, непосредственно прилегающие к данной точке. Эти две точки могут быть выбраны среди других предполагаемых контурных точек (например, точек с координатами $i \pm 1, j \pm 1$) по признаку максимального значения нормы градиента. В крайнем случае, если этому при-

знаку удовлетворяют более двух прилегающих точек, должны быть выбраны две любые точки, например, первые из числа рассматриваемых. Это, хотя и может в конечном итоге привести к незначительным погрешностям в определении координат контурных точек, но позволит избежать более существенных аномальных ошибок, связанных с искажением формы и с очень значительными погрешностями при вычислении периметра объекта.

В частном случае, при обработке бинарных изображений, то есть изображений, каждый элемент которых может принимать одно из двух значений «0» или «1» («чёрное» или «белое»), процедура выделения границ объектов существенно упрощается и может быть сведена к простым логическим операциям

$$i, j \in \omega_{\text{гр.}} : [E_{i,j} = a] \& [E_{i,(j-1)} = b \vee E_{(i-1),j} = b \vee E_{i,(j+1)} = b \vee E_{(i+1),j} = b] \quad (2.16)$$

или

$$i, j \in \omega_{\text{гр.}} : [E_{i,j} = a] \& [E_{(i-1),(j-1)} = b \vee E_{(i-1),(j+1)} = b \vee E_{(i+1),(j-1)} = b \vee E_{(i+1),(j+1)} = b]. \quad (2.17)$$

Здесь $i, j \in \omega_{\text{гр.}}$ — координаты точек, принадлежащих границам объектов; a, b — возможные значения функции $E_{i,j}$ («0» или «1»); $\&$ и \vee — символы логических операций «И» и «ИЛИ» соответственно.

2.4. Алгоритмы трансформирования исходных изображений на основе ортогональных преобразований

В некоторых случаях, для сокращения объёма данных или облегчения процедуры выделения признаков объектов на последующих этапах распознавания, целесообразно предварительно преобразовывать исходный двумерный массив $[E_{i,j}]$ в массив значений коэффициентов $[F_{u,v}]$, имеющий такой же формат $M \times N$, как и исходное изображение.

Вторичный массив или иначе матрица коэффициентов $[F_{u,v}]$ называется **трансформантой**. Один из видов ортогональных преобразований — дискретное преобразование Фурье. В случае преобразования Фурье трансформанта является ничем иным, как двумерным пространственным спектром изображения.

В общем случае любое преобразование исходного изображения на основе ортогональных операторов можно рассматривать как операцию разложения изображения в обобщенный двумерный спектр, а коэффициенты (т.е. элементы трансформанты) — как амплитуды соответствующих спектральных составляющих. Отметим, что если при этом в качестве базисных функций используются негармонические функции, то понятие **пространственной частоты** следует обобщить и использовать понятие **секвенты**.

Секвентой называется величина, равная половине среднего числа пересечений нуля в единицу времени или на единицу длины.

В процессе ортогональных преобразований изображения, имеющего сильные корреляционные связи между соседними элементами, происходит декорреляция (отбеливание). Таким образом, значения элементов трансформанты оказываются практически некоррелированными. В отличие от исходного массива, для которого характерно в среднем равномерное распределение энергии сигнала между элементами, распределение энергии сигнала в трансформанте крайне неравномерно. Основная доля энергии приходится на элементы с малыми порядковыми номерами (т.е. на низкие пространственные секвенты) и лишь небольшая доля – на прочие (см. рис 2. 3).

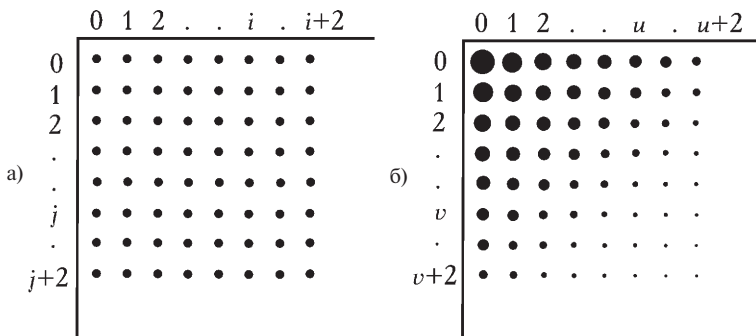


Рис. 2. 3. Распределение энергии сигнала между отдельными элементами в исходном массиве (а) и в трансформанте (б).

Это обстоятельство позволяет либо вообще отбросить (т.е. считать равными нулю) большую часть элементов трансформанты (что означает, по существу, низкочастотную пространственную фильтрацию), либо квантовать их на малое число уровней с использованием минимального числа разрядов двоичного кода.

Рассмотрим некоторые наиболее распространённые виды ортогональных преобразований, применяемых при цифровой обработке изображений.

Дискретное преобразование Фурье

$$F_{u,v} = \frac{1}{M \cdot N} \sum_{i=1}^M \sum_{j=1}^N E_{i,j} \cdot \exp\{-j2\pi[u_i/M + v_j/N]\}. \quad (2.18)$$

Здесь коэффициенты $F_{u,v}$ в общем случае являются комплексными числами

$$F_{u,v} = A_{u,v} + jB_{u,v} \quad (2.19)$$

Каждый комплексный коэффициент можно заменить двумя действительными составляющими. Эти составляющие характеризуют, соответственно, пространственные дискретные спектры амплитуд и фаз и определяются следующим образом:

$$\begin{aligned} |F_{u,v}| &= [B_{u,v}^2 + A_{u,v}^2]^{1/2}; \\ \arg F_{u,v} &= \arctg [B_{u,v} / A_{u,v}]. \end{aligned} \quad (2.20)$$

Основной недостаток дискретного преобразования Фурье — сравнительно большой объём вычислений, а также необходимость сохранения большого числа составляющих трансформанты по сравнению с другими ортогональными преобразованиями при одинаковых ошибках восстановления изображения (т.е. при одинаковых потерях информации). Кроме того, для хранения отдельных составляющих комплексных коэффициентов, требуется больший объём памяти, чем для действительных значений элементов исходного массива. Говоря о дискретном преобразовании Фурье, следует упомянуть о возможности применения специально разработанных алгоритмов *быстрого преобразования Фурье* [1], а также о специализированных вычислительных устройствах для их реализации — так называемых *систолических процессорах* [2].

Преобразование Уолша (при $M = N$)

$$U_{u,v} = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E_{i,j} \cdot \prod_{k=1}^l (-1)^a, \quad (2.21)$$

где $a = b_k(i) \cdot b_{l-k}(u) + b_k(j) \cdot b_{l-k}(v)$.

В свою очередь, коэффициенты $b_k(Z)$ определяются следующим образом: $b_k(Z)$ равен значению k -того разряда двоичного кода числа Z , состоящего из l двоичных разрядов. Если, например, $Z = 10$, т.е. $10_{10} = 1010_2$, то $b_0 = 0$; $b_1 = 1$; $b_2 = 0$; $b_3 = 1$.

Преобразование Адамара (при $M = N$)

$$A_{u,v} = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N [E_{i,j} (-1)^a], \quad (2.22)$$

где $a = \sum_{k=1}^l [b_k(i) \cdot b_k(u) + b_k(j) \cdot b_k(v)]$,

b_k — определяются в соответствии с правилом их определения в преобразовании Уолша.

Очевидно, что *все виды ортогональных преобразований являются обратимыми, т.е., используя процедуру обратного преобразования, можно из трансформанты восстановить исходное изображение.*

Пусть $[E_{i,j}]$ – массив исходного изображения форматом $N \times N$, где j – номер строки, i – номер столбца элементов (номер элементов в строке); $[F_{u,v}]$ – трансформанта изображения, которая имеет тот же формат $N \times N$, где u и v соответственно номер строки и номер столбца элементов трансформанты. Тогда, в общем случае, независимо от вида ортогонального преобразования, запишем

$$\begin{aligned} F_{u,v} &= \sum_{i=1}^N \sum_{j=1}^N [E_{i,j} \cdot a(i,j,u,v)]; \\ E_{i,j} &= \sum_{u=1}^N \sum_{v=1}^N [F_{u,v} \cdot b(i,j,u,v)] , \end{aligned} \quad (2.23)$$

где $a(i,j,u,v)$ и $b(i,j,u,v)$ – базисные функции прямого и обратного преобразований соответственно.

С практической точки зрения важно отметить, что все рассмотренные выше виды ортогональных преобразований являются разделимыми по переменным. Таким образом, вычисление прямых и обратных двумерных ортогональных преобразований удаётся свести к последовательному выполнению одномерных преобразований

$$\begin{aligned} F_{u,v} &= \sum_{i=1}^N a_{\text{стр.}}(i,u) \cdot \sum_{j=1}^N E_{i,j} \cdot a_3(j,v); \\ E_{i,j} &= \sum_{u=1}^N b_{\text{стр.}}(i,u) \cdot \sum_{v=1}^N F_{u,v} \cdot b_3(j,v). \end{aligned} \quad (2.24)$$

Здесь $a_{\text{стр.}}(i,u)$, $b_{\text{стр.}}(i,u)$ и $a_3(j,v)$, $b_3(j,v)$ – базисные функции прямого и обратного преобразований, соответственно вдоль направления строк и столбцов.

Для удобства записи и вычислений целесообразно использовать матричный аппарат

$$[F] = [A_3] \cdot [E] \cdot [A_{\text{стр.}}]^T; \quad [E] = [B_3] \cdot [F] \cdot [B_{\text{стр.}}]^T. \quad (2.25)$$

Здесь $[A_3]$ и $[A_{\text{стр.}}]$ – матрицы прямого преобразования; $[B_3]$ и $[B_{\text{стр.}}]$ – матрицы обратного преобразования; $[A_{\text{стр.}}]^T$ и $[B_{\text{стр.}}]^T$ – матрицы, полученные в результате транспонирования матриц $[A_{\text{стр.}}]$ и $[B_{\text{стр.}}]$.

Разумеется, независимо от формы математического представления, прямое и обратное ортогональные преобразования двумерных массивов требуют, в общем случае, значительных вычислительных затрат. Это следует учитывать при проектировании

АТСН, работающих в реальном масштабе времени. Однако, при цифровой обработке бинарных изображений, процедуры ортогональных преобразований существенно упрощаются, особенно в случае использования бинарных базисных функций (преобразования Уолша, Адамара и др.).

2.5. Алгоритмы выделения признаков контролируемых объектов

Выделение признаков позволяет иногда упростить реализацию последующего этапа — этапа распознавания или идентификации объектов. Путём выделения признаков удаётся создать сжатое описание объекта в выбранной системе признаков.

При выборе наиболее информативных признаков необходимо учитывать как свойства самих объектов, так и возможности телевизионных датчиков — первичных формирователей сигнала изображения с точки зрения их разрешающей способности. Необходимо также принимать во внимание степень сложности процедуры выделения признаков за ограниченное время анализа.

Примечание. Далее пойдёт речь об обработке монохромных (не цветных) изображений. Следует заметить, что цветные изображения в большинстве случаев после соответствующего декодирования сигналов можно представить совокупностью трёх отдельных монохромных составляющих, каждую из которых можно подвергнуть обработке в соответствии с ниже рассматриваемыми алгоритмами.

В автоматизированных телевизионных системах наблюдения наиболее предпочтительными являются геометрические признаки объектов:

- площадь и периметр изображения объекта;
- число отверстий в теле объекта;
- размеры вписанных и описанных простейших геометрических фигур (окружностей, прямоугольников, треугольников и др.);
- число и взаимное расположение углов;
- моменты инерции изображений объектов.

Важной особенностью большинства геометрических признаков является их инвариантность относительно разворота изображения объекта. Кроме того, путём нормирования геометрических признаков друг относительно друга, достигается инвариантность относительно масштаба изображения объекта.

Определение площади и периметра.

Площадь изображения объекта вычисляется путём простого подсчёта числа элементов, относящихся к объекту

$$A = \sum_{i=1}^M \sum_{j=1}^N S_{i,j}; \quad S_{i,j} = \begin{cases} 1, & (i,j) \in L \\ 0, & (i,j) \notin L, \end{cases} \quad (2.26)$$

где L – множество координат массива $[E_{i,j}]$, принадлежащих объекту.

Периметр изображения объекта вычисляется после того, как на предварительном этапе выделены границы объекта (см. п. 2.3.5)

$$P = \sum_{i=1}^M \sum_{j=1}^N K_{i,j}; \quad K_{i,j} = \begin{cases} 1, & (i,j) \in a_{\text{гр}} \\ 0, & (i,j) \notin a_{\text{гр}} \end{cases}, \quad (2.27)$$

где $a_{\text{гр}}$ – множество граничных (контурных) точек изображения объекта.

На основе выделенных признаков можно сформировать обобщенный нормированный признак, инвариантный к масштабу изображения

$$U = A/P^2 \text{ или } V = P/A^{1/2}. \quad (2.28)$$

Определение радиусов вписанных и описанных окружностей.

Процедура складывается из двух этапов (рис 2.4).

1. Определение координат геометрического центра изображения объекта

$$X_{\text{цр}} = \frac{\sum_{i=1}^M \sum_{j=1}^N S_{i,j} \cdot x_{i,j}}{\sum_{i=1}^M \sum_{j=1}^N S_{i,j}}; \quad Y_{\text{цр}} = \frac{\sum_{i=1}^M \sum_{j=1}^N S_{i,j} \cdot y_{i,j}}{\sum_{i=1}^M \sum_{j=1}^N S_{i,j}} \quad (2.29a)$$

где $x_{i,j}$; $y_{i,j}$ – координаты точек изображения объекта, которые могут быть заменены соответствующими номерами столбцов и строк, содержащих данный элемент $x_{i,j} = i$; $y_{i,j} = j$.

2. Вычисление минимального и максимального расстояний от центра до границ изображения объекта, выделенных на предварительном этапе (см. раздел 2.3)

$$r_{i,j} = \sqrt{(x_{i,j} - X_{\text{цр}})^2 + (y_{i,j} - Y_{\text{цр}})^2} \quad (2.29б)$$

$$R_{\text{max}} = r_{i,j(\text{max})}; \quad R_{\text{min}} = r_{i,j(\text{min})}, \text{ где: } i,j \in a_{\text{гр}}$$

Очевидно, что нормированный признак $R' = R_{\text{max}} / R_{\text{min}}$ всегда является инвариантным к масштабу изображения объекта.

Определение сторон описанного прямоугольника.

Это – одна из простейших процедур.

1. Надо определить максимальные и минимальные значения абсцисс и ординат изображения объекта i_{max} и i_{min} ; j_{max} и j_{min} .

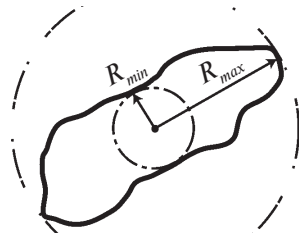


Рис. 2. 4. Определение радиусов описанной и вписанной окружностей.

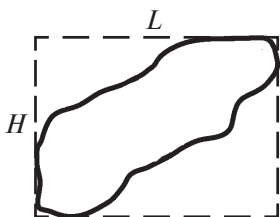


Рис. 2. 5. Определение сторон описанного прямоугольника.

2. Высота и основание прямоугольника определяются следующим образом

$$L = i_{max} - i_{min}; H = j_{max} - j_{min}. \quad (2.30)$$

Отметим, что данный признак (в отличие от предыдущих) не является инвариантным к развороту изображения объекта (рис. 2.5).

Определение числа и взаимного положения углов

На предварительном этапе должны быть выделены и пронумерованы элементы контура объекта.

Классический способ определения угловых точек изображения объекта заключается в анализе небольшого фрагмента контура в окрестностях данной точки и в определении радиуса её кривизны. Если этот радиус окажется меньше установленного порога – это угловой элемент, в противном случае – нет. Однако, такой способ связан с очень большим объёмом вычислений. С практической точки зрения в быстродействующих АТСН, работающих в реальном масштабе времени, предпочтительным представляется более простой алгоритм. Он заключается в оценке расстояний между начальной и конечной точками фрагмента контура, т.е. между элементами контура с порядковыми номерами $k - 2$ и $k + 2$ (рис 2.6).

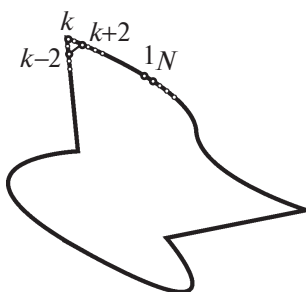


Рис. 2. 6. Определение числа и взаимного положения углов.

Пусть $x(k)$ и $y(k)$ абсцисса и ордината контурных элементов соответственно. Тогда решающее правило может выглядеть следующим образом

$$\{|x(k-2) - x(k+2)| + |y(k-2) - y(k+2)| \leq H\}. \quad (2.31)$$

Если условие (2.31) выполняется, тогда данная точка контура принадлежит множеству угловых точек L . Здесь H – пороговое значение, выбираемое с учётом свойств изображения объектов данного класса.

При реализации вычислительной процедуры необходимо соблюдать следующие правила.

1. Если, в соответствии с условием (2.31), оказываются выделенными несколько смежных элементов контура, то решающее правило должно предусматривать выбор только одного элемента в качестве

углового, например, по минимуму значения модуля разности, а в случае совпадения значений – любой из этих элементов. Это, разумеется, может привести к некоторой ошибке в определении координат углового элемента, но позволит избежать более существенной (аномальной) ошибки, связанной с неправильным определением числа углов и, следовательно, формы объекта.

2. Выделенным угловым элементам целесообразно присваивать порядковые номера, которые могут быть использованы на последующем этапе распознавания и определения ориентации объекта.

3. Процедуру анализа контурных элементов удобно осуществлять в цикле, однако два первых и два последних элемента приходится осуществлять вне цикла, так как для них не удаётся задать значения переменной $k + 2$ и $k - 2$.

Определение моментов инерции изображения объекта

Примечание.

Термин «моменты инерции изображения объекта» здесь, разумеется, не имеет отношения к механике. Его использование оправдано в том смысле, что для вычисления указанного признака используются математические выражения, аналогичные тем, что и при вычислении механических моментов инерции материального тела, если вместо значений масс отдельных точек тела подставлять значение освещенностей в соответствующих точках его изображения.

Моменты инерции являются довольно информационными признаками для последующего этапа распознавания образов, но их определение является не такой уж простой задачей. Вместе с тем, в некоторых случаях могут использоваться промежуточные результаты вычислений, например, для определения угловой ориентации изображения объекта относительно приборной системы координат (см. раздел 3.2).

Обозначим главные искомые моменты инерции изображения объекта через J_1 и J_2 (рис. 2.7б). Однако, чтобы найти J_1 и J_2 , необходимо предварительно определить так называемые *промежуточные моменты* J_x и J_y , т.е. моменты инерции относительно вертикальной и горизонтальной осей приборной системы координат, а также *смешанный момент* $J_{x,y}$ (рис. 2.7 а).

Вычисление осуществляется в следующем порядке.

1. Определяются координаты центра «тяжести» (энергетического центра) изображения объекта

$$X_{цэ} = \frac{\sum_{i=1}^M \sum_{j=1}^N (E_{i,j} \cdot x_{i,j})}{\sum_{i=1}^M \sum_{j=1}^N E_{i,j}} ; \quad Y_{цэ} = \frac{\sum_{i=1}^M \sum_{j=1}^N (E_{i,j} \cdot y_{i,j})}{\sum_{i=1}^M \sum_{j=1}^N E_{i,j}} . \quad (2.32)$$

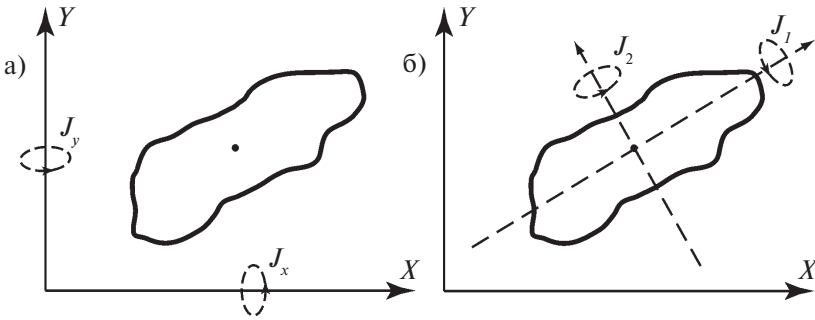


Рис. 2.7. Определение промежуточных (а) и главных (б) моментов инерции.

2. Определяются промежуточные моменты J_x, J_y, J_{xy}

$$\begin{aligned}
 J_x &= \sum_{i=1}^M \sum_{j=1}^N [(x_{i,j} - X_{цэ})^2 \cdot E_{i,j}]; & J_y &= \sum_{i=1}^M \sum_{j=1}^N [(y_{i,j} - Y_{цэ})^2 \cdot E_{i,j}]; \\
 J_{xy} &= \sum_{i=1}^M \sum_{j=1}^N [(x_{i,j} - X_{цэ}) \cdot (y_{i,j} - Y_{цэ}) \cdot E_{i,j}].
 \end{aligned}
 \tag{2.33}$$

3. Рассчитываются главные моменты

$$J_{1,2} = \frac{J_x + J_y}{2} \pm \sqrt{\frac{1}{4}(J_x - J_y)^2 + J_{xy}^2}.
 \tag{2.34}$$

Вопросы для самопроверки:

1. Какие задачи решаются на этапе предварительной обработки изображений? Какие методы могут использоваться для решения этих задач?
2. Чем следует руководствоваться при выборе оптимальной размерности и вида сглаживающего массива в случае применения алгоритма анизотропной фильтрации?
3. В чём отличие свойств алгоритмов медианной и анизотропной фильтрации?
4. Как решается задача автоматического выделения контурных элементов при обработке полутоновых и бинарных изображений объектов?
5. С какой целью могут использоваться алгоритмы трансформирования исходных изображений на основе ортогональных преобразований? Что общего и в чём отличия между дискретным преобразованием Фурье и другими видами ортогональных преобразований?
6. Какими свойствами должны обладать геометрические признаки изображений, которые используются для автоматического распознавания объектов? Назовите наиболее распространённые из них в порядке возрастания сложности их выделения.
7. Что такое промежуточные и главные моменты инерции изображения объекта?
8. Как определяются геометрический и энергетический центры изображения объекта?

3. МЕТОДЫ ОБНАРУЖЕНИЯ, РАСПОЗНАВАНИЯ И ИЗМЕРЕНИЯ ПАРАМЕТРОВ ОБЪЕКТОВ НАБЛЮДЕНИЯ

Подробное рассмотрение теоретических аспектов данной темы не является задачей настоящего пособия. Более полную информацию по этим вопросам желающие могут получить в других источниках (см. например, [3, 8, 9]). Ниже даются практические рекомендации по реализации отдельных методов, непосредственно связанных с наиболее типичными задачами, которые решаются автоматизированными телевизионными системами наблюдения на основе использования алгоритмов, рассмотренных в предыдущей главе.

3.1. Обнаружение объектов

В обобщённой форме операция обнаружения объектов, т.е. операция выявления образов объектов в искажённом шумами и помехами изображении, может быть определена в виде процедуры сравнения с некоторым числом – порогом другого числа, полученного в результате преобразования анализируемого изображения, представленного, в конечном счёте, в виде матрицы-массива целых чисел E_{ij} .

$$L[E_{ij}] \geq \Pi[E_{ij}]. \quad (3.1)$$

Здесь $L[*]$ – оператор преобразования исходного изображения; $\Pi[*]$ – оператор формирования порогового значения.

В случае выполнения условия (3.1) принимается решение о наличии объекта, в противном случае – об отсутствии его. При этом качество обнаружения характеризуется двумя параметрами:

– вероятностью правильного обнаружения, которая равна вероятности выполнения условия (3.1), при наличии объекта в анализируемом изображении;

– вероятностью ложной тревоги, равной вероятности выполнения условия (3.1) при отсутствии объекта в анализируемом изображении.

Конкретный вид операторов $L[*]$, $\Pi[*]$, а также качество обнаружения зависят от наличия априорных сведений об ожидаемых объектах, шумах помехах и искажениях. В общем случае научной основой для определения оптимальных параметров решающего правила является теория статистических решений [8, 9].

Здесь в качестве примера рассмотрим случай обнаружения малоразмерных (точечных) объектов, наблюдаемых АТСН на неоднородном по яркости фоне и при наличии, кроме того, маскирующих помех в виде аддитивного нормального шума, который является след-

ствием случайных флуктуаций, действующих в различных звеньях аппаратной структуры.

На рис. 3.1а условно показан фрагмент строки, содержащий как сигналы от точечных объектов, так и от участков изображения, соответствующих различным уровням яркости подстилающего фона.

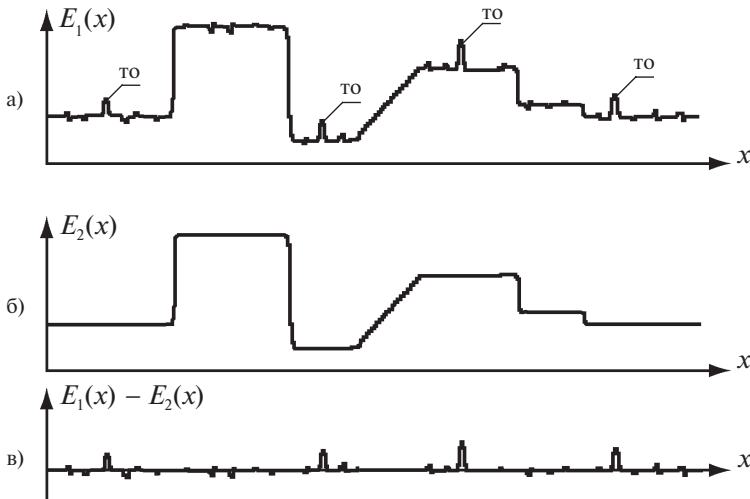


Рис. 3.1 Фрагмент строки, содержащей изображения точечных объектов (то) на неоднородном фоне при наличии шумов (а); тот же фрагмент после сглаживания медианным фильтром (б); разностный сигнал (в).

В данном случае точечные объекты занимают значительно меньшую площадь в зоне наблюдения, чем области участков различных уровней яркости фона. Применяв в качестве сглаживающего фильтра рассмотренный выше алгоритм медианной фильтрации (см. раздел 2.1), можно выделить сначала сигнал, соответствующей фоновой составляющей (рис. 3.1б), а затем, путем его вычитания из исходного сигнала (рис. 3.1а), сформировать разностный сигнал уже не содержащий фоновой составляющей (рис. 3.1в).

Заметим, что при наблюдении АТСН так называемого точечного объекта (т.е. объекта размерами которого можно пренебречь по сравнению с размерами поля зрения системы) форма изображения объекта однозначно определяется функцией пятна рассеяния объектива, которую можно считать известной. Таким образом, дальнейшая обработка разностного сигнала фактически сводится к классической процедуре обнаружения сигнала известной формы на фоне аддитивных нормальных шумов с нулевым средним значением. В этом случае в качестве оператора преобразования исходного изображения выступает корреляционный интеграл, вычисляемый

с использованием заданного описания известного изображения объекта [8]. Тогда параметры, характеризующие качество обнаружения определяются достаточно просто.

Вероятность правильного обнаружения

$$P_{\text{прав.}} = \Phi[(L_c - \Pi_0)/\sigma_L]; \quad (3.2)$$

вероятность ложной тревоги

$$P_{\text{л.т.}} = 1 - \Phi[\Pi_0/\sigma_L]; \quad (3.3)$$

где: L_c – среднее значение корреляционного интеграла при наличии объекта; σ_L – среднеквадратическое отклонение корреляционного интеграла, обусловленное наличием помех; Π_0 – порог обнаружения;

$$\Phi[v] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^v e^{-\frac{x^2}{2}} dx \text{ – интеграл вероятностей.}$$

На практике для определения оптимальных размеров и формы сглаживающей апертуры медианного фильтра, а также для определения оптимальных параметров решающего правила при последующей обработке разностного сигнала бывает полезно воспользоваться методом математического компьютерного моделирования (см. ниже).

3.2. Идентификация и классификация объектов в АТСН

Идентификация заключается в сравнении изображения одного объекта со всеми эталонами заданного класса. По наилучшему совпадению выносится решение об объекте.

Классификация предполагает наличие изображений нескольких различных объектов. Путём сравнения этих изображений с одним из эталонов по наилучшему совпадению выбирается тот, который соответствует данному классу. Затем оставшиеся изображения сравниваются с другими эталонами и так далее, пока не будут исчерпаны все изображения объектов. Разумеется, на каждом шаге классификации должно выделяться изображение одного объекта, иначе задача становится практически не разрешимой.

3.2.1. Способ прямого сравнения изображения объекта с эталонным изображением.

Пусть $[E_{i,j}]$ – исходное изображение объекта; $[F_{i,j}]$ – эталонное изображение. Тогда алгоритм прямого сравнения имеет вид

$$T = \sum_{i=1}^M \sum_{j=1}^N [E_{i,j} - F_{i,j}]^2 \leq D, \quad (3.4)$$

где D – заданное пороговое различие.

Если указанное условие выполняется, то объект идентифицирован, если нет – надо перейти к следующему объекту (или эталону).

Такой способ чрезвычайно прост, особенно, если вычисление суммы квадратов разности заменить вычислением суммы модулей разности. Алгоритм может быть легко реализован и чисто аппаратными средствами за время, практически не превышающее время сканирования кадра телевизионным преобразователем.

Однако, при наличии в реальных условиях дестабилизирующих факторов надёжность такого способа невелика, вследствие интегрального характера алгоритма. Очевидно, что при большем значении порога D различные объекты могут удовлетворять условию (3.4), и, следовательно, могут возникнуть ошибки, связанные с неправильной идентификацией объекта (*ошибки первого рода*). При уменьшении D – наоборот, могут возникнуть ошибки типа пропуска объекта (*ошибки второго рода*). Регулируя величину D , можно лишь менять соотношение между вероятностями возникновения ошибок первого и второго рода в соответствии с заданным критерием оптимальности.

3.2.2. Корреляционный метод

Этот метод основан на вычислении взаимно-корреляционных сумм между объектом и всеми эталонами (или между имеющимися объектами и каждым из эталонов). Из множества альтернативных вариантов выбирается тот объект (или тот эталон), при котором получается максимальное значение взаимно-корреляционной суммы

$$K(k) = \sum_{i=1}^M \sum_{j=1}^N [E_{(k)i,j} \cdot F_{(k)i,j}], \quad (3.5)$$

где k – номер альтернативы.

Очевидно, что при $F_{i,j} = E_{i,j}$

$$K = K_{max} = \sum_{i=1}^N \sum_{j=1}^N [F_{i,j}^2]. \quad (3.6)$$

Удобно пользоваться нормированным значением взаимно-корреляционной суммы – коэффициентом корреляции

$$R(k) = K(k)/K_{max}. \quad (3.7)$$

Корреляционный метод более надёжен, однако он требует и значительно большего объёма вычислений, так как для каждой точки изображения требуется вычисление произведений $E_{i,j} \times F_{i,j}$. Но при обработке бинарных изображений вычисление произведений практически не связано с затратами времени, поскольку перемножение однобитовых чисел сводится к простой логической операции «И».

Следует иметь в виду, что оба рассмотренных выше метода требуют выполнения ряда условий: одинаковой ориентации изображений объекта и эталона, совмещения их по пространственным координатам и выдерживания одинаковых масштабов. Всё это может потребовать дополнительных вычислительных затрат. Наряду с этими недостатками укажем на необходимость хранения в памяти АТСН большого объема данных, особенно при многоальтернативном варианте решения задачи.

3.2.3. Методы распознавания, основанные на использовании системы признаков

В данном случае также используются эталоны объектов. Однако, в качестве непосредственных элементов сравнения выступают не элементы изображений объекта и эталона, а признаки объекта и эталона. Использование признаков в качестве элементов сравнения позволяет резко сократить объём данных, хранящихся в памяти системы, и время обработки информации по сравнению с вышерассмотренными методами.

Однако следует иметь в виду, что на практике в реальных условиях наблюдения выделение признаков объектов всегда осуществляется с некоторой погрешностью. Путём моделирования работы АТСН на стадии проектирования (или на этапе обучения-калибровки системы) следует выявить характер и степень возможного рассеяния оценок используемых признаков для каждого из ожидаемых объектов (рис. 3.2).

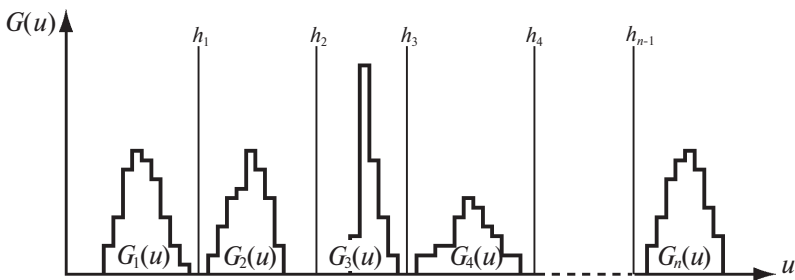


Рис. 3.2 $G_1(u), G_2(u), \dots, G_n(u)$ — гистограммы распределения признака u соответственно для 1, 2, ..., n -го объектов;
 h_1, h_2, \dots, h_{n-1} — границы диапазонов значения признака x , соответствующие областям опознавания объектов 1, 2, ..., n .

С учётом вида полученных гистограмм устанавливаются оптимальные границы диапазонов значений того или иного признака, которые используются при формировании решающего

правила. Очевидно, что подобные семейства гистограмм должны быть заблаговременно получены для каждого из рабочих признаков, используемых при идентификации объектов.

При большом числе возможных вариантов может быть рекомендован многоступенчатый (иерархический) алгоритм. При этом на каждой ступени распознавания используется какой-либо из признаков объекта (площадь, периметр, радиусы вписанных и описанных окружностей, моменты инерции, число и расположение углов и т.д.).

Пусть возможны K решений (K альтернатив). Тогда алгоритм распознавания можно представить в виде дерева (графа) (рис. 3.3).

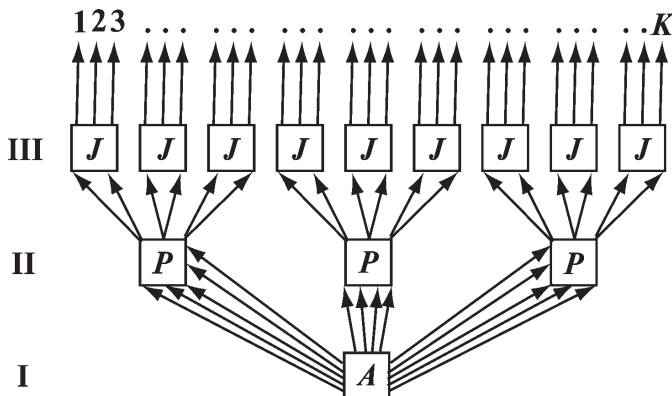


Рис. 3.3. Многоступенчатый алгоритм (граф) распознавания.

Здесь I, II, III, .. – уровни распознавания; A, P, J, .. – алгоритмы сравнения по площади, периметру, моментам инерции и т.д., включая выделение соответствующих признаков изображений объектов; 1, 2, 3, .. K – номера возможных решений.

Важно отметить, что наибольший эффект сокращения времени обработки информации достигается при рациональном распределении типов используемых признаков по уровням распознавания. Так, на нижних уровнях, когда приходится иметь дело с максимальным числом вариантов, следует привлекать признаки, не требующие больших вычислительных затрат на их определение (например, площади и периметры объектов), а наиболее информативные (такие, например, как моменты инерции) – применять на верхнем уровне, где число альтернатив минимально.

В разделе 4 мы более подробно познакомимся с подобным подходом на конкретном примере реализации алгоритма распознавания изображения участка звёздного неба в автономной системе астроориентации.

3.3. Об измерении параметров объектов наблюдения

Очевидно, **измерение размеров** объекта может осуществляться посредством выделения на этапе предварительной обработки соответствующих геометрических признаков, например, площади и периметра изображения, размеров вписанных и описанных геометрических фигур (см. раздел 2.5).

Измерение пространственного положения объекта сводится к измерению координат характерных точек его изображения в приборной системе координат. Такими точками могут, например, служить геометрический или энергетический центры изображения (ф. 2.29 и ф.2.32). В случае необходимости измерения координат точечных объектов в сложных условиях наблюдения при малых отношениях сигнал/шум эффективным может оказаться более сложный алгоритм интерполяции видеосигнала по методу наименьшего среднеквадратического отклонения, подробно описанный в разделе 4.1.

Как указывалось выше, иногда в задачах распознавания возникает необходимость компенсации взаимного разворота изображений объекта и эталона. При этом приходится решать вспомогательную задачу – **определение угла наклона** главной энергетической оси

изображения объекта в приборной системе координат (рис. 3.4).

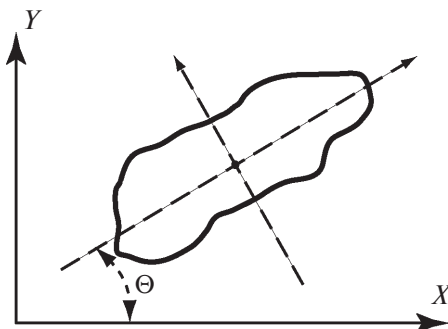


Рис. 3.4. Определение угла наклона изображения объекта.

$$\Theta = \frac{1}{2} \arctg \frac{2J_{xy}}{J_x - J_y}, \quad (3.8)$$

где J_x , J_y , J_{xy} – промежуточные моменты инерции изображения объекта, способ вычисления которых рассмотрен в разделе 2.5 (ф. 2.33)

3.4. Методы моделирования на этапе проектирования АТСН

Методы математического и физического моделирования проектируемой системы помогают решать задачи, связанные с уточнением параметров решающих правил при реализации различных алгоритмов обработки сигналов в АТСН. Они способствуют выявлению обоснованных требований к отдельным звеньям системы особенно в тех случаях, когда аналитические расчётные методики оказываются мало эффективными или достаточно сложными.

На начальном этапе разработки АТСН самым доступным, дешёвым, но вместе с тем достаточно гибким и эффективным средством представляется *математическое (имитационное) компьютерное моделирование*. В качестве непосредственного объекта исследования оно предполагает использование некоторой программы, представляющей собой комплексную математическую модель.

Эта модель обычно включает в себя модели основных звеньев системы: изображения объекта, оптической системы, фотоприёмного узла (анализатора изображения), различных дестабилизирующих факторов (помех) и др., а также модель используемого алгоритма цифровой обработки сигнала.

К числу несомненных достоинств метода математического моделирования следует отнести возможность получения за короткое время и без существенных материальных затрат большого объема данных, характеризующих поведение будущей системы, её метрологические характеристики (характеристики обнаружения, распознавания объектов) в зависимости от каждого из интересующих параметров в отдельности.

Однако математическое моделирование не может полностью гарантировать от ошибок, связанных с неточным заданием исходных данных и с некоторыми упрощениями, допущенными при формировании модели.

Наибольшее приближение к реальным условиям функционирования проектируемой системы даёт *физическое моделирование*. Физическая модель АТСН обычно реализуется на базе универсальных технических средств, включающих реальный телевизионный датчик, блок АЦП, контроллер сопряжения, ЦВУ (например, персональный компьютер), другие функциональные узлы, а также образцы наблюдаемых объектов (или хотя бы их изображений). Заметим, что физическая модель, как правило, не является конструктивной моделью, макетом или тем более опытным образцом проектируемой системы (прибора). К такой модели не предъявляется особых требований минимизации габаритов и энергопотребления, она может быть достаточно громоздкой. Важно лишь, чтобы модель обеспечивала максимум функциональных возможностей и позволяла достаточно легко получать объективные результаты испытаний, сопоставимые с результатами математического моделирования.

К недостаткам физического моделирования можно отнести недостаточную гибкость и меньшую информативность по сравнению с математическим моделированием. При физическом моделирова-

нии, например, не удаётся исследовать влияние параметров различных звеньев в отдельности на качественные характеристики проектируемой системы. Не удаётся исключить или существенно уменьшить влияние отдельных дестабилизирующих факторов, влияющих на качественные характеристики системы. Это связано с тем, что в физической модели используются реальные функциональные узлы: телевизионный датчик, блок АЦП и другие, улучшить параметры которых можно только путём их замены, что часто сопряжено со значительными материальными затратами.

Однако, путём сопоставления некоторых частных результатов, полученных при физическом моделировании с соответствующими результатами, полученными при математическом компьютерном моделировании, можно с высокой степенью вероятности доказать адекватность обеих моделей. Таким образом, именно совокупность обоих методов моделирования может дать наибольший эффект с точки зрения сочетания достоверности и полноты полученной информации.

Практические навыки по применению методов моделирования АТСН могут быть получены в процессе выполнения специального комплекса лабораторных работ, посвященных этой теме [6]

Вопросы для самопроверки:

1. Поясните, в чём заключаются методы идентификации объектов, которые не требуют предварительного выделения геометрических признаков. Укажите достоинства и недостатки каждого из них. Укажите условия, соблюдение которых необходимо при использовании этих методов.
2. В чём заключаются преимущества методов идентификации и классификации объектов, основанных на выделении геометрических признаков?
3. В чём состоит принцип реализации многоступенчатого (иерархического) алгоритма распознавания объектов на основе использования геометрических признаков? Каким образом следует распределять используемые признаки на различных уровнях многоступенчатого алгоритма распознавания?
3. Какие алгоритмы обработки дискретных изображений, связанные с выделением геометрических признаков, могут быть использованы для решения задач оценки параметров объектов наблюдения: определения их размеров, пространственного положения или пространственной ориентации?
4. Поясните преимущества и недостатки математического (имитационного) компьютерного и физического моделирования, которые используются на стадии проектирования.

4. ПРИМЕРЫ ПОСТРОЕНИЯ АВТОМАТИЗИРОВАННЫХ ТЕЛЕВИЗИОННЫХ СИСТЕМ НАБЛЮДЕНИЯ

4.1. Оптико-электронные системы астроориентации и астронавигации.

ОЭС астроориентации и астронавигации решают задачи определения пространственного положения объекта по астроориентирам. Такими объектами могут быть космические аппараты (КА), самолёты, морские объекты и др. В качестве астроориентиров могут использоваться Солнце, Луна, а также планеты и звёзды. Особый интерес в качестве астроориентиров представляют звёзды, поскольку их положение на небесной сфере является наиболее стабильным.

В настоящее время всё большее внимание уделяется разработке так называемых *автономных систем астроориентации и астронавигации*.

Автономные астронавигационные системы (АНС) способны длительное время работать без связи с наземными пунктами. При этом основным условием работы автономных АНС является необходимость определения пространственного положения КА в произвольный момент времени по данным собственных измерений, независимо от того, сколько времени АНС была в выключенном (законсервированном) состоянии. То есть, ставится задача определения пространственной ориентации КА даже при отсутствии данных предшествующих измерений.

В такой постановке задача распадается на три этапа.

1. **Обнаружение и селекция** рабочих астроориентиров, т.е. выделение из числа астроориентиров, попавших в поле зрения АНС, тех, которые соответствуют выбранному диапазону звёздных величин.

2. **Распознавание** участка звёздного неба, попавшего в поле зрения АНС, по выбранным на первом этапе астроориентирам.

3. **Измерение** с высокой точностью (до единиц угловых секунд) угловых координат самой яркой звезды, попавшей в поле зрения АНС, из числа выделенных на первом этапе.

В качестве приёмника оптического излучения и, одновременно, анализатора изображения в АНС могут использоваться различные виды телевизионных преобразователей, например, вакуумные

трубки: диссектор, суперкремникон и др. Однако, наиболее современным и перспективным решением является применение матричных ПЗС и ПЗИ-структур [4, 7].

Преимущества ПЗС и ПЗИ перед другими видами телевизионных преобразователей очевидны: малые габариты и вес, малое энергопотребление, высокая надёжность, а также «жёсткий» растр, т.е. жёсткая геометрическая привязка фоточувствительных элементов к приборной системе координат. Кроме того, ПЗС и ПЗИ обладают достаточно высокой чувствительностью, позволяющей их уверенно использовать для работы со звёздами до 5-й – 7-й величин.

Архитектура АНС принципиально не отличается от архитектуры других оптико-электронных АТСН. Однако, характер задач, решаемых АНС, обладает такой спецификой, что их целесообразно рассмотреть отдельно. К тому же задачи, решаемые АНС на различных этапах, являются хорошей иллюстрацией практического применения некоторых описанных выше алгоритмов.

4.1.1. Этап обнаружения и селекции рабочих астроориентиров

Цель этапа – выделение из общего числа объектов, случайным образом попавших в поле зрения АНС, только тех, которые относятся к заданному диапазону звёздных величин и предусмотрены в бортовом каталоге АНС. Практически это означает двухэтапную обработку исходного массива $[E_{ij}]$ в соответствии с алгоритмом.

1. Формирование нового массива $[E_{ij}^*]$

$$[E_{ij}^*] = [E_{ij} \cdot \delta_{ij}], \quad (4.1)$$

где $[E_{ij}^*]$ – новый массив; $\delta_{ij} = \begin{cases} 1, & E_{ij} \geq E_{min}; \\ 0, & E_{ij} < E_{min}. \end{cases}$

Такой алгоритм по существу является алгоритмом обнаружения астроориентиров в поле зрения АНС, сигналы которых превышают заданный порог. Следует однако заметить, что все значения сигналов E_{ij} исходного массива определяются за вычетом сигнала от фона и темновых токов накопления. Поскольку отдельные элементы ПЗС или ПЗИ-структуры могут обладать некоторым отклонением чувствительности и отклонением уровня темнового тока накопления от соответствующих средних значений, то, с целью коррекции так называемого «геометрического» шума, необходимо хранение в памяти АНС параметров каждого из элементов фотоприёмника.

2. Определение наиболее освещенного элемента изображения $E_{k,l}^* = E_{max}^*$ и сравнение его сигнала с верхним порогом.

Эта операция необходима для того, чтобы исключить из обрабатываемого массива сигналы от планет и других, более мощных источников излучения, случайно попавших в поле зрения АНС. Очевидно, что попадание в поле зрения АНС прямого солнечного излучения должно исключаться специальной системой блокировки канала. В этом случае информация получается с других, параллельно работающих аналогичных каналов.

Если $E_{\max}^* \leq E_{\text{порог}}$, то этап селекции завершён. Если $E_{\max}^* > E_{\text{порог}}$, то массив подвергается дополнительной обработке в соответствии с алгоритмом

$$[E_{ij}^{**}] = [E_{ij}^* \cdot \delta_{ij}], \quad (4.2)$$

где $\delta_{ij} = \begin{cases} 0, & (k-n \leq i \leq k+n) \ \& \ (l-n \leq j \leq l+n); \\ 1, & (i > k+n) \vee (i < k-n) \vee (j > l+n) \vee (j < l-n); \end{cases}$
 $n = 3 \div 5$.

4.1.2. Этап опознавания участка звёздного неба

Как было сказано выше, одним из классических методов опознавания является корреляционный метод (см. раздел 3.1.2). Однако, корреляционный метод, предполагающий вычисление взаимно-корреляционных сумм при использовании сигналов от всех элементов фотоприёмного устройства и всех элементов массивов, хранящихся в бортовом каталоге для всех возможных фрагментов звёздного неба, является практически непригодным. Такая процедура потребовала бы слишком большого числа вычислительных операций и объемов памяти. К тому же корреляционный метод весьма чувствителен к взаимному развороту сравниваемых фрагментов.

Вместе с тем, можно отметить следующую особенность изображений, регистрируемых АНС.

Изображения являются сравнительно малоинформативными, т.е. в большом числе элементов полезный сигнал просто отсутствует. Информация же, содержащаяся в изображениях, может быть закодирована путём задания значений сигнала от каждого астроориентира и координат его изображения, выраженных, например, через номера строк и столбцов матрицы фоточувствительных элементов ПЗС-структуры. Это обстоятельство позволяет значительно сократить объём обрабатываемых и запоминаемых данных и предложить следующую иерархическую схему алгоритма опознавания, который реализуется на трёх уровнях.

1. Сравнение сигнала от самой яркой звезды в поле зрения с данными каталога, в котором хранятся значения сигналов самых ярких (опорных) звёзд, соответствующих выбранному диапазону.

Если выполняется условие

$$\left| \sum_{i=k-n}^{k+n} \sum_{j=l-n}^{l+n} E_{ij}^* - A(J) \right| \leq Z_{A1}, \quad (4.3)$$

то осуществляется переход к следующему уровню. (Здесь Z_{A1} – некоторое пороговое значение модуля разности на первом уровне иерархического алгоритма опознавания; $A(J)$ – значение сигнала J -той звезды из числа опорных звезд, взятых из каталога). В противном случае процедура повторяется для $J+1$ -ой звезды из каталога.

2. После предварительной идентификации опорной звезды на первом этапе (хотя и с определённой вероятностью ошибки) АНС переходит ко второму этапу. Он заключается в анализе угловых расстояний между опорной J -той звездой и менее яркими K -тыми звёздами её окружения, предусмотренными бортовым каталогом (рис.4.1).

Оценка разности угловых координат астроориентиров в конечном счёте сводится к оценке расстояний между центрами их изображений в приборной системе координат в соответствии с алгоритмом

$$\left| R(J,K) - R_k(J,K) \right| \leq Z_{R2}, \quad (4.4)$$

где: $R_k(J,K)$ – константы возможных расстояний, которые берутся из каталога; $X(J,K)$ и $Y(J,K)$ – расстояния между опорной и окружающими звёздами, определяемые соответственно вдоль направления строк и столбцов матричной структуры фотоприёмника; Z_{R2} – некоторое пороговое значение модуля разности на втором уровне опознавания.

Заметим, что расчет $R(J,K)$ и проверка условия (4.4) должны выполняться для всех звёзд, находящихся в поле зрения АНС. При выполнении условия (4.4) для трёх или более звёзд в поле зрения АНС осуществляется переход к третьему (заключительному) уровню идентификации опорной звезды, в противном случае – возврат к первому уровню и повторение описанных процедур для других опорных звёзд каталога.

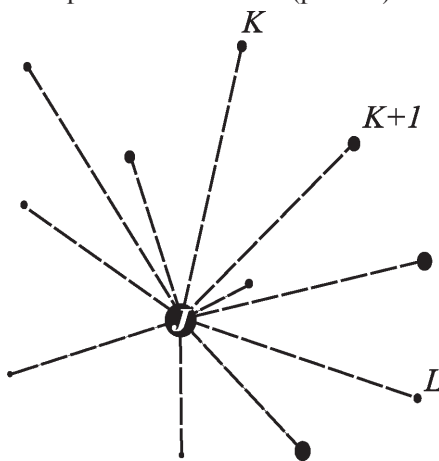


Рис. 4.1. К пояснению второго этапа алгоритма идентификации опорной звезды.

3. Третий (последний) этап идентификации заключается в вычислении взаимных расстояний между всеми звёздами, окружающими опорную звезду (рис. 4.2), и в сравнении этих расстояний с данными каталога

$$|R(K,L) - R_k^*(K,L)| \leq Z_{R3}, \quad (4.5)$$

где $R(K,L) = \sqrt{X(K,L)^2 + Y(K,L)^2}$;

$$R_k^*(K,L) = \sqrt{[X_k(J,K) - X_k(J,L)]^2 + [Y_k(J,K) - Y_k(J,L)]^2}.$$

Значения $X_k(J,K)$, $Y_k(J,K)$ и $X_k(J,L)$, $Y_k(J,L)$ берутся из каталога.

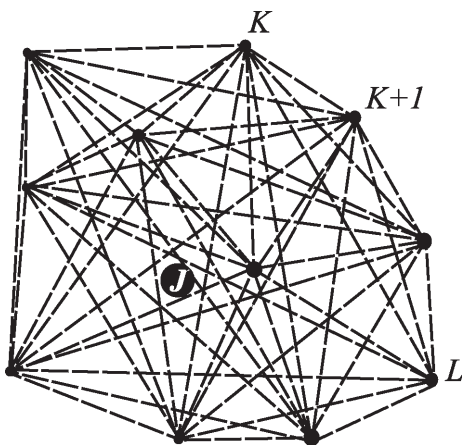


Рис. 4.2. К пояснению третьего этапа алгоритма идентификации опорной звезды.

Процедура идентификации опорной звезды считается законченной, если хотя бы для 3-х звёзд в поле зрения АНС выполняется условие (4.5). Если это не так, то осуществляется возврат к первому уровню и повторение описанных процедур для других звёзд каталога.

При наличии нескольких каналов вероятность пропуска или условная вероятность неопознавания определяется как вероятность пропуска более чем в p из q независимых каналов

$$P_{\text{проп.}} = \sum_{j=p}^q C_q^j [P_{\text{проп.1}}]^j [1 - P_{\text{проп.1}}]^{q-j} \quad (4.6)$$

где: q – число каналов; p – число совпадений результатов опознавания соответствующих участков звёздного неба, предопределяющее правило окончательного решения о завершении этапа опознавания; $P_{\text{проп.1}}$ – вероятность пропуска в одном из каналов.

Вероятность пропуска в каждом канале зависит не только от уровня помех, но и от выбранных порогов сравнения Z_{A1} , Z_{R2} , Z_{R3} .

Так вероятность пропуска на первом этапе

$$P_{\text{проп.1(I)}} = \int_{Z_{\text{д}}/\sigma_{\text{ш}}}^{\infty} p(x) dx, \quad (4.7)$$

где: $p(x)$ – плотность распределения вероятности ошибки при сравнении амплитуд. В частном случае, если ошибка подчиняется нормальному закону распределения, $P_{\text{проп.1(I)}}$ можно оценить следующим образом

$$P_{\text{проп.1(I)}} = \frac{1}{\sqrt{2\pi}} \int_{Z_{\text{д}}/\sigma_{\text{ш}}}^{\infty} e^{-\frac{x^2}{2}} dx, \quad (4.8)$$

где $\sigma_{\text{ш}}$ – среднеквадратическое значение шумового напряжения в разностном сигнале.

Для следующих этапов распознавания можно записать аналогичные выражения

$$P_{\text{проп.1(II,III)}} = \frac{1}{\sqrt{2\pi}} \int_{Z_{R(\text{II,III})}/\sigma_{(\text{II,III})}}^{\infty} e^{-\frac{x^2}{2}} dx \quad (4.9)$$

где $Z_{R(\text{II,III})}$ – величины порогов сравнения на втором и третьем этапах; $\sigma_{R(\text{II,III})}$ – среднеквадратические значения ошибок при сравнении расстояний на втором и третьем этапах.

Вероятность непропуска, т.е. условная вероятность правильного опознавания в одном из каналов определяется выражением

$$P_{\text{непр.1}} = 1 - P_{\text{проп.1}} = [1 - P_{\text{проп.1(I)}}] \cdot [1 - P_{\text{проп.1(II)}}]^3 \cdot [1 - P_{\text{проп.1(III)}}]^3 \quad (4.10)$$

Можно показать, что вероятность ошибочного опознавания при использовании описанного трёхэтапного алгоритма может быть пренебрежимо малой. Однако, разумеется, количественное определение вероятностных характеристик обнаружения и выбор на их основе оптимальных параметров алгоритма могут осуществляться путём детального моделирования АНС на начальном этапе разработки.

Оценим объём памяти бортового каталога, необходимый для нормальной работы АНС. Каталогное описание окружения любой из N ярких звёзд (например, звёзд величин $m \leq 5$, выбираемых в качестве опорных) должно охватывать область звёздного неба, ограниченную радиусом, равным диагонали фоточувствительной площадки матричного ПЗС или ПЗИ фотоприёмника. Здесь имеется в виду фрагмент звёздного неба, спроецированный на плоскость анализа изображения при выбранном фокусном расстоянии (рис. 4.3).

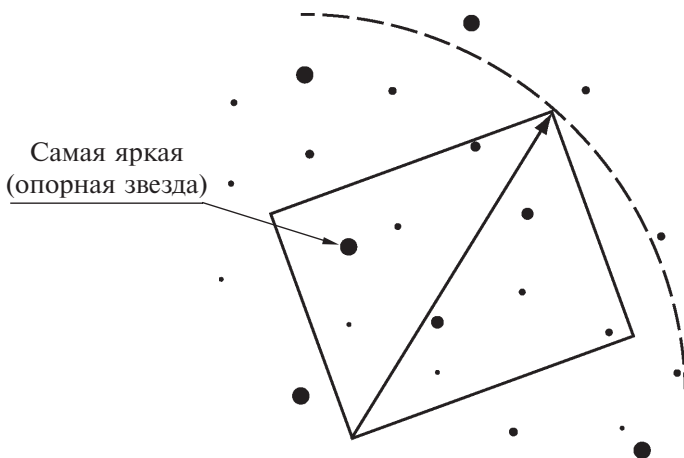


Рис. 4.3. К пояснению оценки объёма памяти бортового каталога АНС.

При этом для гарантии попадания хотя бы 3-х звёзд в поле зрения АНС необходимо, учитывая случайный характер захвата и неравномерность распределения звёзд по небесной сфере, хранить в бортовом каталоге информацию хотя бы о 12-ти звёздах, окружающих каждую из опорных.

Таким образом, каталог звёздного неба требует следующего объёма памяти.

1. Для хранения амплитуд сигналов N опорных звёзд

$$M_1 = N - 8\text{-ми разрядных слов } (N \text{ байт}).$$

2. Для хранения массивов $[X_k(J, \dots)]$ и $[Y_k(J, \dots)]$

$$M_2 = 2 \cdot 12 \cdot N \text{ байт.}$$

3. Для хранения массива $[R_k(J, \dots)]$.

$M_3 = 12 \cdot N - 4 \div 5$ байтных слов (так как это должны быть реальные числа с плавающей точкой).

Так, например, можно показать, что необходимый объём памяти бортового каталога при использовании в качестве опорных звёзд величин $m \leq 5$, что соответствует $N \approx 1500$, полный объём памяти бортового каталога составит всего

$$M = M_1 = M_2 = M_3 \approx 160 \div 200 \text{ Кбайт.}$$

Следует иметь в виду, что при выборе астроориентиров и формировании бортового каталога приходится учитывать спектральный класс звёзд, спектральную чувствительность реального фотоприёмника и вероятность попадания необходимого числа опорных звёзд в поле зрения АНС.

4.1.3. Измерение угловых координат опорной звезды

После окончательной идентификации опорной звезды осуществляется измерение её угловых координат с максимально возможной точностью, предопределяющей точность решения задачи астроориентации в целом.

Очевидно, что угловые координаты ψ и χ бесконечно удалённого источника однозначно связаны с линейными координатами x и y его изображения на фоточувствительной площадке ФПЗС

$$x = f \operatorname{tg} \psi \approx f \cdot \psi; \quad y = f \operatorname{tg} \chi \approx f \chi, \quad (4.11)$$

где f – фокусное расстояние объектива.

Жёсткая геометрическая привязка каждого фоточувствительного элемента на кристалле ФПЗС в сочетании с линейной световой характеристикой позволяет регистрировать линейные смещения изображений звёзд с погрешностью, не превышающей десятых долей элемента. Один из простейших широко известных алгоритмов измерения координат малоразмерных изображений объектов заключается в вычислении энергетического центра (см. ф. 2.29). Однако, при малых отношениях сигнал/шум ($\sigma_{\text{ш}} < 10$) более эффективным является алгоритм интерполяции сигналов, снимаемых с отдельных элементов ПЗС-структуры по методу наименьшего среднеквадратического отклонения (НСКО). При работе ФПЗС в режиме малых сигналов от звёзд величин $m \geq 5$ этот метод позволяет достичь большей точности измерений.

На рис. 4.4 показано распределение освещённости $E(x)$ вдоль строки элементов ФПЗС, которое преобразуется в последовательность электрических сигналов (видеоимпульсов) $Q(x_i)$, где i – номер ячейки, с которой снимается сигнал.

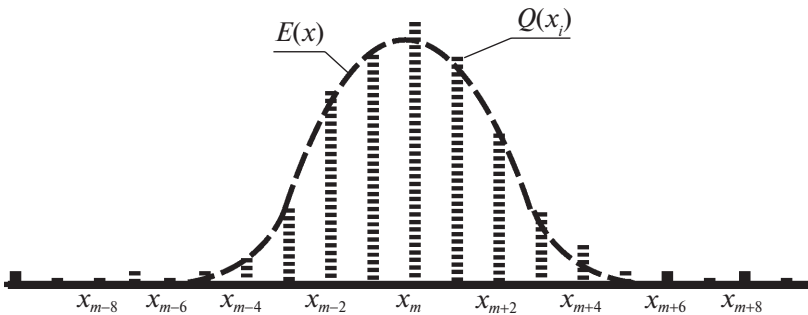


Рис. 4.4. $E(x)$ – распределение освещённости вдоль строки элементов ФПЗС.
 $Q(x_i)$ – последовательность электрических сигналов (видеоимпульсов), снимаемых с отдельных элементов.

Измерение координат изображения (которые в данном случае мы связываем с координатами максимума интерполирующей функции) сводится к восстановлению непрерывной функции $Q(x) \approx E(x)$ по методу НСКО, вычислению её производной $dQ(x)/dx$ и определению координаты x из условия

$$dQ(x)/dx = 0. \quad (4.12)$$

Восстановление непрерывной функции $Q(x)$ может осуществляться путём её аппроксимации полиномом четвёртой степени по методу НСКО

$$Q(x) = K_4x^4 + K_3x^3 + K_2x^2 + K_1x + K_0. \quad (4.13)$$

Известно, что весовая функция оптической системы, содержащей более четырёх поверхностей, хорошо аппроксимируется гауссоидой вращения, поэтому аналогичный полином может быть использован и для восстановления функции вдоль оси Y .

Известно, что для нахождения коэффициентов полинома $Q(x)$ необходимо решить систему линейных уравнений вида

$$\begin{aligned} K_0N + K_1\sum_{i=1}^Nx_i + K_2\sum_{i=1}^Nx_i^2 + K_3\sum_{i=1}^Nx_i^3 + K_4\sum_{i=1}^Nx_i^4 &= \sum_{i=1}^N Q(x_i) \\ K_0\sum_{i=1}^Nx_i + K_1\sum_{i=1}^Nx_i^2 + K_2\sum_{i=1}^Nx_i^3 + K_3\sum_{i=1}^Nx_i^4 + K_4\sum_{i=1}^Nx_i^5 &= \sum_{i=1}^N [Q(x_i)x_i] \\ K_0\sum_{i=1}^Nx_i^2 + K_1\sum_{i=1}^Nx_i^3 + K_2\sum_{i=1}^Nx_i^4 + K_3\sum_{i=1}^Nx_i^5 + K_4\sum_{i=1}^Nx_i^6 &= \sum_{i=1}^N [Q(x_i)x_i^2] \\ K_0\sum_{i=1}^Nx_i^3 + K_1\sum_{i=1}^Nx_i^4 + K_2\sum_{i=1}^Nx_i^5 + K_3\sum_{i=1}^Nx_i^6 + K_4\sum_{i=1}^Nx_i^7 &= \sum_{i=1}^N [Q(x_i)x_i^3] \\ K_0\sum_{i=1}^Nx_i^4 + K_1\sum_{i=1}^Nx_i^5 + K_2\sum_{i=1}^Nx_i^6 + K_3\sum_{i=1}^Nx_i^7 + K_4\sum_{i=1}^Nx_i^8 &= \sum_{i=1}^N [Q(x_i)x_i^4] \end{aligned} \quad (4.14)$$

где: N – количество отсчётов, по которым определяются коэффициенты K_4, K_3, K_2, K_1, K_0 полинома $Q(x)$; x_i – координата геометрического центра данного i -го элемента по горизонтали или вертикали, отсчитываемая от начала приборной системы координат, например, от первого нижнего элемента светочувствительной области ПЗС; $Q(x_i)$ – выраженная в численной форме величина электрического сигнала, соответствующего значению средней облучённости i -той ячейки ПЗС.

В представленном виде (4.14) для составления и решения системы уравнений требуется выполнение большого числа математических операций. Вместе с тем, при реализации алгоритма, работающего

в реальном масштабе времени, часто возникает вопрос оценки и оптимизации вычислительных затрат, с целью достижения предельного быстродействия измерительной системы.

В данном случае существенное уменьшение числа вычислительных операций может быть достигнуто за счёт того, что решение системы линейных уравнений можно производить сначала в собственной нормированной системе координат, привязанной к элементу наибольшего сигнала. При этом координаты элементов ФПЗС, с которых снимаются сигналы, отсчитываются от координаты x_m наиболее освещённого элемента матрицы путём записи целого числа периодов ПЗС-структуры, разделяющих данный элемент и элемент с координатой x_m . Кроме того, распределение освещённости от точечного источника сосредоточено лишь на небольшом участке матрицы в окрестностях наиболее освещённого элемента, поэтому число отсчётов, по которым строится полином $Q(x)$, можно ограничить небольшим числом, например, десятью $N = 10$. Таким образом, приняв в нормированной системе координат $x_m = 5$, координаты других освещённых элементов запишутся в виде

$$\begin{aligned} x_{m-4} &= 1; x_{m-3} = 2; x_{m-2} = 3; x_{m-1} = 4; \\ x_m &= 5; \\ x_{m+1} &= 6; x_{m+2} = 7; x_{m+3} = 8; x_{m+4} = 9; x_{m+5} = 10; \end{aligned} \quad (4.15)$$

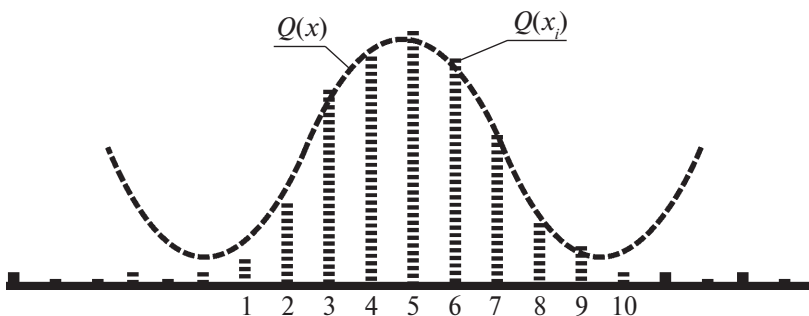


Рис. 4.5. К пояснению алгоритма интерполяции видеоимпульсов по методу НСКО. $Q(x)$ — аппроксимирующий полином 4-й степени

По найденной оценке \hat{X} положения максимума освещённости в нормированной системе координат легко найти оценку координаты центра изображения точечного источника в системе координат ФПЗС по формуле

$$\hat{x} = x_m + (\hat{X} - 5) \cdot \Delta x, \quad (4.16)$$

где Δx — шаг элементов ПЗС-структуры.

В нормированной системе координат (4.15) система уравнений (4.14) переписывается в виде

$$\begin{aligned}
 K_0 \cdot 10 + K_1 \sum_{i=1}^{10} i + K_2 \sum_{i=1}^{10} i^2 + K_3 \sum_{i=1}^{10} i^3 + K_4 \sum_{i=1}^{10} i^4 &= \sum_{i=1}^{10} Q(i) \\
 K_0 \sum_{i=1}^{10} i + K_1 \sum_{i=1}^{10} i^2 + K_2 \sum_{i=1}^{10} i^3 + K_3 \sum_{i=1}^{10} i^4 + K_4 \sum_{i=1}^{10} i^5 &= \sum_{i=1}^{10} [Q(i) \cdot i] \\
 K_0 \sum_{i=1}^{10} i^2 + K_1 \sum_{i=1}^{10} i^3 + K_2 \sum_{i=1}^{10} i^4 + K_3 \sum_{i=1}^{10} i^5 + K_4 \sum_{i=1}^{10} i^6 &= \sum_{i=1}^{10} [Q(i) \cdot i^2] \\
 K_0 \sum_{i=1}^{10} i^3 + K_1 \sum_{i=1}^{10} i^4 + K_2 \sum_{i=1}^{10} i^5 + K_3 \sum_{i=1}^{10} i^6 + K_4 \sum_{i=1}^{10} i^7 &= \sum_{i=1}^{10} [Q(i) \cdot i^3] \\
 K_0 \sum_{i=1}^{10} i^4 + K_1 \sum_{i=1}^{10} i^5 + K_2 \sum_{i=1}^{10} i^6 + K_3 \sum_{i=1}^{10} i^7 + K_4 \sum_{i=1}^{10} i^8 &= \sum_{i=1}^{10} [Q(i) \cdot i^4]
 \end{aligned} \tag{4.17}$$

В отличие от системы уравнений (4.14), система (4.17) имеет вполне детерминированную левую часть. Поэтому большая часть вычислений может быть выполнена заранее (т.е. на этапе разработки алгоритма), и решение системы в процессе измерения координат сведётся к простой последовательности арифметических операций. Отметим, что на этапе разработки алгоритма величины i^2 , i^3 , i^4 также могут быть вычислены заранее и храниться как константы. Таким образом, для составления системы непосредственно на этапе измерения (4.17) потребуется 45 операций сложения и 40 операций умножения, причём все операции над целочисленными переменными.

С целью уменьшения ошибок округления, при последующих вычислениях удобно использовать алгоритм Гаусса по следующей схеме.

1. Все уравнения системы (4.17) делятся на соответствующие множители при K_0 .
2. Формируется новая система из четырёх уравнений путём вычитания из второго уравнения первого, из третьего – второго, из четвёртого – третьего и из пятого – четвёртого.
3. Каждое из уравнений новой системы делится на соответствующие множители при K_1 , полученные на предыдущем шаге.
4. Вычитая из второго уравнения первое, из третьего – второе, из четвёртого – третье, получают третью систему из трёх уравнений.
5. Аналогичным образом получают четвёртую систему из двух уравнений и, наконец, одно уравнение с одним неизвестным коэффициентом K_4 , который определяется путём деления правой части уравнения на множитель при K_4 .

6. На последующих трёх этапах рассчитываются коэффициенты K_3, K_2, K_1 в результате последовательной подстановки известных коэффициентов в последние уравнения четвертой, третьей и второй систем.

Заметим, что значение коэффициента K_0 не имеет практического значения, поскольку максимум интерполирующего полинома будет находиться путём приравнивания его производной к нулю

$$Q'(x) = 4K_4x^3 + 3K_3x^2 + 2K_2x + K_1 = 0. \quad (4.18)$$

Следовательно, для реализации описанной процедуры решения системы уравнений достаточно 16 операций типа сложения-вычитания и 24 операции типа умножения-деления.

Следующий этап – нахождение максимума интерполирующей функции $Q(x)$ в соответствии с (4.18). Однако, поскольку интервал, в котором следует искать значение корня, заранее определён ($4 \leq X \leq 5$), то целесообразно воспользоваться вычислительной схемой Ньютона, которая обеспечивает быструю сходимость вычислительного процесса. Применительно к рассматриваемому случаю она выразится в виде рекуррентного алгоритма

$$\hat{X} = \hat{X} - (b_3 \hat{X}^3 + b_2 \hat{X}^2 + b_1 \hat{X} + b_0) / (a_2 \hat{X}^2 + a_1 \hat{X} + a_0), \quad (4.19)$$

где: $b_3 = 4K_4$; $b_2 = 3K_3$; $b_1 = 2K_2$; $b_0 = K_1$; $a_2 = 3b_3$; $a_1 = 2b_2$; $a_0 = b_1$.

Причём процесс вычислений целесообразно производить по схеме Гарнера

$$\hat{X} = \hat{X} - (\hat{X} \cdot (\hat{X} \cdot (\hat{X} \cdot b_3 + b_2) + b_1) + b_0) / (\hat{X} \cdot (\hat{X} \cdot a_2 + a_1) + a_0). \quad (4.20)$$

Таким образом, каждый шаг, выполняемый по схеме Ньютона, потребует выполнения 6 операций сложения-вычитания, 6 операций умножения-деления. Кроме того, расчёт коэффициентов b_3, b_2, b_1, a_2, a_1 на предварительном этапе связан с выполнением дополнительно 5-ти операций умножения. Анализ сходимости решения уравнения (4.20) показал, что для достижения точности определения значения корня не хуже 10^{-6} достаточно 5 раз воспользоваться схемой Ньютона.

Итак, для формирования системы уравнений (4.17), её решения и нахождения максимума аппроксимирующего полинома, необходимо выполнить 91 операцию типа сложения-вычитания сложения, причём 45 из них над целочисленными переменными и 99 операций типа умножение-деление, причём 48 над целочисленными переменными. Очевидно, что такое количество операций достаточно



Рис. 4.5. Блок-схема алгоритма интерполяции.

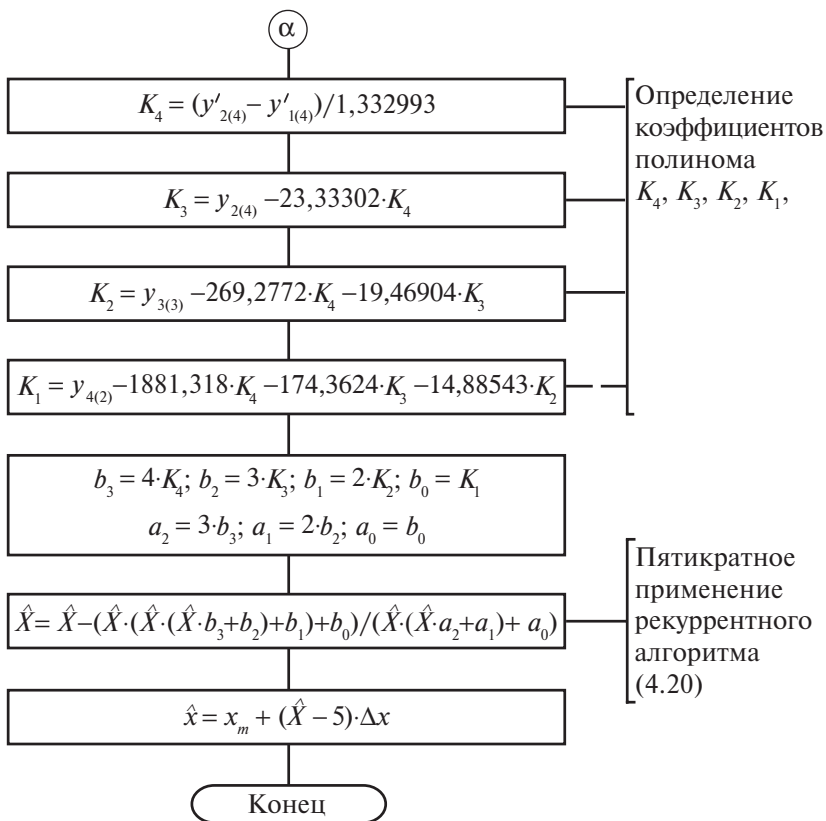


Рис. 4.6. Блок-схема алгоритма интерполяции (окончание).

быстро (за доли секунды) может быть выполнено на базе современных микропроцессорных систем даже сравнительно невысокого быстродействия. Поэтому данный алгоритм может быть использован не только в АНС, но и в других автоматизированных телевизионных системах наблюдения, работающих в реальном масштабе времени.

Очевидно, что измерение координаты вдоль оси Y может быть выполнено совершенно аналогичным образом при обработке сигналов с десяти элементов вдоль столбца матрицы ФПЗС в окрестностях наиболее освещённого элемента с координатами x_m, y_m .

На рис. 4.5 и 4.6 приведена блок-схема описанного алгоритма.

Как показал анализ, рассмотренному методу вычисления координат максимума освещённости в изображении точечного объекта кроме шумовой может быть присуща методическая составляющая погрешности измерения.

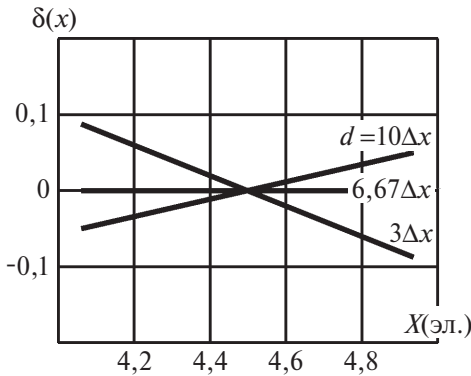


Рис. 4.7. Зависимости методической составляющей погрешности $\delta(\hat{X})$ от положения максимума освещенности в пределах элемента при различных значениях эффективного диаметра изображения d .

ти от положения максимума освещенности в пределах элемента носит линейный характер. Таким образом, методическая составляющая погрешности может быть уменьшена до величины второго порядка малости путём несложной коррекции

$$\hat{X}_{\text{испр.}} = \hat{X} + \delta(\hat{X}) \quad (4.21),$$

где $\hat{X}_{\text{испр.}}$ — исправленная после коррекции методической составляющей погрешности оценка координаты изображения точечного источника.

На рис. 4.8 представлена полученная в процессе моделирования зависимость случайной составляющей погрешности измерения σ от отношения сигнал/шум μ . Там же для сравнения приведена аналогичная зависимость, характерная для

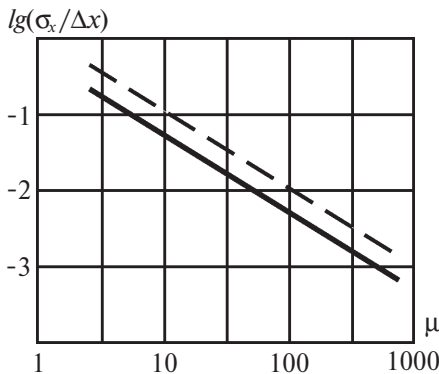


Рис.4.8. Зависимость случайной составляющей погрешности измерения σ от отношения сигнал/шум μ . Пунктиром показана аналогичная зависимость для более простого алгоритма (ф. 2.32).

простейшего алгоритма вычисления координат энергетического центра изображения точечного источника (см., например, ф. 2.32). Сравнительный анализ показывает, что наибольший относительный выигрыш в точности измерения координат достигается при малых отношениях сигнал/шум. Так, например, при $\mu = 10$ рассмотренный алгоритм позволяет измерять координату изображения точечного объекта с погрешностью в 1,5 раза меньшей, чем при использовании более простого алгоритма (ф. 2.32).

В заключение заметим, что рассмотренный алгоритм может найти применение не только в системах астроориентации, но и во многих других оптико-электронных системах, в которых решается задача измерения координат малоразмерных объектов в сложных условиях наблюдения.

4.2. Принципы построения обучаемых АТСН

В обучаемых АТСН окончательная зависимость выходных сигналов (выходных управляющих воздействий) от регистрируемых входных оптических сигналов закладывается не в процессе разработки и изготовления системы, а на этапе настройки и адаптации готового прибора или системы к реальным условиям функционирования. Этот процесс настройки и адаптации в общем случае будем называть периодом или этапом обучения.

Обучаемыми могут быть самые различные по назначению АТСН: всевозможные измерительные приборы, системы технического зрения, астронавигационные системы, тепловизионные обзорно-поисковые системы и т.д. То есть, с точки зрения функционального назначения и характера используемых сигналов, — это не какой-то новый класс приборов и систем. Отличие обучаемых АТСН заключается в способе формирования окончательного решающего правила, предопределяющего зависимость выходных сигналов от входных. Такая зависимость, в общем случае, может быть нелинейной.

Принцип построения обучаемых АТСН проиллюстрируем на двух конкретных примерах.

4.2.1. Применение алгоритма обучения в оптико-электронном угломере.

Оптико-электронные угломеры — приборы, предназначенные для контроля пространственной ориентации контролируемого объекта относительно системы координат, задаваемой другим (базовым) объектом. Один из вариантов оптической схемы угломера показан на рис. 4.9.

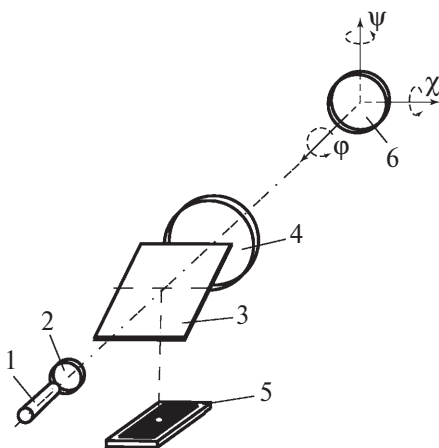


Рис.4.9. Оптическая схема угломера.

Угломер состоит из двух основных частей: опико-электронного датчика (ОЭД), определяющего базовую систему координатных осей X, Y, Z и контрольного элемента, размещаемого на контролируемом объекте.

Основными элементами ОЭД являются: светодиод (1), диафрагма (2), светоделитель (3), объектив (4), позиционно-чувствительный фотоприёмник, например, ПЗС-матрица (5), а также элементы электронного тракта

предварительной обработки сигнала (на рисунке не показаны).

Контрольным элементом (6) может служить плоское зеркало или более сложный по структуре пассивный отражатель, жестко связанный с контролируемым объектом.

Диафрагма, представляющая собой пластину с небольшим отверстием в несколько десятков или сотен микрон, располагается в фокальной плоскости объектива, поэтому ОЭД формирует пучок лучей слабой расходимости. На площадке ПЗС, также расположенной в фокальной плоскости, строится изображение диафрагмы. Координаты изображения зависят только от углов разворота ψ и χ контролируемого объекта относительно коллимационных осей X и Y и не зависят от его линейных смещений вдоль любой из координатных осей.

Таким образом, задача измерения угловых разворотов контролируемого объекта сводится к задаче измерения линейных координат центра изображения диафрагмы на фоточувствительной площадке ПЗС.

Один из возможных алгоритмов обработки сигнала от малоразмерного (точечного) источника заключается в вычислении координат его энергетического центра (ф. 2.32). Другой алгоритм, заключающийся в интерполяции сигнала с отдельных элементов матричного фотоприёмника, с последующим определением координат максимума интерполирующей функции, подробно описан в разделе 4.1.

При реализации высокоточных (с погрешностью, не превышающей единиц угловых секунд) и широкодиапазонных (до единиц и

десятков градусов) угломеров приходится учитывать множество факторов, влияющих на характер статической характеристики ОЭД. К их числу относятся: неравномерность чувствительности отдельных участков ПЗС-структуры; неравномерное распределение плотности темновых токов по кристаллу ПЗС, влияние искажений сигнала, возникающих вследствие неэффективности переноса зарядов в ПЗС; неперпендикулярность падения лучей на плоскую фоточувствительную площадку на краях измерительного диапазона; влияние аберраций оптической системы. Совокупность всех этих факторов приводит к появлению значительной систематической составляющей погрешности помимо случайной шумовой составляющей погрешности измерения.

На практике возможны различные способы коррекции систематической погрешности, реализуемые на этапе цифровой обработки сигнала. Например, путём аппроксимации заранее определённой статической характеристики какой-либо аналитической функцией, возможна последующая коррекция результатов каждого измерения за счет вычисления соответствующей поправки. Однако, это приводит к увеличению времени на каждое измерение, и, поэтому такой путь не всегда является оптимальным решением. Кроме того, при замене каких-либо элементов оптической схемы (объектива, контрольного элемента, фотоприёмника) или при изменении условий работы (например дистанции) возникает необходимость существенной коррекции алгоритма, а, следовательно, программного обеспечения измерительной системы.

Остановимся здесь на возможности использования алгоритма обучения, позволяющего добиться предельного быстродействия измерительной системы в рабочем режиме при достаточной гибкости её адаптации на предварительном этапе калибровки в случае изменения условий работы ОЭД.

На рис. 4.10 показан оптико-электронный угломер, содержащий ОЭД, а также контрольный элемент, расположенный на контрольно-измерительном стенде, используемом на этапе обучения, т.е. на этапе калибровки измерительной системы. Задание точного разворота контрольного элемента, например вокруг оси X , осуществляется оператором с помощью микрометрического винта. На рис. 4.11 в качестве примера приведена статическая характеристика канала измерения угла χ . По оси абсцисс отложены истинные значения измеряемого угла χ , а по оси ординат – усреднённые значения измеряемого угла $\bar{\chi}$. Пунктиром показана идеальная статическая характеристика оптико-электронного угломера.

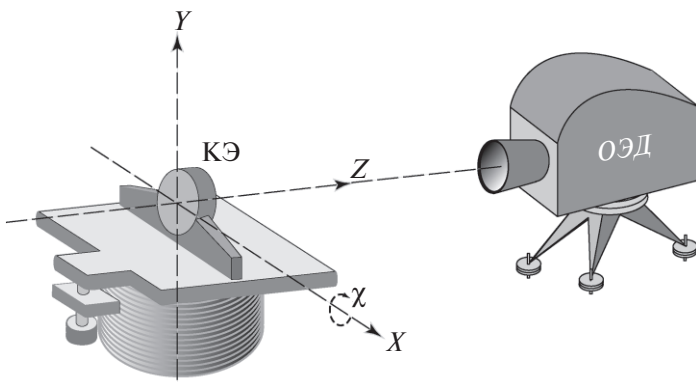


Рис.4.10. Опико-электронный датчик (ОЭД) и контрольный элемент (КЭ) на контрольно-измерительном стенде.

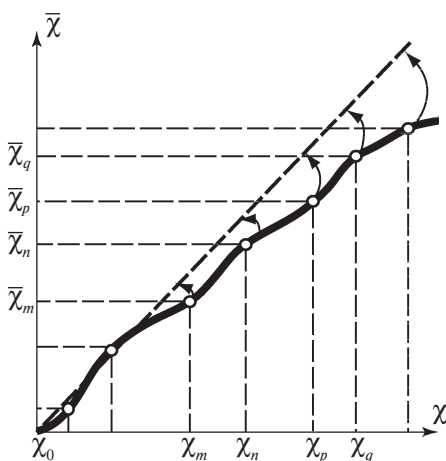


Рис.4.11. Статическая характеристика канала измерения угла χ .

Структурная схема рассматриваемой измерительной системы приведена на рис. 4.12.

На этапе обучения-калибровки с помощью микрометрического винта последовательно задаётся определённое число значений угла разворота контрольного элемента $\chi_m, \chi_n, \dots, \chi_q$. При этом погрешность установки контрольного элемента должна быть значительно меньше величины систематической погрешности измерения.

Для контроля точности установки углов можно воспользоваться теодолитом и вспомогательным зеркалом, закреплённым на координатном столике.

На этапе калибровки переключатели, показанные на рис. 4.12, должны быть переведены в верхнее положение. При задании оператором определённого значения χ_i на выходе ОЭД появляется двоичный параллельный код соответствующей оценки χ_i^* , который посредством устройства ввода/вывода (УВВ) ЦВУ заносится в его

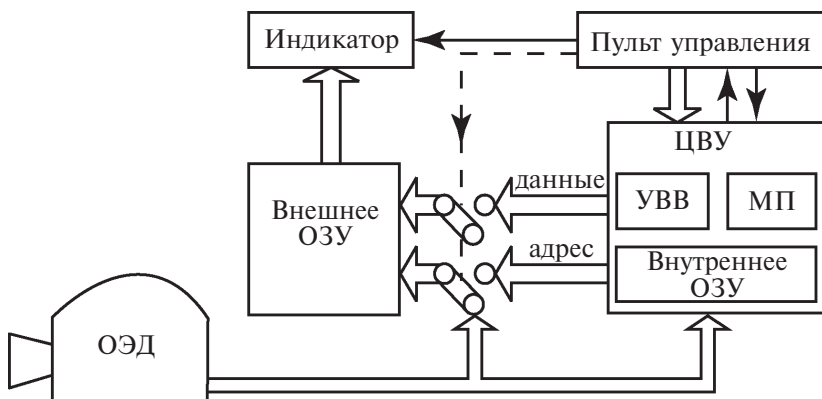


Рис.4.12. Структурная схема рассматриваемой измерительной системы.

собственную оперативную память. Для уменьшения на этапе калибровки влияния случайной составляющей погрешности измерения, при каждом положении контрольного элемента в ЦВУ обеспечивается ввод не однократного значения оценки измеряемого угла χ_i^* , а целый массив значений оценок, состоящий из 100 или более реализаций. Этот массив в дальнейшем обрабатывается ЦВУ с целью вычисления усреднённой оценки $\bar{\chi}_i$. Значение усредненной оценки используется в дальнейшем как адрес той ячейки памяти внешнего оперативного запоминающего устройства (ОЗУ), в которую должно быть записано истинное значение χ_i , установленное оператором. Это же значение χ_i оператор набирает на пульте управления обучением, и значение χ_i записывается во внешнее ОЗУ по адресу $\bar{\chi}_i$.

Аналогичным образом формируются адреса $\bar{\chi}_m, \bar{\chi}_n, \dots, \bar{\chi}_q$ и данные $\chi_m, \chi_n, \dots, \chi_q$, записываемые в другие ячейки ОЗУ (рис. 4.13).

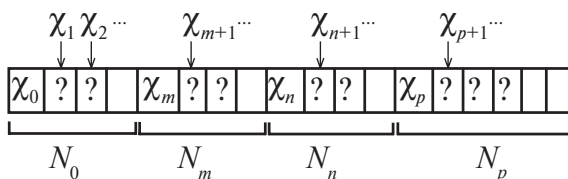


Рис. 4.13. Формирование массива данных во внешнем ОЗУ на этапе обучения (калибровки).

На следующем этапе, в соответствии с предусмотренной программой, ЦВУ осуществляет интерполяцию значений углов в промежуточных

точках статической характеристики, согласно ниже приведённому алгоритму

$$\left. \begin{array}{l}
 \chi_0 = \chi_0 - \text{калибровка} \\
 \chi_1 = \chi_0 + \Delta\chi_0 \\
 \chi_2 = \chi_1 + \Delta\chi_0 \\
 \dots\dots\dots \\
 \chi_{m-1} = \chi_{m-2} + \Delta\chi_0
 \end{array} \right\} \Delta\chi_0 = \frac{\chi_m - \chi_0}{N_0} = \frac{\chi_m - \chi_0}{\bar{\chi}_m - \bar{\chi}_0}$$

$$\left. \begin{array}{l}
 \chi_m = \chi_m - \text{калибровка} \\
 \chi_{m+1} = \chi_m + \Delta\chi_m \\
 \chi_{m+2} = \chi_{m+1} + \Delta\chi_m \\
 \dots\dots\dots \\
 \chi_{n-1} = \chi_{n-2} + \Delta\chi_m
 \end{array} \right\} \Delta\chi_m = \frac{\chi_n - \chi_m}{N_m} = \frac{\chi_n - \chi_m}{\bar{\chi}_n - \bar{\chi}_m}$$

$$\left. \begin{array}{l}
 \chi_n = \chi_n - \text{калибровка} \\
 \chi_{n+1} = \chi_n + \Delta\chi_n \\
 \chi_{n+2} = \chi_{n+1} + \Delta\chi_n \\
 \dots\dots\dots \\
 \chi_{p-1} = \chi_{p-2} + \Delta\chi_n \\
 \dots\dots\dots
 \end{array} \right\} \Delta\chi_n = \frac{\chi_p - \chi_n}{N_n} = \frac{\chi_p - \chi_n}{\bar{\chi}_p - \bar{\chi}_n} \quad (4.22)$$

Все рассчитанные значения $\chi_1, \chi_2, \dots, \chi_{m-1}$ и др. записываются в промежуточные, пока ещё свободные ячейки внешнего ОЗУ (рис. 4.13).

Таким образом, нелинейная статическая характеристика (рис. 4.11) аппроксимируется кусочно-линейной функцией с различной крутизной наклона на различных участках. Эта функция в виде таблицы-массива хранится во внешнем ОЗУ измерительной системы.

Число калибруемых точек определяется степенью нелинейности исходной статической характеристики ОЭД и требуемой точностью коррекции систематической погрешности. Объём памяти ОЗУ зависит от числа возможных отсчётов в пределах измерительного диапазона

$$M = n \cdot N \quad (4.23)$$

где: $N = \chi_{\max} / \Delta\chi$; χ_{\max} – максимальное значение измеряемого угла; $\Delta\chi$ – шаг дискретизации отсчётов, определяемый выбранной разрядностью n двоичного кода ОЭД

$$n = \log_2 N \quad (4.24)$$

После завершения цикла обучения-калибровки ключи переводятся в нижнее положение, и система подготавливается к режиму измерения. Теперь двоичный код каждой однократной оценки χ_i^* , формируемый ОЭД, поступает непосредственно на вход дешифратора адреса внешнего ОЗУ, и из соответствующей ячейки памяти сразу считывается скорректированное значение усреднённой оценки, заранее вычисленное на этапе калибровки. Очевидно, что таким образом удаётся скорректировать систематическую составляющую погрешности измерения, которая может быть весьма значительной. В то же время случайная составляющая погрешности, обусловленная действием шумов, не корректируется.

Рассмотренный принцип построения измерительной системы обеспечивает достаточную гибкость корректировки её статической характеристики и предельное быстродействие системы в рабочем режиме измерения.

4.2.2. Принцип построения обучаемой телевизионной системы для автоматизации контроля заготовок микросхем.

Одним из наиболее сложных для автоматизации звеньев в цепи технологического процесса производства микросхем является участок промежуточного контроля внешнего вида заготовок. Во многих случаях именно эта операция предопределяет производительность всей технологической линии. Большинство «жёстких» алгоритмов, предлагаемых для оптико-электронных методов автоматизации визуального контроля, к сожалению, оказываются малоприспособленными для условий реального производства. Их можно условно разделить на две группы.

1. Относительно простые алгоритмы, основанные на сравнении контролируемых изделий с эталоном, обладающие достаточным быстродействием, но малой устойчивостью к воздействию дестабилизирующих факторов в условиях производства.

2. Значительно более сложные алгоритмы распознавания образов, учитывающие особенности контролируемых изделий и многообразие возможных дефектов. Они обладают недостаточным быстродействием и требуют значительных затрат на перепрограммирование даже при незначительных изменениях в объектах контроля.

Ниже рассматривается один из вариантов построения быстродействующей системы контроля на основе обучаемого автомата, в которой простота решающего правила сочетается с устойчивостью и гибкостью системы при изменении вида контролируемых изделий.

Система контроля (рис. 4.14) может решать задачи обнаружения и определения вида дефектов металлизации плоских заготовок микросхем. В основу её работы также положен известный принцип сравнения с эталоном. Однако, режиму автономного функционирования должен предшествовать период обучения системы при временном участии оператора.

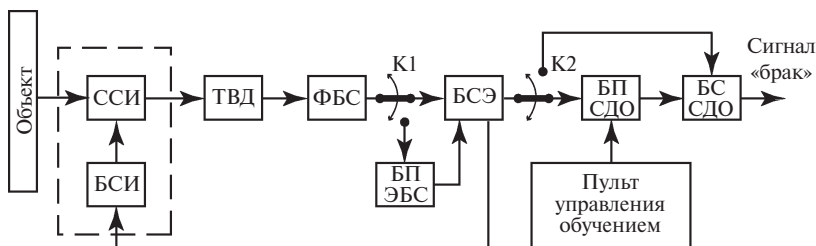


Рис. 4.14. Структурная схема системы автоматизированного контроля заготовок микросхем

Цикл обучения заключается в следующем.

Вначале системе предъявляется, в качестве эталона, годное изделие из контролируемой партии, которое удовлетворяет требованиям качества (рис. 4.15а). Изображение эталона посредством оптической системы (ОС) и телевизионного датчика (ТВД) преобразуется сначала в аналоговый видеосигнал, а затем, с помощью формирователя бинарного сигнала (ФБС), в эталонный бинарный сигнал, фиксируемый в специальном блоке памяти (БП ЭБС). При этом ключ К1 должен быть переведён в нижнее положение.

Затем с пульта управления обучением система переводится в режим сравнения (ключ К1 переводится в верхнее положение) и осуществляется предъявление системе других изделий из контролируемой партии. Оператор оценивает качество каждого изделия и нажимает кнопку на пульте обучения при отсутствии брака. При этом в блоке сравнения с эталоном (БСЭ) формируется сигнал разности между предъявляемым изделием и эталоном (рис. 4.15б), который затем фиксируется в блоке памяти сигнала допустимых отклонений (БП СДО). Далее цикл повторяется несколько десятков раз. Сигнал разности накапливается в БП СДО, формируя своеобразное «поле допусков» на данный вид изделий (рис. 4.15в).

После накопления в БП СДО достаточной информации о границах поля допусков система контроля переводится в основной режим автономного функционирования (ключ К2 переводится в верхнее положение). В этом режиме первоначально происходит

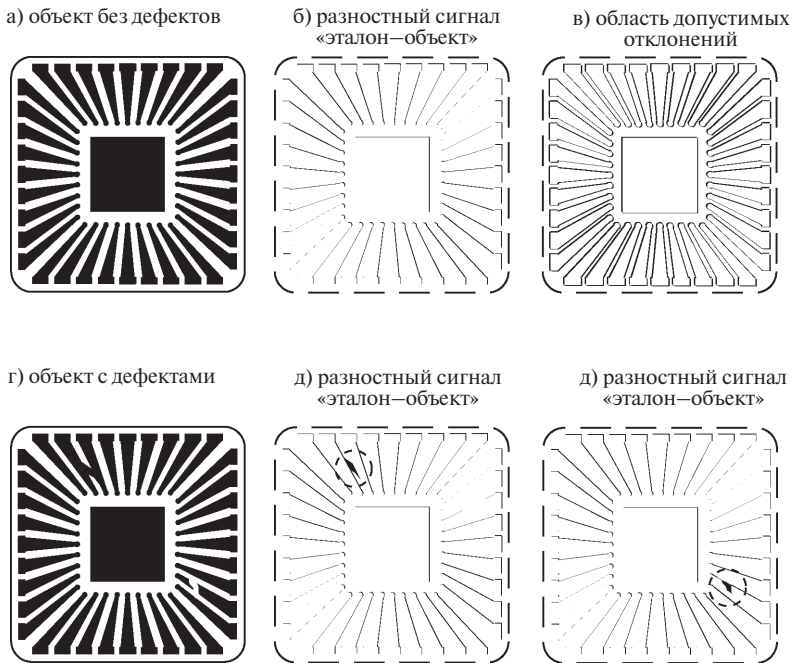


Рис. 4.15. К пояснению принципа работы системы контроля заготовок микросхем на основе обучаемого автомата.

сравнение предъявляемого объекта с эталоном. Результирующий разностный сигнал сравнивается с полем допусков, сформированным в процессе обучения при участии оператора. Если в процессе сравнения разностный сигнал лежит в пределах поля допусков, то изделие признаётся годным. В противном случае формируется сигнал «брак». По виду разностного сигнала, не соответствующего области допустимых отклонений, можно судить о характере дефекта и контролировать работу технологической линии (рис. 4.15г, д, е).

Разумеется, некоторые из функциональных узлов рассматриваемой системы могут быть выполнены не только в виде аппаратных модулей, но и на программном уровне. Принцип работы системы дополнительно поясняют блок-схемы алгоритмов, реализуемых на этапе обучения (рис. 4.16) и на этапе автоматического контроля (рис. 4.17).

Следует отметить, что в случае замены объекта контроля система может быть переориентирована путём проведения повторного этапа обучения. При этом надёжность функционирования системы практически не зависит от сложности изображения контролируемого

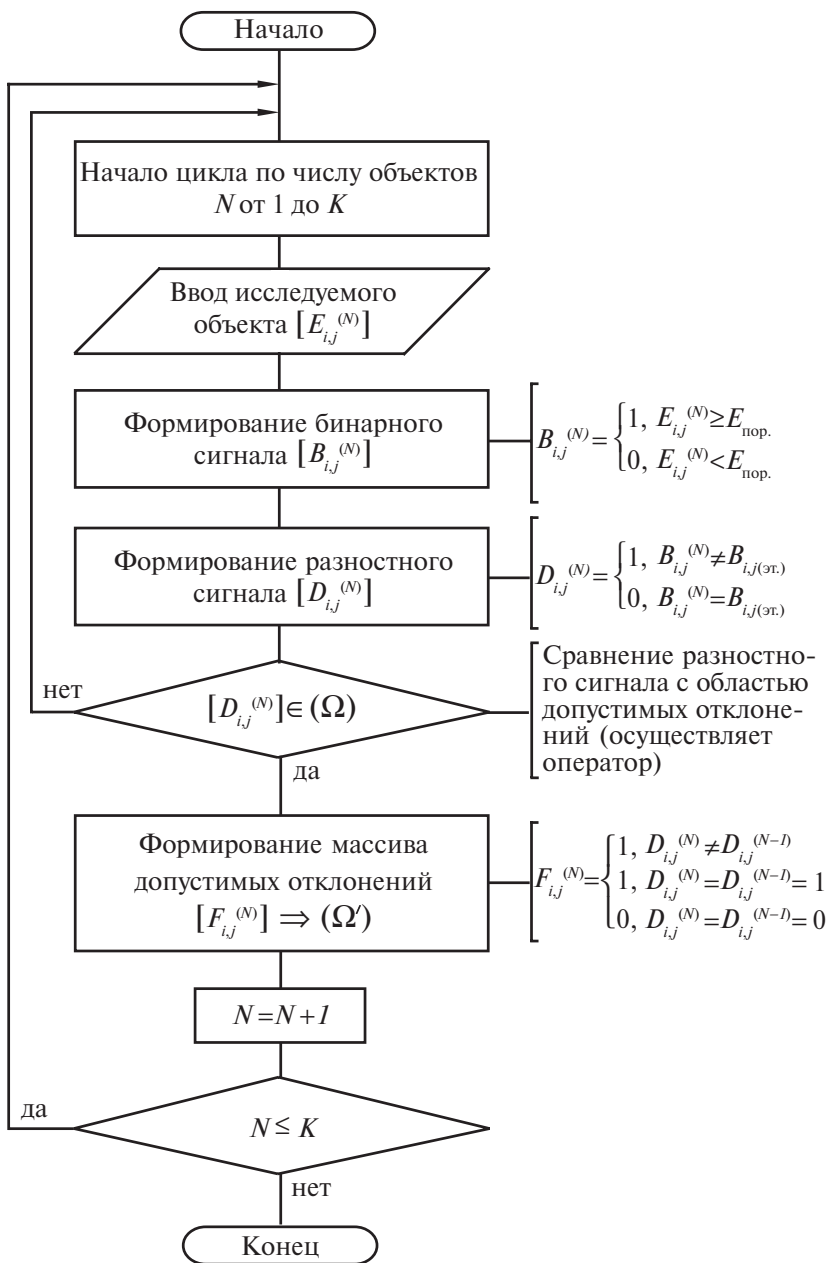


Рис. 4.16. Блок-схемы алгоритма, реализуемого на этапе обучения.

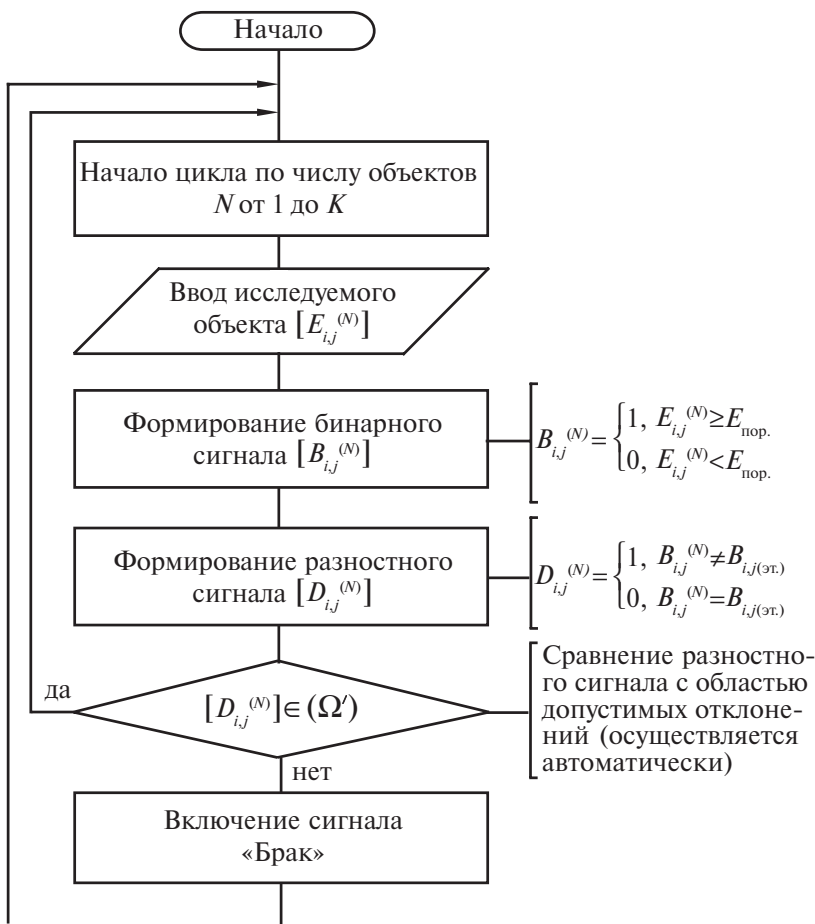


Рис. 4.17. Блок-схемы алгоритма на этапе автоматического контроля

объекта, если только не возникнут ограничения, связанные с недостаточной разрешающей способностью телевизионного датчика. Очевидно, что при необходимости контроля значительных площадей заготовок с высокой разрешающей способностью можно использовать не один, а несколько параллельно работающих каналов наблюдения (на нескольких телевизионных датчиках). Причем зоны наблюдения, охватываемые каждым из каналов, должны немного перекрываться.

Вопросы для самопроверки:

1. В чём состоит специфика автономных систем астроориентации, предопределяющая основные этапы обработки изображений в таких системах?

2. Назовите основные этапы решения задачи определения пространственной ориентации КА по данным бортовых измерений. Поясните этап обнаружения и селекции рабочих астроориентиров.

3. Поясните этапы решения задачи опознавания участка звёздного неба, случайно попадающего в поле зрения автономной системы астроориентации.

4. Как оценивается объём памяти бортового каталога, необходимый для нормальной работы АНС?

5. В чём заключается алгоритм интерполяции видеосигнала, используемый на этапе измерения угловых координат выбранного астроориентира? Каковы предпосылки для сокращения вычислительных затрат при реализации данного алгоритма?

6. Поясните общие принципы построения обучаемых АТСН.

7. В чём заключается процедура обучения, на этапе калибровки оптико-электронного угломера?

8. Поясните принцип работы обучаемого автомата для контроля заготовок микросхем на этапе обучения и на этапе автономного функционирования.

4.3. Основные этапы проектирования и реализации АТСН

Процесс проектирования и создания АТСН можно условно разбить на следующие этапы.

1. Формулировка задачи. Определение исходных данных, входных воздействий и тактико-технических требований, предъявляемых к системе.

2. Анализ известных технических решений и выбор на их основе общей структуры разрабатываемой системы.

3. Выбор и обоснование оптической схемы.

4. Выполнение габаритно-энергетического расчёта, в результате которого должен быть выбран тип приёмников оптического излучения, определены параметры оптической системы и других специфических узлов АТСН.

5. Выбор и обоснование математического метода обработки сигналов, используемого в разрабатываемой системе на основе анализа его характеристик по заданному критерию. Расчёт и оптимизация отдельных параметров. Следует заметить, что на этом этапе весьма эффективным средством может служить математическое моделирование входных воздействий, передаточных характеристик отдельных звеньев и возможных алгоритмов обработки аналоговых и цифровых сигналов в проектируемой АТСН.

6. Детальная разработка будущего алгоритма, включая определение состава операций, операндов и констант, а также способа обмена данными между ЦВУ и другими узлами системы.

На последующих этапах могут параллельно выполняться работы по проектированию и реализации аппаратной и программной частей.

№ этапа	Разработка аппаратной части	Разработка программной части
7 (уточнение структуры)	Определение состава устройств предварительной (допроцессорной) обработки сигналов. Разработка структурной схемы устройства сопряжения ЦВУ с другими узлами АТСН.	Выбор типа ЦВУ (микрочипа, микроконтроллера, микропроцессора) на основе анализа условий сопряжения с другими звеньями с учётом возможности выполнения требуемых операций по обработке сигналов за ограниченное время анализа.
8 (детализация)	Разработка функциональной схемы устройства сопряжения.	Разработка и отладка рабочей программы алгоритма на одном из возможных языков программирования. Моделирование и редактирование программы.
9 (определение аппаратных и вычислительных затрат)	Определение эксплуатационных характеристик надёжности, потребляемой мощности, габаритов, степени унификации и т.п. Принятие решения о необходимости корректировки предыдущих этапов.	Определение количества микроопераций, операндов и констант, а также необходимых объёмов памяти ОЗУ и ПЗУ. Принятие решения о необходимости корректировки предыдущих этапов.
10	Разработка принципиальной схемы устройства сопряжения и других узлов аппаратной части АТСН. Разработка конструкторской и другой технической документации.	Трансляция рабочей программы в машинные коды используемого ЦВУ (микрочипа).
11 (изготовление компонентов АТСН)	Изготовление аппаратной части разработанной АТСН.	Занесение рабочей программы в ПЗУ.

12. Отладка в процессе совместной работы аппаратной и программной частей АТСН, проведение её испытаний.

ЗАКЛЮЧЕНИЕ

В настоящем учебном пособии основное внимание уделено наиболее общим вопросам, связанным с проектированием автоматизированных телевизионных систем наблюдения (АТСН). Проведённый анализ возможных вариантов аппаратной структуры, используемой элементной базы и наиболее характерных алгоритмов цифровой обработки изображений, а также примеры реализации некоторых видов АТСН имеют, главным образом, практическую направленность. Более подробное изложение теоретических вопросов, касающихся методов описания и преобразования сигналов дискретных изображений, можно найти в других источниках (см. например, [3, 8, 9]).

Практические навыки по применению методов моделирования АТСН, с целью определения обоснованных требований к отдельным звеньям, а также с целью выявления оптимальных параметров решающих правил, могут быть получены в процессе выполнения специального комплекса лабораторных работ, посвященных этим вопросам [6].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Быстрые алгоритмы цифровой обработки изображений/Т.С. Хуанг, Дж.-О. Эклунд, Г.Дж. Нуссбаумер и др.; Под ред. Т.С. Хуанга: Пер. с англ. –М.: Радио и связь, 1984. –224с., ил.
2. Кухарев Г.А. и др. Систолические процессоры для обработки сигналов/Г.А. Кухарев, А.Ю. Тропченко, В.П. Шмерко. – Мн.: Беларусь, 1988. – 127 с.
3. Лукин С.Б. Конспект лекций по курсу ОЭС. – СПб, СПбГУИТМО, 2004. – 161 с.
4. Источники и приёмники излучения: Учебное пособие для студентов оптических специальностей вузов/Г.Г. Ишанин, Э.Д. Панков, А.Л. Андреев, Г.В. Польшиков. – СПб.: Политехника, 1991. – 240 с.
5. Сташин В.В и др. Проектирование цифровых устройств на однокристалльных микроконтроллерах/ В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. – М.: Энергоатомиздат, 1990. – 224 с.
6. Аппаратные и программные средства оптико-электронных приборов с телевизионными датчиками на ФПЗС.: Методические указания/А.Л. Андреев, С.Н. Ярышев, А.Р. Стрелков. – СПб.: ИТМО, 1995. – 49 с.
7. Источники и приёмники излучения.: Методические указания/ А.Л. Андреев, В.Л. Мусяков, С.Н. Ярышев, А.Р. Стрелков. – СПб.: ИТМО, 1998. – 49 с.
8. Системы технического зрения (принципиальные основы, аппаратное и математическое обеспечение)/А.Н. Писаревский, А.Ф. Чернявский, Г.К. Афанасьев и др.; Под общ. ред. А.Н. Писаревского, А.Ф. Чернявского. – Л.: Машиностроение, 1088. – 424 с.
9. Прэрт У. Цифровая обработка изображений: в 2 кн. М.: Мир, 1982. – 790 с.

ИЗ ИСТОРИИ КАФЕДРЫ ОПТИКО-ЭЛЕКТРОННЫХ ПРИБОРОВ И СИСТЕМ

История кафедры началась в 1936 году с организации в ЛИТМО кафедры военных оптических приборов. Первым заведующим кафедрой был К.Е. Солодилов, до этого возглавлявший ЦКБ ВООМП. Преподавателями кафедры стали сотрудники этого ЦКБ – М.А. Резунов, М.Я. Кругер, С.Т. Цуккерман, В.А. Егоров, Б.М. Кулежнов.

В годы Великой Отечественной войны кафедра была эвакуирована в Черепаново, где ее объединили с кафедрой оптико-механических приборов под руководством профессора А.И. Захарьевского. После возвращения в Ленинград кафедрой в 1945-46 годах по совместительству заведовал начальник КБ ГОИ М.А. Резунов.

В начале 1947 года кафедру возглавил профессор С.Т. Цуккерман, который руководил ею до 1972 года. В 1958 году кафедра была реорганизована в кафедру специальных оптических приборов, а в 1967 году – в кафедру оптико-электронных приборов (ОЭП). Издание С.Т. Цуккерманом в предвоенные годы книги «Точные механизмы» (М.: Оборонгиз, 1941) является значительным вкладом в развитие отечественного точного приборостроения.

В начале 60-х годов Г.Г. Ишанин занимался разработкой фотометрической аппаратуры, предназначенной для паспортизации оптико-электронных приборов и систем различного назначения. В 1965 году Г.Г. Ишаниным разработана теория, методы расчета и проектирования, технология и конструктивные решения приемников на основе термоупругого эффекта в кристаллическом кварце. Приемники на термоупругом эффекте внедрены в серийное производство. Научно-исследовательские работы, проведенные в указанном направлении, легли в основу учебной дисциплины «Источники и приемники излучения».

Значительное влияние на содержание подготовки специалистов и научных исследований оказало привлечение к работе на кафедре выдающегося специалиста в области оптико-электронного приборостроения, члена-корреспондента РАН, Героя Социалистического Труда, лауреата Ленинской премии профессора М.М. Мирошникова, который, работая на кафедре ОЭП с 1969 года по 1976 год в должности профессора по совместительству, поставил и читал курс «Теория оптико-электронных приборов».

С 1972 по 1992 год кафедрой ОЭП заведовал Заслуженный деятель науки и техники РСФСР, профессор Л.Ф. Порфирьев, известный специалист в области автоматических ОЭПиС в комплексах навигации и управления авиационной и космической техникой. Соответственно тематика выполнения научно-исследовательских работ на кафедре приобрела новые направления, существенно увеличилось число тем, носящих поисковый фундаментальный характер. Были разработаны новый учебный план и программы учебных дисциплин. Л.Ф. Порфирьев является автором 19 учебников, учебных пособий и монографий. Результаты его работ можно оценить как значительный вклад в разработку общей теории оптико-электронных систем. В 70-90 годах под руководством Л.Ф. Порфирьева был разработан ряд астродатчиков, систем астроориентации и космической навигации (В.И. Калинин, А.Л. Андреев, С.Н. Ярышев).

Г.Н. Грязин, перешедший на кафедру с радиотехнического факультета в конце 60-х годов, продолжил свои работы в области прикладного телевидения, в частности, по разработке систем наблюдения за быстро движущимися объектами и быстропротекающими процессами. Благодаря этим работам проводились исследования явлений кавитации гребных валов судов, а также были начаты работы по созданию системы контроля качества текстильной продукции. Таким образом были заложены основы телевидения быстропротекающих процессов.

С 1975 года заведующим отраслевой лабораторией стал А.Н. Тимофеев, который продолжил исследования по разработке методов и средств контроля пространственного положения объектов с помощью ОЭП с оптической равносигнальной зоной для машиностроения, энергетики, строительства, судостроения и железнодорожного транспорта.

С 1975 года, после увольнения в запас, из ЛВИКА им. А.Ф. Можайского на кафедру пришел работать в должности профессора С.П. Авдеев, известный специалист в области ОЭПиС космических аппаратов. Он поставил курсы и читал лекции по учебным дисциплинам «Оптико-электронные приборы», «Оптико-электронные приборы систем управления», «Оптико-электронные приборы для научных исследований».

Существенное влияние на содержание подготовки специалистов и научных исследований оказало привлечение к работе на кафедре лауреата Ленинской и Государственной премий профессора Б.А. Ермакова, известного специалиста в области физической оптики и оптико-электронного приборостроения. Б.А. Ермаков работал на кафедре ОЭП с 1979 года по 1992 год в должности профессора по совместительству и поставил курс «Оптико-электронные приборы с лазерами».

В 70-80 годах под руководством Е.Г. Лебедько проводились исследования законов отражения лазерного излучения от нестационарных поверхностей и протяженных объектов, исследования в области теории идентификации объектов по их излучению в сложной фоновой ситуации. Создан комплекс для лазерной локации крупногабаритных морских объектов сложной конфигурации и водной поверхности. В этих работах принимали участие доценты О.П. Тимофеев и С.Б. Лукин. Результаты, полученные в рамках этих исследований, нашли применение в учебных курсах: «Оптико-электронные системы локации и связи», «Оптоэлектронные системы автоматизации технологических процессов» и «Прикладная оптика».

С 1992 г. заведующим кафедрой является Заслуженный деятель науки Российской Федерации, профессор Э.Д. Панков. В 1992 году кафедра была переименована в кафедру оптико-электронных приборов и систем (ОЭПиС). Под руководством Э.Д. Панкова в 70-90-х годах были проведены разработки ряда оптико-электронных приборов и систем специального и гражданского применения, нашедших практическое внедрение и способствующих научно-техническому прогрессу и укреплению обороноспособности нашей страны. В частности, исследования и разработки в области линейных и угловых измерений позволили приступить к решению общей проблемы согласования отсчетных баз на нестационарно деформируемых объектах с помощью оптико-электронных систем.

По результатам научных работ сотрудниками кафедры ОЭПиС выпущено в свет 15 монографий, 11 учебников и учебных пособий. На кафедре подготовлено 14 докторов наук, а также более 110 кандидатов наук.

На разработки кафедры получены авторские свидетельства СССР и патенты Российской Федерации на более чем 200 изобретений. Наибольший вклад в изобретательскую деятельность внес Э.Д. Панков - автор 123 изобретений, из которых 33 внедрены в промышленности.

За все время существования кафедры подготовлено более 3000 инженеров.

Андрей Леонидович Андреев

Автоматизированные телевизионные системы наблюдения

Часть II. Арифметико-логические основы и алгоритмы

Учебное пособие для курсового и дипломного проектирования

В авторской редакции

Компьютерный набор и верстка

Дизайн обложки

Ю.А. Смирнов

Т.П. Крутихин

Редакционно-издательский отдел СПб ГИТМО (ТУ)

Зав. редакцией

Лицензия ИД №00408 от 05.11.99

Подписано к печати

Тираж 100 экз.

Заказ №

Н.Ф. Гусарова

Отпечатано на ризографе